Creating custom tasks on a jump list

devblogs.microsoft.com/oldnewthing/20131223-00

December 23, 2013

Raymond Chen

Today's Little Program adds a custom task to the application's jump list. Take the scratch program and make the following changes. (Remember, Little Programs do very little error checking because that's how they roll.)

```
#include <shlobj.h>
#include <propkey.h>
#include <wrl/client.h>
using namespace Microsoft::WRL;
ComPtr<IShellLink>
CreateShellLinkForTask(
    PCWSTR pszTitle,
    PCTSTR pszArgs,
    int idIcon)
{
    ComPtr<IShellLink> spsl;
    CoCreateInstance(CLSID_ShellLink, nullptr, CLSCTX_ALL, IID_PPV_ARGS(&spsl));
    wchar_t szBuf[MAX_PATH];
    GetModuleFileName(g_hinst, szBuf, ARRAYSIZE(szBuf));
    spsl->SetPath(szBuf);
    spsl->SetArguments(pszArgs);
    spsl->SetIconLocation(szBuf, idIcon);
    PROPVARIANT pvar;
    pvar.vt = VT_LPWSTR;
    pvar.pwszVal = const_cast<PWSTR>(pszTitle);
    ComPtr<IPropertyStore> spps;
    spsl.As(&spps);
    spps->SetValue(PKEY_Title, pvar);
    spps->Commit();
    return spsl;
}
```

This helper function creates an in-memory shell link object with the specified title, command line arguments, and icon. The underlying executable is assumed to be the running executable.

```
BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
  ComPtr<ICustomDestinationList> spcdl;
  CoCreateInstance(CLSID_DestinationList, nullptr, CLSCTX_ALL,
                   IID_PPV_ARGS(&spcdl));
  ComPtr<IObjectCollection> spoc;
  UINT cMinSlots;
  spcdl->BeginList(&cMinSlots, IID_PPV_ARGS(&spoc));
  spoc->Clear();
  spoc->AddObject(CreateShellLinkForTask(L"New frob",
                   TEXT("/frob"), -2).Get());
  spcdl->AddUserTasks(spoc.Get());
  spcdl->CommitList();
  return TRUE;
}
```

When our window is created, we get the destination list for our application and ask it for an object collection so we can fill it with tasks. We empty the existing collection and add a single shortcut called "New frob" and which passes the `/frob` command line argument. The icon here is given as a negative number to indicate that it is an icon ID rather than an icon index. We then tell the destination list that this is our new task collection.

Before we forget, let's add the icon to our resource file.

```
// scratch.rc
1 ICON icon1.ico
2 ICON icon2.ico
```

I'll leave you to find some icons to use. Icon number 2 is the one that will be used for the jump list. (Icon number 1 I left to represent the application itself.)

Finally, we respond to the command line switch.

```
int WINAPI WinMain(HINSTANCE hinst, HINSTANCE hinstPrev,
                   LPSTR lpCmdLine, int nShowCmd)
{
  if (strcmp(lpCmdLine, "/frob") == 0) {
    MessageBox(nullptr, L"Frob!", L"Title", MB_OK);
    return 0;
  }
  ...
}
```

If the command line switch `/frob` is passed, then we say something silly. In real life, we would create a new frob, possibly by looking for an existing running copy of the program and asking it to do the creation.

Okay, run this program and then right-click on the taskbar icon. Observe that there is now a *New frob* task, and if you select it, you get the silly message.

Raymond Chen

Follow