What if I have two programs that are logically a single application, and I want them to be treated as a single group on the taskbar?

devblogs.microsoft.com/oldnewthing/20150810-00

August 10, 2015

Raymond Chen

Suppose you have two programs that are logically one application, and you want them to be treated as a single group on the taskbar, share jump lists, and all that stuff. Basically, even though they are physically separate executables, they should be logically be treated as two facets of the same application.

This the opposite of What if my application is really two applications bundled into a single file, and I want them collected into two groups on the taskbar in Windows 7?. But the solution is the same.

As noted in the linked article, the central concept is the Application User Model ID. The shell uses the Application User Model ID to identify applications. If you give two processes the same Application User Model ID, then they will be treated as two instances of the same application. If you give them different Application User Model IDs, then they will be treated as instances of separate applications.

In the linked article, we had two processes running from the same executable, but we wanted them to be treated as separate applications, so we gave them different Application User Model IDs. Today, we have two processes running from separate executables, but we want them to be treated as the same application. Solution: Give them the same Application User Model ID.

Start with our scratch program and make these changes:

Tricked you! We’re not changing anything. Just compile and run it. Then copy `scratch.exe` to `scratch2.exe` and run that. Observe that the two windows are not grouped together in the taskbar.

Okay, now let’s make these changes:

```
#include <shlobj.h>

...
int WINAPI WinMain(HINSTANCE hinst, HINSTANCE hinstPrev,
                   LPSTR lpCmdLine, int nShowCmd)
{
    SetCurrentProcessExplicitAppUserModelID(L"Contoso.Designer");
    ...
}
```

Compile and run this program, then copy `scratch.exe` to `scratch2.exe` and run the second copy. This time, the two windows are grouped together because they have the same Application User Model ID.

Bonus chatter: If you create a shortcut to the application, you should also assign the custom Application User Model ID to the shortcut. That way, if the user calls up the jump list for the shortcut, they get the jump list for the correct Application User Model ID.

Raymond Chen

Follow