

How can I write a program that monitors another window for a change in size or position?

 devblogs.microsoft.com/oldnewthing/20210104-00

January 4, 2021



Raymond Chen

A few days ago, a reader bemoaned, “There’s no way to update the position/aspect ratio of live window thumbnails unless you poll.”

Today’s Little Program monitors another window for a change in its size and position, without polling. It’s basically another variation on the basic “window monitoring” pattern. This time, instead of monitoring the title, we monitor the location (which is the combination of size, position, and shape).

```

#define UNICODE
#define _UNICODE
#define STRICT
#include <windows.h>
#include <stdio.h>

HWND g_hwndMonitor;

void CALLBACK WinEventProc(
    HWINEVENTHOOK hook,
    DWORD event,
    HWND hwnd,
    LONG idObject,
    LONG idChild,
    DWORD idEventThread,
    DWORD time)
{
    if (hwnd == g_hwndMonitor &&
        idObject == OBJID_WINDOW &&
        idChild == CHILDID_SELF &&
        event == EVENT_OBJECT_LOCATIONCHANGE) {
        RECT rc;
        if (GetWindowRect(g_hwndMonitor, &rc)) {
            printf("window rect is (%d,%d)-(%d,%d)\n",
                rc.left, rc.top, rc.right, rc.bottom);
        }
    }
}

int __cdecl main(int, char**)
{
    g_hwndMonitor = FindWindow(L"Awesome Program", nullptr);
    DWORD processId;
    DWORD threadId = GetWindowThreadProcessId(g_hwndMonitor, &processId);
    HWINEVENTHOOK hook = SetWinEventHook(
        EVENT_OBJECT_LOCATIONCHANGE,
        EVENT_OBJECT_LOCATIONCHANGE,
        nullptr,
        WinEventProc,
        processId,
        threadId,
        WINEVENT_OUTOFCONTEXT);
    MessageBox(nullptr, L"Press OK when bored", L"Title", MB_OK);

    UnhookWinEvent(hook);
    return 0;
}

```

The program starts by identifying the window it wants to monitor. This is your application's business logic, so I'll just fake it with a `Find Window` .

We get the thread and process ID for the window and use it to register a thread-specific accessibility event hook, filtered to location changes.

In the event callback, we see if the notification is for the window we are monitoring. If so, we get the window bounds and print it. The attempt to get the window bounds might fail if the window has been destroyed, so watch out for that.¹

¹ One way to track object destruction is with, yup, another accessibility hook, this time watching for `EVENT_OBJECT_DESTROY`.

Raymond Chen

Follow

