How to host an IContextMenu, part 5 – Handling menu messages

devblogs.microsoft.com/oldnewthing/20040927-00

September 27, 2004

Raymond Chen

One bug that was called out immediately in our first attempt at displaying the context menu to the user is that the Open With and Send To submenus don't work.

The reason for this is that these submenus are delay-generated (which explains why they don't contain anything interesting when you expand them) and owner-drawn (which you can't notice yet because of the first problem, but trust me, they are).

This is where the IContextMenu2::HandleMenuMsg and IContextMenu3::HandleMenuMsg2 methods are used.

Historical note: IContextMenu2::HandleMenuMessage is on its own interface rather than being merged with the base interface IContextMenu because it was added late in Windows 95 development, so it was considered safer to add a derived interface than to make everybody who had been writing Windows 95 shell extensions go back and rewrite their code. IContextMenu3::HandleMenuMessage2 was added in Internet Explorer 4 (I think) when it became clear that the ability for a context menu extension to override the default message return value was necessary in order to support keyboard accessibility in owner-drawn context menus.

In a "real program", these two variables would be class members associated with the window, but this is just a sample program, so they are globals. **When you write your own programs, don't use global variables here** because they will result in mass mayhem once you get a second window, since both of them will try to talk to the interface even though only the window displaying the context menu should be doing so.

```
IContextMenu2 *g_pcm2;
IContextMenu3 *g_pcm3;
```

These two new variables track the IContextMenu2 and IContextMenu3 interfaces of the active tracked popup menu. We need to initialize and uninitalize them around our call to TrackPopupMenuEx:

```
      pcm->QueryInterface(IID_IContextMenu2, (void**)&g_pcm2);
      pcm->QueryInterface(IID_IContextMenu3, (void**)&g_pcm3);
      int iCmd = TrackPopupMenuEx(hmenu, TPM_RETURNCMD, pt.x, pt.y, hwnd, NULL);
      if (g_pcm2) {
        g_pcm2->Release();
        g_pcm2 = NULL;
      }
      if (g_pcm3) {
        g_pcm3->Release();
        g_pcm3 = NULL;
      }
```

And finally we need to invoke the HandleMenuMessage/HandleMenuMessage methods in the window procedure:

```
LRESULT CALLBACK
WndProc(HWND hwnd, UINT uiMsg, WPARAM wParam, LPARAM lParam)
{
    if (g_pcm3) {
        LRESULT lres;
        if (SUCCEEDED(g_pcm3->HandleMenuMsg2(uiMsg, wParam, lParam, &lres))) {
          return lres;
        }
    } else if (g_pcm2) {
        if (SUCCEEDED(g_pcm2->HandleMenuMsg(uiMsg, wParam, lParam))) {
          return 0;
        }
    }


    switch (uiMsg) {
    ….
```

In the window procedure, we ask the context menu whether it wishes to handle the menu message. If so, then we stop and return the desired value (if HandleMenuMsg2) or just zero (if HandleMenuMsg).

With these changes, run the scratch program again and observe that the Open With and Send To submenus now operate as expected.

Next time: Getting menu help text.

Raymond Chen

Follow