

# How to query properties of the taskbar

 [devblogs.microsoft.com/oldnewthing/20050506-32](http://devblogs.microsoft.com/oldnewthing/20050506-32)

May 6, 2005



Raymond Chen

Occasionally, people want to query properties of the taskbar. I don't quite understand why; you should just get on with your life and let the taskbar get on with its life. After all, there might not even be a taskbar, as we discussed last time.

But if you really want to know (perhaps you're collecting usability data), here's how:

```
#include <stdio.h>
#include <windows.h>
#include <shellapi.h>
int __cdecl main(int argc, const char* argv[])
{
    APPBARDATA abd = { sizeof(abd) };
    UINT uState = (UINT)SHAppBarMessage(ABM_GETSTATE, &abd);
    printf("Taskbar on top? %s\n",
        (uState & ABS_ALWAYSONTOP) ? "yes" : "no");
    printf("Taskbar autohide? %s\n",
        (uState & ABS_AUTOHIDE) ? "yes" : "no");
    return 0;
}
```

This little program uses the ABM\_GETSTATE message of the SHAppBarMessage function to get the “Always on top” and “Auto-hide” properties of the taskbar.

Since you're using the `SHAppBarMessage` function, if a future version of Windows changes the way it maintains the taskbar state (or perhaps even changes the name of the taskbar to something else), your program will still work because the `SHAppBarMessage` function will be kept in sync with whatever changes happen to the taskbar.

You can also use the ABM\_SETSTATE message to change these states. Note that doing so from a program is discouraged; these state bits belong to the user's preferences. A program shouldn't be messing with the user's preferences. (Well, unless the whole point of the program is to change the user's preferences, of course. But the frequency with which I see this question makes me wonder whether there really are that many settings-tweaking programs out there. I suspect people are using this power for evil, not for good.)

And to stave off follow-up questions: These are the only two properties of the taskbar that are programmable. Exposing a programmable interface for something as highly visible as the taskbar is a very sensitive issue, because once you grant programmatic access to something, there is a very strong temptation for programs to start abusing it.



Raymond Chen

**Follow**