You didn't know you could add properties by atom, and it's a good thing you didn't know

devblogs.microsoft.com/oldnewthing/20080502-00

May 2, 2008

Raymond Chen

As I noted a few days ago, there is weirdness associated with properties added by atom. This weirdness stems from the fact that adding properties by atom is really a hole in the original implementation rather than something designed on purpose.

The original 16-bit code for adding and removing properties went roughly like this:

```
BOOL SetProp(HWND hwnd, LPSTR pszName, HANDLE hValue)
{
    ... let's look only at the part that adds a new property ...
    ATOM atm = HIWORD(pszName) ? GlobalAddAtom(pszName) : LOWORD(pszName);
    if (atm == 0) return FALSE;
    ... add the atom "atm" to the property list ...
}
HANDLE RemoveProp(HWND hwnd, LPSTR pszName)
{
    ATOM atm = HIWORD(pszName) ? GlobalFindAtom(pszName) : LOWORD(pszName);
    if (atm == 0) return NULL;
    ... look for the atom "atm" in the property list and remove it ...
    if (!found) return NULL;
    // clean up the atom
    if (HIWORD(pszName)) GlobalDeleteAtom(atm);
}
void CleanPropertiesWhenWindowIsDestroyed(HWND hwnd)
{
    for (each property on the window) {
        if (atm >= MAXINTATOM) GlobalDeleteAtom(atm);
    }
    .. delete memory used for recording properties ...
}
```

First, let’s look at properties set and removed via integer atoms. These are simple: When setting the property, we just add it to the property list, and when removing the property, we remove it. Nothing fancy going on here.

Similarly, there's nothing particularly exciting going on if a property is set and removed by name. When setting the property, we use `GlobalAddAtom` to convert the string to an atom (incrementing the reference count), and when removing it, we use `GlobalDeleteAtom` to clean it up (decrementing the reference count and removing the atom if the reference count goes to zero).

Finally, when a window is destroyed with outstanding properties, we clean them up by calling `GlobalDeleteAtom` on all the string atoms, counteracting the `GlobalAddAtom` we performed when we added the property.

So what's the big deal? Looks great, right?

See if you can find the hole in this implementation.

Hint 1: There are actually *three* ways of adding and removing properties from a window, not the two I led you to believe. <!– You can use an integer atom (one whose numerical value is less than `MAXINTATOM`), you can use a string atom (an atom whose numerical value is between `MAXINTATOM` and `0xFFFF`), or you can use a string. –>

Hint 2: What happens if you mix and match these three methods?

Hint 3: What happens to each of the three types of properties when the window manager is forced to clean them up?

These problems with properties were fixed a long time ago, but old-timers remain wary of adding named properties by string atom. It's one of those superstitions.

Raymond Chen

Follow