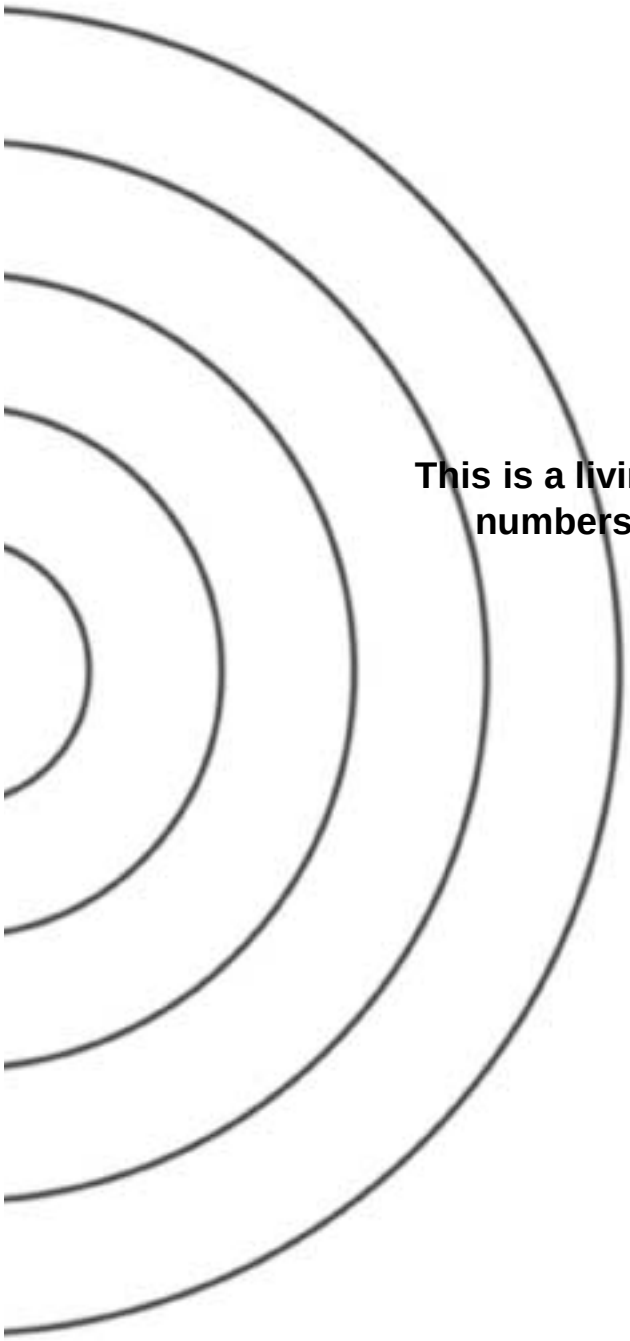
Bluetooth Assigned Numbers



This is a living document that lists assigned numbers, codes and identifiers in the Bluetooth standard

Bluetooth™

Version 1.1
February 22 2001

Revision History

The Revision History is shown in on page 29.

Contributors

The persons who contributed to this specification are listed on page 29.

Web Site

This document can also be found on the Bluetooth website:
http://www.bluetooth.org/assigned-numbers.htm

Disclaimer and Copyright Notice

Each Member hereby acknowledges that products equipped with the Bluetooth™ technology ("Bluetooth™ Products") may be subject to various regulatory controls under the laws and regulations of various governments worldwide. Such laws and regulatory controls may govern, among other things, the combination, operation, use, implementation and distribution of Bluetooth™ Products. Examples of such laws and regulatory controls include, but are not limited to, airline regulatory controls, telecommunications regulations, technology transfer controls and health and safety regulations. Each Member is solely responsible for the compliance by their Bluetooth™ Products with any such laws and regulations and for obtaining any and all required authorizations, permits, or licenses for their Bluetooth™ Products related to such regulations within the applicable jurisdictions. Each Member acknowledges that nothing in the Specification provides any information or assistance in connection with securing such compliance, authorizations or licenses. NOTHING IN THE SPECIFICATION CREATES ANY WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING SUCH LAWS OR REGULATIONS.

ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS OR FOR NONCOMPLIANCE WITH LAWS, RELATING TO USE OF THE SPECIFICATION IS EXPRESSLY DISCLAIMED. BY USE OF THE SPECIFICATION, EACH MEMBER EXPRESSLY WAIVES ANY CLAIM AGAINST BLUETOOTH SIG AND ITS PROMOTER MEMBERS RELATED TO USE OF THE SPECIFICATION.

Bluetooth SIG reserves the right to adopt any changes or alterations to the Specification as it deems necessary or appropriate and to adopt a process for adding new Bluetooth™ profiles after the release of the Specification.

CONTENTS

1 BLUETOOTH BASEBAND

1.1 THE GENERAL- AND DEVICE-SPECIFIC INQUIRY ACCESS CODES (DIACS)

The Inquiry Access Code is the first level of filtering when finding Bluetooth devices and services. The main purpose of defining multiple IACs is to limit the number of responses that are received when scanning devices within range.

#	LAP value	Usage
0	0x9E8B33	General/Unlimited Inquiry Access Code (GIAC)
1	0x9E8B00	Limited Dedicated Inquiry Access Code (LIAC)
2-63	0x9E8B01-0x9E8B32, 0x9E8B34-0x9E8B3F	RESERVED FOR FUTURE USE

Table 1.1: The Inquiry Access Codes

The Limited Inquiry Access Code (LIAC) is only intended to be used for limited time periods in scenarios where both sides have been explicitly caused to enter this state, usually by user action. For further explanation of the use of the LIAC, please refer to the Generic Access Profile [7].

In contrast it is allowed to be continuously scanning for the General Inquiry Access Code (GIAC) and respond whenever inquired.

1.2 THE CLASS OF DEVICE/SERVICE FIELD

The Class of Device/Service (CoD) field has a variable format. The format is indicated using the ‘Format Type field’ within the CoD. The length of the Format Type field is variable and ends with two bits different from '11'. The version field starts at the least significant bit of the CoD and may extend upwards.

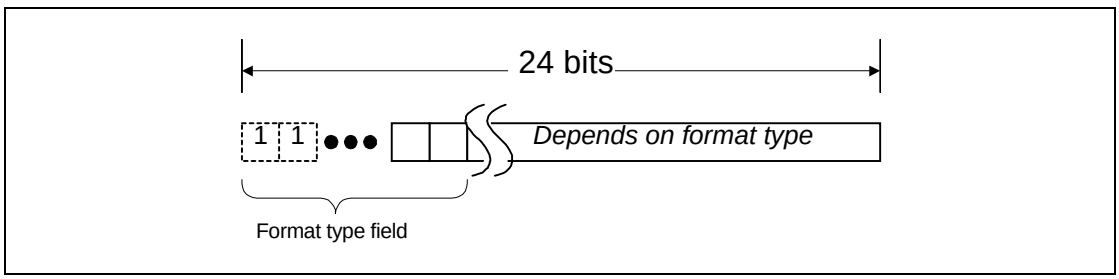


Figure 1.1: General format of Class of Device/Service

In the ‘format #1’ of the CoD (Format Type field = 00), 11 bits are assigned as a bit-mask (multiple bits can be set) each bit corresponding to a high level generic category of service class. Currently 7 categories are defined. These

are primarily of a ‘public service’ nature. The remaining 11 bits are used to indicate device type category and other device-specific characteristics.

Any reserved but otherwise unassigned bits, such as in the Major Service Class field, should be set to 0.

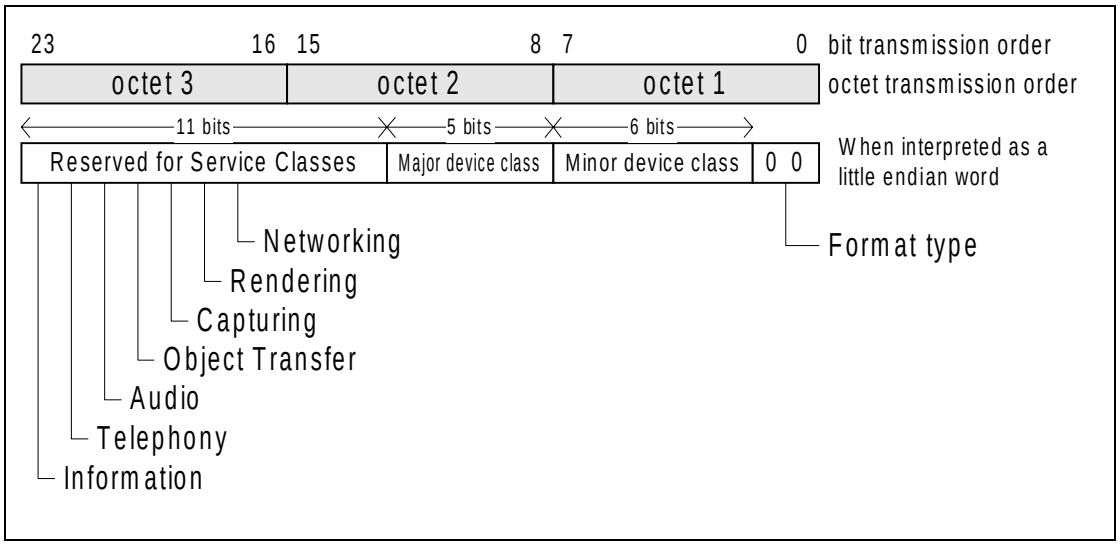


Figure 1.2: The Class of Device/Service field (format type 1). Note the order in which the octets are sent on the air and stored in memory.

1.2.1 Major Service Classes

Bit no	Major Service Class
13	Limited Discoverable Mode[1]
14	(reserved)
15	(reserved)
16	(reserved)
17	Networking (LAN, Ad hoc, …)
18	Rendering (Printing, Speaker, …)
19	Capturing (Scanner, Microphone, …)
20	Object Transfer (v-Inbox, v-Folder, …)
21	Audio (Speaker, Microphone, Headset service, …)
22	Telephony (Cordless telephony, Modem, Headset service, …)
23	Information (WEB-server, WAP-server, …)

Table 1.2: Major Service Classes

1. As defined in [7]

1.2.2 Major Device Classes

The Major Class segment is the highest level of granularity for defining a Bluetooth Device. The main function of a device is used to determine the major class grouping. There are 32 different possible major classes. The assignment of this Major Class field is defined in Table 1.3.

Code (bits)					Major Device Class
12	11	10	9	8	bit no of CoD
0	0	0	0	0	Miscellaneous[1]
0	0	0	0	1	Computer (desktop, notebook, PDA, organizers, …)
0	0	0	1	0	Phone (cellular, cordless, payphone, modem, …)
0	0	0	1	1	LAN Access Point
0	0	1	0	0	Audio (headset, speaker, stereo, …)
0	0	1	0	1	Peripheral (mouse, joystick, keyboards, …)
x	x	x	x	x	Range 0x06 to 0x1E reserved
1	1	1	1	1	Unclassified, specific device code not assigned

Table 1.3: Major Device Classes

1. Used where a more specific Major Device Class code is not suited (but only as specified in this document. Devices that do not have a major class code assigned can use the all-1 code until 'classified')

1.2.3 The Minor Device Class field

The 'Minor Device Class field' (bits 7 to 1 in the CoD), are to be interpreted only in the context of the Major Device Class (but independent of the Service Class field). Thus the meaning of the bits may change, depending on the value of the 'Major Device Class field'. When the Minor Device Class field indicates a device class, then the primary device class should be reported, e.g. a cellular phone that can also work as a cordless handset should use 'Cellular' in the minor device class field.

1.2.4 Minor Device Class field - Computer Major Class

Code (bits)						Minor Device Class
7	6	5	4	3	2	bit no of CoD
0	0	0	0	0	0	Unclassified, code for device not assigned
0	0	0	0	0	1	Desktop workstation
0	0	0	0	1	0	Server-class computer
0	0	0	0	1	1	Laptop
0	0	0	1	0	0	Handheld PC/PDA (clam shell)
0	0	0	1	0	1	Palm sized PC/PDA
x	x	x	x	x	x	Range 0x06-0x7F reserved

Table 1.4: Sub Device Class field for the 'Computer' Major Class

1.2.5 Minor Device Class field - Phone Major Class

Code (bits)						Minor Device Class
7	6	5	4	3	2	bit no of CoD
0	0	0	0	0	0	Unclassified, code not assigned
0	0	0	0	0	1	Cellular
0	0	0	0	1	0	Cordless
0	0	0	0	1	1	Smart phone
0	0	0	1	0	0	Wired modem or voice gateway
x	x	x	x	x	x	Range 0x05-0x7F reserved

Table 1.5: Sub Device Classes for the 'Phone' Major Class

1.2.6 Minor Device Class field - LAN Access Point Major Class

Code (bits)			Minor Device Class
7	6	5	bit no of CoD
0	0	0	Fully available
0	0	1	1-17% utilized
0	1	0	17 - 33% utilized
0	1	1	33 - 50% utilized
1	0	0	50 - 67% utilized
1	0	1	67 - 83% utilized
1	1	0	83 - 99% utilized
1	1	1	No Service Available[1]

Table 1.6: The LAN Access Point Load Factor field

1. “Device is fully utilized and cannot accept additional connections at this time, please retry later”

The exact loading formula is not standardized. It is up to each LAN Access Point implementation to determine what internal conditions to report as a utilization percentage. The only requirement is that the number reflects an ever-increasing utilization of communication resources within the box. As a recommendation, a client that locates multiple LAN Access Points should attempt to connect to the one reporting the lowest load.

Code (bits)			Minor Device Class
4	3	2	bit no of CoD
0	0	0	Unclassified (use this value if no other apply)
x	x	x	range 0x01-0x0F reserved

Table 1.7: Reserved sub-field for the LAN Access Point

1.2.7 Minor Device Class field - Audio Major Class

Code (bits)						Minor Device Class
7	6	5	4	3	2	bit no of CoD
0	0	0	0	0	0	Unclassified, code not assigned
0	0	0	0	0	1	Device conforms to the Headset profile [9]
x	x	x	x	x	x	Range 0x02-0x7F reserved

Table 1.8: Sub Device Classes for the 'Audio' Major Class

2 LINK MANAGER PROTOCOL (LMP)

2.1 THE LINK MANGER VERSION PARAMETER

Parameter name	Assigned values	
VersNr	0	Bluetooth LMP 1.0, [2]
	1	Bluetooth LMP 1.1
	2-255	(reserved)

Table 2.1: The LMP Version Parameter Values

2.2 THE LMP_COMPID PARAMETER CODES

This is the parameter used in the LMP Version procedure. All companies that create a unique implementation of the Link Manager shall have their own LMP_COMPID.

Code	Company
0	Ericsson Mobile Communications
1	Nokia Mobile Phones
2	Intel Corp.
3	IBM Corp.
4	Toshiba Corp.
5	3Com
6	Microsoft
7	Lucent
8	Motorola
9	Infineon Technologies AG
10	Cambridge Silicon Radio
11	Silicon Wave
12	Digianswer
13 - 65534	(reserved)
65535	For use in internal and interoperability tests before a Company ID has been assigned. May not be used in products.

Table 2.2: The LMP_CompId parameter codes

3 LOGICAL LINK CONTROL AND ADAPTATION PROTOCOL (L2CAP)

Please see Section 4.3 for assigned PSM values.

3.1 CHANNEL IDENTIFIERS

Destination CID	Protocol/usage	Reference
0x0000	Illegal, should not be used	[3]
0x0001	L2CAP signalling channel	[3]
0x0002	L2CA connection less data	[3]
0x0003 - 0x003F	(reserved)	

Table 3.1: Pre-defined L2CAP Channel Identifiers

3.2 PROTOCOL AND SERVICE MULTIPLEXOR (PSM)

Protocol	PSM	Reference
SDP	0x0001	[4]
RFCOMM	0x0003	[5]
TCS-BIN	0x0005	[6]
TCS-BIN-CORDLESS	0x0007	[6]

Table 3.2: Assigned Protocol and Service Multiplexor values (PSM)

4 SERVICE DISCOVERY PROTOCOL (SDP)

4.1 UNIVERSALLY UNIQUE IDENTIFIER (UUID) SHORT FORMS

The Bluetooth Service Discovery Protocol (SDP) specification defines a way to represent a range of UUIDs (which are nominally 128-bits) in a shorter form. A *reserved* range of 2^{32} values can be represented using 32-bits (denoted uuid32). Of these, a sub-range of 2^{16} values can be represented using only 16-bits (denoted uuid16). Any value in the 2^{32} range that is not assigned in this document is reserved pending future revisions of this document. In other words, no value in this range may be used except as specified in this or future revisions of this document. UUID values outside of this range can be allocated as described in [19] for any purpose the allocater desires.

4.2 BASE UNIVERSALLY UNIQUE IDENTIFIER (UUID)

The Base UUID is used for calculating 128-bit UUIDs from 'short UUIDs' (uuid16 and uuid32) as described in the SDP Specification [4].

Mnemonic	UUID
BASE_UUID	00000000-0000-1000-8000-00805F9B34FB

4.3 PROTOCOLS

Mnemonic	UUID	Name	Ref.
SDP	uuid16: *0x0001*[1]	sdp.bt	[4]
RFCOMM	uuid16: *0x0003*	com.bt	[5]
TCS-BIN	uuid16: *0x0005*	tcs.bt	[6]
L2CAP	uuid16: *0x0100*		[3]
IP	uuid16: *0x0009*		
UDP	uuid16: *0x0002*		
TCP	uuid16: *0x0004*		
TCS-AT	uuid16: *0x0006*	modem	
OBEX	uuid16: *0x0008*	obex	
FTP	uuid16: *0x000A*	ftp	
HTTP	uuid16: *0x000C*	http	
WSP	uuid16: *0x000E*	wsp	

Table 4.1: Protocol Universally Unique Identifiers and Names

1. 'Short UUID'

4.4 SERVICE CLASSES

Mnemonic	UUID	Profile[1]	AbstractName
ServiceDiscoveryServerServiceClassID	uuid16: *0x1000*		
BrowseGroupDescriptorServiceClassID	uuid16: *0x1001*		
PublicBrowseGroup	uuid16: *0x1002*		
SerialPort	uuid16: *0x1101*	[7]	serial.bt
LANAccessUsingPPP	uuid16: *0x1102*		
DialupNetworking	uuid16: *0x1103*	[13]	
IrMCSync	uuid16: *0x1104*	[17]	
OBEXObjectPush	uuid16: *0x1105*	[16]	
OBEXFileTransfer	uuid16: *0x1106*	[15]	
IrMCSyncCommand	uuid16: *0x1107*	[17]	
Headset	uuid16: *0x1108*	[7]	headset
CordlessTelephony	uuid16: *0x1109*	[10]	
Intercom	uuid16: *0x1110*	[11]	
Fax	uuid16: *0x1111*	[12]	
HeadsetAudioGateway	uuid16: 0x1112	[7]	
WAP	uuid16: 0x1113	[20]	
WAP_CLIENT	uuid16: 0x1114	[20]	
PnPInformation	uuid16: *0x1200*		
GenericNetworking	uuid16: *0x1201*	n/a	
GenericFileTransfer	uuid16: *0x1202*	n/a	
GenericAudio	uuid16: *0x1203*	n/a	
GenericTelephony	uuid16: *0x1204*	n/a	

Table 4.2: Service Class Identifiers and Names

1. If the specified Service Class directly and exactly implies a certain Profile, the Profile is indicated here (i.e. for concrete Service Classes). Leave empty for abstract Service Classes.

The Profile column in Table 4.2 indicates which Service Class identifiers that also directly corresponds to a Bluetooth Profile. It is not allowed to use the Service Class UUID unless the service complies with the specified Profile. These UUIDs might also appear as Profile Identifiers in the BluetoothProfileDescriptorList attribute.

4.5 ATTRIBUTE IDENTIFIER CODES

Mnemonic	Attribute ID	Reference
ServiceRecordHandle	0x0000	[4] *Bluetooth Service Discovery Protocol (SDP)*, Bluetooth SIG
ServiceClassIDList	0x0001	
ServiceRecordState	0x0002	
ServiceID	0x0003	
ProtocolDescriptorList	0x0004	
BrowseGroupList	0x0005	
LanguageBaseAttributeIDList	0x0006	
ServiceInfoTimeToLive	0x0007	
ServiceAvailability	0x0008	
BluetoothProfileDescriptorList	0x0009	
DocumentationURL	0x000A	
ClientExecutableURL	0x000B	
Icon10	0x000C	
IconURL	0x000D	
Reserved	0x000E-0x01FF	
ServiceName	0x0000 + b[1]	
ServiceDescription	0x0001 + b	
ProviderName	0x0002 + b	
VersionNumberList	0x0200	
ServiceDatabaseState	0x0201	
GroupID	0x0200	
Remote audio volume control	0x0302[2]	[7]
External network	0x0301	[10]
Service Version	0x0300	
Supported Data Stores List	0x0301	[17]
Supported Formats List	0x0303	[16]

Table 4.3: Attribute Identifiers

Mnemonic	Attribute ID	Reference
Fax Class 1 Support	0x0302	[12]
Fax Class 2.0 Support	0x0303	
Fax Class 2 Support	0x0304	
Audio Feedback Support	0x0305	
NetworkAddress	0x0306	[20]
WAPGateWay	0x0307	[20]
HomePageURL	0x0308	[20]
WAPStackType	0x0309	[20]

Table 4.3: Attribute Identifiers

1. 'b' in this table represents a base offset as given by the LanguageBaseAttributeIDList attribute. For the primary language, 'b' must be equal to 0x0100 as described in the SDP specification.
2. Items in *italic* are tentative values in this version of the document.

4.6 PROTOCOL PARAMETERS

Protocol	Parameter mnemonic	Index
L2CAP	PSM	1
TCP or UDP	Port	1
RFCOMM	Channel	1

Table 4.4: Protocol Parameters

4.7 HOST OPERATING ENVIRONMENT IDENTIFIERS

4.7.1 ClientExecutableURL substitution strings

The operating environment identifier strings have the following format[1]:

<cpu_type>-<manufacturer>-[<kernel>-]<os>[<version>][-<object_format>]

1. It is based on a format used by the GNU AutoConfig tools

The general rule is that is that a new identifier should only be defined as required to differentiate incompatible operating environments concerning an executable file image. That is, for example different <version>-tags should not be used for compatible versions of the same operating system.

Currently defined tags:

CPU-Type ID	Description
alpha	Digital Alpha* compatible
arm	ARM* core or compatible
i86	Any Intel* 80x86-family compatible CPU
i960	Intel* i960 compatible
jvm	Java Virtual Machine*
mips	MIPS MIPS* compatible
ppc	IBM/Motorola PowerPC* compatible
sh3	Hitachi SH-3* compatible
sh4	Hitachi SH-4* compatible
sparc	Sun Sparc* compatible
Kernel ID	**Description**
chorus, linux, javaos, os9, qnx, vxworks	
<os>	An 'OS identifier' as listed below, might appear in the <kernel> field when the requested OS platform is Java based.
OS+Version-Identifiers	
amigaos, beos4.5, ejava, epocc, epoce, epocq, epocs, gnu, jre1.1, jre1.2, macos, macosx, os2, os9, palmos, pjava, pjava1.1, photon, plan9, qnx, rtjava, win95, win98, win2000, wince, winnt4	
Object Format Identifiers[1]	
aout, bout, coff, elf, jar	
Manufacturer Identifiers	
amiga*, apple*, be*, ericsson*, ibm*, intel*, lucent*, microsoft*, microware*, motorola*, nokia*, palm*, psion*, qnx*, sun*, symbian*, toshiba*, unknown[2]	

1. Only applicable when the object format is not otherwise uniquely implied by the identifier string.
2. Use when no other applies.

For Linux, the ’manufacturer’ field may be used to indicate Linux distribution if so required (in which case <version> indicates the version of the distribution). Otherwise use ’unknown’.

Linux Distribution Identifiers
caldera, debian, dlx, doslinux, linuxpro, linuxware, mandrake, mklinux, redhat, slackware, stampede, suse, turbolinux, yggdrasil

Example Operating Environment Identifier Strings		
i86-microsoft-win32	ppc-apple-macos	i86-redhat-linux-gnu6
i86-microsoft-win98	m68k-apple-macos	ppc-mklinux-linux-gnu
i86-microsoft-winnt4	ppc-apple-macosx	
alpha-microsoft-winnt4	i86-apple-macosx	
i86-microsoft-win2000	m68k-amiga-amigaos	
	ppc-amiga-amigaos	
i86-be-beos4.5	jvm-sun-jre1.2	
ppc-be-beos4.5	jvm-sun-pjava1.1	
arm-symbian-epoc3	jvm-sun-ejava	
i86-unknown-linux-gnu	m68k-palm-palmos-coff	
sh3-microsoft-wince	ppc-ibm-vxworks-pjava1.2	
arm-microsoft-wince	sparc-sun-javaos-jre1.2	

4.7.2 IconURL substitution strings

The IconURL operating environment identifier strings have the following general format:

<horizontal_pixels>x<vertical_pixels>x<color_depth>[m].<file_format>

The optional tag 'm' indicates monochrome or grayscale. The host is free to try to match/request any graphics file format as indicated by a <file_format> tag, however at a minimum files conforming to the Portable Network Graphic standard [18] should be made available at the resulting URL (indicated by <file_format>=png)[2].

File format tag	Description
png	Portable Network Graphics [18]
gif	Graphics Interchange File format
bmp	Windows bitmap

Currently defined IconURL Icon format identifier strings:

Example Icon format Identifier Strings	
32x32x8.png	256 color 32 by 32 icon (or 255 colors + transparent)
16x16x8.png	
16x16x1m.png	Black and white (or monochrome + transparent)
10x10x2m.png	4 gray-scales

2. The use of PNG, and whether a subset of PNG should be required, is currently pending further investigation.

5 REFERENCES

[1] *Bluetooth Baseband Specification*, Bluetooth SIG

[2] *Bluetooth Link Manager Specification,* Bluetooth SIG

[3] *Logical Link Control and Adaptation Protocol Specification*, Bluetooth SIG

[4] *Bluetooth Service Discovery Protocol (SDP)*, Bluetooth SIG

[5] *RFCOMM with TS 07.10,* Bluetooth SIG

[6] *Bluetooth Telephony Control Specification / TCS Binary* , Bluetooth SIG

[7] *Generic Access Profile*, Bluetooth SIG

[8] *Serial Port Profile*, Bluetooth SIG

[9] *Headset Profile*, Bluetooth SIG

[10] *Cordless Telephony Profile*, Bluetooth SIG

[11] *Intercom Profile*, Bluetooth SIG

[12] *Fax Profile*, Bluetooth SIG

[13] *Dial-up Networking Profile*, Bluetooth SIG

[14] *IrDA Interoperability*, Bluetooth SIG

[15] *File Transfer Profile*, Bluetooth SIG

[16] *Object Push Profile*, Bluetooth SIG

[17] *Synchronization Profile*, Bluetooth SIG

[18] *Portable Network Graphics (PNG),* http://www.w3.org/Graphics/PNG

[19] *UUIDs and GUIDs*, P. J. Leach et al, http://www.ietf.org/internet-drafts/draft-leach-uuids-guids-01.txt

[20] *Interoperability Requirements for Bluetooth as a WAP*, Bluetooth SIG

6 TERMS AND ABBREVIATIONS

LMP	Link Management Protocol
L2CA	Logical Link Control and Adaptation, protocol multiplexer layer for Bluetooth
MTU	Maximum Transmission Unit
SAP	Service Access Points
Baseband	Baseband Protocol
Service Discovery	The ability to discover the capability of connecting devices or hosts.
PnP	Plug and Play
SAR	Segmentation and Reassembly
IP	Internet Protocol
IrDA	InfraRed Data Association
PPP	Point-to-Point Protocol
IETF	Internet Engineering Task Force
RFC	Request For Comments

7 LIST OF FIGURES

8 LIST OF TABLES

REVISION HISTORY

Rev	Date	Comments
1.0 draft	July 5th 1999	• Renamed some of the service class mnemonics. Added a bit to indicate if a node is in Limited Discoverable Mode (useful in a reply to GIAC). • Updated after review. Added the IAC LAP codes. • Minor editoral and clarifications, including section "Universally Unique Identifier (UUID) short forms".
1.0B	Dec 1st 1999	• Revised from a linguistic point of view. • Errata items previously published on the web has been included. These corrections and clarifications are marked with correction bars.
1.1	Dec 1st 2000	*In order to keep this section more up-to-date, it has been removed from the Core Specification and is now a separate "living" document available on the member website.* • Errata items previously published on the web have been included. These corrections and clarifications are marked with correction bars.

CONTRIBUTERS

Johannes Elg (owner)	Ericsson Mobile Communications AB
Numerous SIG members contributed directly or indirectly to this revision