
Subject: Re: How to set a global (system) keyboard hook?

Posted by [tojocky](#) on Fri, 18 Sep 2009 14:03:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

About global hotkey!

FOR WIN32:

acourding by simple example from

[http://msdn.microsoft.com/en-us/library/ms646309\(VS.85\).aspx?ppud=4](http://msdn.microsoft.com/en-us/library/ms646309(VS.85).aspx?ppud=4)

first we need to register hot key by calling:

```
BOOL RegisterHotKey(  
    HWND hWnd,  
    int id,  
    UINT fsModifiers,  
    UINT vk  
);
```

and in U++ is already created a thread that execute function DWORD WINAPI Ctrl::Win32OverwatchThread(LPVOID). In this function is:

```
DWORD WINAPI Ctrl::Win32OverwatchThread(LPVOID)  
{  
    .....  
    MSG Msg;  
=>>HERE while(GetMessage(&Msg, NULL, 0, 0) > 0) {  
    TranslateMessage(&Msg);  
==>>HERE we can process our keyboard event  
    if(IsWindowUnicode(Msg.hwnd))  
        DispatchMessageW(&Msg); // send to aprpopiate pthead  
    else  
        DispatchMessage(&Msg);  
    }  
    .....  
}
```

The problem is that I do not get keycode property from Msg. From where can I get it?

for linux X11:

acording by documentation from:

1. <http://www.xfree.org/current/XGrabKey.3.html#toc2>

The simple code like this:

Window root;

```

XEvent ev;

Display * dpy = XOpenDisplay(0);

if(!dpy) return 1;

root = DefaultRootWindow(dpy);

char * key_string = "F3";

KeyCode key = XKeysymToKeycode(dpy, XStringToKeysym(key_string));

XGrabKey(dpy, key , AnyModifier, root,
        True, GrabModeAsync, GrabModeAsync);

for(;;)
{
    XNextEvent(dpy, &ev);
    if(ev.type == KeyPress && ev.xkey.keycode == key){
        ==>> Do our action
    }
}

```

First we need register global hotkey by calling:

```

int XGrabKey(Display *display, int keycode, unsigned int
modifiers, Window grab_window, Bool owner_events, int pointer_mode, int keyboard_mode);

```

In u++ we can get message from void Ctrl::EventLoop0(Ctrl *ctrl)

which call void Ctrl::ProcessEvent(XEvent *event)

The good news is that I have keycode from the returned message. like this "ev.xkey.keycode".

The Question:

2. Is possible to add in u++ cross-os method for register global hotkey event?

For first I propose like this:

```

static bool Ctrl::AddGlobalHotkey(Key p_key, callback
p_func)

```

2. If is possible for register global hotkey event, maybe it is possible to unregister global hotkey event.

Maybe I was not understandable.

UPDATE:

Fro linux I found interesting example:

```
void
xstuff_grab_key_on_all_screens (int    keycode,
    guint  modifiers,
    gboolean grab)

{
    GdkDisplay *display;
    int        n_screens;
    int        i;

    display = gdk_display_get_default ();
    n_screens = gdk_display_get_n_screens (display);

    for (i = 0; i < n_screens; i++) {
        GdkWindow *root;

        root = gdk_screen_get_root_window (
            gdk_display_get_screen (display, i));

        if (grab)
            XGrabKey (gdk_x11_display_get_xdisplay (display),
                keycode, modifiers,
                gdk_x11_drawable_get_xid (root),
                True, GrabModeAsync, GrabModeAsync);
        else
            XUngrabKey (gdk_x11_display_get_xdisplay (display),
                keycode, modifiers,
                gdk_x11_drawable_get_xid (root));
    }
}
```

UPDATE 2:

Good news is for win32 too. It returns keycode in Mgs->IParam.

Ion Lupascu (tojocky).
