

# A timed context menu

 [devblogs.microsoft.com/oldnewthing/20050307-00](http://devblogs.microsoft.com/oldnewthing/20050307-00)

March 7, 2005



Raymond Chen

This is sort of in the same spirit as our previous exercise in writing a timed message box, but this is much easier. Here, we use the handy-dandy [WM\\_CANCELMODE message](#) to get us out of menu mode.

```
void CALLBACK
MenuTooLateProc(HWND hwnd, UINT uiMsg, UINT idEvent, DWORD dwTime)
{
    SendMessage(hwnd, WM_CANCELMODE, 0, 0);
}
BOOL
TimedTrackPopupMenuEx(HMENU hMenu, UINT uFlags, int x, int y,
    HWND hwnd, LPTMPARAMS pTpm, DWORD dwTimeout)
{
    UINT idTimer = SetTimer(NULL, IDT_TOOLATE, dwTimeout, MenuTooLateProc);
    BOOL fResult = TrackPopupMenuEx(hMenu, uFlags, x, y, hwnd, pTpm);
    if (idTimer) KillTimer(NULL, idTimer);
    return fResult;
}
```

Before displaying the menu, we set a timer. (And we use a thread timer because we don't own the `hwnd` window and therefore don't know what timer IDs are safe to use.) If the timer fires, we send ourselves a `WM_CANCELMODE` message to cancel menu mode. This causes the system to act as if the user had dismissed the menu without selecting anything, either by hitting `ESC` or clicking outside the menu. The call to [the `TrackPopupMenuEx` function](#) returns once the user has selected something (or the timeout has elapsed), at which point we clean up by destroying our timer before returning.

[Raymond Chen](#)

**Follow**

