Open Firmware Recommended Practice:

Universal Serial Bus

Version 1

June 1, 1998

Published by the Open Firmware Working Group

This document is a voluntary-use recommended practice of the Open Firmware Working Group. The Open Firmware Working Group is an ad hoc committee composed of individuals interested in Open Firmware as defined by IEEE 1275-1994, related standards, and their application to various computer systems.

The Open Firmware Working Group is involved both in IEEE sanctioned standards activities, whose final results are published by IEEE, and in informal recommendations such as this, which are published on the Internet at:

http://playground.sun.com/pub/1275

Membership in the Open Firmware Working Group is open to all interested parties. The working group meets at regular intervals at various locations. For more information send email to:

p1275-wg@risc.sps.mot.com

Revision History

1. Overview and References

This document describes the application of Open Firmware to the Universal Serial Bus (USB) and addresses nodes representing USB devices.

Since there is no provision for FCode on USB devices, this binding restricts itself to providing a common framework for a USB device tree. Implementations that support USB devices as boot, input, and output devices, will define their own methods and properties to communicate with USB endpoints. Those methods are beyond the scope of this document.

1.1. References

 [1] IEEE Standard 1275-1994 Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices

[2] Universal Serial Bus Specification, 1.0

1.2. Definitions of Terms

combined node: A device tree node that represents both a USB device and a USB interface.

device node: A device tree node that represents a USB device.

USB host controller: A hardware device that interfaces between a computer system and a USB bus.

host controller node: A device tree node that represents a USB host controller.

hub node: 1. A device tree node that represents a USB hub. 2. A combined node whose USB interface is a USB hub.

interface node: A device tree node that represents a USB interface.

low speed: 1.5 Mbs, rather than the normal 12.0 Mbs signalling speed. See [2].

transaction: The delivery of service to an endpoint. See [2].

transfer: One or more transactions. See [2].

USB device: A hardware device that connects to a USB bus.

USB hub: A USB interface that provides additional connections to the USB bus.

USB interface: An independent piece of functionality of a USB device.

1.2.1. USB Descriptors

[2] defines a number of data structures that are referenced in this document. They are:

DEVICE

See [2] section 9.6.1. Fields referenced by this binding: bDeviceClass, bDeviceSubClass, bDeviceProtocol, idVendor, idProduct, bcdDevice, bNumConfigurations.

CONFIGURATION

See [2] section 9.6.2. Fields referenced by this binding: bNumInterfaces, bConfigurationValue.

INTERFACE

See [2] section 9.6.3. Fields referenced by this binding: bInterfaceNumber, bAlternateSetting, bInterfaceClass, bInterfaceSubClass, bInterfaceProtocol.

2. Bus Characteristics

USB is not a memory-mapped bus. An operation with a USB device is performed by executing a transfer consisting of one or more bus transactions to move information between the host system and the device. A bus transaction consists of multiple packets, including a token packet, possibly a data packet, and possibly a handshake packet. The specific packets are allowed and/or required based on the transaction type. Four transfer types are defined: control, interrupt, bulk, and isochronous.

2.1. Bus-specific Configuration Variables

None.

2.2. Format of a Probe List

None.

2.3. FCode Interpretation Semantics

None (USB has no provision for device identification via FCode).

3. Device Tree Structure

This document defines four types of device tree nodes for USB: host controller nodes, device nodes, interface nodes, and combined nodes.

Generally, USB devices are represented as two levels of device tree nodes: a device node representing the entire USB device, with one or more child interface nodes representing the individual USB interfaces on the device. For special cases the device and interface nodes are combined into a single combined node.

3.1. Host Controller Nodes

3.1.1. Host Controller Node Properties

3.1.1.1. Open Firmware-defined Properties for Host Controller Nodes

The following standard properties, as defined in [1], have special meaning or interpretation for host controller nodes.

```
"#address-cells"
Standard property to define the address format. Its value shall be 1, encoded as with encode-int.
```

"#size-cells" Standard prop-name to define the package's address size format.

prop-encoded-array: 0, encoded as with encode-int.

The value of "#size-cells" for host controller nodes shall be 0, representing the fact that host controller addresses are an enumeration rather than memory-like address ranges.

3.1.1.2. Bus-specific Properties for Host Controller Nodes

None.

3.1.2. Host Controller Node Methods

S

S

3.1.2.1. Open Firmware-defined Methods for Host Controller Nodes

A package implementing a host controller node shall implement the following standard methods as defined in [1], with physical address representation as specified in section 3.2.1.

decode-unit Convert text unit-string to physical address.	(addr len port)	М
encode-unit Convert physical address to text unit-string.	(port addr len)	М

3.1.2.2. Bus-specific Methods for Host Controller Nodes

None.

3.2. Device Nodes

Device nodes represent USB devices. USB devices may require device-specific support to enable choice of the USB device configuration and/or to enable operation of multiple USB interfaces in a coordinated fashion.

Unless otherwise specified, a device node shall be created for each USB device.

3.2.1. Device Node Address Representation

The textual representation of a device node unit address shall be the number of the USB hub port or the USB host controller port to which this USB device is attached, in lower case hexadecimal with leading zeroes suppressed.

3.2.2. Device Node Properties

3.2.2.1. Open Firmware-defined Properties for Device Nodes

The following properties, as defined in [1], have special meanings or interpretations for device nodes. These properties shall be created for each device node created.

The following notation is used:

From the DEVICE descriptor for this USB device:

VID	idVendor
PID	idProduct
REV	bcdDevice
DC	bDeviceClass
DSC	bDeviceSubClass
DPROTO	bDeviceProtocol

From the CONFIGURATION descriptor for this USB device:

CN

bConfigurationValue for the selected configuration.

The textual representation of VID, PID, REV, DC, DSC, DPROTO, and CN shall be in lower case hexadecimal with leading zeroes suppressed.

"#address-cells"

Standard property to define the address format. Its value shall be 2, encoded as with encode-int.

S

Standard prop-name 4 prop-encoded-array; a list of strings, each encoded with encode-string and concatenated with encode+. 5 Construct encoded strings as enumerated in the following list, All of the applicable strings shall be included in the value of " compact Lible". Their relative ouder shall be as in the list, string 1) before string 2), and so forth. 9 Strings containing a device class (5.10) shall be omitted if the device class is 0. 11 1) usbVID.PID.REV 13 2) usbVID.PID.REV 14 14 1) usbVID.PID.REV 15 15 4) usbVID.PID.REV 16 16 5) usbVID.classDC.DSC.DPROTO 16 17 usbclassDC.DSC.DPROTO 17 18 usbclassDC.DSC.DPROTO 18 19 usbclassDC.DSC 19 21 1) usbclassDC 21 22 11 usbclassDC.DSC 21 23 usbclassDC.DSC 21 21 24 Note: entry 11), "usb.device", provides a mechanism to bind a generic driver to a USB device. 25 * * 5 26 * * 5 27 * name * S 5<	1	"compatible'	n				S
3 prop-encoded-array; a list of strings, each encoded with encode-string and concatenated with encode+. 5 Construct encoded strings as enumerated in the following list. All of the applicable strings shall be included in the value of * compactible*. Their relative order shall be as in the list, string 1) before string 2), and so forth. 6 Construct encoded strings as commerated in the following list. All of the applicable strings ontaining a device class (5.10) shall be omitted if the device class is 0. 11 usbvID.PJD.REV.configCN 2 usbvID.PJD.REV 3 usbvID.PJD.REV 4 3) usbvID.PJD.REV 5 usbvID.PJD.REV 6 yusbvID.PLBASDC.DSC.DPROTO 6 yusbclassDC.DSC.DPROTO 7 usbvID.elassDC 8 usbclassDC.DSC 9 usbclassDC.DSC 10 usbclassDC.DSC 21 10 11 usbclassDC.DSC 22 Note: entry 11), "usb.device", provides a mechanism to bind a generic driver to a USB device. 23 prop-encoded-array: a string encoded with encode-string. 24 Prop-encoded-array: a string encoded with encode-string. 25 fmame* S 26 storage <t< td=""><td>2</td><td>Standard prop-</td><td>name</td><td></td><td></td><td></td><td></td></t<>	2	Standard prop-	name				
5 Construct encoded strings as enumerated in the following list. All of the applicable strings shall be included in the value of "compatible". Their relative order shall be as in the list, string 1) before string 2), and so forth. 9 Strings containing a configuration number (1 and 3) shall be omitted for a single-configuration USB device. 10 usbVID.PID.REV.configCN 2 usbVID.PID.REV 11 usbVID.PID.REV 23 usbVID.PID.REV 14 usbVID.PID.REV 15 usbVID.PID.REV 16 \$) usbVID.pID.REV 17 usbVID.PID.REV 18 usbVID.PID.CONFIGCN 19 usbVID.PID.CONC 10 usbclassDC.DSC.DPROTO 10 usbclassDC 21 usbclassDC.DSC 21 10 usbclassDC \$ 23 usbclassDC 24 Note: entry 11), "usb.device", provides a mechanism to bind a generic driver to a USB device. 25 ************************************	3 4	prop-encoded-a	array; a list of strin	igs, each encoded	with encode-s	string and concatenated wit	h encode+.
9 Strings containing a corfiguration number (1 and 3) shall be omitted if a single-configuration USB device. 10 Strings containing a device class (510) shall be omitted if the device class is 0. 11 1) usbVID.PID.REV.configCN 12 2) usbVID.PID.configCN 13 10 usbVID.PID.configCN 14 3) usbVID.PID.configCN 15 4) usbVID.pID.configCN 16 9) usbVID.classDC.DSC 17 usb.classDC.DSC 10 10 usb.classDC.DSC 210 11 usb.classDC.DSC 2111 11 usb.classDC.DSC 2111 11 usb.classDC 2111 11 usb.classDC 2111 11 usb.classDC 2111 11 usb.classDC 2111 12 11 usb.classDC 21 10 usb.classDC 22 110 usb.classDC 23 rentry 11, "usb.device", provides a mechanism to bind a generic driver to a USB device. 24 Note: entry 11, "usb.device", provides a mechanism to bind a generic driver to a USB de	5 6 7	Construct enco value of "com	ded strings as enum patible". Their	nerated in the follo relative order shal	owing list. All l be as in the l	l of the applicable strings s list, string 1) before string 2	hall be included in the 2), and so forth.
1) usbVID.PID.REV.configCN 2) usbVID.PID.REV 14 3) usbVID.PID.REV 14 3) usbVID.PID.REV 14 3) usbVID.PID.REV 15 4) usbVID.PID.REV 16 5) usbVID.classDC.DSC.DPROTO 17 10 usb.classDC.DSC 19 8) usb.classDC.DSC 21 10) usb.classDC.DSC 22 11) usb.device* 23 Note: entry 11), "usb.device", provides a mechanism to bind a generic driver to a USB device. 24 Note: entry 11), "usb.device", provides a mechanism to bind a generic driver to a USB device. 25 *name* S 26 *name* S 27 *name* S 30 prop-encoded-array: a string encoded with encode-string. S 31 The name of the node should be chosen from the following table, using the first name applicable: 32 pop-encoded-array: a string encoded with encode-string. 33 1 any storage 34 DeviceClass DeviceSubclass DeviceTotool Name <td>8 9 10</td> <td>Strings contair Strings contain</td> <td>ning a configuration ing a device class (</td> <td>on number (1 and 510) shall be om</td> <td>3) shall be of the deviation of the devi</td> <td>omitted for a single-confi vice class is 0.</td> <td>guration USB device.</td>	8 9 10	Strings contair Strings contain	ning a configuration ing a device class (on number (1 and 510) shall be om	3) shall be of the deviation of the devi	omitted for a single-confi vice class is 0.	guration USB device.
11 1) usbVID.PID.REV.configCN 13 2) usbVID.PID.configCN 14 3) usbVID.PID.configCN 15 4) usbVID.classDC.DSC.DPROTO 16 5) usbvLassDC.DSC 17 6) usbvLassDC.DSC 18 9) usb.classDC 19 8) usb.classDC.DSC 10 10) usb.classDC 21 11) usb.device 22 11) usb.device 23 *name * 24 Note: entry 11, "usb.device", provides a mechanism to bind a generic driver to a USB device. 25 *name * 26 Standard prop-name. 27 *name * S 28 Standard prop-name. 29 prop-encoded-array: a string encoded with encode-string. 31 The name of the node should be chosen from the following table, using the first name applicable: 34 bDeviceClass bDeviceSubclass bDeviceProtocol Name 37 [MASS] 1 any storage 38 [MASS] 2 any cdrom 39 MASS] 3 any storage 40	11	8	8				
13 2) usbVID.PID.REV 14 3) usbVID.PID.PID 15 4) usbVID.PID 16 5) usbVID.classDC.DSC.DPROTO 17 0) usb.classDC.DSC 18 1) usb.classDC.DSC 19 8) usb.classDC.DSC 21 10) usb.classDC.DSC 22 11) usb.dexice 23 10) usb.classDC 24 Note: entry 11), "usb.device", provides a mechanism to bind a generic driver to a USB device. 25 11) usb.device 26 "name" 27 "name" 28 Standard prop-name. 29 prop-encoded-array: a string encoded with encode-string. 31 11 32 bDeviceClass 34 bDeviceClass 35 bDeviceClass 36 9 37 IMASSI 38 1 39 IMASSI 39 MASSI 39 MASSI 39 MASSI 30 any any 39 MASSI any	12	1) usbVID,PI	D.REV.configCN				
14 3) usbVID:PID configCN 15 4) usbVID:PID 16 5) usbVID:classDC.DSC.DPROTO 17 0; usbVID:classDC.DSC 18 7) usb/classDC.DSC 19 8) usb,classDC.DSC 20 9) usb,classDC.DSC 21 10) usb,classDC 22 11) usb,device 23 7 24 Note: entry 11), "usb,device", provides a mechanism to bind a generic driver to a USB device. 25 7 26 7 7 ''name" 27 Standard prop-name. 28 Standard prop-name. 29 any any 31 The name of the node should be chosen from the following table, using the first name applicable: 34 bDeviceClass bDeviceProtocol Name 36 9 any any storage 37 [MASS] 1 any storage 38 MASS] 2 any cdrom 39 [MASS] 3 any storage 40 [MASS] 4 <	13	2) usbVID,PI	D.REV				
15 4) usbVID.PID 16 5) usbVID.classDC.DSC.DPROTO 17 usbVID.classDC.DSC.DPROTO 18 10, usb.classDC.DSC.DPROTO 19 8) usb.classDC.DSC 21 10) usb.classDC 22 11) usb.device 23 Note: entry 11), "usb.device", provides a mechanism to bind a generic driver to a USB device. 24 Note: entry 11), "usb.device", provides a mechanism to bind a generic driver to a USB device. 25 Standard prop-name. 26 "name" S 27 "name" S 28 Standard prop-name. S 29 prop-encoded-array: a string encoded with encode-string. S 31 The name of the node should be chosen from the following table, using the first name applicable: S 34 bDeviceClass bDeviceProtocol Name 35	14	3) usbVID,PI	D.configCN				
16 5) usbVID,classDC.DSC 17 6) usbvID,classDC.DSC 18 0) usb,classDC.DSC 19 8) usb,classDC.DSC 100 usb,classDC.DSC 110 usb,classDC 22 11) usb,device 23 11) usb,device 24 Note: entry 11), "usb,device", provides a mechanism to bind a generic driver to a USB device. 25 **name" S 26 **name" S 27 *name" S 28 Standard prop-name. S 29 prop-encoded-array: a string encoded with encode-string. S 31 The name of the node should be chosen from the following table, using the first name applicable: S 34 bDeviceClass bDeviceProtocol Name 35	15	4) usbVID,PI	D				
17 6) usbVID,classDC.DSC 18 7) usbVID,classDC 19 8) usb,classDC.DSC.DPROTO 20 9) usb,classDC 21 10) usb,classDC 22 11) usb,device 23 Note: entry 11), "usb,device", provides a mechanism to bind a generic driver to a USB device. 24 Note: entry 11), "usb,device", provides a mechanism to bind a generic driver to a USB device. 25 "name" \$ 26 "name" \$ 27 "name" \$ 28 Standard prop-name. \$ 29 prop-encoded-array: a string encoded with encode-string. \$ 30 prop-encoded-array: a string encoded with encode-string. \$ 31 the name of the node should be chosen from the following table, using the first name applicable: \$ 32 The name of the node should be chosen from the following table, using the first name applicable: \$ 34 bDeviceClass bDeviceProtocol Name \$ 35 1 any storage \$ 36 9 any storage \$ 37 [MASS] 1 any storage	16	5) usbVID,cla	assDC.DSC.DPRO	ТО			
18 7) usbVID,classDC 19 8) usb,classDC.DSC.DPROTO 10) usb,classDC.DSC 10) usb,classDC 21 11) usb,device 22 11) usb,device 23 Note: entry 11), "usb,device", provides a mechanism to bind a generic driver to a USB device. 24 Note: entry 11), "usb,device", provides a mechanism to bind a generic driver to a USB device. 25 "rame" S 26 "rame" S 27 "name" S 28 Standard prop-name. S 29 prop-encoded-array: a string encoded with encode-string. The name of the node should be chosen from the following table, using the first name applicable: 31 The name of the node should be chosen from the following table, using the first name applicable: 32 bDeviceClass bDeviceProtocol 34 bDeviceClass bDeviceProtocol Name 35 Image: any any tape Image: any any tape Image: any any any solid-state 38 [MASS] 3 any tape Image: any any any device Image: any any any device 34 Image: any any any any device S S S 39 </td <td>17</td> <td>6) usbVID,cla</td> <td>assDC.DSC</td> <td></td> <td></td> <td></td> <td></td>	17	6) usbVID,cla	assDC.DSC				
 s) usb.classDC.DSC.DPROTO 9) usb.classDC.DSC 10) usb.classDC 11) usb.device Note: entry 11), "usb.device", provides a mechanism to bind a generic driver to a USB device. "name" S Standard prop-name. prop-encoded-array: a string encoded with encode-string. The name of the node should be chosen from the following table, using the first name applicable: bDeviceClass bDeviceSubclass bDeviceProtocol Name <u>9</u> any any hub [MASS] 1 any storage [MASS] 2 any cdrom [MASS] 3 any tape [MASS] 4 any solid-state [MASS] 4 any solid-state [MASS] any any device Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class, specification is complete. "reg" \$ Standard prop-name. prop-encoded-array: one integer, encoded as with encode-int. 	18	7) usbVID.cla	assDC				
 b) usbclassDC.DSC 10) usbclassDC.DSC 11) usb,device Note: entry 11), "usb,device", provides a mechanism to bind a generic driver to a USB device. "name" S Standard prop-name. prop-encoded-array: a string encoded with encode-string. The name of the node should be chosen from the following table, using the first name applicable: bDeviceClass bDeviceSubclass bDeviceProtocol Name <u>many any hub</u> MASS 1 any storage [MASS] 2 any cdrom [MASS] 3 any tape [MASS] 4 any solid-state [MASS] any any device Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete. "reg" S Standard prop-name. Standard prop-name. The "reg" S 	19	8) usb.classD	C.DSC.DPROTO				
10) usb,classDC 11) usb,device 11) usb,device 11) usb,device 11) usb,device 11) usb,device 111	20	9) usb.classD	C.DSC				
22 1) usb,device 23 Note: entry 11), "usb,device", provides a mechanism to bind a generic driver to a USB device. 24 Note: entry 11), "usb,device", provides a mechanism to bind a generic driver to a USB device. 25 "name" S 26 "name" S 27 "name" S 28 Standard prop-name. S 30 prop-encoded-array: a string encoded with encode-string. S 31 The name of the node should be chosen from the following table, using the first name applicable: S 32 bDeviceClass bDeviceProtocol Name 33 bDeviceClass bDeviceProtocol Name 34 bDeviceClass bDeviceProtocol Name 35	21	10) usb classD					
Note: nrrr 11), "usb,device", provides a mechanism to bind a generic driver to a USB device. "name" S Standard prop-name. S prop-encoded-array: a string encoded with encode-string. The name of the node should be chosen from the following table, using the first name applicable: bDeviceClass bDeviceSubclass bDeviceClass bDeviceProtocol NASS 1 any hub IMASS 2 MASS 3 any solid-state IMASS any any any solid-state IMASS any any any solid-state IMASS any any any solid-state IMASS any any any any any any any should be used, as appropriate, when the Mass Storage Device Class specification is complete. "reg" S Standard prop-name. S standard prop-name. S standard prop-nam	22	11) ush device					
Note: entry 11), "usb,device", provides a mechanism to bind a generic driver to a USB device. "name" S Standard prop-name. S generic driver to a USB device. S standard prop-name. S generic driver to a driver to a driver to a USB device. S generic driver to a driver to a driver to a driver to a USB device. S generic driver to a driver to a driver to a USB device. S generic driver to a driver to a driver to a USB device. S generic driver to a driver to a USB device. S generic driver to a driver to a USB device. S generic driver to a driver to a driver to a driver to a USB device. S generic driver to a driver to a driver to a driver to a USB device. S generic driver to a use the driver to a driver to	22						
26 "name" S 27 "name" S 28 Standard prop-name. S 29 prop-encoded-array: a string encoded with encode-string. Image: Standard prop-name. 31 The name of the node should be chosen from the following table, using the first name applicable: Image: Standard prop-name. 32 DeviceClass bDeviceSubclass bDeviceProtocol Name 33 Image: Standard prop-name. Image: Standard prop-name. Image: Standard prop-name. 34 bDeviceClass bDeviceSubclass bDeviceProtocol Name 35 Image: Standard prop-name. Image: Standard prop-name. Image: Standard prop-name. Image: Standard prop-name. 36 9 any any storage Image: Standard prop-name. Image: Standard prop-name. 37 IMASS] 4 any storage Standard prop-name. Standard prop-name. 38 Image: Treg " S Standard prop-name. S 38 Image: Treg " S S Standard prop-name. 39 Image: Treg " S S S <	24 25	Note: entry 11), "usb,device", pro	ovides a mechanisi	n to bind a ge	neric driver to a USB devi	ce.
28 Standard prop-name. 29 prop-encoded-array: a string encoded with encode-string. 31 The name of the node should be chosen from the following table, using the first name applicable: 32 The name of the node should be chosen from the following table, using the first name applicable: 33 bDeviceClass bDeviceSubclass bDeviceProtocol Name 34 bDeviceClass bDeviceSubclass bDeviceProtocol Name 35	26 27	"name"					S
29 prop-encoded-array: a string encoded with encode-string. 31 The name of the node should be chosen from the following table, using the first name applicable: 32 The name of the node should be chosen from the following table, using the first name applicable: 33 bDeviceClass bDeviceSubclass bDeviceProtocol Name 34 bDeviceClass bDeviceSubclass bDeviceProtocol Name 35	28	Standard prop-	name.				
30 prop-encoded-array: a string encoded with encode-string. 31 The name of the node should be chosen from the following table, using the first name applicable: 32 The name of the node should be chosen from the following table, using the first name applicable: 34 bDeviceClass bDeviceProtocol Name 35	29						
31 The name of the node should be chosen from the following table, using the first name applicable: 32 bDeviceClass bDeviceProtocol Name 33	30	prop-encoded-a	array: a string enco	oded with encode-s	string.		
32 The name of the node should be chosen from the following table, using the first name applicable: 33 bDeviceClass bDeviceProtocol Name 34 bDeviceClass bDeviceProtocol Name 35	31						
33 bDeviceClass bDeviceSubclass bDeviceProtocol Name 35	32	The name of th	e node should be c	hosen from the fol	lowing table,	using the first name applic	able:
35	33 34	bDeviceClass	bDeviceSubclass	bDeviceProtocol	Name		
36 9 any any bub 37 [MASS] 1 any storage 38 [MASS] 2 any cdrom 39 [MASS] 3 any tape 40 [MASS] 4 any solid-state 41 [MASS] any any storage 42 any any any device 43 44 Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete. 47 48 "reg" S 50 Standard prop-name. S 51 prop-encoded-array: one integer, encoded as with encode-int. S 52 prop-encoded-array: one integer, encode as with encode-int. S	35						
37 [MASS] 1 any storage 38 [MASS] 2 any cdrom 39 [MASS] 3 any tape 40 [MASS] 4 any solid-state 41 [MASS] any any storage 42 any any any device 43 44 Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete. 47 48 * * 49 "reg" \$ \$ 50 Standard prop-name. \$ \$ 51 prop-encoded-array: one integer, encoded as with encode-int. \$ 52 The "reg" nonperty for a device node shall consist of the number of the USB hub port or the USB hort	36	9	any	any	hub		
38 [MASS] 2 any cdrom 39 [MASS] 3 any tape 40 [MASS] 4 any solid-state 41 [MASS] any any storage 42 any any any device 43 44 Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete. 47 48 "reg" S 50 Standard prop-name. S S 51 prop-encoded-array: one integer, encoded as with encode-int. S 52 prop-encoded-array: one integer, encoded as with encode-int. S	37	[MASS]	1	any	storage		
39 [MASS] 3 any tape 40 [MASS] 4 any solid-state 41 [MASS] any any storage 42 any any any device 43 Attribute Any device 44 Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete. 47 48	38	[MASS]	2	any	cdrom		
40 [MASS] 4 any solid-state 41 [MASS] any any storage 42 any any any device 43 Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete. 47 48 49 "reg" S 50 Standard prop-name. S 51 prop-encoded-array: one integer, encoded as with encode-int. S 52 The "reg" nonperty for a device node shall consist of the number of the USB bub port or the USB hert	39	[MASS]	3	any	tape		
 [MASS] any any storage any any any device any any any device Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete. "reg" S Standard prop-name. prop-encoded-array: one integer, encoded as with encode-int. The "reg" property for a device node shall consist of the number of the USB bub port or the USB hest 	40	[MASS]	4	any	solid-state		
 42 any any any device 43 44 Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete. 47 48 49 "reg" 49 50 Standard prop-name. 51 52 prop-encoded-array: one integer, encoded as with encode-int. 53 54 The "reg" nonerty for a device node shall consist of the number of the USB hub port or the USB host 	41	[MASS]	any	any	storage		
 43 44 Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete. 47 48 49 "reg" S 50 Standard prop-name. 51 52 prop-encoded-array: one integer, encoded as with encode-int. 53 54 The "reg" nonperty for a device node shall consist of the number of the USB bub port or the USB heret 	42	any	any	any	device		
 144 Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete. 47 48 49 "reg" 49 Standard prop-name. 51 52 prop-encoded-array: one integer, encoded as with encode-int. 53 54 The "reg" nonperty for a device node shall consist of the number of the USB bub port or the USP host 	43	Notes The USI	Davias Warking	Crown has not ust	completed at	and ardization of the Mass	Storage Device Class
 and so a device class code has not yet been assigned and the subclass codes are not final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete. 47 48 49 "reg" S 50 Standard prop-name. 51 52 prop-encoded-array: one integer, encoded as with encode-int. 53 54 The "reg" property for a device node shall consist of the number of the USB hub port or the USB host 	44 4F	note. The USE	Device working	oroup has not yet	completed sta	andardization of the Mass	Storage Device Class,
 should be used, as appropriate, when the Mass Storage Device Class specification is complete. 47 48 49 "reg" S 50 Standard prop-name. 51 52 prop-encoded-array: one integer, encoded as with encode-int. 53 54 The "reg" property for a device node shall consist of the number of the USB hub port or the USB host 	40			not yet been assig	neu anu me	subclass codes are not in	
 47 48 49 "reg" S 50 Standard prop-name. 51 52 prop-encoded-array: one integer, encoded as with encode-int. 53 54 The "reg" property for a device node shall consist of the number of the USB hub port or the USB host 	46	snould be used	, as appropriate, wr	ien the Mass Stora	ge Device Cla	ass specification is complet	te.
 48 49 "reg" S 50 Standard prop-name. 51 52 prop-encoded-array: one integer, encoded as with encode-int. 53 54 The "reg" property for a device node shall consist of the number of the USB hub port or the USB host 	4/						
 49 "reg" S 50 Standard prop-name. 51 52 prop-encoded-array: one integer, encoded as with encode-int. 53 54 The "reg" property for a device node shall consist of the number of the USB hub port or the USB host 	48						_
 50 Standard prop-name. 51 52 prop-encoded-array: one integer, encoded as with encode-int. 53 54 The "reg" property for a device node shall consist of the number of the USB hub port or the USB host 	49	"reg"					S
 51 52 prop-encoded-array: one integer, encoded as with encode-int. 53 54 The "reg" property for a device node shall consist of the number of the USB hub port or the USB host. 	50	Standard prop-	name.				
 52 prop-encoded-array: one integer, encoded as with encode-int. 53 54 The "reg" property for a device node shall consist of the number of the USB hub port or the USB host. 	51						
53 54 The "reg" property for a device node shall consist of the number of the USR hub port or the USP best	52	prop-encoded-a	array: one integer, e	encoded as with er	code-int.		
5.4 The "reg" property for a device node shall consist of the number of the USR hub port or the USP best	53						
JT THE TEST Property for a device node shall consist of the number of the OSD hub port of the OSD host	54	The "reg" p	roperty for a devi	ce node shall con	nsist of the n	number of the USB hub	port or the USB host
55 controller port to which this USB device is attached. As specified in [2] section 11.11.2.1, port numbers range	55	controller port	to which this USE	B device is attache	ed. As specifi	ied in [2] section 11.11.2.	1, port numbers range
56 from 1 to 255.	56	from 1 to 255.			I		. 0
	57						
	57						

S

"#size-cells" Standard prop-name.

prop-encoded-array: 0, encoded as with encode-int.

The value of "#size-cells" for device nodes shall be 0, representing the fact that USB device addresses are an enumeration rather than memory-like address ranges.

3.2.2.2. Bus-specific Properties for Device Nodes

```
"assigned-address"
```

prop-name to indicate the assigned USB bus address.

prop-encoded-array: one integer, encoded as with encode-int.

This property supplies the USB bus address assigned to this USB device. Each USB device that is assigned a USB address shall be assigned a USB address that is unique among all USB devices on this USB bus.

"configuration#" prop-name to indicate the selected configuration.

prop-encoded-array: one integer, encoded as with encode-int.

This property indicates the bConfigurationValue contained in the CONFIGURATION descriptor for the configuration selected for this USB device.

"low-speed"

prop-name to indicate that the USB device is low speed.

prop-encoded-array: None; presence or absence of the property conveys the information.

This property, if present, indicates that the USB device is low speed.

3.2.3. Device Node Methods

3.2.3.1. Open Firmware-defined Methods for Device Nodes

A package implementing a device node shall implement the following standard methods as defined in [1], with physical address representation as specified in section 3.3.1.

decode-unit Convert text unit-string to physical address.	(addr len config# interface#)	М
encode-unit Convert physical address to text unit-string.	(config# interface# addr len)	М

3.2.3.2. Bus-specific Methods for Device Nodes

None.

3.3. Interface Nodes

Interface nodes represent the USB interfaces present on a USB device. Normally, each interface represents independently controlled functionality, although for some device classes (e.g. Communications, Audio) one interface may provide "out of band" control for another.

Unless otherwise specified, an interface node shall be created for each USB interface.

3.3.1. Interface Node Address Representation

The textual representation of an interface node unit address shall be constructed as follows:

If the bConfigurationValue of the CONFIGURATION descriptor associated with this USB interface is equal to 1, the textual representation shall be the bInterfaceNumber of the INTERFACE descriptor with value 0 for bAlternateSetting associated with this USB interface, expressed in lower case hexadecimal with leading zeroes suppressed.

If the bConfigurationValue is not equal to 1, the textual representation shall be the bInterfaceNumber of the INTERFACE descriptor with value 0 for bAlternateSetting associated with this USB interface, a comma, and the bConfigurationValue, with both values expressed in lower case hexadecimal with leading zeroes suppressed.

3.3.2. Interface Node Properties

3.3.2.1. Open Firmware-defined Properties for Interface Nodes

The following properties, as defined in [1], have special meanings or interpretations for interface nodes. These properties shall be created for each interface node created.

The following notation is used:

29 30		From the DEV	VICE descriptor for this USB interface:	
31		VID	idVendor	
32		PID	idProduct	
33		REV	bcdDevice	
34		DC	bDeviceClass	
35		DSC	bDeviceSubClass	
36		DPROTO	bDeviceBrotocol	
37		DIROTO		
38		From the INT	ERFACE descriptor with the value of 0 for bAlternateSetting for this	USB interface:
39				
40		IN	bInterfaceNumber	
41		IC	bInterfaceClass	
42		ISC	bInterfaceSubClass	
43		IPROTO	bInterfaceProtocol	
44				
45 46 47	The textual repr case hexadecim	resentation of V al with leading z	VID, PID, REV, DC, DSC, DPROTO, IN, IC, ISC, and IPROTO shares suppressed.	all be in lower
47 48				_
49	"compatible	2 "		S
50	Standard prop	o-name.		
51 52	prop-encoded	l-array; a list of	strings, encoded with encode-string and concatenated with encode+.	
53	Construct end	coded strings as	enumerated in the following list. All of the applicable strings shall be	e included in the
54 55	value of "con	mpatible".T	heir relative order shall be as in the list, string 1) before string 2), and	so forth.
56 57	Strings contai	ining an interfac	e class (38) shall be omitted if the interface class is 0.	
<i>.</i> ,	8			6/2/98
	0			$0, \mu, \mathbf{y}_0$

Open Firmware Recommended Practice

1					
2	1) usbifVID,P	ID.REV.config	CN.IN		
3	2) usbifVID.P	ID.configCN.II	Ń		
4	3) usbifVID cl	assIC ISC IPR	ОТО		
5	4) usbifVID cl	lassIC ISC	010		
5	5) usbifVID cl				
0 7	6) usbif class	C ISC IPPOTO)		
0	7) usbif classi	C ISC.II KOTC	,		
0	() usbif, classi	C.ISC			
10	o) usuii,ciassi	C			
11					
					C
⊥∠ 1 2	"name"				3
13	Standard prop	-name.			
14			1 1 1 1	1	
15	prop-encoded-	array: a string	encoded with enco	ode-string.	
16					
17	The name of the	he node should	be chosen from th	e following table, using the first name applicable	e:
18					
19	bInterface	bInterface	bInterface		
20	Class	Subclass	Protocol	Name	
21					
22	1	1	any	sound-control	
23	1	2	any	sound	
24	1	3	any	midi	
25	1	anv	anv	sound	
26	3	1	1	kevboard	
27	3	1	2	mouse	
28	7	anv	anv	printer	
29	9	any	any	hub	
30	[POWFR]	any	any	nower	
21		any	any	display control	
27		1	any	modem	
2∠ 22		1	ally	modem	
22		2	ally	talambana	
34		5	any		
35		any	any		
36	[DATA]	any	any	data	
37	any	any	any	interface	
38					
39	Note: The US	B Device Wo	rking Group has i	not yet completed standardization of the Audic	, Power, Monitor,
40	Communicatio	ons, or Data Ir	iterface Classes, a	and so interface class codes have not yet been	assigned and the
41	subclass codes	s are not final.	The names above	should be used, as appropriate, when the various	s specifications are
42	complete.				
43					
44					
45	"reg"				S
46	Standard prop	-name.			
47					
48	prop-encoded-	array: two inte	gers, each encoded	d as with encode-int.	
49		-	-		
50	The "reg" p	roperty for an in	nterface node shall	l be constructed as follows:	
51	- J F	1 2			
52	The first inte	eger shall con	tain the bInterfac	ceNumber from the INTERFACE descriptor	with value 0 for
52	bAlternateSett	ing associated	with this USB inte	erface.	
54					
51	The second in	nteger shall co	ntain the bConfig	urationValue from the CONFIGURATION de	scriptor associated
55	with this USR	interface	inam the ocoining		serptor associated
50		mortuee.			
51					

3.3.2.2. Bus-specific Properties for Interface Nodes

None.

3.3.3. Interface Node Methods

3.3.3.1. Open Firmware-defined Methods for Interface Nodes

None.

3.3.3.2. Bus-specific Methods for Interface Nodes

None.

3.4. Combined Nodes

A combined node is a special case node, combining some of the properties of both device nodes and interface nodes, and is typically used to simplify the representation of a simple USB device, with a single configuration and a single interface.

Neither a device node (see section 3.2.) nor an interface node (see section 3.3.) shall be created when a USB device reports in its DEVICE descriptor the following:

(1) bDeviceClass is 0 or 9, and

(2) bNumConfigurations is 1,

and reports in its CONFIGURATION descriptor the following:

(3) bNumInterfaces is 1.

Instead, a combined node shall be created.

3.4.1. Combined Node Address Representation

The textual representation of a combined node unit address shall be the number of the USB hub port or USB host controller port to which this USB device is attached, in lower case hexadecimal with leading zeroes suppressed.

3.4.2. Combined Node Properties

3.4.2.1. Open Firmware-defined Properties for Combined Nodes

The following standard properties, as defined in [1], have special meaning or interpretation for combined nodes.

The following notation is used:

From the DEVICE descriptor for this USB device:

VID	idVendor
PID	idProduct
REV	bcdDevice
DC	bDeviceClass
DSC	bDeviceSubClass
DPROTO	bDeviceProtocol

From the INTERFACE descriptor with the value of 0 for bAlternateSetting for this USB interface:

IC	bInterfaceClass
ISC	bInterfaceSubClass
IPROTO	bInterfaceProtocol

1 2 The textual representation of VID, PID, REV, DC, DSC, DPROTO, IC, ISC, and IPROTO shall be in lower case 3 hexadecimal with leading zeroes suppressed. 4 5 "compatible" S б Standard prop-name. 7 8 prop-encoded-array; a list of strings, encoded with encode-string and concatenated with encode+. 9 10 Construct encoded strings as enumerated in the following list. All of the applicable strings shall be included in the 11 value of "compatible". Their relative order shall be as in the list, string 1) before string 2), and so forth. 12 13 Strings containing a device class (3..8) shall be omitted if the device class is 0. Strings containing an interface class (9..14) shall be omitted if the interface class is 0. 14 15 16 1) usbVID,PID.REV 17 2) usbVID,PID 18 3) usbVID,classDC.DSC.DPROTO 19 4) usbVID,classDC.DSC 20 5) usbVID,classDC 21 6) usb,classDC.DSC.DPROTO 22 7) usb,classDC.DSC 8) usb,classDC 23 24 9) usbifVID,classIC.ISC.IPROTO 10) usbifVID,classIC.ISC 25 26 11) usbifVID, classIC 27 12) usbif, classIC.ISC.IPROTO 28 13) usbif, classIC.ISC 29 14) usbif, classIC 30 31 32 "name" S 33 Standard prop-name. 34 35 prop-encoded-array: one string, encoded with encode-string. 36 37 The name of the node should be chosen from the following table, using the first name applicable: 38 39 bInterface bInterface bInterface 40 Class Subclass Protocol Name 41 _____ 42 1 1 sound-control any 43 1 2 any sound 3 44 1 midi any 45 1 any sound any 46 3 1 1 keyboard 47 3 1 2 mouse 48 7 any any printer 49 9 hub any any [POWER] 50 power any any display-control 51 [MONITOR] any any 52 [COMM] 1 modem any 53 2 modem [COMM] any 54 [COMM] 3 any telephone 55 communications [COMM] any any

[DATA]

any

any

data

56

1 2	bDeviceClass	bDeviceSubclass	bDeviceProtocol	Name
3				
4	9	any	any	hub
5	[MASS]	1	any	storage
6	[MASS]	2	any	cdrom
7	[MASS]	3	any	tape
8	[MASS]	4	any	solid-state
9	[MASS]	any	any	storage
10	any	any	any	device
$\perp \perp$				

Note: The USB Device Working Group has not yet completed standardization of the Audio, Power, Monitor, Communications, or Data Interface Classes, and so interface class codes have not yet been assigned and the subclass codes are not yet final. The names above should be used, as appropriate, when the various specifications are complete.

Note: The USB Device Working Group has not yet completed standardization of the Mass Storage Device Class, and so a device class code has not yet been assigned and the subclass codes are not yet final. The names above should be used, as appropriate, when the Mass Storage Device Class specification is complete.

"reg"

Standard prop-name.

S

prop-encoded-array: one integer, encoded as with encode-int.

The "reg" property for a combined node shall be the number of the USB hub port or the USB host controller port to which this USB device is attached. As specified in [2] section 11.11.2.1, port numbers range from 1 to 255.

3.4.2.2. Bus-specific Properties for Combined Nodes

"assigned-address"

prop-name to indicate the assigned USB bus address.

prop-encoded-array: one integer, encoded as with encode-int.

This property supplies the USB bus address assigned to this USB device. Each USB device that is assigned a USB address shall be assigned a USB address that is unique among all USB devices on this USB bus.

"low-speed"

prop-name to indicate that the USB device is low speed.

prop-encoded-array: None; presence or absence of the property conveys the information.

This property, if present, indicates that the USB device is low speed.

3.4.3. Combined Node Methods

3.4.3.1. Open Firmware-defined Methods for Combined Nodes

None.

3.4.3.2. Bus-specific Methods for Combined Nodes

None.

S

S

4. Hub Nodes

This section includes special requirements for devices implementing a hub node.

4.1. Common Requirements for Hub Nodes

This section contains the common requirements for hub nodes.

4.1.1. Hub Node Properties

4.1.1.1. Open Firmware-defined Properties for Hub Nodes

The following standard properties, as defined in [1], have special meaning or interpretation for hub nodes.

```
"#address-cells"
Standard property to define the address format. Its value shall be 1, encoded as with encode-int.
```

"#size-cells" Standard prop-name to define the package's address size format.

prop-encoded-array: 0, encoded as with encode-int.

The value of "#size-cells" shall be 0, representing the fact that child device addresses are an enumeration rather than memory-like address ranges.

4.1.1.2. Bus-specific Properties for Hub Nodes

None.

4.1.2. Hub Node Methods

4.1.2.1. Open Firmware-defined Methods for Hub Nodes

A package implementing a hub node shall implement the following standard methods as defined in [1], with physical address representations as specified in section 3.2.1.

decode-unit Convert text unit-string to physical address.	(addr len port)	М
encode-unit	(port addr len)	М

Convert physical address to text unit-string.

4.1.2.2. Bus-specific Methods for Hub Nodes

None.

4.2. Requirements for Combined Node Hubs

A combined node (see section 3.4.) which reports in its INTERFACE descriptor with value 0 for bAlternateSetting the following:

bInterfaceClass is 9,

is a combined node hub.

A combined node hub shall include all the properties, methods, and requirements of "Common Requirements for

Hub Nodes" as described in section 4.1

4.3. Requirements for Interface Node Hubs

An interface node (see section 3.3.) which represents a USB hub shall include all the properties, methods, and requirements of "Common Requirements for Hub Nodes" as described in section 4.1