How can I tell that I have a shell folder that represents My Computer?

devblogs.microsoft.com/oldnewthing/20130920-00

September 20, 2013

Raymond Chen

You have in your hands an `IShellFolder`, and you want to know whether this is an `IShellFolder` that represents *My Computer*. There are a few ideas that may occur to you.

One is to ask the folder for its current location and compare it to `CSIDL_DRIVES`.

```
#define STRICT_TYPED_ITEMIDS
#include <shlobj.h>
PIDLIST_ABSOLUTE GetIDListViaPersistFolder(IUnknown *punk)
{
  PIDLIST_ABSOLUTE pidl = NULL;
  IPersistFolder2 *ppf;
  if (SUCCEEDED(punk->QueryInterface(IID_PPV_ARGS(&ppf)))) {
    ppf->GetCurFolder(&pidl);
    ppf->Release();
  }
  return pidl;
}
HRESULT CompareAbsoluteIDLists(
    LPARAM lParam,
    PCUIDLIST_ABSOLUTE pidl1,
    PCUIDLIST_ABSOLUTE pidl2,
    int *piResult)
{
  *piResult = 0;
  IShellFolder *psfDesktop;
  HRESULT hr = SHGetDesktopFolder(&psfDesktop);
  if (SUCCEEDED(hr)) {
    hr = psfDesktop->CompareIDs(lParam,
    reinterpret_cast<PCUIDLIST_RELATIVE>(pidl1),
    reinterpret_cast<PCUIDLIST_RELATIVE>(pidl2));
    if (SUCCEEDED(hr)) {
     *piResult = (short)HRESULT_CODE(hr);
    }
    psfDesktop->Release();
  }
  return hr;
}
BOOL IsMyComputerFolder(IUnknown *punk)
{
  BOOL fIsMyComputer = FALSE;
  PIDLIST_ABSOLUTE pidl = GetIDListViaPersistFolder(punk);
  if (pidl) {
    PIDLIST_ABSOLUTE pidlMyComputer;
    if (SUCCEEDED(SHGetSpecialFolderLocation(NULL,
                                 CSIDL_DRIVES, &pidlMyComputer)))
    {
      int iCompare;
      fIsMyComputer = SUCCEEDED(CompareAbsoluteIDLists(
                               SHCIDS_CANONICALONLY,
                               pidl, pidlMyComputer, &iCompare)) &&
                      iCompare == 0;
      CoTaskMemFree(pidlMyComputer);
    }
    CoTaskMemFree(pidl);
  }
  return fIsMyComputer;
}
```

Okay, we have a lot of moving parts here. Let's look at them one at a time.

The `GetIDListViaPersistFolder` function takes an object and asks `IPersistFolder2::GetCurFolder` what folder it represents. Since we don't actually use any methods on the object beyond what is provided by `IUnknown`, we weaken the parameter requirement to simply `IUnknown`.

The `CompareAbsoluteIDLists` function compares two absolute ID lists according to the criteria specified by the `lParam`.

The `IsMyComputerFolder` combines these two function: It takes the object you pass in and gets the ID list it represents. It then gets the ID list for the My Computer folder. And then it compares the two via `SHCIDS_CANONICALONLY`, which means "I just want to see if they represent the same object. Don't worry about getting the sort order absolutely right." And again, since we don't use any methods on the object other than `IUnknown::QueryInterface`, we weaken the parameter requirements to simply `IUnknown`.

Now, this code could be simplified or at least tweaked to take advantage of `IShellItem`. For example, we could use `SHGetKnownFolderItem` to get the `FOLDERID_ComputerFolder` and then use `IShellItem::Compare`.

But I'm not going to bother, because there is an underlying algorithmic problem with this technique: It is checking whether you have a folder to *My Computer* specifically at its default location. If somebody creates a *My Computer* folder at a custom location, say via a folder shortcut, or a folder with a magic name, then this code will not recognize it as *My Computer* because these alternate locations for *My Computer* will not match the standard location.

If you want to identify *My Computer* no matter where it winds up, then instead of checking the path, you can check its class.

```
HRESULT GetObjectCLSID(IUnknown *punk, CLSID *pclsid)
{
  *pclsid = CLSID_NULL;
  IPersist *pp;
  HRESULT hr = punk->QueryInterface(IID_PPV_ARGS(&pp));
  if (SUCCEEDED(hr)) {
    hr = pp->GetClassID(pclsid);
    pp->Release();
  }
  return hr;
}
BOOL IsMyComputerFolder(IUnknown *punk)
{
  CLSID clsid;
  GetObjectCLSID(punk, &clsid);
  return clsid == CLSID_MyComputer;
}
```

We ask the object directly, “Hey, what’s your CLSID?” and if it replies, “I am `CLSID_MyComputer`,” then we say, “Pleased to meet you, *My Computer*.”

Raymond Chen

Follow