Windows Platform Design Notes

Designing Hardware for the Microsoft® Windows® Family of Operating Systems

Keyboard Scan Code Specification

Abstract: This specification details the PS/2 Scan Codes and USB Usage Tables that are validated for compliance to the Microsoft® Windows® Logo Program testing standard. This document details the alternative make and break PS/2 scan code and USB code response for the Windows Logo Key and Application Keys, plus Advanced Configuration and Power Interface (ACPI) power controls.

This specification was previous published, with the same content, as "Windows Hardware Quality Labs Keyboard Specification" and also referred to as "Windows Keys Specification" and "New Keys Specification."

Revision 1.3a — March 16, 2000

Contents

Notice of Change

This version of the Keyboard Scan Code Specification details the PS/2 Scan Codes and USB Usage Tables that are validated for compliance to the Microsoft Windows Logo Program testing standard. This document details the alternative make and break PS/2 scan code and USB code response for the Windows Logo key, Application Keys, and Advanced Configuration & Power Interface (ACPI) power controls.
/
This document supercedes all versions of the New Keys Specifications for the Windows Keys and Windows Hardware Quality Labs WHQL Keyboard Specification. There is no change required for developers of Windows-based applications to support any compatible implementation.

***IMPORTANT ***
As of August 1, 1996, your keyboard must pass the Windows keyboard testing at the Windows Hardware Quality Labs either prior to, or in conjunction with, the Windows Key Logo testing.

The requirements for compatibility testing are not changed, and a keyboard vendor can implement these new keys successfully following the 1.0, 1.1, or 1.2 versions of the New Keys Specification. There is no change required for developers of Windows-based applications to support any compatible implementation.

Revision History

1.3a	3/16/00	Adjusted title and related references
1.3	2/23/99	Apps key removed as modifier, Hid audio control usages added
1.2	6/25/98	ACPI Codes corrected.
1.1	5/1/98	Scan Code Set 3 requirement removed.
1.0	3/27/98	Initial proposal for consideration.

Windows Keys Support for Windows Software Applications

The three Windows Keys report the following key codes in the Microsoft Windows 95, Windows 98, and Windows NT® operating systems, and future versions of Windows operating systems.

Key Codes for Scan Code Set 1:

Windows Key	Make	Break	Windows Virtual Key
Left Windows	E0 5B	E0 DB	**5B**
Right Windows	E0 5C	E0 DC	**5C**
Application	E0 5D	E0 DD	**5D**

Key Codes for Scan Code Set 2:

Windows Key	Make	Break	Windows Virtual Key
Left Windows	E0 1F	E0 F0 1F	**5B**
Right Windows	E0 27	E0 F0 27	**5C**
Application	E0 2F	E0 F0 2F	**5D**

Key Codes for USB Usage Tables:

Windows Key	Usage Page	Usage Index (Dec)	Usage Index (Hex)	Typical AT-101 position
Left Windows	0x07	227	E3	127
Right Windows	0x07	231	E7	128
Application	0x07	101	65	129

What Software Applications Vendors Should Support and How

To provide support for the Windows Keys, an application should support the application key virtual key scan code (5D) as a context menu event similar to a right mouse button click in some applications today.

Applications vendors are encouraged to extend their support beyond the application key to include support for application key combinations like CTRL+Application, ALT+Application and SHIFT+Application. These key combinations are reserved for applications to support. Support for Windows+Application is reserved for the operating system.

Application Key Combinations	Recommended Support
CTRL + Application Key	unspecified, application specific
ALT + Application Key	unspecified, application specific
SHIFT + Application Key	unspecified, application specific
Windows + Application Key	*reserved for operating system*
Application Key + <alpha numeric keys>	not supported as a modifier key, support must be supplied by application

Suggestions for Support of Application Key Combinations

- Context menu at the location of the text cursor (instead of the mouse pointer)
- Launch automated help agent
- Pop up list of last 5 actions (or list of commonly accessed functions)
- Enable/Disable macro recording or other user controllable features
- Switch to next open window in a multiple document
- User assignable key through an application UI

Application Key Support in the Microsoft Windows Operating Systems

The application key will primarily function to bring up a context menu at the selection or mouse pointer. This functionality is the same as the right mouse button click in some applications today.

Operating System	Application Key Response Supported by the Operating System
Windows 3.x	no response1
Windows NT 3.5x	no response1
Windows NT 4.0	context menu on selection
Windows NT 4.0	context menu on selection
Windows 95	context menu on selection
Windows 98	context menu on selection

1 Windows still reports a 5D virtual key code which means that applications can enable functionality

Using the Windows Logo Key Logo and Application Key Logo in Documentation

The Windows Logo key and the Application Logo key may be used by OEMs, ISVs and IHVs in documentation that describes the functionality of the Logo keys, provided such description is consistent with Microsoft's guidelines for use of the Logo keys.

- It is not necessary to obtain a Logo license from Microsoft in order to refer to the Logo keys in documentation. This is the only permissible non-licensed use of the Windows Logo.

- The Logos may not be used in any way other than as specified in the Logo license guidelines.

- Upon request, Microsoft will provide camera-ready artwork of the Logos to be used in documentation. Send your e-mail request to Microsoft Windows Hardware Quality Labs at whqlinfo@microsoft.com with "Artwork Request" in your subject line. Alternatively, you may use the Windows flag symbol provided in the "Wingdings®" font. This character is available by pressing ALT + 0255 on the numeric keypad.

- You may not alter the Logos in any way.

- The Logos should be typed with an empty space before and after each symbol, followed by a plus sign and another space, followed by the modifier if appropriate. Examples are shown:
  Windows Logo key + B or   + B

  Application key + S or   + + S

Windows Logo Key Support (Reserved for Operating System Use)

The Windows Logo keys are reserved for system level functions. Software developers should not implement support for the Windows Logo keys in Windows 95, Windows 98, or Windows NT-based applications. The Windows Logo keys are supported in the operating system and provide system level functionality to the end user. The following table lists a few of the Windows Logo key combinations and their functionality. This list is not exhaustive and additional combinations will be used by the Microsoft Windows 95, Windows 98, or Windows NT operating systems.

Windows Logo Key Combination	Functionality in Windows 95/98
Windows + F1	Display the popup menu for the selected object.
Windows + TAB	Activate next Taskbar button.
Windows + E	Explore My Computer.
Windows + F	Find Document.
Windows + CTRL + F	Find Computer.
Windows + M	Minimize All.
SHIFT + Windows + M	Undo Minimize All.
Windows + R	Display Run dialog box.
Windows + PAUSE (Break)	Perform a system function.

Windows Keys Support for OEMs and Keyboard IHVs

OEMs that buy their keyboards with Windows Logo keys from a licensed keyboard manufacturer and do not modify the keyboard product except to add their Logo and other cosmetic changes are not required to license the Windows Logo keys from Microsoft. If the OEM product changes the layout or BIOS of an already licensed keyboard the OEM must go through the same licensing and certification procedure as described in the next section of this document.

Software Support of the Windows Keys in the Windows 95/98 Operating System

The Windows Keys are supported in the Windows 95, Windows 98, or Windows NT operating systems user interface. Application vendors will enable functionality for the Application key in their applications. 10 Windows Logo key combinations are reserved for OEM use.

Key Event Supported	Windows Support	Application Support	OEM System Utilities
Windows Logo Key	open the Start menu		
Windows Logo key Combinations	9 shortcut functions		Win + 1-0 reserved for OEM
Application Logo Key	context menu on selection	context menu on selection	context menu on selection
CTRL, ALT, SHIFT+Application		varies by application	varies by utility

Special Keys Reserved for OEM Usage

Ten (10) Windows Logo keys combinations have been reserved for OEM use. These keys can be used by OEMs to provide keyboard hotkey controls for Speaker Volume, Monitor/LCD Brightness or Contrast, Password, or other value added functions.

Windows Logo Key Combination	Status
Windows + 1	reserved for OEMs
Windows + 2	reserved for OEMs
Windows + 3	reserved for OEMs
Windows + 4	reserved for OEMs
Windows + 5	reserved for OEMs
Windows + 6	reserved for OEMs
Windows + 7	reserved for OEMs
Windows + 8	reserved for OEMs
Windows + 9	reserved for OEMs
Windows + 0	reserved for OEMs

Windows + all other key combinations	*reserved for operating system*

Multiple Key Operation Requirements

Valid 2-Key Combinations

The following list combined with the "Valid Final Keys" table, defines the valid 2-key combinations. When any one of the keys below is pressed, followed by a 2nd key from the "Valid Final Keys" table, the keyboard must properly indicate that those 2 keys are pressed. There are no exceptions to these 2-Key Combinations.

- LWIN
- RWIN
- LCTRL
- RCTRL
- LSHIFT
- RSHIFT
- LALT
- RALT

Valid Final Keys

The following table lists the final keys that are required to work for valid two and three key combinations. Any key from the Valid 2-Key Combinations list combined with any key from the table below must work properly. Any 2 keys indicated in a row from the Valid 3-Key Combinations table combined with any key from the table below must work properly. Exceptions to this requirement are listed in Combinations of 3-Keys which are unavailable. Invalid key combinations will generate a Key Error code.

Esc	Tab	c	Left Arrow
F1	q	v	Right Arrow
F2	w	b	Up Arrow
F3	e	n	Down Arrow
F4	r	m	
F6	y	.	Enter
F7	u	/	Print Screen
F8	i	Insert	Scroll Lock
F9	o	Delete	Pause
F10	p	Home	Num Lock
F11	[	End	Numeric 0
F12	]	Page Up	Numeric 1
`	\	Page Down	Numeric 2
1	a		Numeric 3
2	s		Numeric 4
3	d		Numeric 5
4	f		Numeric 6
5	g		Numeric 7
6	h		Numeric 8
7	j		Numeric 9
8	k		Numeric /
9	l		Numeric *
0	;		Numeric -
-	'		Numeric +
=	z		Numeric Enter
Backspace	x		Numeric .

Indicates Traditional IBM Final keys.

Valid 3-Key Combinations

The following table combined with the "Valid Final Keys" table, defines the valid 3-key combinations. For each row in the table below, an "X" indicates that key is selected. When the two keys in a row are pressed followed by a 3rd key from the "Valid Final Keys" table, the keyboard must properly indicate that those 3 keys are pressed. Exceptions to these tables are listed below. Any 3-key combinations not listed are not required to work, but may do so at the keyboard manufacturer's discretion.

LWIN	RWIN	LCTRL	RCTRL	LSHIFT	RSHIFT	LALT	RALT
X							
X		X					
X			X				
X				X			
X					X		
X						X	
X							X
	X						
	X	X					
	X		X				
	X			X			
	X				X		
	X					X	
	X						X
X	X						
		X		X			
		X			X		
			X	X			
			X		X		
		X				X	
		X					X
			X			X	
			X				X
				X		X	
				X			X
					X	X	
					X		X

Indicates Traditional IBM 3-Key combinations.

Combinations of 3-Keys which are Unavailable

Based upon the standard industry accepted 16 X 8 scanning matrix, several keys will be "ghosted", meaning not uniquely detectable by the firmware during the key switch scanning of 3-Key combinations. To accommodate this, the following table details the key combinations, which are excluded from the above tables. These 3-key combinations are not required to work, but may do so at the keyboard manufacturer's discretion.

1st Key	2nd Key	3rd Key
LWIN	RWIN	Up Arrow
RWIN	LALT	key 56

RWIN	RALT	key 42
RWIN	RCTRL	Caps Lock
RWIN	LCTRL	Caps Lock
LWIN	LALT	key 107
LCTRL	LSHIFT	Pause, Caps Lock
RCTRL	RSHIFT	Pause, Caps Lock

Additional General Requirements

LWIN and RWIN Keys as modifiers

This keyboard must generate unique scan codes for make and break. The LWIN and RWIN keys must be treated as modifier keys, much in the same way as ALT, and Control keys are handled today. It will be up to the operating system to decide on an implementation scheme for the modifier key functionality.

Building a Keyboard with the Windows Keys

Tools Needed to Build a Keyboard with the Windows Keys

- Keyboard Scan Code Specification (this document).
- Microsoft Windows Logo License Agreement
- Windows Logo Key License Agreement.
- Windows Exhibit A from the Device Testing Agreement.
- Windows Exhibit B from the "Designed for Microsoft" Logo License Agreement.
- Windows Keyboard Test Kit: Test Procedures and Test Tools.

How to Get the WHQL Keyboard Test Kit and Keys Specification

The latest version of the Keyboard Scan Code Specification is available at http://www.microsoft.com/hwdev/desinit/scancode.htm.

The latest version of the WHQL Keyboard Test Kit is available electronically directly from the Windows Hardware Quality Labs Web Site at http://www.microsoft.com/hwtest/testkits/. You will need to individually download the Test Procedures and Test Tools document.

Microsoft Windows Legal Agreements

Complete and sign the Microsoft Logo License Agreement and the Testing Agreement. If your company previously signed these agreements, you only need to include a newly completed "Exhibit A" of the Testing Agreement with your Test Submission.

Note: The following WHQL legal agreements, previously available through the Microsoft FAX Service, can now be downloaded from the WHQL web site at http://www.microsoft.com/hwtest/testkits/ :

- Microsoft Windows Logo License Agreement - "Exhibit A5" (Windows Logo Agreement.tif)
- Testing Agreement (Testing Agreement.tif)
- "Exhibit A" of the Testing Agreement (Exhibit A.tif)

If you are unable to open these agreements, please send e-mail to whqlinfo@microsoft.com with **AGREEMENT REQUEST** in the Subject line.

Send blank e-mail to the auto reply at <u>whqlinfo@microsoft.com</u> for a complete list of administrative contacts.

Steps to Ship a Keyboard with the Windows Logo Keys

Develop your keyboard according to the tables in Appendix A of this specification.

Compatibility Requirement

1. Keyboards are required to be compatible at the Windows virtual key code level. You can accomplish this by either:
    a) Developing a keyboard with the electronics and firmware that comply with the specification or by,
    b) Developing a software driver for use with Windows 3.1 and Windows 95 that maps the proper virtual key codes to keyboard events on your keyboard.
2. Test your implementation using the WHQL Keyboard Test Procedures to ensure that single-key, two-key and three-key combinations work correctly, including the Windows Logo keys (if implemented) and report the proper virtual scan codes in Windows.
3. Once you are satisfied that your keyboard passes the tests, run the full Test Suite and use it to generate log files of encoded test results that must be included with the other information required for your Test Submission. Full testing instructions are provided in the WHQL Keyboard Test Procedures.

Retest Requirements

Any time that you implement the Windows Keys on a new keyboard technology, you must go through this testing and certification procedure. You will also need to submit a retest any time that you change the key layout or the BIOS firmware on a certified keyboard.

Recommended Physical Locations of the Windows Keys

For full travel desktop keyboards Microsoft recommends that the Windows Logo keys be located near the current modifier keys like CTRL, SHIFT and ALT since the Windows Logo keys must also function as modifier keys. The Application key can be located wherever it appropriately fits. In the attached diagram the keys were placed on the bottom row on either side of the space bar. By changing the current size of the space bar the Windows Keys can be located in a similar position on a standard keyboard.
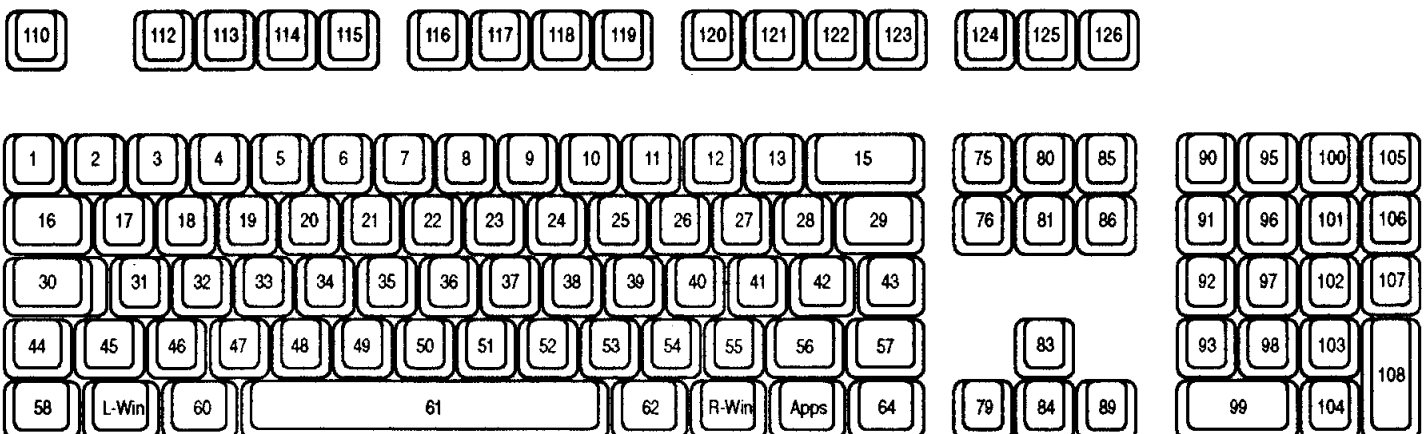
Recommendations for Laptop Keyboards

Given the crowded nature of laptop keyboards it is difficult to add three Windows Keys. A minimal implementation of the Windows Keys would require the addition of one Windows Logo Key and the replacement of the [Fn] key for the Application key.

The two Windows Logo Keys are not distinguished individually by the Microsoft Windows 95 operating system and the IntelliType software only uses a Left Windows Logo key + Right Windows Logo key press for the Logon/Logoff function.

Design Requirements:

● The Windows Logo Key needs to function as a modifier key (CTRL, SHIFT and ALT).
● There can be no other icons or words associated with the Windows Flag trademark on the keycap.

Design Options:

● There do not need to be Left and Right Windows Logo keys to get full functionality in Windows 95.
● The Application key can be a dual function key and can be used to replace the Fn. key.

ACPI Power Management Keys

The three ACPI Power Management keys report the following key codes in the Microsoft Windows 95, Windows 98, and Windows NT and operating systems.

Key Codes for Scan Code Set 1:

ACPI key	Make	Break	Windows Virtual Key
Power Event	E0 5E	E0 DE	**N/A**
Sleep Event	E0 5F	E0 DF	**N/A**
Wake Event	E0 63	E0 E3	**N/A**

Key Codes For Scan Code Set 2:

ACPI key	Make	Break	Windows Virtual Key
Power Event	E0 37	E0 F0 37	**N/A**
Sleep Event	E0 3F	E0 F0 3F	**N/A**
Wake Event	E0 5E	E0 F0 5E	**N/A**

Key Codes for USB Usage Tables:

ACPI Key	Usage Page	Usage Index (Dec)	Usage Index (Hex)	Typical AT-101 position
Power Event	0x01	129	81	**N/A**
Sleep Event	0x01	130	82	**N/A**
Wake Event	0x01	131	83	**N/A**

Usage Page and Usages for Audio Control

A device wanting to be to recognized as a HID audio control device must declare itself as being a **Consumer Control** device (usage 0x01), as defined in the **Consumer Page** (page 0x0C) in

the *Universal Serial Bus HID Usage Tables Version 1.0 specification*. This means that its top-level application collection should be Usage Page (Consumer), Usage (Consumer Control).

When such a device is enumerated by the operating system, the supporting software (HIDSERVE.EXE) is installed and loaded on the host system. Table 1 outlines the Consumer Page audio controls that are supported in Windows 2000.

Table 1. Consumer Page audio controls supported in Windows 2000.

Usage	Usage Name	Usage Type
0xE0	*Volume**	Linear Control (LC)
0xE2	*Mute**	On/Off Control (OOC)
0xE3	Bass	Linear Control (LC)
0xE4	Treble	Linear Control (LC)
0xE5	*Bass Boost**	On/Off Control (OOC)
0xE7	Loudness	On/Off Control (OOC)
0xE9	*Volume Increment**	Re-trigger Control (RTC)
0xEA	*Volume Decrement**	Re-trigger Control (RTC)
0x152	Bass Increment	Re-trigger Control (RTC)
0x153	Bass Decrement	Re-trigger Control (RTC)
0x154	Treble Increment	Re-trigger Control (RTC)
0x155	Treble Decrement	Re-trigger Control (RTC)

*) These controls are supported in Windows 98 (original release and Service Pack 1 release).

The Volume, Bass, and Treble usages (of type LC) should be deployed for controls that generate both increment and decrement data represented by a 2-bit value, whereas the associated Increment and Decrement usages (of type RTC) should be deployed for pairs of one-bit controls (traditional button controls). The hardware design and implementation determines what usage types are appropriate for the HID firmware implementation for a particular device.

It's also important to notice that any re-triggering of events should be done by software timers in the host system, and not by hardware timers in the device itself. For example, if the user keeps pressing the Volume Increment button, the device should only generate one input report with this state information. Host software will perform the actual re-triggering of events that will lead to continuous increments of the volume until the device generates another input report indicating that the button has been released or until the maximum volume has been reached.

Important Design Aspects

The supported HID audio controls:

● Affect audio playback only. There's no support for HID-based controls that affect audio recording, etc.

● Affect audio playback by the default system audio device (same device as sndvol32.exe controls).

● Should be declared right beneath the top-level application collection in the HID firmware in order to be discovered and utilized by the host system. Important: do not wrap these controls in sub-collections.

Sample Firmware Designs

The following two samples illustrate how firmware that uses the audio controls listed above can be implemented. The first sample uses the Volume usage (of type LC) for the volume control, and the second sample uses the Volume Increment and Volume Decrement usages (of type RTC). Notice that the mute button in both of these samples is implemented as a Relative, Preferred State type of control, or a toggle, which means that 0-to-1 transitions will toggle mute on/off.

Sample 1.

```
Usage Page (Consumer)
Usage (Consumer Control)
Collection (Application)
    Usage (Volume)
    Logical Minimum (-1)
    Logical Maximum (1)
    Report Size (2)
    Report Count (1)
    Input (Data, Variable, Relative, Preferred)
    Usage (Mute)
    Logical Minimum (0)
    Logical Maximum (1)
    Report Size (1)
    Report Count (1)
    Input (Data, Variable, Relative, Preferred)
    Report Count (5)
    Input (Constant)
End Collection
```

Sample 2.

```
Usage Page (Consumer)
Usage (Consumer Control)
Collection (Application)
    Logical Minimum (0)
    Logical Maximum (1)
    Usage (Volume Increment)
    Usage (Volume Decrement)
    Report Size (1)
    Report Count (2)
    Input (Data, Variable, Absolute, Preferred)
    Usage (Mute)
    Report Count (1)
    Input (Data, Variable, Relative, Preferred)
    Report Count (5)
    Input (Constant)
End Collection
```

Appendix A: Windows Standard PS/2 Scan Codes

General Requirements

This keyboard must be downward compatible with an enhanced type 101 keyboard. This keyboard must be fully compatible with the PS/2 Keyboard Controller command set. It must support Scan Set 2. Three additional scan codes are to be generated from the three Windows Keys (Left Windows, Right Windows, and Application).

The virtual key codes described in the following tables are specifically for Windows operation only and are not generated by the keyboard directly. The keyboard manufacturer will need to provide a Windows Logo keyboard driver to support the three virtual keys for use with the Windows v. 3.1 operating system (not required for Logo purposes). The support will be built into drivers shipped with future versions of the Windows and Windows NT operating systems.

Typematic Characteristics

Typematic is the term used for keys that will automatically repeat after a specified time delay. In scan code 1 and scan code 2, key 126 is the only key that is not Typematic. (This key is the only one that does not send a break code as well).

Scan Code Assignments:

Under all Microsoft operating systems, all keyboards actually transmit Scan Code Set 2 values down the wire from the keyboard to the keyboard port. These values are translated to Scan Code Set 1 by the i8042 port chip.[1]   The rest of the operating system, and all applications that handle scan codes expect the values to be from Scan Code Set 1. Scan Code Set 3 is not used or required for operation of Microsoft operating systems.

The following recommendations are made for those who wish to define a new scan code for a proprietary purpose. It is the manufacturer's responsibility to implement a software driver that supports any such scan codes. New scan codes must not be chosen from those currently recognized by the operating system. Keys that are called out with a Key Number or Key Name in the Scan Code Table are currently recognized by and used by the operating system.

Single-Byte Scan Codes

The values discussed below are all Scan Code Set 1 values.

Avoid Set 1 scan codes above 0x79, since the release (key up) code would be 0xFA or greater, and the keyboard driver is known to interpret such values as responses from the 8042 chip, not as keystrokes (scan codes). Specifically, 0x7A, 0x7E and 0x7F are problematic for the Windows NT keyboard driver in 3.51 and 4.0 since 0xFE is RESEND, 0xFA. is ACKNOWLEDGE and 0xFF seems to be simply just swallowed up. The effect is that key releases would be lost, and error logs fill up.

---

[1] This mode is set by issuing a 0xF0 command byte to the keyboard, and turning the translate bit (0x40) in the 8042 command byte on. On Intel and Power PC machines, it is the firmware that initializes the keyboard and 8042 chip this way, while Windows NT® explicitly sets this mode for MIPS and Alpha.
In the very early days of Windows NT®, an attempt was made to use the much more orthogonal Scan Code Set 3, but due to bugs in the implementation of this Scan Code Set on numerous OEM keyboards, the idea was abandoned.

- **Avoid 0x60 and 0x61**, since the release (up key) code would be 0xE0 and 0xE1, which are reserved prefix bytes.
- **Avoid 0x00**, since that is likely to have some special meaning to code from driver level through to application level.

Prefixed Scan Codes

Some keys on standard 101/102-key keyboards (and the Microsoft Natural keyboard amongst others) emit a sequence of two bytes, where the 1st byte is either 0xE0 or 0xE1. This method is used primarily to distinguish between left and right versions of the same key, e.g., Left Alt is 0x38 while Right Alt is 0xE0 0x38. The 0xE0 prefix is indicated as the "extended bit" (bit 24) in the lParam of messages such as WM_KEYDOWN. The 0xE1 prefix is much rarer, but operates similarly to the 0xE0 prefix. It's presence or absence is not indicated through the API in any way.

If you use scan codes from the 0xE0 set, make sure the second byte is suitable in the same way as single byte scan code values. In other words:

> Not greater than 0x79
>
> Not 0x60 or 0x61
>
> Not 0x00

Scan Code Table

The following table lists the full set of Scan Codes as presently recognized by the Microsoft operating systems. The US Key assignments are for reference to a type 101/102 Enhanced keyboard as supported by the Type 4 Keyboard layout. If there is no entry in the 101/102 Enhanced keyboard column, this scan code is currently not recognized by the operating system. The Key Location field has been added to aid in the placement of keys as illustrated in the Recommended Physical Locations of the Windows Keys on page .

key location	101/102 Enhanced Keyboard	scan 1 make	scan 1 break	scan 2 make	scan 2 brake
	DO NOT USE	00	80	00	F0 00
	DO NOT USE	E0 00	E0 80	E0 00	E0 F0 00
1	~ `	29	A9	0E	F0 0E
		E0 29	E0 A9	E0 0E	E0 F0 0E
2	! 1	02	82	16	F0 16
		E0 02	E0 82	E0 16	E0 F0 16
3	@ 2	03	83	1E	F0 1E
		E0 03	E0 83	E0 1E	E0 F0 1E
4	# 3	04	84	26	F0 26
		E0 04	E0 84	E0 26	E0 F0 26
5	$ 4	05	85	25	F0 25
		E0 05	E0 85	E0 25	E0 F0 25
6	% 5	06	86	2E	F0 2E
		E0 06	E0 86	E0 2E	E0 F0 2E
7	^ 6	07	87	36	F0 36
		E0 07	E0 87	E0 36	E0 F0 36
8	& 7	08	88	3D	F0 3D
		E0 08	E0 88	E0 3D	E0 F0 3D
9	* 8	09	89	3E	F0 3E
		E0 09	E0 89	E0 3E	E0 F0 3E
10	(9	0A	8A	46	F0 46
		E0 0A	E0 8A	E0 46	E0 F0 46
11	) 0	0B	8B	45	F0 45
		E0 0B	E0 8B	E0 45	E0 F0 45
12	_ -	0C	8C	4E	F0 4E
		E0 0C	E0 8C	E0 4E	E0 F0 4E

key location	101/102 Enhanced Keyboard	scan 1 make	scan 1 break	scan 2 make	scan 2 brake
13	+ =	0D	8D	55	F0 55
		E0_0D	E0_8D	E0_55	E0_F0 55
15	Backspace	0E	8E	66	F0 66
		E0_0E	E0_8E	E0_66	E0_F0 66
16	Tab	0F	8F	0D	F0 0D
		E0_0F	E0_8F	E0_0D	E0_F0 0D
17	Q	10	90	15	F0 15
		E0_10	E0_90	E0_15	E0_F0 15
18	W	11	91	1D	F0 1D
		E0_11	E0_91	E0_1D	E0_F0 1D
19	E	12	92	24	F0 24
		E0_12	E0_92	E0_24	E0_F0 24
20	R	13	93	2D	F0 2D
		E0_13	E0_93	E0_2D	E0_F0 2D
21	T	14	94	2C	F0 2C
		E0_14	E0_94	E0_2C	E0_F0 2C
22	Y	15	95	35	F0 35
		E0_15	E0_95	E0_35	E0_F0 35
23	U	16	96	3C	F0 3C
		E0_16	E0_96	E0_3C	E0_F0 3C
24	I	17	97	43	F0 43
		E0_17	E0_97	E0_43	E0_F0 43
25	O	18	98	44	F0 44
		E0_18	E0_98	E0_44	E0_F0 44
26	P	19	99	4D	F0 4D
		E0_19	E0_99	E0_4D	E0_F0 4D
27	{ [	1A	9A	54	F0 54
		E0_1A	E0_9A	E0_54	E0_F0 54
28	}]	1B	9B	5B	F0 5B
		E0_1B	E0_9B	E0_5B	E0_F0 5B
29*	| \	2B	AB	5D	F0 5D
		E0_2B	E0_AB	E0_5D	E0_F0 5D
30	Caps Lock	3A	BA	58	F0 58
		E0_3A	E0_BA	E0_58	E0_F0 58
31	A	1E	9E	1C	F0 1C
		E0_1E	E0_9E	E0_1C	E0_F0 1C
32	S	1F	9F	1B	F0 1B
		E0_1F	E0_9F	E0_1B	E0_F0 1B
33	D	20	A0	23	F0 23
		E0_20	E0_A0	E0_23	E0_F0 23
34	F	21	A1	2B	F0 2B
		E0_21	E0_A1	E0_2B	E0_F0 2B
35	G	22	A2	34	F0 34
		E0_22	E0_A2	E0_34	E0_F0 34
36	H	23	A3	33	F0 33
		E0_23	E0_A3	E0_33	E0_F0 33
37	J	24	A4	3B	F0 3B
		E0_24	E0_A4	E0_3B	E0_F0 3B
38	K	25	A5	42	F0 42
		E0_25	E0_A5	E0_42	E0_F0 42
39	L	26	A6	4B	F0 4B
		E0_26	E0_A6	E0_4B	E0_F0 4B
40	: ;	27	A7	4C	F0 4C
		E0_27	E0_A7	E0_4C	E0_F0 4C
41	" '	28	A8	52	F0 52
		E0_28	E0_A8	E0_52	E0_F0 52
42**		2B	AB	5D	F0 5D
		E0_2B	E0_AB	E0_5D	E0_F0 5D
43	Enter	1C	9C	5A	F0 5A
44	L SHIFT	2A	AA	12	F0 12
		E0_2A	E0_AA	E0_12	E0_F0 12
45**		56	D6	61	F0 61

key location	101/102 Enhanced Keyboard	scan 1 make	scan 1 break	scan 2 make	scan 2 brake
		E0_56	E0_D6	E0_61	E0_F0 61
46	Z	2C	AC	1A	F0 1A
		E0_2C	E0_AC	E0_1A	E0_F0 1A
47	X	2D	AD	22	F0 22
		E0_2D	E0_AD	E0_22	E0_F0 22
48	C	2E	AE	21	F0 21
		E0_2E	E0_AE	E0_21	E0_F0 21
49	V	2F	AF	2A	F0 2A
		E0_2F	E0_AF	E0_2A	E0_F0 2A
50	B	30	B0	32	F0 32
		E0_30	E0_B0	E0_32	E0_F0 32
51	N	31	B1	31	F0 31
		E0_31	E0_B1	E0_31	E0_F0 31
52	M	32	B2	3A	F0 3A
		E0_32	E0_B2	E0_3A	E0_F0 3A
53	< ,	33	B3	41	F0 41
		E0_33	E0_B3	E0_41	E0_F0 41
54	> .	34	B4	49	F0 49
		E0_34	E0_B4	E0_49	E0_F0 49
55	? /	35	B5	4A	F0 4A
		E0_35	E0_B5	E0_4A	E0_F0 4A
56***		73	F3	51	F0 51
		E0_73	E0_F3	E0_51	E0_F0 51
57	R SHIFT	36	B6	59	F0 59
		E0_36	E0_B6	E0_59	E0_F0 59
58	L CTRL	1D	9D	14	F0 14
60	L ALT	38	B8	11	F0 11
		E0_38	E0_B8	E0_11	E0_F0 11
61	Space Bar	39	B9	29	F0 29
		E0_39	E0_B9	E0_29	E0_F0 29
62	R ALT	E0 38	E0 B8	E0 11	E0 F0 11
64	R CTRL	E0 1D	E0 9D	E0 14	E0 F0 14
75	Insert	Note 1	Note 1	Note 2	Note 2
76	Delete	Note 1	Note 1	Note 2	Note 2
79	L Arrow	Note 1	Note 1	Note 2	Note 2
80	Home	Note 1	Note 1	Note 2	Note 2
81	End	Note 1	Note 1	Note 2	Note 2
83	Up Arrow	Note 1	Note 1	Note 2	Note 2
84	Dn Arrow	Note 1	Note 1	Note 2	Note 2
85	Page Up	Note 1	Note 1	Note 2	Note 2
86	Page Down	Note 1	Note 1	Note 2	Note 2
89	R Arrow	Note 1	Note 1	Note 2	Note 2
90	Num Lock	45	C5	77	F0 77
		E0_45	E0_C5	E0_77	E0_F0 77
91	Numeric 7	47	C7	6C	F0 6C
92	Numeric 4	4B	CB	6B	F0 6B
93	Numeric 1	4F	CF	69	F0 69
95	Numeric /	Note 3	Note 3	Note 3	Note 3
96	Numeric 8	48	C8	75	F0 75
97	Numeric 5	4C	CC	73	F0 73
98	Numeric 2	50	D0	72	F0 72
99	Numeric 0	52	D2	70	F0 70
100	Numeric *	37	B7	7C	F0 7C
		E0_37	E0_B7	E0_7C	E0_F0 7C
101	Numeric 9	49	C9	7D	F0 7D
102	Numeric 6	4D	CD	74	F0 74
103	Numeric 3	51	D1	7A	F0 7A
104	Numeric .	53	D3	71	F0 71
105	Numeric -	4A	CA	7B	F0 7B
106	Numeric +	4E	CE	79	F0 79
107***		7E	FE	6D	F0 6D
	DO NOT USE	E0_7E	E0_FE	E0_6D	E0_F0 6D

key location	101/102 Enhanced Keyboard	scan 1 make	scan 1 break	scan 2 make	scan 2 brake
108	Numeric Enter	E0 1C	E0 9C	E0 5A	E0 F0 5A
110	Esc	01	81	76	F0 76
		E0_01	E0_81	E0_76	E0_F0 76
112	F1	3B	BB	05	F0 05
		E0_3B	E0_BB	E0_05	E0_F0 05
113	F2	3C	BC	06	F0 06
		E0_3C	E0_BC	E0_06	E0_F0 06
114	F3	3D	BD	04	F0 05
		E0_3D	E0_BD	E0_04	E0_F0 05
115	F4	3E	BE	0C	F0 0C
		E0_3E	E0_BE	E0_0C	E0_F0 0C
116	F5	3F	BF	03	F0 03
		E0_3F	E0_BF	E0_03	E0_F0 03
117	F6	40	C0	0B	F0 0B
		E0_40	E0_C0	E0_0B	E0_F0 0B
118	F7	41	C1	83	F0 83
		E0_41	E0_C1	E0_83	E0_F0 83
119	F8	42	C2	0A	F0 0A
		E0_42	E0_C2	E0_0A	E0_F0 0A
120	F9	43	C3	01	F0 01
		E0_43	E0_C3	E0_01	E0_F0 01
121	F10	44	C4	09	F0 09
		E0_44	E0_C4	E0_09	E0_F0 09
122	F11	57	D7	78	F0 78
123	F12	58	D8	07	F0 07
124	Print Screen	Note 4	Note 4	Note 4	Note 4
125	Scroll Lock	46	C6	7E	F0 7E
		E0 46	E0_C6	E0 7E	E0_F0 7E
126	Pause	Note 5	Note 5	Note 5	Note 5
		59	D9	0F	F0 0F
		E0 59	E0_D9	E0 0F	E0_F0 0F
		5B	DB	1F	F0 1F
	Left Win	E0_5B	E0_DB	E0 1F	E0_F0 1F
		5C	DC	27	F0 27
	Right Win	E0_5C	E0_DC	E0 27	E0_F0 27
		5D	DD	2F	F0 2F
	Application	E0_5D	E0_DD	E0 2F	E0_F0 2F
		5E	DE	37	F0 37
	ACPI Power	E0_5E	E0_DE	E0 37	E0_F0 37
		5F	DF	3F	F0 3F
	ACPI Sleep	E0_5F	E0_DF	E0 3F	E0_F0 3F
	DO NOT USE	60	E0	47	F0 47
	DO NOT USE	E0_60	E0_E0	E0 47	E0_F0 47
	DO NOT USE	61	E1	4F	F0 4F
	DO NOT USE	E0_61	E0_E1	E0_4F	E0_F0 4F
		62	E2	56	F0 56
		E0_62	E0_E2	E0_56	E0_F0 56
		63	E3	5E	F0 5E
	ACPI Wake	E0_63	E0_E3	E0_5E	E0_F0 5E
		64	E4	08	F0 08
		E0_64	E0_E4	E0_08	E0_F0 08
		65	E5	10	F0 10
		E0_65	E0_E5	E0_10	E0_F0 10
		66	E6	18	F0 18
		E0_66	E0_E6	E0_18	E0_F0 18
		67	E7	20	F0 20
		E0_67	E0_E7	E0_20	E0_F0 20
		68	E8	28	F0 28
		E0_68	E0_E8	E0_28	E0_F0 28
		69	E9	30	F0 30
		E0_69	E0_E9	E0_30	E0_F0 30
		6A	EA	38	F0 38

key location	101/102 Enhanced Keyboard	scan 1 make	scan 1 break	scan 2 make	scan 2 brake
		E0_6A	E0_EA	E0_38	E0_F0 38
		6B	EB	40	F0 40
		E0_6B	E0_EB	E0_40	E0_F0 40
		6C	EC	48	F0 48
		E0_6C	E0_EC	E0_48	E0_F0 48
		6D	ED	50	F0 50
		E0_6D	E0_ED	E0_50	E0_F0 50
		6E	EE	57	F0 57
		E0_6E	E0_EE	E0_57	E0_F0 57
		6F	EF	6F	F0 6F
		E0_6F	E0_EF	E0_6F	E0_F0 6F
	DBE_KATAKANA‡	70	F0	13	F0 13
		E0_70	E0_F0	E0_13	E0_F0 13
		71	F1	19	F0 19
		E0_71	E0_F1	E0_19	E0_F0 19
		72	F2	39	F0 39
		E0_72	E0_F2	E0_39	E0_F0 39
		74	F4	53	F0 53
		E0_74	E0_F4	E0_53	E0_F0 53
		75	F5	5C	F0 5C
		E0_75	E0_F5	E0_5C	E0_F0 5C
		76	F6	5F	F0 5F
		E0_76	E0_F6	E0_5F	E0_F0 5F
	DBE_SBCSCHAR‡	77	F7	62	F0 62
		E0_77	E0_F7	E0_62	E0_F0 62
		78	F8	63	F0 63
		E0_78	E0_F8	E0_63	E0_F0 63
	CONVERT‡	79	F9	64	F0 64
		E0_79	E0_F9	E0_64	E0_F0 64
	DO NOT USE	7A	FA	65	F0 65
	DO NOT USE	E0_7A	E0_FA	E0_65	E0_F0 65
	NONCONVERT‡	7B	FB	67	F0 67
	DO NOT USE	E0_7B	E0_FB	E0_67	E0_F0 67
	DO NOT USE	7C	FC	68	F0 68
	DO NOT USE	E0_7C	E0_FC	E0_68	E0_F0 68
	DO NOT USE	7D	FD	6A	F0 6A
	DO NOT USE	E0_7D	E0_FD	E0_6A	E0_F0 6A
	DO NOT USE	7F	FF	6E	F0 6E
	DO NOT USE	E0_7F	E0_FF	E0_6E	E0_F0 6E

Note *   Key 29 is available on the US and not on the International Keyboard.
Note **   Keys 42 and 45 are available on the International Keyboard and not on the US Keyboard.
Note *** Keys 56 and 107 are used on Brazilian and some Far East keyboards. They not available on US Keyboards.
Note ‡   Keys are used for Far East keyboards.

Note 1 for Scan Code 1:

These keys may have additional "shift" and/or "unshift" scan codes preceding the **Base Make** code and following the **Base Break** code, depending upon the current state of **Num Lock** and the state of **Shift** key/s (when multiple keys are held down at the same time).

These Keys are Typematic:

Key Location	US key assignment	Base Make	Base Break
75	Insert	E0 52	E0 D2
76	Delete	E0 53	E0 D3

79	Left Arrow	E0 4B	E0 CB
80	Home	E0 47	E0 C7
81	End	E0 4F	E0 CF
83	Up Arrow	E0 48	E0 C8
84	Dn Arrow	E0 50	E0 D0
85	Page Up	E0 49	E0 C9
86	Page Down	E0 51	E0 D1
89	Right Arrow	E0 4D	E0 CD

Num Lock ON	**Precede Base Make code with**	**follow Base Break code with**
Final Key only	E0 2A	E0 AA
LSHIFT down + **Final Key**		
RSHIFT down **+ Final Key**		
Both **LSHIFT** and **RSHIFT** down **+ Final Key**		

Num Lock OFF	**Precede Base Make code with**	**follow Base Break code with**
Final Key only		
LSHIFT down + **Final Key**	E0 AA	E0 2A
RSHIFT down **+ Final Key**	E0 B6	E0 36
Both **LSHIFT** and **RSHIFT** down **+ Final Key**	E0 AA E0 B6	E0 36 E0 2A

As an example, consider the case where **Num Lock** is **OFF** and you choose to push and hold **Left SHIFT**, then push and hold **Right SHIFT**, and finally push **Insert.** The scan codes sent would be:

2A	(**LSHIFT** make)
36	(**RSHIFT** make)
E0 AA E0 B6 E0 52	(**Insert** make)

When these keys are released, the keys still held down at the time a key is released determine it's scan code. Assume **Insert** is released first then **Right SHIFT** and last **Left SHIFT**. The scan codes would be:

E0 D2 E0 36 E0 2A	(**Insert** break)
B6	(**RSHIFT** break)
AA	(**LSHIFT** break)

Note 2 for Scan Code 2:

These keys may have additional "shift" and/or "unshift" scan codes preceding the **Base Make** code and following the **Base Break** code, depending upon the current state of **Num** Lock and the state of **SHIFT** key/s (when multiple keys are held down at the same time).

These Keys are Typematic

key location	US key assignment	Base Make	Base Break
75	Insert	E0 70	E0 F0 70
76	Delete	E0 71	E0 F0 71
79	Left Arrow	E0 6B	E0 F0 6B
80	Home	E0 6C	E0 F0 6C
81	End	E0 69	E0 F0 69
83	Up Arrow	E0 75	E0 F0 75

84	Dn Arrow	E0 72	E0 F0 72
85	Page Up	E0 7D	E0 F0 7D
86	Page Down	E0 7A	E0 F0 7A
89	Right Arrow	E0 74	E0 F0 74

Num Lock ON	**Precede Base Make code with**	**follow Base Break code with**
Final Key only	E0 12	E0 F0 12
LSHIFT down + **Final Key**		
RSHIFT down + **Final Key**		
Both **LSHIFT** and **RSHIFT** down + **Final Key**		

Num Lock OFF	**Precede Base Make code with**	**follow Base Break code with**
Final Key only		
LSHIFT down + **Final Key**	E0 F0 12	E0 12
RSHIFT down + **Final Key**	E0 F0 59	E0 59
Both **LSHIFT** and **RSHIFT** down + **Final Key**	E0 F0 12 E0 F0 59	E0 59 E0 12

Note 3 Key 95 for Scan Codes 1 and 2:

This Key is Typematic **(Numeric / on US keyboards)**

Scan Code	base	L-SHIFT + base	R-SHIFT + base	L-SHIFT + R-SHIFT + base
1 make	E0 35	E0 AA E0 35	E0 B6 E0 35	E0 AA E0 B6 E0 35
1 break	E0 B5	E0 B5 E0 2A	E0 B5 E0 36	E0 B5 E0 36 E0 2A
2 make	E0 4A	E0 F0 12 E0 4A	E0 F0 59 E0 4A	E0 F0 12 E0 F0 59 E0 4A
2 break	E0 F0 4A	E0 F0 4A E0 12	E0 F0 4A E0 59	E0 F0 4A E0 59 E0 12

Note 4 Key 124 for Scan Codes 1 and 2:

This Key is Typematic **(Print Screen on US keyboards)**

Scan Code	Make	Break	LCtrl or RCtrl + LSHIFT or RSHIFT + Make	LCtrl or RCtrl + LSHIFT or RSHIFT + Break	LAlt or RAlt + Make	LAlt or RAlt + Break
1	E0 2A E0 37	E0 B7 E0 AA	E0 37	E0 B7	54	D4
2	E0 12 E0 7C	E0 F0 7C E0 F0 12	E0 7C	E0 F0 7C	84	F0 84

Note 5 Key 126 for Scan Codes 1 and 2:

This Key is not Typematic and has no Break Code **(Pause on US keyboards)**

Scan Code	Make	L-Ctrl or R-Ctrl + Make
1	E1 1D 45 E1 9D C5	E0 46 E0 C6
2	E1 14 77 E1 F0 14 F0 77	E0 7E E0 F0 7E

Additional General Requirements

Power On Reset Routine

During a Power On Reset condition, this keyboard must perform a self-test and respond with a diagnostic completion code (AA for pass, otherwise return an error code). Completion codes must be sent within 450 msecs and 2.5 seconds after the Power On Reset. After Reset, this keyboard defaults to Scan Code Set 2.

Reset Command

Upon receiving a FF (soft Reset command) from the host, this keyboard responds with a completion code, within 300 and 500 msecs after acknowledging the Reset command.

Keyboard Command Acknowledge

All keyboard commands are acknowledged with an FA (ACK) or FE (Re-send). The Acknowledge response time must be within 20 msecs.

Appendix B: Device Class Power Management v1.0a

Scope

This specification defines the behavior of the input device class as it relates to power management, and, specifically, to the four device power states defined for the OnNow Architecture. This specification applies to standard types of input devices such as keyboards, keypads, mice, pointing devices, joysticks, game pads, to devices that combine these kinds of input functionality (composite devices, etc.), and to new types of input devices such as virtual reality devices, simulation devices, etc. It is intended that input device vendors will be able to design consistent power-manageable products, and that OS vendors will be able to implement an appropriate input device power management policy based on the contents of this specification.

General Device Power Management Considerations

In the OnNow architecture, power management of individual devices is the responsibility of a policy owner in the operating system, generally a class-specific driver. This policy-owner will implement a power conservation policy that is appropriate for devices in its class. The policy will operate in conjunction with a global system power policy implemented in the operating system (i.e., is the system Working or Sleeping?). In general, the device-class power conservation policy strives to reduce power consumption while the system is Working by transitioning amongst various available power states according to device usage.

Since the policy-owner in the operating system has very specific knowledge of when a device is in use, or potentially in use, there is no need for hardware timers or such to determine when to make these transitions. Similarly, this level of understanding of device usage makes it possible to use fewer device power states. Generally, intermediate states attempt to draw a compromise between latency and consumption due to the uncertainty of actual device usage. With the increased knowledge in the OS, crisp decisions can be made about whether the device is needed at all. With this ability to turn devices off more frequently, the benefit of having intermediate states diminishes.

The policy-owner also determines what class-specific events can cause the system to transition from Sleeping to Working, and enables this functionality based on application or user requests. Note that the definition of the wake-up events that each class supports will influence the system's global power policy in terms of the level of power conservation the Sleeping state can attain while still meeting wake-up latency requirements set by applications or the user.

In the OnNow architecture, bus drivers also implement power policy for their bus class (e.g. PCI, USB, etc.). In general, the Bus driver has responsibility for tracking the device power states of all devices on its bus, and transitioning the Bus itself to only those power states that are consistent with those of its devices. This means that the Bus state can be no lower than the highest state of one of it's devices. However, enabled wake-up events can affect this as well. For example if a particular device is in the D2 state and set to wake-up the system, and the bus can only forward wake-up requests while in the D1 state, then the Bus must remain in the D1 state even if all devices are in a lower state.

Device power state transitions are explicitly commanded by the driver and invoked through bus-specific mechanisms (e.g., ATA Standby command, USB Suspend, etc.). In some cases, bus-specific mechanisms are not available and device-specific mechanisms must be used. Note that the explicit command for entering the D3 state may be the removal of power.

The following definitions apply to devices of all classes:

- **D0**:   Device is on and running. It is receiving full power from the system, and is delivering full functionality to
the user.

- **D1**: Class-specific low-power state (defined below) in which device context may or may not be lost.
- **D2**: Class-specific low-power state (defined below) in which device context may or may not be lost. Attains
  greater power savings than D1.
- **D3**: Device is off and not running. Device context is lost. Power may be removed from the device.

Input Device Power State Definitions

D0

Input device is on and running:
- device is receiving full power from its power source,
- device is delivering full functionality to the user,
- device is preserving applicable context and state information.

D1

Input device power consumption is greatly reduced:
- in general, device is in a power conservation state and is not delivering any functionality to the user except wake-up functionality if applicable,
- device status, state, or other information indicators (e.g., LED's, LCD displays, etc.) are turned off to save power,
- the following device context and state information should be preserved by device driver software:
  - Keyboard: Num, Caps, Scroll Lock states – and Compose and Kana states if applicable - and associated LED/indicator states, Repeat delay, Repeat rate,
  - Joystick: forced feedback effects (if applicable),
  - Any input device: all context and state information that cannot be preserved by the device when it's conserving power.

D2

This state is not defined for input devices, use D1 as the power conservation state instead.

D3   (Power may be removed)

Input device is off and not running:
- in general, device is off and is not delivering any functionality to the user except wake-up functionality if applicable,
- device context and state information is lost.

Input Device Power Conservation Policy

The runtime power conservation policy for input devices is simple and straightforward:
- devices are by default always on (D0).
- devices transition to other power states only when requested by the system (or when power is removed).
- devices with wake-up capability enabled can request a power state transition by generating a wake-up event.

D3	D0	● Requested by system.
D0	D1 or D3*	● Requested by system (e.g. system goes to sleep w/ wake-up enabled).
D0, D1	D3	● Requested by system (e.g. system goes to sleep w/ wake-up disabled).

		● Power is removed.
D1, D3	D0	● Device with wake-up capability enabled request transition by generating a wake-up event; request is ultimately granted by system.
		● Requested by system.

* Depends on capability of device (if device features D1 or D3 wake-up capability or not); device will be put in state with lowest possible power consumption.

Input Device Wake-up Events

It is recommended, but not required, that input devices implement and support bus-specific wake-up mechanisms if these are defined for their bus type. This is recommended because a user typically uses an input device of some kind to wake up the system when it is in a power conservation state, for example, when the system is sleeping.

The actual input data (particular button or key pressed, etc.) that's associated with a wake-up event should never be discarded by the device itself, but should always be passed along to device driver(s) for further interpretation. It is up to the device driver(s) to implement a policy for how this input data should be interpreted, and decide what should be passed along to higher-level software, etc.

It is recommended that the device button(s) or key(s) used for power management purposes are clearly labeled with text and/or icons. This is recommended for keyboards and other input devices on which all buttons or keys are typically labeled with text and/or icons to identify their usage.

Examples of wake-up events include (but are not limited to):

Keyboards:
- press special purpose power management button (e.g., POWER button) to generate wake-up event, OR
- press designated dual usage key (e.g., Spacebar key) to generate wake-up event.

Mice and pointing devices:
- press any button (e.g., left or right button) to generate wake-up event.

Any other type of input device:
- press special purpose power management button (e.g., POWER button) to generate wake-up event, OR
- press designated dual usage button or key to generate wake-up event.

Examples of more advanced wake-up events include keyboard wake-up signaling when any key is pressed, mouse wake-up signaling on detection of X/Y motion, joystick wake-up signaling on X/Y motion, etc. However, in order to avoid accidental or unintentional wake-up of the system, and to give the user some control over which input events will result in a system wake-up, it's suggested that more advanced types of wake-up events are implemented as features that can be turned on/off by the user via, for example, a control panel type of software application.

Minimum Input Device Power Capabilities

D0	Mandatory.
D1	Optional.
D2	Not defined for input devices.
D3	Mandatory.

Recommendations for Human Interface Devices

Input devices compliant with the Human Interface Devices (HID) firmware specification v1.0 should deploy usages defined in the supplemental HID Usage Table specification v1.0 for power management buttons, i.e. POWER and SLEEP buttons. The Generic Desktop Page (page 0x01) has been extended with the following usages compared to what's defined in the core HID 1.0 specification:

- **System Control** (Usage ID 0x80),
- System Power Down (Usage ID 0x81),
- System Sleep (Usage ID 0x82),
- System Wake Up (Usage ID 0x83).

These usages should be used in following manner with the power management buttons:

POWER button:

- Should send usage "System Power Down" when the button is pressed to power down the system. The system power policy manager will look up the POWER button action in the current power policy (by default, shutdown) and take that action.

- Should send usage "System Wake Up" when the button is pressed again to power up the system.

SLEEP button:

- Should send usage "System Sleep" when the button is pressed to put the system to sleep. The system power policy manager will look up the SLEEP button action in the current power policy (by default, sleep) and take that action.

- Should send usage "System Wake Up" when the button is pressed again to wake up the system.

The power management buttons should be reported in a **System Control** application level collection (a.k.a. "top level" collection) in order to be interpreted correctly by the host system software. Power management buttons appearing in other application level collections will be ignored.

Note: In previous revisions of this specification, the "Keyboard Power" usage (index 102dec/66hex) as defined in the Keyboard/Keypad Page (page 0x07) in the core HID 1.0 specification was recommended as the way to implement a POWER button on a keyboard. However, this recommendation should not be followed anymore and has been replaced with the recommendations above.

Recommendations for i8042 keyboards

i8042-Based keyboards should deploy the following scan codes for power management buttons, i.e., POWER and SLEEP buttons:

Power event
Set1:   Make = E0, 5E Break = E0, DE
Set2:   Make = E0, 37 Break = E0, F0, 37

Sleep event
Set1:   Make = E0, 5F Break = E0, DF
Set2:   Make = E0, 3F Break = E0, F0, 3F

Wake event

Set1:   Make = E0, 63 Break = E0, E3
Set2:   Make = E0, 5E Break = E0, F0, 5E

The Power, Sleep, and Wake event scan codes are the i8042 equivalents to the System Power Down, System Sleep, and System Wake Up HID usages as defined above.

Appendix C: USB Keyboard/Keypad Page (0x07)

This section is the **Usage Page** for key codes to be used in implementing a USB keyboard. A **Boot Keyboard** (84-, 101- or 104-key) should at a minimum support all associated usage codes as indicated in the "Boot" column below.

Note: Due to the variation of keyboards from language to language, it is not feasible to specify exact key mappings for every language. Where this list is not specific for a key function in a language, the closest equivalent key position should be used, so that a keyboard may be modified for a different language by simply printing different keycaps. One example is the Y key on a North American keyboard. In Germany, this is typically Z. Rather than changing the keyboard firmware to put the Z Usage into that place in the descriptor list, the vendor should use the Y Usage on both the North American and German keyboards. This continues to be the existing practice in the industry, in order to minimize the number of changes to the electronics to accommodate other languages.

Usage index (dec)	Usage Index (hex)	Usage	Ref:typical AT-101 position	PC-AT	Mac-intosh	UNIX	Boot
0	00	Reserved (no event indicated) [9]	N/A	√	√	√	84/101/104
1	01	Keyboard ErrorRollOver [9]	N/A	√	√	√	84/101/104
2	02	Keyboard POSTFail [9]	N/A	√	√	√	84/101/104
3	03	Keyboard ErrorUndefined [9]	N/A	√	√	√	84/101/104
4	04	Keyboard a and A [4]	31	√	√	√	84/101/104
5	05	Keyboard b and B	50	√	√	√	84/101/104
6	06	Keyboard c and C [4]	48	√	√	√	84/101/104
7	07	Keyboard d and D	33	√	√	√	84/101/104
8	08	Keyboard e and E	19	√	√	√	84/101/104
9	09	Keyboard f and F	34	√	√	√	84/101/104
10	0A	Keyboard g and G	35	√	√	√	84/101/104
11	0B	Keyboard h and H	36	√	√	√	84/101/104
12	0C	Keyboard i and I	24	√	√	√	84/101/104
13	0D	Keyboard j and J	37	√	√	√	84/101/104
14	0E	Keyboard k and K	38	√	√	√	84/101/104
15	0F	Keyboard l and L	39	√	√	√	84/101/104
16	10	Keyboard m and M [4]	52	√	√	√	84/101/104
17	11	Keyboard n and N	51	√	√	√	84/101/104
18	12	Keyboard o and O [4]	25	√	√	√	84/101/104
19	13	Keyboard p and P [4]	26	√	√	√	84/101/104
20	14	Keyboard q and Q [4]	17	√	√	√	84/101/104
21	15	Keyboard r and R	20	√	√	√	84/101/104
22	16	Keyboard s and S [4]	32	√	√	√	84/101/104
23	17	Keyboard t and T	21	√	√	√	84/101/104
24	18	Keyboard u and U	23	√	√	√	84/101/104
25	19	Keyboard v and V	49	√	√	√	84/101/104
26	1A	Keyboard w and W [4]	18	√	√	√	84/101/104
27	1B	Keyboard x and X [4]	47	√	√	√	84/101/104
28	1C	Keyboard y and Y [4]	22	√	√	√	84/101/104
29	1D	Keyboard z and Z [4]	46	√	√	√	84/101/104
30	1E	Keyboard 1 and ! [4]	2	√	√	√	84/101/104
31	1F	Keyboard 2 and @ [4]	3	√	√	√	84/101/104

Usage index (dec)	Usage Index (hex)	Usage	Ref:typical AT-101 position	PC-AT	Mac-intosh	UNIX	Boot
32	20	Keyboard 3 and #[4]	4	√	√	√	84/101/104
33	21	Keyboard 4 and $[4]	5	√	√	√	84/101/104
34	22	Keyboard 5 and %[4]	6	√	√	√	84/101/104
35	23	Keyboard 6 and ^[4]	7	√	√	√	84/101/104
36	24	Keyboard 7 and &[4]	8	√	√	√	84/101/104
37	25	Keyboard 8 and *[4]	9	√	√	√	84/101/104
38	26	Keyboard 9 and ([4]	10	√	√	√	84/101/104
39	27	Keyboard 0 and)[4]	11	√	√	√	84/101/104
40	28	Keyboard Return(ENTER)[5]	43	√	√	√	84/101/104
41	29	Keyboard ESCAPE	110	√	√	√	84/101/104
42	2A	Keyboard DELETE (Backspace)[13]	15	√	√	√	84/101/104
43	2B	Keyboard Tab	16	√	√	√	84/101/104
44	2C	Keyboard Spacebar	61	√	√	√	84/101/104
45	2D	Keyboard - and (underscore)[4]	12	√	√	√	84/101/104
46	2E	Keyboard = and+[4]	13	√	√	√	84/101/104
47	2F	Keyboard [and {[4]	27	√	√	√	84/101/104
48	30	Keyboard] and }[4]	28	√	√	√	84/101/104
49	31	Keyboard \ and		29	√	√	√	84/101/104
50	32	Keyboard Non-US# and ~[2]	42	√	√	√	84/101/104
51	33	Keyboard [4]	40	√	√	√	84/101/104
52	34	Keyboard ' and "[4]	41	√	√	√	84/101/104
53	35	Keyboard Grave Accent and Tilde[4]	1	√	√	√	84/101/104
54	36	Keyboard , and <[4]	53	√	√	√	84/101/104
55	37	Keyboard . and >[4]	54	√	√	√	84/101/104
56	38	Keyboard / and ?[4]	55	√	√	√	84/101/104
57	39	Keyboard CapsLock[11]	30	√	√	√	84/101/104
58	3A	Keyboard F1	112	√	√	√	84/101/104
59	3B	Keyboard F2	113	√	√	√	84/101/104
60	3C	Keyboard F3	114	√	√	√	84/101/104
61	3D	Keyboard F4	115	√	√	√	84/101/104
62	3E	Keyboard F5	116	√	√	√	84/101/104
63	3F	Keyboard F6	117	√	√	√	84/101/104
64	40	Keyboard F7	118	√	√	√	84/101/104
65	41	Keyboard F8	119	√	√	√	84/101/104
66	42	Keyboard F9	120	√	√	√	84/101/104
67	43	Keyboard F10	121	√	√	√	84/101/104
68	44	Keyboard F11	122	√	√	√	101/104
69	45	Keyboard F12	123	√	√	√	101/104
70	46	Keyboard PrintScreen[1]	124	√	√	√	101/104
71	47	Keyboard ScrollLock[11]	125	√	√	√	84/101/104
72	48	Keyboard Pause[1]	126	√	√	√	101/104
73	49	Keyboard Insert[1]	75	√	√	√	101/104
74	4A	Keyboard Home[1]	80	√	√	√	101/104

Usage index (dec)	Usage Index (hex)	Usage	Ref:typical AT-101 position	PC-AT	Mac-intosh	UNIX	Boot
75	4B	Keyboard PageUp[1]	85	√	√	√	101/104
76	4C	Keyboard Delete Forward[1]	76	√	√	√	101/104
77	4D	Keyboard End[1]	81	√	√	√	101/104
78	4E	Keyboard PageDown[1]	86	√	√	√	101/104
79	4F	Keyboard RightArrow[1]	89	√	√	√	101/104
80	50	Keyboard LeftArrow[1]	79	√	√	√	101/104
81	51	Keyboard DownArrow[1]	84	√	√	√	101/104
82	52	Keyboard UpArrow[1]	83	√	√	√	101/104
83	53	Keypad NumLock and Clear[11]	90	√	√	√	101/104
84	54	Keypad /[1]	95	√	√	√	101/104
85	55	Keypad *	100	√	√	√	84/101/104
86	56	Keypad -	105	√	√	√	84/101/104
87	57	Keypad +	106	√	√	√	84/101/104
88	58	Keypad ENTER5	108	√	√	√	101/104
89	59	Keypad 1 and End	93	√	√	√	84/101/104
90	5A	Keypad 2 and Down Arrow	98	√	√	√	84/101/104
91	5B	Keypad 3 and PageDn	103	√	√	√	84/101/104
92	5C	Keypad 4 and Left Arrow	92	√	√	√	84/101/104
93	5D	Keypad 5	97	√	√	√	84/101/104
94	5E	Keypad 6 and Righ tArrow	102	√	√	√	84/101/104
95	5F	Keypad 7 and Home	91	√	√	√	84/101/104
96	60	Keypad 8 and Up Arrow	96	√	√	√	84/101/104
97	61	Keypad 9 and PageUp	101	√	√	√	84/101/104
98	62	Keypad 0 and Insert	99	√	√	√	84/101/104
99	63	Keypad . and Delete	104	√	√	√	84/101/104
100	64	Keyboard Non-US\ and |[3;6]	45	√	√	√	84/101/104
101	65	Keyboard Application[10]	129	√		√	104
102	66	Keyboard Power[9]			√	√	
103	67	Keypad =			√		
104	68	Keyboard F13			√		
105	69	Keyboard F14			√		
106	6A	Keyboard F15			√		
107	6B	Keyboard F16					
108	6C	Keyboard F17					
109	6D	Keyboard F18					
110	6E	Keyboard F19					
111	6F	Keyboard F20					
112	70	Keyboard F21					
113	71	Keyboard F22					
114	72	Keyboard F23					
115	73	Keyboard F24					
116	74	Keyboard Execute				√	
117	75	Keyboard Help				√	
118	76	Keyboard Menu				√	
119	77	Keyboard Select				√	
120	78	Keyboard Stop				√	

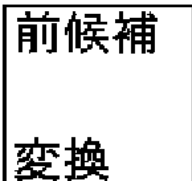| Usage index (dec)	Usage Index (hex)	Usage	Ref:typical AT-101 position	PC-AT	Mac-intosh	UNIX	Boot
121	79	Keyboard Again				√	
122	7A	Keyboard Undo				√	
123	7B	Keyboard Cut				√	
124	7C	Keyboard Copy				√	
125	7D	Keyboard Paste				√	
126	7E	Keyboard Find				√	
127	7F	Keyboard Mute				√	
128	80	Keyboard Volume Up				√	
129	81	Keyboard Volume Down				√	
130	82	Keyboard Locking Caps Lock[12]				√	
131	83	Keyboard Locking Num Lock[12]				√	
132	84	Keyboard Locking Scroll Lock [12]				√	
133	85	Keypad Comma					
134	86	Keypad Equal Sign					
135	87	Keyboard Kanji1[15]					
136	88	Keyboard Kanji2[16]					
137	89	Keyboard Kanji3[17]					
138	8A	Keyboard Kanji4[18]					
139	8B	Keyboard Kanji5[19]					
140	8C	Keyboard Kanji6[20]					
141	8D	Keyboard Kanji7[21]					
142	8E	Keyboard Kanji8[22]					
143	8F	Keyboard Kanji9[22]					
144	90	Keyboard LANG1[8]					
145	91	Keyboard LANG2[8]					
146	92	Keyboard LANG3[8]					
147	93	Keyboard LANG4[8]					
148	94	Keyboard LANG5[8]					
149	95	Keyboard LANG6[8]					
150	96	Keyboard LANG7[8]					
151	97	Keyboard LANG8[8]					
152	98	Keyboard LANG9[8]					
153	99	Keyboard AlternateErase[7]					
154	9A	Keyboard SysReq/Attenti[1]					
155	9B	Keyboard Cancel					
156	9C	Keyboard Clear					
157	9D	Keyboard Prior					
158	9E	Keyboard Return					
159	9F	Keyboard Separator					
160	A0	Keyboard Out					
161	A1	Keyboard Oper					
162	A2	Keyboard Clear/Again					
163	A3	Keyboard CrSel/Props					
164	A4	Keyboard ExSel					
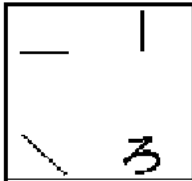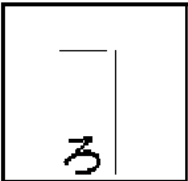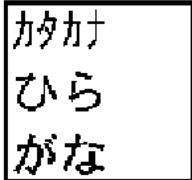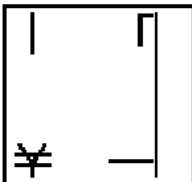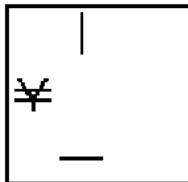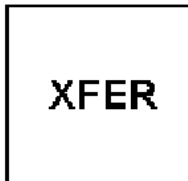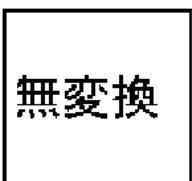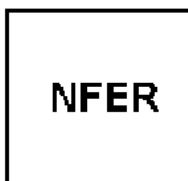
Usage index (dec)	Usage Index (hex)	Usage	Ref:typical AT-101 position	PC-AT	Mac-intosh	UNIX	Boot
165-223	A5-DF	Reserved					
224	E0	Keyboard LeftControl	58	√	√	√	84/101/104
225	E1	Keyboard LeftShift	44	√	√	√	84/101/104
226	E2	Keyboard LeftAlt	60	√	√	√	84/101/104
227	E3	Keyboard Left GUI[10;23]	127	√	√	√	104
228	E4	Keyboard RightControl	64	√	√	√	101/104
229	E5	Keyboard RightShift	57	√	√	√	84/101/104
230	E6	Keyboard RightAlt	62	√	√	√	101/104
231	E7	Keyboard Right GUI[10;24]	128	√	√	√	104
232-255	E8-FF	Reserved					

Footnotes

1. Usage of keys is not modified by the state of the Control, Alt, Shift or Num Lock keys. That is, a key does not send extra codes to compensate for the state of any Control, Alt, Shift or Num Lock keys.
2. Typical language mappings: US: \| Belg: µ`£ FrCa: <}> Dan:'* Dutch: <> Fren:*µ Ger: #' Ital: ù§ LatAm: }`] Nor:,* Span:}Ç Swed: ,* Swiss: $£ UK: #~.
3. Typical language mappings: Belg:<\> FrCa:«°» Dan:<\> Dutch:]|[Fren:<> Ger:<|> Ital:<> LatAm:<> Nor:<> Span:<> Swed:<|> Swiss:<\> UK:\| Brazil: \|.
4. Typically remapped for other languages in the host system.
5. Keyboard Enter and Keypad Enter generate different Usage codes.
6. Typically near the Left-Shift key in AT-102 implementations.
7. Example, Erase-Eaze™ key.
8. Reserved for language-specific functions, such as Front End Processors and Input Method Editors.
9. Reserved for typical keyboard status or keyboard errors. Sent as a member of the keyboard array. Not a physical key.
10. Microsoft Windows key for Microsoft Windows 95 and "Compose."
11. Implemented as a non-locking key; sent as member of an array.
12. Implemented as a locking key; sent as a toggle button. Available for legacy support; however, most systems should use the non-locking version of this key.
13. Backs up the cursor one position, deleting a character as it goes.
14. Deletes one character without changing position.
15. See next page
16. See next page
17. See next page
18. See next page
19. See next page
20. See next page
125708044.   Toggle Double-Byte/Single-Byte mode.
125708045.   Undefined, available for other Front End Language Processors.
125708046.   Windowing environment key, examples are Microsoft Left Win key, Macintosh Left Apple key, Sun Left Meta key
125708047.   Windowing environment key, examples are Microsoft Right Win key, Macintosh Right Apple key,
Sun Right Meta key.

Footnotes 15–20

Note	AT-104	DOS/V-109 (suggested)	PC98 (suggested)
15	No function	＿ ｜ ＼ ろ	＿｜ ろ
16	No function	カタカナ ひら がな	かな
17	No function	｜ 「 ￥ ＿	｜ ￥ ＿
18	No function	前候補 変換	XFER
19	No function	無変換	NFER
20	No function	No function	, (カンマ)