
Subject: K_CTRL_SEMICOLON and K_CTRL_PERIOD keys share the same constant value on Win32

Posted by [Oblivion](#) on Fri, 04 Dec 2020 19:32:18 GMT

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

Hi,

As the title explains, the Ctrl+semicolon and Ctrl+Period keys share the same constant value and this results in error in Win32 if they are both used in the same switch statement.

See, Win32Keys.h, In: 118, 119:

```
K_CTRL_PERIOD = K_CTRL|0xbe|K_DELTA,  
K_CTRL_SEMICOLON = K_CTRL|0xbe|K_DELTA,
```

Best regards,
Oblivion

Subject: Re: K_CTRL_SEMICOLON and K_CTRL_PERIOD keys share the same constant value on Win32

Posted by [mirek](#) on Fri, 11 Dec 2020 15:14:00 GMT

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

Thanks, fixed.

Subject: Re: K_CTRL_SEMICOLON and K_CTRL_PERIOD keys share the same constant value on Win32

Posted by [Oblivion](#) on Mon, 14 Dec 2020 19:47:20 GMT

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

Hello Mirek,

There seems to be another problem with function key handling, specifically on X11:

In X11Proc.cpp, Key release the "keysym" variable (which is unsigned long) is either uninitialized or can return values > 64K. This messes the upper word of K_ keys where the keyflags reside and basically renders the modified function keys ([SHIFT]CTRL/ALT+FN) unusable on X11.

For example, CTRL + F5:

```
K_CTRL_F5|K_KEYUP = 0x13ffc2  
actual key       = 0xfffffc2
```

This seems to be a quick work-around, but unfortunately, I have little time now to investigate its possible side-effects (I believe, we don't use the upper word of keySYM at this point):

CtrlCore/X11Proc.cpp:270

```
keySYM &= 0xFFFF; <---- ADDED
if(keySYM >= '0' && keySYM <= '9' && (chr == 0 || GetCtrl() || GetAlt())) {
    DispatchKey(KEYtoK(keySYM - '0' + K_0)|up, count);
    return;
}
if(chr >= 1 && chr < 32) {
    DispatchKey(KEYtoK(chr - 1 + K_CTRL_A)|up, count);
    return;
}
if(keySYM >= 0xff80 && keySYM <= 0xffb9 && chr) {
    DispatchKey(KEYtoK(chr)|up, count);
    return;
}
if(keySYM >= 0xff00 && chr < 128 ||
    (GetCtrl() || GetAlt()) && keySYM >= 0x20 && keySYM < 0x7f) {
    if(keySYM >= 'a' && keySYM <= 'z')
        keySYM = keySYM - 'a' + 'A';
    DispatchKey(KEYtoK(keySYM|K_DELTA)|up, count);
    return;
}
```

Best regards,
Oblivion

Subject: Re: K_CTRL_SEMICOLON and K_CTRL_PERIOD keys share the same constant value on Win32

Posted by [Oblivion](#) on Mon, 14 Dec 2020 21:48:28 GMT

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

Below code seems to allow numpad, editpad, cursor keys with modifiers on X11:

```
static struct { KeySym keySYM; dword key; } tab[] = {
    { XK_ISO_Left_Tab, K_TAB|K_SHIFT },
    { XK_BackSpace,    K_BACKSPACE },
    { XK_Tab,          K_TAB      },
```

```

{ XK_Return,    K_ENTER  },
{ XK_KP_Enter,  K_ENTER  },
{ XK_Escape,    K_ESCAPE },
{ XK_space,     K_SPACE  },
{ XK_Pause,     K_BREAK  },
{ XK_Scroll_Lock, K_SCROLL },
{ XK_Home,      K_HOME   },
{ XK_End,       K_END    },
{ XK_Left,      K_LEFT   },
{ XK_Up,        K_UP     },
{ XK_Right,     K_RIGHT  },
{ XK_Down,      K_DOWN   },
{ XK_Prior,     K_PRIOR  },
{ XK_Next,      K_NEXT   },
{ XK_Page_Up,   K_PAGEUP },
{ XK_Page_Down, K_PAGEDOWN },
{ XK_Insert,    K_INSERT },
{ XK_KP_Space,  K_SPACE  },
{ XK_KP_Tab,    K_TAB    },
{ XK_KP_Enter,  K_ENTER  },
{ XK_KP_F1,     K_F1     },
{ XK_KP_F2,     K_F2     },
{ XK_KP_F3,     K_F3     },
{ XK_KP_F4,     K_F4     },
{ XK_KP_Home,   K_HOME   },
{ XK_KP_Left,   K_LEFT   },
{ XK_KP_Up,     K_UP     },
{ XK_KP_Right,  K_RIGHT  },
{ XK_KP_Down,   K_DOWN   },
{ XK_KP_Page_Up, K_PAGEUP },
{ XK_KP_Page_Down, K_PAGEDOWN },
{ XK_KP_End,    K_END    },
{ XK_KP_Begin,  K_HOME   },
{ XK_KP_Insert, K_INSERT },
{ XK_KP_Delete, K_DELETE },
{ XK_KP_Multiply, K_MULTIPLY },
{ XK_KP_Add,    K_ADD    },
{ XK_KP_Separator, K_SEPARATOR },
{ XK_KP_Subtract, K_SUBTRACT },
{ XK_KP_Decimal, K_DECIMAL },
{ XK_KP_Divide, K_DIVIDE },
};
for(int i = 0; i < __countof(tab); i++)
if(tab[i].keysym == keysym) {
    DispatchKey(KEYtoK(tab[i].key)|up, count);
    return;
}
if(GetShift() && chr == 0) {

```

```

static dword k[] = { 41, 33, 64, 35, 36, 37, 94, 38, 42, 40 };
for(int i = 0; i < 10; i++)
    if(keysym == k[i]) {
        DispatchKey(KEYtoK(i + K_0)|up, count);
        return;
    }
}
#endif PLATFORM_OSX11
if(GetCtrl() || GetAlt()) { // fix Ctrl+Shift+1 etc...
    keysym = decode((int)event->xkey.keycode, 0xa, '1', 0xb, '2', 0xc, '3', 0xd, '4',
        0xe, '5', 0xf, '6', 0x10, '7', 0x11, '8', 0x12, '9', 0x13, '0',
        keysym);
}
#endif
// DLOG("keysym: " << keysym << " " << (char)keysym);
keysym &= 0xFFFF;
if(keysym >= '0' && keysym <= '9' && (chr == 0 || GetCtrl() || GetAlt())) {
    DispatchKey(KEYtoK(keysym - '0' + K_0)|up, count);
    return;
}
if(chr >= 1 && chr < 32) {
    DispatchKey(KEYtoK(chr - 1 + K_CTRL_A)|up, count);
    return;
}
if(keysym >= 0xff80 && keysym <= 0xffb9 && chr) {
    DispatchKey(KEYtoK(chr)|up, count);
    return;
}
if(keysym >= 0xff00 && chr < 128 ||
    (GetCtrl() || GetAlt()) && keysym >= 0x20 && keysym < 0x7f) {
    if(keysym >= 'a' && keysym <= 'z')
        keysym = keysym - 'a' + 'A';
    DispatchKey(KEYtoK(keysym|K_DELTA)|up, count);
    return;
}

if((chr == 32 || chr == 9 || chr == 13) && !pressed)
    DispatchKey(chr|K_KEYUP, count);
if(chr && pressed) {
    DispatchKey(chr, count);
    for(int ii = 1; ii < wtext.GetLength(); ii++)
        DispatchKey(wtext[ii], count);
}
}
break;

```

Best regards,
Oblivion

Subject: Re: K_CTRL_SEMICOLON and K_CTRL_PERIOD keys share the same constant value on Win32

Posted by [mirek](#) on Fri, 18 Dec 2020 14:29:11 GMT

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

Thanks for investigating this. Do you think it would be possible to post here a complete set of changed files to fix X11?

Mirek

Subject: Re: K_CTRL_SEMICOLON and K_CTRL_PERIOD keys share the same constant value on Win32

Posted by [Oblivion](#) on Fri, 18 Dec 2020 17:12:23 GMT

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

Hi,

I've attached the patch.

After further investigating, the problem turned out to be simpler to solve. The culprit seems to be `decode()`.

When `keysym` is returned as default, it is truncated and the higher bytes were invalidated (filled with `0xff`).

Changing this:

```
keysym = decode((int)event->xkey.keycode, 0xa, '1', 0xb, '2', 0xc, '3', 0xd, '4',
              0xe, '5', 0xf, '6', 0x10, '7', 0x11, '8', 0x12, '9', 0x13, '0',
              keysym);
```

to this, solved the problem:

```
keysym = decode((int)event->xkey.keycode,
              0x0a, 0x31,
              0x0b, 0x32,
```

```
0x0c, 0x33,  
0x0d, 0x34,  
0x0e, 0x35,  
0x0f, 0x36,  
0x10, 0x37,  
0x11, 0x38,  
0x12, 0x39,  
0x13, 0x30, keySYM);
```

Best regards,
Oblivion

File Attachments

1) [X11Proc.cpp.patch](#), downloaded 125 times

Subject: Re: K_CTRL_SEMICOLON and K_CTRL_PERIOD keys share the same constant value on Win32

Posted by [mirek](#) on Tue, 22 Dec 2020 14:47:06 GMT

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

Thank you, patch applied.

Subject: Re: K_CTRL_SEMICOLON and K_CTRL_PERIOD keys share the same constant value on Win32

Posted by [Oblivion](#) on Sun, 31 Jan 2021 11:07:07 GMT

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

Hello Mirek,

I have further investigated the key handling on X11, and found that numpad virtual keycodes are not propagated properly:

Here is the current situation:

- When Numlock is ON, the characters corresponding to the numpad keys (0-9, +/*,) are dispatched. This is fine, nothing wrong here.

- However, when the same keys are pressed with CTRL or ALT modifier, they SHOULD send numpad keycodes, e.g. K_NUMPAD0-K_NUMPAD_9, and operators (K_ADD, K_SUBTRACT, etc.). This is the default behaviour of Upp on Windows and GTK (I don't have a Mac)

- But this is not happening on X11. Character codes are dispatched regardless of modifiers

So, I made a patch to remedy this problem. The patch, I believe, is at least a fix for some regression, if not an overall improvement in X11 key handling code.

Please review.

Best regards,
Oblivion

File Attachments

1) [CtrlCore.patch](#), downloaded 117 times

Subject: Re: K_CTRL_SEMICOLON and K_CTRL_PERIOD keys share the same constant value on Win32

Posted by [mirek](#) on Sun, 07 Feb 2021 09:25:44 GMT

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

Thanks, applied.

Mirek
