

Some other places atoms (and the magical 0xC000) arise

 devblogs.microsoft.com/oldnewthing/20080430-00

April 30, 2008



Raymond Chen

The moment the Windows developers got a system for converting strings into numbers, they could use it anywhere they need to, well, convert a string into a number. Sometimes these integers are officially declared as atoms, but most of the time they are just integers that happen to be atoms under the covers. I'll start with registered window messages, created by the `RegisterWindowMessage` function. These are not officially atoms; they are just integers that happen to lie in the range `0xC000` to `0xFFFF`, just like atoms. But yeah, internally, they're atoms. Of course, you shouldn't rely on it since it's not contractual. Think of it as a fantastic coincidence. Registered clipboard formats created by the `RegisterClipboardFormat` message are also not officially atoms; they're just `UINT`s. The numeric range for registered clipboard formats isn't even specified; that they hang out in the `0xC000` range is just an implementation detail. Someday, registered clipboard formats may have values like `0x1234`, who knows. Window properties are also stored in the form of atoms, but unlike the other examples above, the atomic nature of window properties is contractual. You can set a property either by passing the property name `SetProp(hwnd, TEXT("PropertyName"), hData)` or by passing the property atom `SetProp(hwnd, MAKEINTATOM(atm), hData)`, where `atm` was obtained from an earlier call to `GlobalAddAtom`. There is additional weirdness with the way these atoms are tracked, which I'll defer to Friday's article, though it is hinted at in the documentation for `SetProp` which cautions that you need to remove all the properties from a window before it is destroyed.

Window classes also have atoms. The return value of the `RegisterClass` function is an `ATOM`, and you can also retrieve the atom later by calling `GetClassWord(GCW_ATOM)`. We'll see more about that atom next time.

[Raymond Chen](#)

Follow

