

Adding a sound to the Alt+Tab window

 devblogs.microsoft.com/oldnewthing/20140623-00

June 23, 2014



Raymond Chen

Today's Little Program plays a sound when the **Alt + Tab** window appears and disappears.

```

#define STRICT
#include <windows.h>
#include <mmsystem.h>
HWND g_hwndAltTab = nullptr;
void CALLBACK WinEventProc(
    HWINEVENTHOOK hWinEventHook,
    DWORD event,
    HWND hwnd,
    LONG idObject,
    LONG idChild,
    DWORD dwEventThread,
    DWORD dwmsEventTime
)
{
    PCTSTR pszSound = nullptr;
    switch (event) {
    case EVENT_SYSTEM_SWITCHSTART:
        if (!g_hwndAltTab) {
            g_hwndAltTab = hwnd;
            pszSound = "C:\\Windows\\Media\\Speech on.wav";
        }
        break;
    case EVENT_SYSTEM_SWITCHEND:
        if (g_hwndAltTab) {
            g_hwndAltTab = nullptr;
            pszSound = "C:\\Windows\\Media\\Speech sleep.wav";
        }
        break;
    }
    if (pszSound) {
        PlaySound(pszSound, nullptr, SND_FILENAME | SND_ASYNC);
    }
}
int WINAPI WinMain(HINSTANCE hinst, HINSTANCE hinstPrev,
    LPSTR lpCmdLine, int nShowCmd)
{
    HWINEVENTHOOK hWinEventHook = SetWinEventHook(
        EVENT_SYSTEM_SWITCHSTART, EVENT_SYSTEM_SWITCHEND,
        nullptr, WinEventProc, 0, 0,
        WINEVENT_OUTOFCONTEXT | WINEVENT_SKIPOWNPROCESS);
    if (hWinEventHook) {
        MessageBox(nullptr, "Close this window when sufficiently annoyed.",
            "Hello", MB_OK);
        UnhookWinEvent(hWinEventHook);
    }
    return 0;
}

```

The program registers an accessibility event hook for the `EVENT_SYSTEM_SWITCHSTART` and `EVENT_SYSTEM_SWITCHEND` events. The Start event fires when an **Alt + Tab** operation begins, and the End event fires when an **Alt + Tab** operation completes. As noted in the

documentation, you can get spurious End events, so we keep track of our current state to avoid any surprises.

In addition to adding an annoying sound to the **Alt** + **Tab** window, let's also add an annoying sound each time you move focus to a new item.

```

HWINEVENT g_hWinEventHookFocus = nullptr;
void CALLBACK WinEventProc(
    HWINEVENTHOOK hWinEventHook,
    DWORD event,
    HWND hwnd,
    LONG idObject,
    LONG idChild,
    DWORD dwEventThread,
    DWORD dwmsEventTime
)
{
    PCTSTR pszSound = nullptr;
    switch (event) {
    case EVENT_SYSTEM_SWITCHSTART:
        if (!g_hwndAltTab) {
            g_hwndAltTab = hwnd;
            g_hWinEventHookFocus = SetWinEventHook(
                EVENT_OBJECT_FOCUS, EVENT_OBJECT_FOCUS,
                NULL, WinEventProc, 0, 0,
                WINEVENT_OUTOFCONTEXT | WINEVENT_SKIPOWNPROCESS);
            pszSound = "C:\\Windows\\Media\\Speech on.wav";
        }
        break;
    case EVENT_SYSTEM_SWITCHEND:
        if (g_hwndAltTab) {
            g_hwndAltTab = nullptr;
            UnhookWinEvent(g_hWinEventHookFocus);
            g_hWinEventHookFocus = nullptr;
            pszSound = "C:\\Windows\\Media\\Speech sleep.wav";
        }
        break;
    case EVENT_OBJECT_FOCUS:
        if (hwnd == g_hwndAltTab) {
            pszSound = TEXT("C:\\Windows\\Media\\Speech Misrecognition.wav");
        }
        break;
    }
    if (pszSound) {
        PlaySound(pszSound, nullptr, SND_FILENAME | SND_ASYNC);
    }
}

int WINAPI WinMain(HINSTANCE hinst, HINSTANCE hinstPrev,
    LPSTR lpCmdLine, int nShowCmd)
{
    HWINEVENTHOOK hWinEventHook = SetWinEventHook(
        EVENT_SYSTEM_SWITCHSTART, EVENT_SYSTEM_SWITCHEND,
        nullptr, WinEventProc, 0, 0,
        WINEVENT_OUTOFCONTEXT | WINEVENT_SKIPOWNPROCESS);
    if (hWinEventHook) {
        MessageBox(nullptr, "Close this window when sufficiently annoyed.",
            "Hello", MB_OK);
        UnhookWinEvent(hWinEventHook);
    }
}

```

```
if (g_hWinEventHookFocus) {  
    UnhookWinEvent(g_hWinEventHookSelect);  
}  
}  
return 0;  
}
```

Okay, this was a pretty annoying program, but maybe you can use it for something better.

Raymond Chen

Follow

