

Windows Explorer Shell context menu problem

Windows Explorer Shell context menu problemHi,

I have an interesting problem about adding menu items to Windows Explorer context menu. Details below:

I have used a sample (ShellExt) from Microsoft Platform SDK for Windows Server 2003 R2. After some changes, I have built the sample and got ShellExt.dll file. Then, I have registered the extension (using ShellExt.reg).

Till now, everthing goes well. But the interesting thing is:

I have two computers which have the same operating system (Windows XP SP2).

The .dll file runs perfectly at first computer and adds menu items to Windows Explorer Shell context menu, but it does not do the same task at second computer.

At first step, I have examined and compared all related registry entries of two computers. They are same.

Secondly, using a program which is called "ShellExView", I have tried every non-Microsoft products extensions one by one (disabling and enabling them) to clarify whether the failure stems from them or not. Additionally, "ShellExView" shows that the extension ShellExt.dll runs perfectly at the second computer.

DebugView shows all messages at first computer but nothing at the second.

After a long research on internet, I have failed to find any clue about this kind of problem.

May be it seems very strange but the only difference between two computers is:

The background color of context menu is gray at first computer, and white at the second.

Thanks in advance!

Add items to Windows Explorer Shell context (right-click) menu easily – How to add them ?

Add an entry to Windows Explorer context menu easily with Windows Explorer Shell Context Menu. This powerful .Net component for custom items adding to Windows Explorer Shell context menu will add all your application entries to Windows Explorer Shell context menu. It and VB.NET support include detailed C# / VB.NET samples, tutorials 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 all your items to Windows Explorer Shell context menu to be shown in any way - with your 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 files or shown only for files of particular type (for example, only for .DOC , .MP3,.WMA,.AAC , .AVI media files)

- Add your program items to Windows Explorer Shell context menu, sub-menus, sub-sub-menus, sub-menus of unlimited depth and even more

Windows Explorer Shell Context Menu - is a powerful .Net framework component that support all you need to add all your application items to the Windows Explorer Shell context menu - in a fast and easy way. Add your items to Windows Explorer Shell context menu right now – add items to context menu fast and exactly as you want :

Discussion

Before will be provided this question discussion we should inform you that menu item appending to context menu methods described below work 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. Here are the autumn 1997 descussion details:

Can you share the contents of your registration and the location of the binaries on the two machines?

I imagine you have additional registration that you've done on the first computer or a hard-coded path that only applies to the first machine.

First of all, thank you very much for your kind answer.

After your message, I have changed

```
HKEY_CLASSES_ROOT\CLSID\{2C4BBD71-
F932-410A-8603-415085CE28D9}\InProcServer32
```

Default = "shellex.dll" path idenfication
but nothing have changed at second computer.

ShellExt.Reg (I have used the same file on both computers)

REGEDIT4

```
[HKEY_CLASSES_ROOT\CLSID\{2C4BBD71-F932-410A-8603-415085CE28D9}]
@="Shell Extension Sample"
[HKEY_CLASSES_ROOT\CLSID\{2C4BBD71-
F932-410A-8603-415085CE28D9}\InProcServer32]
@="shellex.dll"
```

```
[HKEY_CLASSES_ROOT\.gak]
@="GAKFile"
[HKEY_CLASSES_ROOT\GAKFile]
@="Shell Extension file"
[HKEY_CLASSES_ROOT\GAKFile\shellex\ContextMenuHandlers]
@="GAKMenu"
[HKEY_CLASSES_ROOT\GAKFile\shellex\ContextMenuHandlers\GAKMenu]
@="{2C4BBD71-F932-410A-8603-415085CE28D9}"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Shell
Extensions\Approved]
```

First Computer (shellex works perfectly on this computer):

File path : C:\WINDOWS\system32\ShellExt.dll

Registry entries:

1. HKEY_CLASSES_ROOT\CLSID\{2C4BBD71-F932-410A-8603-415085CE28D9}

Default = "Shell Extension Sample"

2. HKEY_CLASSES_ROOT\CLSID\{2C4BBD71-F932-410A-8603-415085CE28D9}\InProcServer32

Default = "shellex.dll"
ThreadingModel = "Apartment"

3. HKEY_CLASSES_ROOT\gak

Default = "GAKFile"

4. HKEY_CLASSES_ROOT\GAKFile

Default = "Shell Extension file"

5. HKEY_CLASSES_ROOT\GAKFile\shellex\ContextMenuHandlers

Default = "GAKMenu"

6. HKEY_CLASSES_ROOT\GAKFile\shellex\ContextMenuHandlers\GAKMenu

Default = "{2C4BBD71-F932-410A-8603-415085CE28D9}"

7. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2C4BBD71-F932-410A-8603-415085CE28D9}

Default = "Shell Extension Sample"

8. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2C4BBD71-F932-410A-8603-415085CE28D9}\InProcServer32

Default = "shellex.dll"
ThreadingModel = "Apartment"

9. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\GAKFile

Default = "Shell Extension file"

10. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\GAKFile\shellex
\ContextMenuHandlers

Default = "GAKMenu"

11. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\GAKFile\shellex
\ContextMenuHandlers\GAKMenu

Default = "{2C4BBD71-F932-410A-8603-415085CE28D9}"

12. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Shell
Extensions\Approved

{2C4BBD71-F932-410A-8603-415085CE28D9} = "Shell Extension
Sample"

Second Computer

File path : D:\WINDOWS\system32\ShellExt.dll

Registry entries:

1. HKEY_CLASSES_ROOT\CLSID\{2C4BBD71-F932-410A-8603-415085CE28D9}

Default = "Shell Extension Sample"

2. HKEY_CLASSES_ROOT\CLSID\{2C4BBD71-F932-410A-8603-415085CE28D9}\InProcServer32

Default = "shellex.dll"
ThreadingModel = "Apartment"

3. HKEY_CLASSES_ROOT\gak

Default = "GAKFile"

4. HKEY_CLASSES_ROOT\GAKFile

Default = "Shell Extension file"

5. HKEY_CLASSES_ROOT\GAKFile\shellex\ContextMenuHandlers

Default = "GAKMenu"

6. HKEY_CLASSES_ROOT\GAKFile\shellex\ContextMenuHandlers\GAKMenu

Default = "{2C4BBD71-F932-410A-8603-415085CE28D9}"

7. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached

{2C4BBD71-F932-410A-8603-415085CE28D9} {000214E8-0000-0000-C000-000000000046} 0x401 (REG_BINARY) = "01 00 00 00 31 00 39 00 76 ca da 59 df eb c7 01"

8. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2C4BBD71-F932-410A-8603-415085CE28D9}

Default = "Shell Extension Sample"

9. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2C4BBD71-F932-410A-8603-415085CE28D9}\InProcServer32

Default = "shellex.dll"
ThreadingModel = "Apartment"

10. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\GAKFile

Default = "Shell Extension file"

11. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\GAKFile\shellex\ContextMenuHandlers

Default = "GAKMenu"

12. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\GAKFile\shellex\ContextMenuHandlers\GAKMenu

Default = "{2C4BBD71-F932-410A-8603-415085CE28D9}"

13. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Shell Extensions\Approved

```
{2C4BBD71-F932-410A-8603-415085CE28D9} = "Shell Extension Sample"
```

14. HKEY_USERS\S-1-5-21-1935655697-2111687655-839522115-500\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached

```
{2C4BBD71-F932-410A-8603-415085CE28D9} {000214E8-0000-0000-C000-000000000046} 0x401 (REG_BINARY) = "01 00 00 00 31 00 39 00 76 ca da 59 df eb c7 01"
```

You might be running afoul of verclsid (introduced in this update <http://www.microsoft.com/technet/security/Bulletin/MS06-015.msp>). Perhaps you don't have all the dependencies you need on the second machine (for instance, a particular version of the c-runtime). Anyway, verclsid verifies that a COM object is going to do a few basic things without crashing.

I would check this by using depends.exe from the second machine and making sure that you've got all your dependencies. If that works, try debugging verclsid as it checks out your extension. I don't remember the syntax, but you can just debug verclsid whenever it runs a couple times to get an idea for how to invoke it. Image File Execution Options is a reasonable way of doing that (<http://msdn2.microsoft.com/en-us/library/a329t4ed.aspx>).

I don't see anything really wrong with the registration and the fact that verclsid seems to be invoking your stuff is also evidence that it's hooked up right and that there is something wrong with loading your dll.

I have checked all dependencies. The result is same for both machines and no problem about dependencies.

Here, the article tells about manual steps to add shell extensions to the "allow list". I have added the following value to the registry for the extension:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached
{2C4BBD71-F932-410A-8603-415085CE28D9} {000214E8-0000-0000-C000-000000000046} 0x401 (REG_DWORD) = 1
```

But, it did not change the result.

To try another shell extension (newcntx.dll), which gives the same results with ShellExt on both computers, I have used the same method and I have added the following value:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached
{E0147448-0B85-4C6A-A70D-6F54422210F6} {000214E8-0000-0000-C000-000000000046} 0x401 = 1
```

The result was positive at the second computer. newcntx.dll menu items are added to the windows explorer context menu.

All the registry entries for newcntx.dll:

1. HKEY_CLASSES_ROOT*\shellex\ContextMenuHandlers\MyNewCntx
Default = "{E0147448-0B85-4C6A-A70D-6F54422210F6}"

2. HKEY_CLASSES_ROOT\CLSID\{E0147448-0B85-4C6A-A70D-6F54422210F6}
Default = "MyNewCntx"

3. HKEY_CLASSES_ROOT\CLSID\{E0147448-0B85-4C6A-A70D-6F54422210F6}\InProcServer32
Default = "D:\Program Files\My New Context\newcntx.dll"
ThreadingModel = "Apartment"

4. HKEY_CLASSES_ROOT\Directory\shellex\ContextMenuHandlers\MyNewCntx
Default = "{E0147448-0B85-4C6A-A70D-6F54422210F6}"

5. HKEY_CLASSES_ROOT\Folder\shellex\ContextMenuHandlers\MyNewCntx
Default = "{E0147448-0B85-4C6A-A70D-6F54422210F6}"

6. HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Shell
Extensions\Cached
{E0147448-0B85-4C6A-A70D-6F54422210F6} {000214E8-0000-0000-C000-000000000046} 0x401 = "01 00 00 00 31 00 39 00 b4 79 16 41 47 f1 c7 01"

7. HKEY_LOCAL_MACHINE\SOFTWARE\Classes*\shellex\ContextMenuHandlers\MyNewCntx
Default = "{E0147448-0B85-4C6A-A70D-6F54422210F6}"

8. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{E0147448-0B85-4C6A-A70D-6F54422210F6}
Default = "MyNewCntx"

9. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{E0147448-0B85-4C6A-A70D-6F54422210F6}\InProcServer32
Default = "D:\Program Files\My New Context\newcntx.dll"
ThreadingModel = "Apartment"

10. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Directory\shellex\ContextMenuHandlers\MyNewCntx
Default = "{E0147448-0B85-4C6A-A70D-6F54422210F6}"

11. HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Folder\shellex\ContextMenuHandlers\MyNewCntx
Default = "{E0147448-0B85-4C6A-A70D-6F54422210F6}"

12. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Shell
Extensions\Approved
{E0147448-0B85-4C6A-A70D-6F54422210F6} = "MyNewCntx"

13. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Shell
Extensions\Cached
{E0147448-0B85-4C6A-A70D-6F54422210F6} {000214E8-0000-0000-C000-000000000046} 0x401 = 1

14. HKEY_USERS\S-1-5-21-1935655697-2111687655-839522115-500\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached

{E0147448-0B85-4C6A-A70D-6F54422210F6} {000214E8-0000-0000-C000-000000000046} 0x401 = "01 00 00 00 31 00 39 00 b4 79 16 41 47 f1 c7 01"

Consequently, security update MS06-015 is not installed at the first computer. Cverclsid.exe, coming with security update MS06-015, blocks the shellex.dll and newcntx.dll at the second computer. Adding item 13 for newcntx.dll have solved the problem. But this solution is not valid for shellex.dll.

Instead of adding item 13 to the registry, renaming verclsid.exe file to verclsid.old gave the same result.

The most interesting thing is:

Softwares like Winzip, Winrar, Axcrypt, etc. use their extensions without any problem at the second computer and they have same registry entries with newctx.dll except item 13. They need not adding item 13.

Right - the other applications don't need those entries because they are going to pass the verclsid test anyway.

The next thing to do is debug explorer and see if your dll is getting loaded. Probably, you're getting loaded (now that you've faked passing the verclsid test), and you can debug the failure right in explorer - put a breakpoint on your DLLGetClassObject and step out from there. Perhaps you're failing some internal call on the second machine - only debugging can tell (and apparently my guessing hasn't turned up too much yet :).

However, failing the verclsid test means that you've got something pretty fundamental that's broken. So you need to track down where that's failing as well (it may be for the same reason).