Video Electronics Standards Association

LS-EXT™ Standard

VESA
Video Electronics Standards Association
920 Hillview Court, Suite 140
Milpitas, CA 95035

Phone: (408) 957-9270
Fax: (408) 957-9277

VESA ENHANCED EDID LOCALIZED STRING EXTENSION STANDARD

Release A
July 10, 2003

Purpose

This standard defines an extension to the Enhanced EDID (E-EDID) data format used to provide specific types of string information.

Summary

This document describes a 128-byte data structure, provided as an optional extension to the base Enhanced EDID data structure. This structure provides data in the form of character strings used to describe the monitor. These strings are used to supplement or replace similar strings in the base EDID by providing more complete string descriptions and/or string descriptions using different languages or localizations.

Preface

Intellectual Property

While every precaution has been taken in the preparation of this standard, the Video Electronics Standards Association and its contributors assume no responsibility for errors or omissions, and make no warranties, expressed or implied, of functionality or suitability for any purpose.

Trademarks

All trademarks used within this document are the property of their respective owners. VESA, DDC, DPMS, EDID, EVC, P&D and VDIF are trademarks of the Video Electronics Standard Association. I^2C is a trademark owned by Philips.

Patents

VESA proposals and standards are adopted by the Video Electronics Standards Association without regard as to whether their adoption may involve any patents or articles, materials, or processes. Such adoption does not assume any liability to any patent owner, nor does it assume any obligation whatsoever to parties adopting the proposals or standards documents.

Support for this Standard

Clarifications and application notes to support this standard may be written. To obtain the latest standard and any support documentation, contact VESA.

If you have a product, which incorporates EDID, you should ask the company that manufactured your product for assistance. If you are a manufacturer, VESA can assist you with any clarification you may require. All comments or reported errors should be submitted in writing to VESA using one of the following methods.

- Fax: 408-957 9277, *direct this note to Technical Support at VESA*
- e-mail: support@vesa.org
- Mail: Technical Support
  Video Electronics Standards Association
  920 Hillview Court, Suite 140
  Milpitas, CA 95035

Revision History

Release A July 10, 2003
Initial release of the standard.

Acknowledgments

This document would not have been possible without the efforts of the VESA Display Committee. In particular, the following individuals and their companies contributed significant time and knowledge to this edition, and/or previous editions of the EDID document.

Bryan Young	Microsoft
Jack Hosek	NEC-Mitsubishi Electronics Display
Alain d'Hautecourt	ViewSonic
Syed Athar Hussain	ATI Technologies Inc.
George Wiley	Qualcomm

Table of Contents

1. OVERVIEW

1.1 Summary

The E-EDID extension defined in this document defines strings fields that can be used to replace those in the Base EDID structure. The definitions in this case allow the use of different string representations. The string length limit is also increased.

1.2 Background

The Extended Display Identification Data (EDID) standard already contains definitions for character string fields. However these definitions are limited to supporting ASCII representations only. The length of the string is also limited in the current definitions. This limits their use in internationally.

1.3 Standard Objectives

The EDID Standard was developed by VESA to meet, exceed and/or complement certain criteria. These criteria are set forth as Standard Objectives as follows:

1.4 Reference Documents

Note: Versions identified here are current, but users of this standard are advised to ensure they have the latest versions of referenced standards and documents.

- VESA Enhanced Extended Display Identification Data Standard - E-EDID
- Information about Unicode can be found at http://www.unicode.org
- ISO 639-2:ASCII-3: http://www.loc.gov/standards/iso639-2/englangn.html
- ISO 3166-1:Numeric 3 (UN): http://www.unicode.org/unicode/onlinedat/countries.html
- EISA/ISA PNPID: http://www.microsoft.com/hwdev/pnpid.asp
- SCSU Unicode Compression: http://www.unicode.org/unicode/reports/tr6/
- BOCU Unicode Compression: http://oss.software.ibm.com/icu/docs/papers/binary_ordered_compression_for_unicode.html

2. DATA FORMATS

2.1.1 Enhanced EDID High Level Layout

2.1.1.1 Mandatory elements

Block 0 is the only mandatory block.
This table shows the required use of E-EDID blocks. All blocks are 128 bytes in length.

Each extension block is structured according to Section 2.2.1.3. All extension blocks must be sequential, no holes allowed

Block #	**Block Description**
0	EDID 1.3 (or higher)
1	Extension if only 1 extension, otherwise EDID Block map (blocks 2-127)
2	Extension
3	Extension
4	Extension
:	
N	Extension
:	
128	EDID Block map for blocks 129 – 254 if more than 128 blocks used
129	Extension
:	
N =< 254	Extension

Block number 1 is used for Extension data if there is only one extension; otherwise block 1 is used as a block map.

2.1.1.2 EDID Block Map Extension

Byte #	Description	
0	Tag for Block Map	
1	Extension Tag for data in block 2 or block 129	Unused blocks are listed as Extension Tag = 0
2	Extension Tag for data in block 3 or block 130	
N	Extension Tag for data in block N+1 or block N+128	
126	Extension Tag for data in block 127 or block 254	
127	Check sum for this block map	

Block Tag is a byte that identifies the content of the Extension Block. A partial list of defined Tags is listed in Section 2.2.1.4.

2.1.1.3 General Extension Format

Byte #	Description	
0	Extension Tag	
1	Revision number for this tag	One byte binary number. Revisions are backward compatible.
2-126	Extension data	
127	Checksum for this Extension Block	

2.1.1.4 EDID Extension Tags Assigned by VESA

VESA will maintain a list of assigned EDID Extension Tags used to identify VESA Standard EDID Extensions.
For the most current list of EDID Extensions, see the VESA website.

Tag	Description
02h	Timing Extension
20h	EDID 2.0 Extension
40h	Display Information Extension (DI-EXT)
50h	Localized String Extension (LS-EXT)
60h	Microdisplay Interface Extension (MI-EXT)
F0h	Block Map
FFh	Extension defined by monitor manufacturer.

Note: At the time of the publication of this document, several of these extensions were not yet been defined or written. Contact VESA for the latest list of published EDID Extensions.

3. Localized String Extension

The Localized String Extension mechanism allows for the display of user-friendly information in the language and dialect of the user. It also enables a manufacturer to resolve legal issues in other countries by enabling the host operating system (OS) to select the correct trademark in the correct country. LS-EXT accomplishes this by using Unicode for all strings, which contains all the characters of the world, and thus will display correctly on any Unicode-compliant computer.

There can be multiple LS-EXT extension blocks in an E-EDID. In addition, there can be multiple string tables in each extension block. See section 3.5.3, "Language ID Structure", for rules on multiple string tables.

3.1 Extension Overview

Address	No. bytes		Description	Format
00h	**1**	**Bytes**	**Ext ID**	**50h**
01h	**2**	**Bytes**	**Extension Version / Revision**	
01h		1	Version #	Binary
02h		1	Revision #	Binary
03h	**2**	**Bytes**	**Unicode Version**	
03h		1	Major/Minor	Binary
04h		1	Update	Binary
05h	**Upto 122**	**Bytes**	**String Table(s)**	
7Fh	**1**	**Byte**	**Checksum**	**The 1-byte sum of all 128 bytes in this EDID block shall equal zero**

Table 3.1 – LS-EXT Extension Overview

The following sections provide details on each byte of the EDID Version 1 data structure.

3.2 Extension ID

This extension block is assigned the block tag of 50h. This tag is stored at the first byte of the extension block.

3.3 Extension Version/Revision

2	Bytes	LS-EXT Structure Version, Revision	
	1	Version no.	Binary
	1	Revision no.	Binary

Table 3.2 – LS-EXT Structure Version and Revision

The appropriate version and revision numbers shall be stored here. Products compliant with this document shall have Version = 1 and Revision = 0. Major revisions are backwardly compatible. Software should not reject an LS-EXT block due to revision mismatches.

3.4 Unicode Version and Revision Structure: 2 bytes

This structure defines what revision of *The Unicode Standard* this string block complies to. Software should not reject an LS-EXT block based on revision and must comply with *The Unicode Standards* stability policy, http://www.unicode.org/standard/stability_policy.html. This allows software to treat the string encodings in the most displayable manor.

2	Bytes	Bits	Unicode Version	
	1	7-4	Major	Binary
		3-0	Minor	Binary
	1		Update	Binary

Table 3.3 - Unicode Version and Revision Structure

3.5 String Table(s)

The extension is capable of supporting multiple string tables. Each table can supply strings describing the manufacturer, the model, and the serial number. Each table is associated with a localization. Each table follows the format shown below.

	Bytes	String Table	
	1	String Table Size	See section 3.5.1
	1	String Table Header	See section 3.5.2
	4	Language ID structure	See section 3.5.3
	1	Manufacturer name string length	x bytes
	x	Manufacturer name string	
	1	Model name string length	y bytes
	y	Model name string	
	1	Serial number string length	z bytes
	z	Serial number string	

Table 3.4 – String table structure

3.5.1 String Table Size

The size of the string table is given in bytes. This provides a reference point to the start of any additional string tables contained in each extension. Based on the strings types defined in this document the table size can be calculated as:

$$1+1+4+3*1+x+y+z = 9 + x + y + z$$

where:

x = the manufacturer name string length
y = the model name string length
z = the serial number string length

3.5.2 String Table Header

The string table header provides information used to decode the strings in the table.

Bit	Description	Detailed Description
7-3	Reserved	Reserved for future use
2-0	Unicode Transformation Format (UTF) type	Bit2 Bit1 Bit0 UTF type 0 0 0 UTF 8 0 0 1 UTF 16BE 0 1 0 UTF 32BE 1 1 1 Reserved all other combinations reserved

Table 3.5 – String Table Header

A string can be expressed in UTF16LE (Little Endian) or UTF32LE by using the appropriate Byte Order Mark (BOM). The BOM's for different UTFs are summarized below:

Bytes	Encoding Form
00h 00h FEh FFh	UTF-32, big-endian

FFh FEh 00h 00h	UTF-32, little-endian
FEh FFh	UTF-16, big-endian
FFh FEh	UTF-16, little-endian
EFh BBh BFh	UTF-8

Table 3.6 – Unicode Byte Order Markers (BOMs)

In depth-descriptions are in Section 3.8,"Transformations", part D35, Section 2.7, "Byte Order Mark (BOM)", and summarized at http://www.unicode.org/faq/utf_bom.html#2 of *The Unicode Standard 3.0.*

Since the string table is a stream of bytes, a BOM will not be necessary in most cases.

3.5.3 Language ID Structure

The Language ID structure consists of an optional compressed ASCII Language ID and a three-digit code number. All other bit fields are reserved and should be set to 0.

The Language ID bit field, if present, allows the operating system to select a string block that matches the language ID of the current logged in user. When present, the ISO 639-2 ASCII-3 English language ID is used to define the base language, but is stored in a compressed ASCII format. This format is similar to the compressed ASCII format of the PNPID in the base EDID. Compressed ASCII for this standard is a five-bit encoding of the subset of ASCII characters from 65 ('A') to 90 ('Z'), with SPACE mapped to zero. The encoding differs for ASCII codes 91 to 95. These encodings are not valid for this specification. If the language ID field is not present, this field should be filled in with the compressed ASCII form of the SPACE character. The bit pattern to use would be 000000000000000b. ISO 639-2 does not define any codes with spaces in them. Therefore, space cannot be present if a language ID is used in order to avoid confusion with ISO 639-1 ASCII-2 codes. Refer to Section 3.4 of the VESA Enhanced Extended Display Identification Data (E-EDID) Standard.

The country code ID bit field, if present contains an ISO 3166-3 numeric-3 country code. The country code can be used in combination with the language ID in order to handle different spellings, phrases, or dialects that are unique to a country's language usage. The country code can also be used to display different registered trademarks in different countries. If not used, the ISO 3166-3 numeric code field must be set to 0.

Only one string table with no language ID and no country code is permitted. This string table is called the **neutral string table** and must be the first string table encountered in the fist string extension block in the E-EDID. Block priority is determined by the segment register value that corresponds to the block, with smaller values having greater priority. In all cases at least one string in one string table must be selected.

In the event that a country-specific language string is not available, but a base language translation is available, the operating system (OS) should attempt to use the first language implementation found that does not cause problems. The OS is allowed to override these rules if such a match would lead to political problems. If the language ID cannot be matched, then the first string block found should be used that does not offend the country of the user. If no language ID can be matched, but a country code can be matched, then the OS should use the first country-specific implementation. If the country and language ID cannot be matched, and a string table is present that does not have a country and language ID (neutral string table), then that string table should be used. If no neutral string table is present, then the first string table present that does not cause political problems should be used. In all cases the OS must choose a string table to use.

For example, if the locale of a user is Brazilian Portuguese, but only Portugal Portuguese is available, then the Portugal Portuguese would be used. If Brazil and Portugal should go to war, then the OS may decide to choose the first translation available that does not cause political problems.

See section four for a summary of string priority.

Bit	Description	Detailed Description
31-30	Reserved	Set all bits to 0b
29-15	ISO 3166-3 Numeric-3 code	Code number
14-0	ISO 639-2 Compressed ASCII Language ID	Bits 14-10 First char; Bits 9-5 Middle char; Bits 4-0 Last char

Table 3.7 – Language ID encoding type

3.5.4 Manufacturer Name Data

Each table provides space for a manufacture name string. The string length field precedes the string. The length is expressed in bytes. A value of zero in this file indicates that a string of this type is not provided in the table. The string length in characters will vary from this number depending on the UTF type. The string immediately follows the string length filed. The string does not use a null terminator or any other type of terminator. The length field determines where the string ends.

	Bytes	String Table	
	1	Manufacturer string length	x bytes
	X	Manufacturer string	

Table 3.8– Manufacturer data

3.5.5 Model Name Data

Each table provides space for a model name string. The string length field precedes the string. The length is expressed in bytes. A value of zero in this file indicates that a string of this type is not provided in the table. The string length in characters will vary from this number depending on the UTF type. The string immediately follows the string length filed. The string does not use a null terminator or any other type of terminator. The length field determines where the string ends.

	Bytes	String Table	
	1	Model string length	y bytes
	Y	Model string	

Table 3.9 – Model data

3.5.6 Serial Number Data

Each table provides space for a serial number string. The string length field precedes the string. The length is expressed in bytes. A value of zero in this file indicates that a string of this type is not provided in the table. The string length in characters will vary from this number depending on the UTF type. The string immediately follows the string length filed. The string does not use a null terminator or any other type of terminator. The length field determines where the string ends.

	Bytes	String Table	
	1	Serial number string length	z bytes
	Z	Serial number string	

Table 3.10 – Model data

3.6 Additional string tables

Additional string tables based on a different localization ID or different country code can be included as space permits in the extension. Each table immediately follows the previous one and uses the same structure.

If remaining space in the extension block does not allow for a complete table, additional string tables can be provided using another localized string extension block. String tables must be contiguous, with no used bytes between tables. Each string table must reside completely within a single extension. Partial tables cannot be continued in a different extension. Unused space in the extension must be filled with zeros. See figure 3.1 for a diagram.

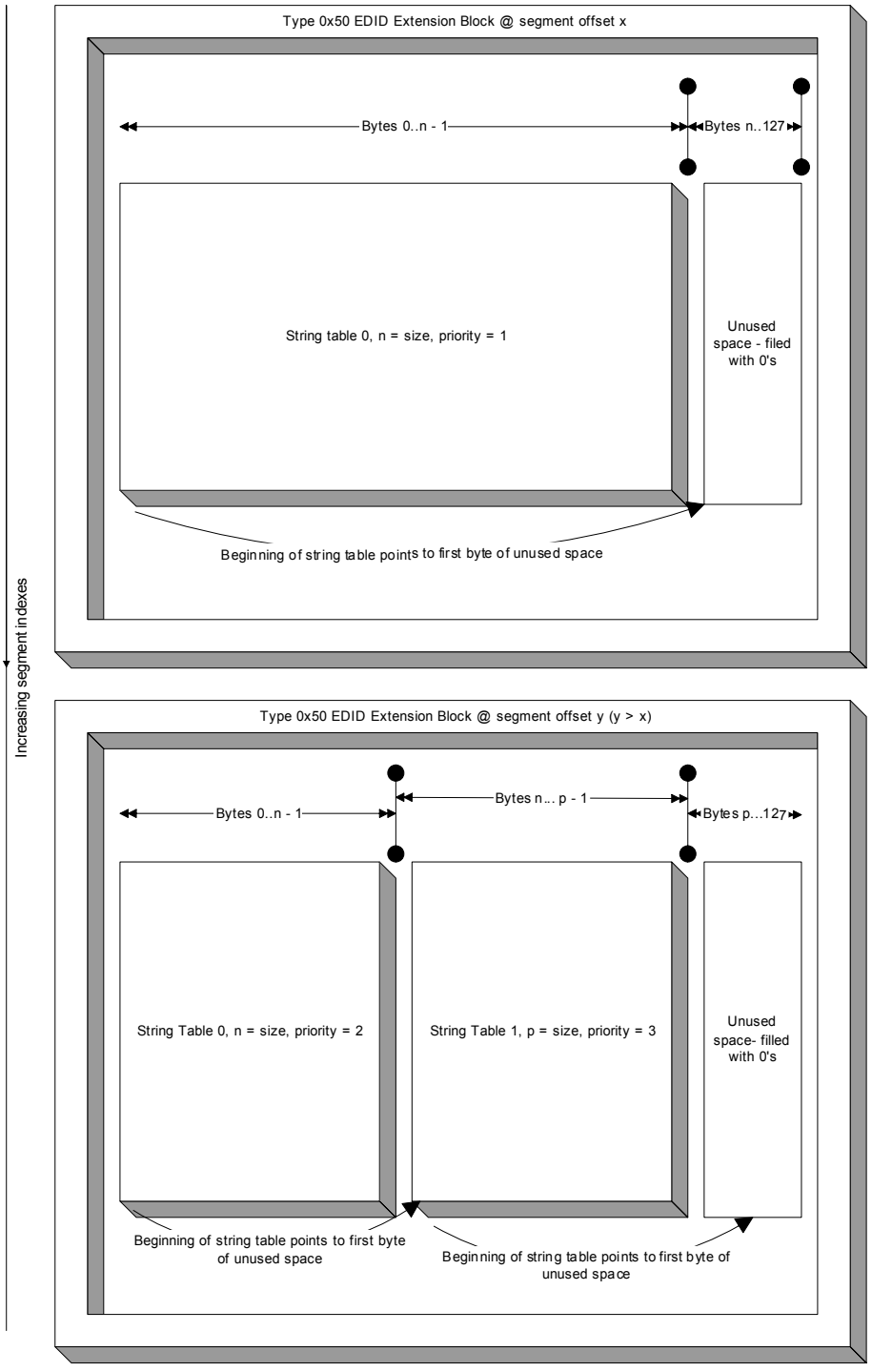

Figure 3.1 String table priority and organization within extension blocks

In this example one type of priority scheme is shown. This priority scheme assigns the first string table a value of 1. Subsequent string tables are assigned the successor integer. This algorithm defines the order to search the string tables in. For this example, string tables would be searched starting at the string table with a priority of 1.

3.7 Unused bytes

In any localized string extension, unused bytes between the end of the last table and the checksum bye shall be set to 0.

3.8 Checksum

1	Bytes	Description	Function
	1	Checksum	This byte should be programmed such that a one-byte checksum of the entire 128-byte EDID equals 0.

Table 3.11 - Checksum

4. String Priority

The following table tells what an OS should do when searching for a particular string table to use.

Case	Language ID state	Country ID state	Neutral Table state	OS Action
Case1	Matched	Matched	Don't care	Use matched string table
Case2	Matched	Not matched	Don't care	Use the first string table that matched the language ID
Case3	Not matched	Matched	Don't care	Use the first string table that matched the country ID
Case4	Not matched	Not matched	Present	Use the neutral string table
Case5	Not matched	Not matched	Not present	Use the first string table found in the first block: the string table that has a priority of 1.

Table 4.1 String Priority Cases

The OS reserves the right to veto these cases if there is a geopolitical issue that would prevent the normal display of the normally selected string tables information.

5. APPENDIX A - Sample Localized string extension data

5.1 UTF 16BE Full Default Neutral Table at Possible Segment ID of 2

Byte # (decimal)	Byte # (hex)	Field Name and *Comments*	Value (hex)	Value (binary)
00	00h	Extension ID	50h	01010000
01	01h	LS-Ext Version number	01h	00000001
02	02h	LS-Ext Revision number	00h	00000000
03	03h	Unicode Version – Major/Minor	32h	00110010
04	04h	Unicode Version – Update	00h	00000000
05	05h	String Table Size	28h	00101000
06	06h	String Table Header – UTF 8	00h	00000000
07	07h	Language ID structure – Default Neutral String Table	00h	00000000
08	08h		00h	00000000
09	09h		00h	00000000
10	0Ah		00h	00000000
11	0Bh	Manufacturer Name Data Length	0Eh	00001110
12	0Ch	‘D’	44h	01000100
13	0Dh	‘i’	69h	01101001
14	0Eh	‘s’	73h	01110011
15	0Fh	‘p’	70h	01110000
16	10h	‘l’	6Ch	01101100
17	11h	‘a’	61h	01100001
18	12h	‘y’	79h	01111001
19	13h	‘s’	73h	01110011
20	14h	‘,’	2Ch	00101100
21	15h	‘ ‘	20h	00100000
22	16h	‘I’	49h	01001001
23	17h	‘n’	6Eh	01101110
24	18h	‘c’	63h	01100011
25	19h	‘.’	2Eh	00101110
26	1Ah	Model Name String Length	06h	01100000
27	1Bh	‘F’	46h	01000110
28	1Ch	‘C’	43h	01000011
29	1Dh	‘1’	31h	00110001
30	1Eh	‘9’	39h	00111001
31	1Fh	‘0’	30h	00110000
32	20h	‘1’	31h	00110001
33	21h	Serial Number Data String Length	0Ch	00001100
34	22h	‘0’	30h	00110000
35	23h	‘3’	33h	00110011
36	24h	‘2’	32h	00110010
37	25h	’5’	35h	00110101
38	26h	‘-‘	2Dh	00101101
39	27h	‘N’	4Eh	01001110
40	28h	‘C’	43h	01000011
41	29h	’-‘	2Dh	00101101
42	2Ah	‘P’	50h	01010000
43	2Bh	‘R‘	52h	01010010
44	2Ch	‘-‘	2Dh	00101101
45	2Dh	‘0’	30h	00110000
46	2Eh	String Table Size	1Bh	00011011

47	2Fh	String Table Header - UTF 16BE	01h	00000000
48	30h	Language ID Structure – JPN/392	00h	00000000
49	31h		C4h	11000100
50	32h		2Ah	00101010
51	33h		0Eh	00001110
52	34h	Manufacturer ID Name Length	14h	00010100
53	35h	‘デ’	C7h	11000111
54	36h		30h	00110000
55	37h	‘ィ’	A3h	10100011
56	38h		30h	00110000
57	39h	‘ス’	B9h	10111001
58	3Ah		30h	00300000
59	3Bh	‘プ’	D7h	11010111
60	3Ch		30h	00110000
61	3Dh	‘レ’	ECh	11101100
62	3Eh		30h	00110000
63	3Fh	‘イ’	A4h	10100100
64	40h		30h	00110000
65	41h	‘株’	2Ah	00101010
66	42h		68h	01101000
67	43h	‘式’	0Fh	00001111
68	44h		5Fh	01011111
69	45h	‘会’	1Ah	00011010
70	46h		4Fh	01001111
71	47h	‘社’	3Eh	00101110
72	48h		79h	01111001
73	49h	Model Name String Length	00h	00000000
74	4Ah	Serial Number String Length	00h	00000000
75	4Bh	0	00h	00000000
76	4Ch	0	00h	00000000
77	4Dh	0	00h	00000000
78	4Eh	0	00h	00000000
79	4Fh	0	00h	00000000
80	50h	0	00h	00000000
81	51h	0	00h	00000000
82	52h	0	00h	00000000
83	53h	0	00h	00000000
84	54h	0	00h	00000000
85	55h	0	00h	00000000
86	56h	0	00h	00000000
87	57h	0	00h	00000000
88	58h	0	00h	00000000
89	59h	0	00h	00000000
90	5Ah	0	00h	00000000
91	5Bh	0	00h	00000000
92	5Ch	0	00h	00000000
93	5Dh	0	00h	00000000
94	5Eh	0	00h	00000000
95	5Fh	0	00h	00000000
96	60h	0	00h	00000000
97	61h	0	00h	00000000
98	62	0	00h	00000000
99	63	0	00h	00000000
100	64	0	00h	00000000

101	65	0	00h	00000000
102	66	0	00h	00000000
103	67	0	00h	00000000
104	68	0	00h	00000000
105	69	0	00h	00000000
106	6A	0	00h	00000000
107	6B	0	00h	00000000
108	6C	0	00h	00000000
109	6D	0	00h	00000000
110	6E	0	00h	00000000
111	6F	0	00h	00000000
112	70	0	00h	00000000
113	71	0	00h	00000000
114	72	0	00h	00000000
115	73	0	00h	00000000
116	74	0	00h	00000000
117	75	0	00h	00000000
118	76	0	00h	00000000
119	77	0	00h	00000000
120	78	0	00h	00000000
121	79	0	00h	00000000
122	7A	0	00h	00000000
123	7B	0	00h	00000000
124	7C	0	00h	00000000
125	7D	0	00h	00000000
126	7E	0	00h	00000000
127	7F	Checksum	5Ah	01000010

6. APPENDIX B - Answers To Commonly Asked Questions

Ref. #	Question	Answer
B1	Where can I find information about Unicode?	Unicode information may be located at http://www.unicode.org
B2	Where can I find information about ISO 639-2?	ISO 639 can be located at http://www.loc.gov/standards/iso639-2/
B3	Where can I find information about ISO 3166-3?	http://userpage.chemie.fu-berlin.de/diverse/doc/ISO_3166.html http://www.iso.ch/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html

7. APPENDIX C – Spec Compliance Checklist

7.1 All reserved bits in all fields are zero.

7.2 ISO 639-2 ASCII-3 code fields, if not used, contain the value 000000000000000b.

7.3 If a country code is not used, the country code field must be zero.

7.4 There are no SPACE characters used in an ASCII-3 code field.

7.5 Localized sting extension block contains Unicode characters from the latest released standard.

7.6 At least one string table contains one string.

7.7 One valid UTF encoding is selected per string table.

7.8 String tables are laid out in correct order by numerical segment index precedence, then order encountered in the block.

7.9 The neutral string table, if present, is the first table in the first block.

7.10 If multiple string tables are present in an extension block, then there are no unused bytes between string tables.

7.11 Unused bytes in an extension block should be filled with zeros.

7.12 There can only be one string table for every given combination of language IDs and country IDs.

If multiple string tables are present in an extension block, then they are complete

8. APPENDIX D – Glossary

8.1 Neutral String Table

The neutral string table is a string table that does not contain a language ID nor a country code.

THIS PAGE INTENTIONALLY LEFT BLANK