[DllImport("user32.dll", SetLastError = true)]

static extern System.IntPtr OpenInputDesktop(uint dwFlags, bool fInherit, uint dwDesiredAccess);

Timer

var handle = OpenInputDesktop(0, false, 0);

string ss = handle.ToString();

if (ss == "0") ss = "Locked";

using Microsoft.Win32;

Con un timer y

string ss = SessionSwitchReason.SessionLock.ToString();

using Microsoft.Win32;

Constrctor()

{

SystemEvents.SessionSwitch += new SessionSwitchEventHandler(SystemEvents\_SessionSwitch);

}

~Constrctor()

{

//Do this during application close to avoid handle leak

Microsoft.Win32.SystemEvents.SessionSwitch –

= new SessionSwitchEventHandler(SystemEvents\_SessionSwitch);

}

void SystemEvents\_SessionSwitch(object sender, SessionSwitchEventArgs e)

        {

            switch (e.Reason)

            {

                case SessionSwitchReason.SessionLogon:

                case SessionSwitchReason.SessionUnlock:

                    break;

                case SessionSwitchReason.SessionLock:

                case SessionSwitchReason.SessionLogoff:

                    break;

            }

        }

using Microsoft.Win32;

public void FooBar()

{

SystemEvents.SessionSwitch += SystemEvents\_SessionSwitch;

SystemEvents.SessionSwitch -= SystemEvents\_SessionSwitch;

}

public void SystemEvents\_SessionSwitch(object sender, SessionSwitchEventArgs e)

{

if (e.Reason == SessionSwitchReason.SessionLock)

{

// Do what you need to here as the system is locked

}

}