

How can I write a program that monitors another window for a title change?

 devblogs.microsoft.com/oldnewthing/20180601-00

June 1, 2018



Raymond Chen

A customer was writing a monitoring application and wanted to be notified if a window's title changes.

Sure, we can use accessibility to do that.

```

#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_NAMECHANGE) {
        printf("title changed\n");
    }
}

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_NAMECHANGE,
        EVENT_OBJECT_NAMECHANGE,
        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. Presumably the customer will use some domain-specific knowledge to find the window, but here, we'll just demonstrate with the `Find Window` function.

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

In the event callback, we see if the notification is for the window we are monitoring. If so, we print a message. The customer's program would presumably do something more interesting than just print a message.

Raymond Chen

Follow

