What's up with the registry key HKEY_CLASSES_ROOT\CLSID\CLSID?

devblogs.microsoft.com/oldnewthing/20131205-00

December 5, 2013

Raymond Chen

While I was looking for spelling errors in the registry, I ran across the strange registry key `HKEY_CLASSES_ROOT\CLSID\CLSID`, which has as its default value the stringified GUID `{0000031A-0000-0000-C000-000000000046}`. I asked the COM folks what the deal was with that.

They explained, "It's the progid for the class moniker."

That was enough to push me in the right direction, and it made my brain explode from the meta-ness of it all. Sort of like when you study Category Theory in math class.

If you want to connect a COM class ID with a programmatic ID, you cross-link them like this:

```
[HKEY_CLASSES_ROOT\CLSID\{clsid}\ProgId]
@="progid"

[HKEY_CLASSES_ROOT\progid\CLSID]
@="{clsid}"
```

The class ID is a child of `HKEY_CLASSES_ROOT\CLSID`, and it in turn has a `ProgId` subkey whose value is the programmatic ID. Meanwhile, the programmatic ID is a child of `HKEY_CLASSES_ROOT`, and it in turn has a `CLSID` subkey whose value is the class ID.

Here's where things get weird: The programmatic ID for the class moniker is `CLSID`.

Plug that into the above formula, and you get

```
[HKEY_CLASSES_ROOT\CLSID\{0000031A-0000-0000-C000-000000000046}\ProgId]
@="CLSID"

[HKEY_CLASSES_ROOT\CLSID\CLSID]
@="{0000031A-0000-0000-C000-000000000046}"
```

So it's all legit. It just looks weird.

It’s like that trick on the 80386 where you set a page directory entry to point back to the page directory itself, thereby allowing you to access page tables through the page table.

Bonus chatter: It frustrates me when I answer a question by merely giving a push in the right direction, and the person doesn’t realize that it was a helpful push and just comes back looking for more hand-holding. I try not to be that person.

Raymond Chen

Follow