

Re: Application monitoring/key logging

Source:

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2006-01/msg01546.html>

- *From:* "Rob Kings" <greeneggsandham@xxxxxxxxxxxxxxxx>
 - *Date:* Tue, 17 Jan 2006 23:19:42 -0000
-

DanS

Thanks for the comprehensive reply. I must confess I was rather hoping for suggestions of a tool rather than VB code, since then I could use it to test and log multiple applications, but I'll take a look and see whether I can apply it.

Thanks

"DanS" <t.h.i.s.n.t.h.a.t@xxxxxxxxxxxxxxxx> wrote in message news:Xns974EAA120EF31idispcom@xxxxxxxxxxxxxxxx
> "Rob Kings" <greeneggsandham@xxxxxxxxxxxxxxxx> wrote in
> news:4353jkF1k2h46U1@xxxxxxxxxxxxxxxx:

>
>> I don't know whether this classes as OT or not, but here goes anyway.
>> I'm looking for a keyboard logger, but it needs to capture mouse
>> activity as well. I want it for testing some software. Occasionally we
>> get a crash, but only after a long session of complex activity. What
>> I'd like to do is run some background task and then after my App has
>> hit its problem I'd have a full log of everything I'd done upto that
>> point, and hence be able to try to reproduce it.
>>
>> Most of the logging tools out there (that I've found) seem to:
>>
>> a) Be written from the point of view of company/parent security (They
>> all have stealth modes so that the person using them doesn't know they
>> are running)
>>
>> b) Capture text only (i.e. When I use my app I see very little logged
>> after my password)
>>
>> c) Are frankly pretty crap.
>>
>> Any ideas or suggestions?
>>
>> Cheers
>>

Re: Application monitoring/key logging

>> Rob

>>

>>

>

> you can implement system-wide keyboard and mouse hooks in VB under XP (2k
> ?) using SetWindowsHook and the WH_MOUSE_LL and WH_KEYBOARD_LL hooks.

>

> If you can copy and paste this (I'm sure to be) horribly wrapped text,
> add it to a new module. It is very simple to use, call
> StartMouseHook/StopMouseHook, or StartKeyboardHook/StopKeyboardHook.

>

> Look in the module for 'KeyboardProc' and 'MouseProc'. The mouse hook has
> a Select/Case for the mouse events hooked, and the keyboard does a
> debug.print of the keycode.

>

> Using a simple hook like this it should be very easy to log to file what
> ever you deem necessary, with or without writing tons more code.

>

>

> Regards,

>

> DanS

>

>

> ----Start Code-----

> Option Explicit

>

> Public Const WH_KEYBOARD_LL = 13

> Private Const WH_MOUSE_LL As Long = 14

>

> Private Const HC_ACTION As Integer = 0

>

> Private Const WM_MOUSEMOVE As Integer = &H200

> Private Const WM_LBUTTONDOWN As Integer = &H201

> Private Const WM_LBUTTONUP As Integer = &H202

> Private Const WM_LBUTTONDBLCLK As Integer = &H203

> Private Const WM_RBUTTONDOWN As Integer = &H204

> Private Const WM_RBUTTONUP As Integer = &H205

> Private Const WM_RBUTTONDBLCLK As Integer = &H206

> Private Const WM_MBUTTONDOWN As Integer = &H207

> Private Const WM_MBUTTONUP As Integer = &H208

> Private Const WM_MBUTTONDBLCLK As Integer = &H209

> Private Const WM_MOUSEWHEEL As Integer = &H20A

>

>

>

> Private Type KBDLLHOOKSTRUCT

> vkCode As Long ' virtual key code

> scanCode As Long ' scan code

> flags As Long ' flags

> time As Long ' time stamp for this message

Re: Application monitoring/key logging

Re: Application monitoring/key logging

```
> dwExtraInfo As Long ' extra info from the driver or keybd_event
> End Type
>
> Private Type Point
> x As Long
> y As Long
> End Type
>
> Private Type MSLLHOOKSTRUCT
> pt As Point
> mouseData As Integer
> flags As Integer
> time As Integer
> dwExtraInfo As Integer
> End Type
>
> Public Declare Function SetWindowsHookEx Lib "user32" Alias
> "SetWindowsHookExA" (ByVal idHook As Long, ByVal lpfn As Long, ByVal hmod
> As Long, ByVal dwThreadId As Long) As Long
> Public Declare Function UnhookWindowsHookEx Lib "user32" (ByVal hHook As
> Long) As Long
>
> Private Declare Function CallNextHookEx Lib "user32" (ByVal hHook As
> Long, ByVal nCode As Long, ByVal wParam As Long, lParam As Any) As Long
> Private Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory"
> (pDest As Any, pSrc As Any, ByVal ByteLen As Long)
>
> Private kb_struct As KBDLLHOOKSTRUCT
> Private mouse_struct As MSLLHOOKSTRUCT
>
> Global kbd_Hook As Long
> Global mouse_Hook As Long
>
> Public Function startKeyboardHook() As Boolean
>
> kbd_Hook = SetWindowsHookEx(WH_KEYBOARD_LL, AddressOf KeyboardProc,
> App.hInstance, ByVal 0&)
> If kbd_Hook <> 0 Then
> startKeyboardHook = True
> End If
>
> End Function
>
> Public Sub stopKeyboardHook()
> UnhookWindowsHookEx kbd_Hook
> End Sub
> Public Function KeyboardProc(ByVal nCode As Long, ByVal wParam As Long,
> ByVal lParam As Long) As Long
> If nCode = HC_ACTION Then
> CopyMemory kb_struct, ByVal lParam, LenB(kb_struct)
> Debug.Print kb_struct.scanCode
```

Re: Application monitoring/key logging

```
> End If
> KeyboardProc = CallNextHookEx(kbd_Hook, nCode, wParam, lParam)
> End Function
>
> Public Function startMouseHook() As Boolean
> mouse_Hook = SetWindowsHookEx(WH_MOUSE_LL, AddressOf MouseProc,
> App.hInstance, ByVal 0&)
> If mouse_Hook <> 0 Then
> startMouseHook = True
> End If
> End Function
>
> Public Sub stopMouseHook()
> UnhookWindowsHookEx mouse_Hook
> End Sub
>
> Public Function MouseProc(ByVal nCode As Long, ByVal wParam As Long,
> ByVal lParam As Long) As Long
> If nCode = HC_ACTION Then
> CopyMemory mouse_struct, ByVal lParam, LenB(mouse_struct)
> Select Case wParam
> Case WM_MOUSEMOVE
>
> Case WM_LBUTTONDOWN
>
> Case WM_LBUTTONUP
>
> Case WM_LBUTTONDBLCLK
>
> Case WM_RBUTTONDOWN
>
> Case WM_RBUTTONUP
>
> Case WM_RBUTTONDBLCLK
>
> Case WM_MBUTTONDOWN
>
> Case WM_MBUTTONDBLCLK
>
> Case WM_MOUSEWHEEL
> End Select
> End If
> MouseProc = CallNextHookEx(mouse_Hook, nCode, wParam, lParam)
> End Function
>
> ----End of Code-----
```

.

Re: Application monitoring/key logging

- **References:**

- ◆ **Application monitoring/key logging**

- ◇ From: Rob Kings

- ◆ **Re: Application monitoring/key logging**

- ◇ From: DanS

- Prev by Date: **Re: Application monitoring/key logging**
- Next by Date: **Re: "The Publisher could not be verified. etc."**
- Previous by thread: **Re: Application monitoring/key logging**
- Next by thread: **Re: Application monitoring/key logging**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**