How can I get notified when the cursor changes?

devblogs.microsoft.com/oldnewthing/20151116-00

November 16, 2015

Raymond Chen

Today's Little Program tracks changs to the cursor. You might want to do this as part of instrumentation, in order to see how often the user is staring at an hourglass, for example. (It's easier to make the Move cursor appear on demand, so I'll use that instead.)

The magic words here are `OBJID_CURSOR` and `GetCursorInfo`.

```
#include <windows.h>
#include <stdio.h>

void log()
{
  CURSORINFO ci = { sizeof(ci) };
  GetCursorInfo(&ci);
  printf("showing = %d, suppressed = %d, pos = (%d, %d), handle = %p\n",
    !!(ci.flags & CURSOR_SHOWING),
    !!(ci.flags & CURSOR_SUPPRESSED),
    ci.ptScreenPos.x,
    ci.ptScreenPos.y,
    ci.hCursor);
}
```

The `log` function prints information about the current cursor. For now, we just dump it to the screen, but obviously you could do something fancier with it. The `CURSOR_SHOWING` flag tells you whether the cursor show count is nonnegative, which is what classically controls whether the cursor is visible on the screen. The `CURSOR_SUPPRESSED` flag tells tells you that nominally visible cursor is not visible to the user because the user touched the screen with a finger or pen.

```
void CALLBACK WinEventProc(
  HWINEVENTHOOK hook,
  DWORD event,
  HWND hwnd,
  LONG idObject,
  LONG idChild,
  DWORD idEventThread,
  DWORD time)
{
  if (hwnd == nullptr &&
      idObject == OBJID_CURSOR &&
      idChild == CHILDID_SELF) {
    switch (event) {
    case EVENT_OBJECT_HIDE:
      printf("cursor hidden\n");
      log();
      break;
    case EVENT_OBJECT_SHOW:
      printf("cursor shown\n");
      log();
      break;
    case EVENT_OBJECT_NAMECHANGE:
      printf("cursor changed\n");
      log();
      break;
    }
  }
}
```

Our event hook procedure checks if we're being notified about the cursor. If so, then we print some information about the event we received, and then log the cursor details.

```
int __cdecl main(int, char**)
{
  printf("Move cursor = %p\n", LoadCursor(nullptr, IDC_SIZEALL));

  HWINEVENTHOOK hook = SetWinEventHook(
    EVENT_OBJECT_SHOW,
    EVENT_OBJECT_NAMECHANGE,
    nullptr,
    WinEventProc,
    GetCurrentProcessId(),
    GetCurrentThreadId(),
    WINEVENT_OUTOFCONTEXT);

  MessageBox(nullptr, TEXT("Press Ok when bored"),
             TEXT("Title"), MB_OK);

  UnhookWinEvent(hook);
  return 0;
}
```

Our main program prints the handle of the Move cursor, just to demonstrate that the handle will match the output. Next, it installs the event hook on its own process and thread. (If you want to monitor the entire process, then pass 0 for the thread ID. If you wanted to monitor all processes on the desktop, then pass 0 for both the process ID and thread ID.) Next, we display a message box to give you a way to exit the program, and to fire up a message pump. After you are bored, we remove the hook and exit.

Now, I chose the Move cursor because it is pretty much the only cursor you can get to from a message box: Press `Alt`+`Space`, then hit `M` for Move. Bingo, a Move cursor. And you can see the program spit out the new cursor handle, and it should match the value printed at the start of the program.

Raymond Chen

Follow