Keyboard Events Simulation using keybd_event() function

Naren Neelamegam
4 Jun 2004 CPOL

A short description of keybd_event() function for beginners.

Introduction

Simulation of a keyboard input is a well known concept for those who are all familiar with Visual Basic. `SendKeys()` in Visual Basic does all the things, if you want to do anything without keyboard. But what is in `SendKeys()` function? What does it do? Can we do such a thing with Visual C++? This article is the answer. I think this will be useful for beginners who are all just trying to do something different with VC++. Let us get into the steps...

Function Syntax

```
void keybd_event(BYTE bVirtualKey, BYTE bScanCode,
                         DWORD dwFlags, DWORD dwExtraInfo);
```

- `bVirtualKey //Virtual Keycode of keys. E.g., VK_RETURN, VK_TAB…`
- `bScanCode //Scan Code value of keys. E.g., 0xb8 for “Left Alt” key.`
- `dwFlags //Flag that is set for key state. E.g., KEYEVENTF_KEYUP.`
- `dwExtraInfo //32-bit extra information about keystroke.`

Function Details:

- `bVirtualKey`

  Virtual keycode that has to be send as key input. The following are the available predefined virtual key codes:

VK_NUMPAD7	0x67	VK_BACK	0x08
VK_NUMPAD8	0x68	VK_TAB	0x09
VK_NUMPAD9	0x69	VK_RETURN	0x0D
VK_MULTIPLY	0x6A	VK_SHIFT	0x10
VK_ADD	0x6B	VK_CONTROL	0x11
VK_SEPARATOR	0x6C	VK_MENU	0x12
VK_SUBTRACT	0x6D	VK_PAUSE	0x13
VK_DECIMAL	0x6E	VK_CAPITAL	0x14
VK_DIVIDE	0x6F	VK_ESCAPE	0x1B

VK_F1	0x70	VK_SPACE	0x20
VK_F2	0x71	VK_END	0x23
VK_F3	0x72	VK_HOME	0x24
VK_F4	0x73	VK_LEFT	0x25
VK_F5	0x74	VK_UP	0x26
VK_F6	0x75	VK_RIGHT	0x27
VK_F7	0x76	VK_DOWN	0x28
VK_F8	0x77	VK_PRINT	0x2A
VK_F9	0x78	VK_SNAPSHOT	0x2C
VK_F10	0x79	VK_INSERT	0x2D
VK_F11	0x7A	VK_DELETE	0x2E
VK_F12	0x7B	VK_LWIN	0x5B
VK_NUMLOCK	0x90	VK_RWIN	0x5C
VK_SCROLL	0x91	VK_NUMPAD0	0x60
VK_LSHIFT	0xA0	VK_NUMPAD1	0x61
VK_RSHIFT	0xA1	VK_NUMPAD2	0x62
VK_LCONTROL	0xA2	VK_NUMPAD3	0x63
VK_RCONTROL	0xA3	VK_NUMPAD4	0x64
VK_LMENU	0xA4	VK_NUMPAD5	0x65
VK_RMENU	0xA5	VK_NUMPAD6	0x66

Character key can be converted into virtual key using `VkKeyScan(TCHAR ch)` function.

- `bScanCode`

  Scan code is the hardware key code for the key (make and break codes). The following are the available scan codes (break code will be used in this parameter).

Key	Make	Break	Key	Make	Break
Backspace	0E	8E	F1	3B	BB
Caps Lock	3A	BA	F2	3C	BC
Enter	1C	9C	F3	3D	BD
Esc	01	81	F4	3E	BE
Left Alt	38	B8	F7	41	C1
Left Ctrl	1D	9D	F5	3F	BF
Left Shift	2A	AA	F6	40	C0
Num Lock	45	C5	F8	42	C2
Right Shift	36	B6	F9	43	C3
Scroll Lock	46	C6	F10	44	C4
Space	39	B9	F11	57	D7
Sys Req (AT)	54	D4	F12	58	D8
Tab	0F	8F			
Key	Make	Break	Key	Make	Break
A	1E	9E	N	31	B1
B	30	B0	O	18	98
C	2E	AE	P	19	99
D	20	A0	Q	10	90
E	12	92	R	13	93
F	21	A1	S	1F	9F
G	22	A2	T	14	94
H	23	A3	U	16	96
I	17	97	V	2F	AF
J	24	A4	W	11	91
K	25	A5	X	2D	AD
L	26	A6	Y	15	95
M	32	B2	Z	2C	AC

Key	Make	Break	Key	Make	Break
1	02	82	-	0C	8C
2	03	83	=	0D	8D
3	04	84	[	1A	9A
4	05	85	]	1B	9B
5	06	86	;	27	A7
6	07	87	'	28	A8
7	08	88	`	29	A9
8	09	89	\	2B	AB
9	0A	8A	,	33	B3
0	0B	8B	.	34	B4
			/	35	B5

Keypad Keys	Make	Break
Pad0(Ins)	52	D2
Pad1(End)	4F	CF
Pad2(D arrow)	50	D0
Pad3(PgDn)	51	D1
Pad4(L arrow)	4B	CB
Pad5	4C	CC
Pad6(R arrow)	4D	CD
Pad7(Home)	47	C7
Pad8(U arrow)	48	C8
Pad9(PgUp)	49	C9
Pad.(Del)	53	D3
Pad*(PrtSc)	37	B7
Pad-	4A	CA
Pad+	4E	CE

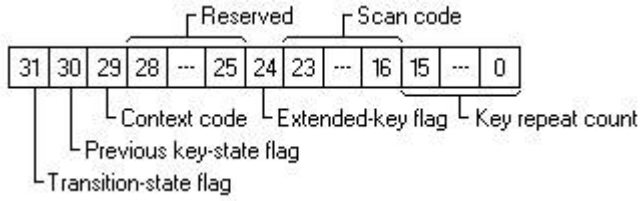
- `dwFlags`

  A set of flag bits that specify various aspects of function operation. An application can use any combination of the following predefined constant values to set the flags.

Value	Meaning
`KEYEVENTF_EXTENDEDKEY`	If specified, the scan code was preceded by a prefix byte having the value 0xE0 (224).
`KEYEVENTF_KEYUP`	If specified, the key is being released. If not specified, the key is being depressed.

- `dwExtraInfo`

  32-bit extra information along with the keyboard input.



Example Code

```
// Simulating a Alt+Tab keystroke
keybd_event(VK_MENU,0xb8,0 , 0); //Alt Press
keybd_event(VK_TAB,0x8f,0 , 0); // Tab Press
keybd_event(VK_TAB,0x8f, KEYEVENTF_KEYUP,0); // Tab Release
keybd_event(VK_MENU,0xb8,KEYEVENTF_KEYUP,0); // Alt Release

// Simulating a Ctrl+A keystroke
keybd_event(VK_CONTROL,0x9d,0 , 0); // Ctrl Press
keybd_event(VkKeyScan('A'),0x9e,0 , 0); // 'A' Press
```

```
keybd_event(VkKeyScan('A'),0x9e, KEYEVENTF_KEYUP,0); // 'A' Release
keybd_event(VK_CONTROL,0x9d,KEYEVENTF_KEYUP,0); // Ctrl Release
```

Conclusion

This article may not be that much detailed. None of the articles can satisfy one's expectations. But, each article should be a seed for your technical growth. Thus, I believe that this would be a seed. Thank you all.

License

About the Author

Naren Neelamegam

Software Developer

India

Naren started coding during 1999 with FORTRAN, then COBOL, PASCAL, C, C++, VC++ ..... C#, Java, ASP so on, till today. He claims himself as techie who loves coding, but he is not even sure which technology or platform he loves, most of the time Windows, some time WinCE, some time Linux, nowadays Android and embedded platforms. He can do some query stuffs with Oracle, SQL Server, MySQL. He strongly believes that C/C++ is a must for all programmers, "if you know C/C++, you can do programming on any language". He is an electronic gadget guy who likes to buy small gadgets all the time, at least he will do window shopping on a gadget shop. His interest always have been with Automation in any form, call it a little automated program sitting in his Laptop or a home automation program runs on his mobile. Apart from coding, he likes to do...???

Comments and Discussions

42 messages have been posted for this article Visit **https://www.codeproject.com/Articles/7305/Keyboard-Events-Simulation-using-keybd-event-funct** to post and view comments on this article, or click **here** to get a print view with messages.