How do I prevent users from pinning my program to the taskbar?

devblogs.microsoft.com/oldnewthing/20110601-00

June 1, 2011

Raymond Chen

A customer wanted to prevent users from pinning their application to the taskbar.

> I have an application that is launched as a helper by a main application. Users shouldn't be launching it directly, but rather should be launching the main application. But since the helper shows up in the taskbar, users may be tempted to right-click on the taskbar icon and select "Pin to taskbar." Unfortunately, this pins the helper program to the taskbar instead of the main application, and launching the helper program directly doesn't work. Is there a way I can prevent users from pinning the helper program?

It so happens that there are a number of ways of marking your helper program as *Don't pin me*. Given the description above, the most direct way is probably to set the `System.AppUserModel.PreventPinning` property on the window created by the helper program.

Take our scratch program and make the following changes:

```
#include <shellapi.h>
#include <propsys.h>
#include <propkey.h>
HRESULT MarkWindowAsUnpinnable(HWND hwnd)
{
 IPropertyStore *pps;
 HRESULT hr = SHGetPropertyStoreForWindow(hwnd, IID_PPV_ARGS(&pps));
 if (SUCCEEDED(hr)) {
  PROPVARIANT var;
  var.vt = VT_BOOL;
  var.boolVal = VARIANT_TRUE;
  hr = pps->SetValue(PKEY_AppUserModel_PreventPinning, var);
  pps->Release();
 }
 return hr;
}

BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
 MarkWindowAsUnpinnable(hwnd);
 return TRUE;
}
```

I set the `PROPVARIANT` manually instead of using `InitPropVariantFromBoolean` just to emphasize that the `boolVal` must be `VARIANT_TRUE` and not `TRUE`. In real life, I probably would have used `InitPropVariantFromBoolean`.

Run this program and observe that "Pin this program to taskbar" does not appear on the menu when you right-click on the taskbar button.

Even better would be to permit pinning, but set the `System.AppUserModel.Relaunch-Command`, `.RelaunchDisplayNameResource` and optionally `.RelaunchIconResource` properties so that if the user tries to pin the helper, it actually pins the main application.

Start with a new scratch program and make these changes:

```
#include <shellapi.h>
#include <propsys.h>
#include <propkey.h>
#include <propvarutil.h>
HRESULT IPropertyStore_SetValue(IPropertyStore *pps,
    REFPROPERTYKEY pkey, PCWSTR pszValue)
{
 PROPVARIANT var;
 HRESULT hr = InitPropVariantFromString(pszValue, &var);
 if (SUCCEEDED(hr))
 {
  hr = pps->SetValue(pkey, var);
  PropVariantClear(&var);
 }
 return hr;
}
BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
 IPropertyStore *pps;
 HRESULT hr = SHGetPropertyStoreForWindow(hwnd, IID_PPV_ARGS(&pps));
 if (SUCCEEDED(hr)) {
  IPropertyStore_SetValue(pps,
    PKEY_AppUserModel_ID, L"Contoso.Scratch");
  IPropertyStore_SetValue(pps,
    PKEY_AppUserModel_RelaunchCommand,
    L"notepad.exe %windir%\\system.ini");
  IPropertyStore_SetValue(pps,
    PKEY_AppUserModel_RelaunchDisplayNameResource,
    L"C:\\full\\path\\to\\scratch.exe,-1");
  // optionally also set PKEY_AppUserModel_RelaunchIconResource
  pps->Release();
 }
 return TRUE;
}
// resource file
STRINGTABLE BEGIN
 1 "Open system.ini"
END
```

I'm pulling a fast one here and pretending that Notepad is my main application. Obviously you'd use your actual main application. (I'm also hard-coding the path to my scratch program.)

When you run this program, right-click on the taskbar button. Observe that the option to run a new copy of the program is called *Open system.ini* and if you pick it (or use the middle-mouse-button shortcut), Notepad runs. If you pin the program, the pinned icon runs Notepad.

Even if you don’t need to redirect the pinned item to another program, you can use this second technique to pass a custom command line for the pinned icon.

Raymond Chen

Follow