

Akshat Kapoor

From: Mervyn Bick [bickmeo@gmail.com]
Sent: Thursday, July 12, 2018 20:47
To: Akshat Kapoor
Subject: Re: Changing from plus 10 to plus 11

Hi Akshat,

You didn't disturb me at all. :-) I'm glad you got your forms working in dBASE 11. There shouldn't be a problem with the keyboard hook if you ever migrate to dBASE 12.

The CALLBACK code in the form's onOpen event handler installs a keyboard hook which grabs all keystrokes meant for the form and makes them available to you in the function before they are passed on to the form. If necessary the monitor function in your form (hookWndproc()) can "swallow" a keystroke so that it doesn't get passed back to Windows. The keystroke can then be used to execute some code in your form

In the original example the intention was to allow the operator to use the numeric keypad for both data entry and, by holding down the Alt key, turn the numeric keys into "hot" keys. With the Alt key down hookWndproc() passes the keystroke to num_key_pressed() and return 1 to Windows. This tells Windows to drop the keystroke. With the Alt key up the keystroke is returned to Windows for further processing.

hookWndproc() sees both keydown and keyup messages. The key is identified by wParam and the high bit is set in lParam on keyup. For keydown this high bit is 0. Since dBASE 11 each keystroke returns hCode=0 and hCode=3 values together with the keydown and keyup values i.e 4 values for each keystroke. By only looking at hCode=0 and the keydown value duplicate actions are avoided.

The lParam value for any given key can also be found by using GetAsyncKeyState() and this is used to monitor the Alt key as it makes the code to test the Alt key less "messy".

You are using the code in a way it wasn't meant to be used. In the code that you've posted here, the NOT in ' if not bitset(GetAsyncKeyState(VK_MENU),31)' reverses the setting so that the code thinks the Alt key is permanently down. It is, therefore, not returning the values to Windows but is, instead, passing the keys to num_key_pressed(). Here you are keyboarding the key character. In other words you are doing something for Windows that Windows would have done anyway. :-)

If you want the numeric keypad to return the values pressed simply remove the NOT. In fact you don't need any of the code associated with the CALLBACK, hookWndproc() and num_key_pressed() if you are not using the numeric pad for hot keys.

Instead of using on key to set escape, pageup, pagedown, end and home as hot keys you could have used the keyboard hook as follows. It's a bit of overkill here as you are not using the numeric pad for hot keys and the on key works anyway but I've included it to try and help you follow the process.

```
function num_key_pressed(num_key)
    if num_key = 0x1b // VK_ESCAPE
        form.refresh_data()
    elseif num_key = 0x21 // VK_PRIOR pgup
        form.plus1()
    elseif num_key = 0x22 // VK_NEXT pgdn
        form.minus1()
    elseif num_key = 0x23 // VK_END
        form.minus10()
    elseif num_key = 0x24 // VK_HOME
        form.plus10()
    endif
    return

function hookWndproc(hCode, wParam, lParam)
    if (wParam = VK_ESCAPE and hCode = 0 and not
bitset(lparam,31)) or ;
        (wParam >= VK_PRIOR and wParam <= VK_HOME) and hcode =
0 and (not bitset(lparam,31))
        class::num_key_pressed(wParam)
        cRet = 1 //Hot key. Don't pass key press on to next hook for processing.
    else
        cRet = CallNextHookEx(this.hook, hCode, wParam, lParam) //Not a hot key. Process key press
    endif
    return cRet
```

Cheers,
Mervyn.

On 2018-07-12 7:50 AM, Akshat Kapoor wrote:

```
> Good morning Mervyn,
> Sorry for disturbing you
>
```

> Set escape off
>
> Solved the issue.
>
> Regards
> Akshat
>