

Taxes: Detecting session state changes, such as a locked workstation

 devblogs.microsoft.com/oldnewthing/20060104-50

January 4, 2006



Raymond Chen

Another developer tax is playing friendly with Fast User Switching and Terminal Services. When the workstation is locked or disconnected, you should turn off non-essential timers, minimize background activities, and generally send your program into a quiet state. If you already used the technique of painting only when your window is visible on the screen, then you get all of this for free, since a locked workstation and a disconnected session do not generate paint messages.

If you have other activities that you need to scale back or shut down when the user has locked the workstation or disconnected, you can register to be notified when the state changes. Knowing about these state changes is also important so that you can tell when your display is local or remote. As we saw last time, drawing on Remote Desktop Connection is much slower than on a local display, since all the bitmaps need to be transferred over the network to the Remote Desktop client.

Since locking a workstation and disconnecting a session remove the ability to use visual feedback to indicate our program's state, we'll use the speaker. Start with our new scratch program and make the following changes:

```

#include <wtsapi32.h>
LRESULT RootWindow::OnCreate()
{
    WTSRegisterSessionNotification(m_hwnd, NOTIFY_FOR_THIS_SESSION);
    return 0;
}
LRESULT RootWindow::HandleMessage(
    UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    switch (uMsg) {
        ...
    case WM_WTSSESSION_CHANGE:
        switch (wParam) {
            case WTS_CONSOLE_DISCONNECT:
            case WTS_REMOTE_DISCONNECT:
            case WTS_SESSION_LOCK:
            case WTS_SESSION_LOGOFF:
                Beep(440, 250); break;
            case WTS_CONSOLE_CONNECT:
            case WTS_REMOTE_CONNECT:
            case WTS_SESSION_UNLOCK:
            case WTS_SESSION_LOGON:
                Beep(880, 250); break;
        }
        break;
        ...
    }
}

```

In this program, we register for a session notification when we create our main window, and listen for the session change message in our window procedure. If we see one of the “going away” messages, we make a low beep; if we see one of the “coming back” messages, we make a high beep.

Run this program and then lock the workstation or use Fast User Switching to switch away. You should be greeted by a low beep (though you may have to listen carefully if you have a sound associated with the action you performed because the low beep will be mixed in with it). When you switch back, you should hear a high beep.

Of course, a real program would respond to the notifications by starting or stopping its background tasks. The purpose of this program was merely to show how to get the notifications in the first place.

Raymond Chen

Follow

