L2/98-355

HEX BYTE PICTURES FOR UNICODE

Frank da Cruz
The Kermit Project
Columbia University
New York City USA
fdc@columbia.edu
http://www.columbia.edu/kermit/

D R A F T # 2

Tue Nov 3 19:08:37 1998

THIS IS A PREFORMATTED PLAIN-TEXT ASCII DOCUMENT. IT IS DESIGNED TO BE VIEWED AS-IS IN A FIXED-PITCH FONT. ITS WIDEST LINE IS 79 COLUMNS. IT CONTAINS NO TABS. IF IT LOOKS MESSY TO YOU, PLEASE FEEL FREE TO PICK UP A CLEAN COPY OF THIS OR THE RELATED PROPOSALS BY ANONYMOUS FTP:

HEX BYTE PICTURES FOR UNICODE (plain text)
ftp://kermit.columbia.edu/kermit/ucsterminal/hex.txt

ADDITIONAL CONTROL PICTURES FOR UNICODE (plain text)
ftp://kermit.columbia.edu/kermit/ucsterminal/control.txt

TERMINAL GRAPHICS FOR UNICODE (plain text)
ftp://kermit.columbia.edu/kermit/ucsterminal/ucsterminal.txt

Glyph Map (PDF, contributed by Michael Everson)
ftp://kermit.columbia.edu/kermit/ucsterminal/terminal-emulation.pdf

Clarification of SNI Glyphs (Microsoft Word 7.0)
ftp://kermit.columbia.edu/kermit/ucsterminal/sni-charsets.doc

Discussion (plain text)
ftp://kermit.columbia.edu/kermit/ucsterminal/mail.txt

(Note, the Exhibits are on paper and not available at the FTP site.)

ABSTRACT

A set of characters is proposed for encoding 8-bit values and for displaying them in single cells for debugging and analysis purposes.

Please refer to the TERMINAL GRAPHICS FOR UNICODE proposal for a discussion of terminal emulation, including motivation for supporting it in Unicode, as well as for acknowledgements to those who helped with this set of proposals.

CONTENTS

NOTATION

. Numbers in (parentheses) are footnote references, keyed to footnotes at the bottom of the section in which they appear.
. Numbers in [brackets] are keyed to the References in Section 3.
. Letter-Digit in brackets refers to an Exhibit in Section 4.

For consistency, the References and Exhibits are the same as those in the accompanying, even though most of the items are not referenced here.

1. THE CASE FOR HEX BYTE CHARACTERS

A set of 256 hex byte-value picture characters is proposed for compatibility with existing terminals, line monitors, and protocol analyzers; for use in debugging of Unicode applications; and for data exchange with non-Unicode applications.

1.1. Hex Byte Pictures in Terminal Emulation

Certain real physical terminals can show byte values as 2 hexadecimal digits in a single screen cell. These include:

. DEC VT220 [5,6] in Display Controls Mode, uses the 32 hex byte pictures, 80-9F, to represent the 32 C1 control characters [A1-A2].

. DEC VT320 and above [7,8,9] use hex bytes 80-83 and 98-9A when displaying C1 control values in these ranges (and mnemonics for the others) [B1,B2,C1].

. Siemens Nixdorf 97801 includes 00 through 1F in its "IBM" character set [E4], and 80-9F in its character ROM [E6].

To emulate these terminals accurately, therefore, requires 32 hex-byte glyphs, 00-1F and 80-9F.

1.2. Hex Byte Pictures for Debugging

The widespread use of hex byte glyphs by protocol analyzers (e.g. see [N1]) and line monitors (increasingly PC-based) suggests a possible motivation for encoding all 256 possible hex bytes. Once encoded, these glyphs could also be used in terminal-emulator debug screens, word processors, file dump and analysis programs, Web browsers, and so on, for unambiguously showing the value of a given byte (or byte pair = Unicode character = 2 hex byte glyphs) in a data stream, buffer, or file.

1.3. Hex Byte Glyphs for Unknown Characters

Hex byte characters offer a solution to the increasingly common problem of unmappable characters when converting to Unicode from another character set.

Presently, unmappable characters are handled (in Web browsers, word processors, etc) in most cases by substituting or displaying the U+FFFD Replacement Character. In many cases this is adequate for display purposes. But developers, help-desk and support personnel, and even end-users could benefit from seeing the actual value. It could aid them, for example, in identifying the source character set and choosing the correct mapping, or in sending precise problem reports to misbehaving websites, etc.

Mapping unknown characters to Unicode characters keyed to their specific byte values would allow corrections to be made in partially converted documents, e.g. by search and replace in a Unicode editor or other

Unicode-based text utility.

Displaying the actual hex byte value in a single character cell allows (in most cases) a mixture of valid and invalid characters in monospaced screen displays without disrupting the formatting, e.g. of tabular information.

1.4. Hex Byte Characters for Data Exchange

When textual information is transferred from a non-Unicode host or application to a Unicode one, and the mapping from source to destination character set is incomplete or unknown, substitution of hex byte-value characters for the unknown source characters allows round-trip integrity without a need for the higher-level protocols that would otherwise be necessary, and which would no doubt proliferate and cause much unneeded labor and confusion.

1.5. Standard Codes Are Needed

A standard and uniform set of hex byte value characters and associated glyphs would allow any maker of Unicode-base software software to include debug / trace / dump or unmappable-character preservation capabilities simply by using standard Unicode characters (which would presumably find their way into standard fonts) for this purpose, rather than having to create mutually incompatible custom encodings and fonts.

This would allow copying and pasting into other applications, including into tech-support email, with the reasonable expectation that the hex bytes would arrive intact and this, in turn, should promote faster problem resolution and increased standards compliance.

Without a standard encoding, problem resolution and technical support in this area will remain the ordeal they are today, especially for the naive end-user.

2. CHARACTER AND GLYPH REPERTOIRE

One glyph is required for each hex byte code 00 through FF; 256 glyphs in all, as shown in Table 2.1, in which the "Code" column shows the temporary reference value for this document, E100-E1FF. Ideally (for efficiency in real-time debugging/display applications), the final 8 bits of the actual code would correspond to the 8-bit value represented by the corresponding glyph, as they do in the sample codes.

Table 2.1: Hex Byte Characters

```
Code  Byte  Description
E100   00   Symbol for Hex Byte 00
E101   01   Symbol for Hex Byte 01
:      :    :
E1FF   FF   Symbol for Hex Byte FF (1)
```

These characters should have the following properties:

```
Case:             No
Combining Class: 0
Combining Jamo:  No
Directionality:  Other Neutral (ON)
Jamo Short Name: No
Numeric Value:   No (2)
Private Use:     No
Surrogate:       No
Mirrored:        No
Mathematical:    No
```

Notes:

(1) Hex byte values can collide with control-character names: FF, D1, D2, D3, D4, etc, from the control-pictures sets proposed in ADDITIONAL CONTROL PICTURES FOR UNICODE. If both hex bytes and control pictures are implemented, the font designer should ensure they are distinct enough visually that they will not be confused.

(2) I do not have a strong opinion as to whether these characters should have the Numeric Value property; a case could be made either way.

To prevent cell-boundary ambiguity, the font designer should employ some visual device to bind the two hex digits together in an unmistakable way, for example by arranging them diagonally within the character cell as shown in Figure 2.1.

Figure 2.1: Suggested Glyph Format

```
 +--+ +--+ +--+      +--+ +--+ +--+ +--+      +--+ +--+      +--+--+
 |0 | |0 | |0 | ...  |0 | |1 | |1 | |1 | ...  |E | |F | ... |F |F |
 | 1| | 2| | 3|      | F| | 0| | 1| | 2|      | F| | 0|     | E| F|
 +--+ +--+ +--+      +--+ +--+ +--+ +--+      +--+ +--+      +--+--+
```

Summary:
256 new characters, U+E100 through U+E1FF.

Status:
Controversial. Should this proposal be rejected, a smaller selection of hex bytes is still required for the C1 control pictures set and for SNI "IBM" character-set glyphs: 00-1F and 80-9F (32 characters).

3. REFERENCES

[1] American National Standards Institute, ANSI X3.4-1986, Code for Information Interchange (ASCII), 1986.

[2] Data General, Programming the Display Terminal: Models D217, D413, and

D463, Westboro, MA, 1991.

[3] Digital Equipment Corporation, VT100 User Guide, EK-VT100-UG-002, Maynard, MA, 1979.

[4] Digital Equipment Corporation, VT102 Video Terminal User Guide, EK-VT102-UG-003, Maynard, MA, 1982.

[5] Digital Equipment Corporation, VT220 Owner's Manual, EK-VT220-UG-003, Maynard, MA, 1984.

[6] Digital Equipment Corporation, VT220 Series Programmer Reference Manual, EK-VT240-RM-002, Maynard, MA, 1984.

[7] Digital Equipment Corporation, VT330/VT340 Programmer Reference Manual, Volume 1: Text Programming, ED-VT3XX-TP-002, Maynard, MA, 1988.

[8] Digital Equipment Corporation, Installing and Using the VT420 Video Terminal EK-VT420-UG.002, Maynard, MA, 1988.

[9] Digital Equipment Corporation, VT520/VT525 Video Terminal Programmer Inforamtion, EK-VT520-RM.A01, Maynard, MA, 1994.

[10] Heathkit Manual for the Video Terminal Model H19, The Heath Company, Benton Harbor, MI, 1979.

[11] Hewlett Packard 2621A/P Interactive Terminal Owner's Manual, 1978.

[12] Hewlett Packard 2648A Graphics Terminal Reference Manual, 1977.

[13] IBM System/360 Principles of Operation, GA22-6821-8, Poughkeepsie, NY, 1970.

[14] IBM National Language Design Guide, Volume 2: National Language Support Reference Manual, 4th Edition, SE09-8002-03, North York ON, 1994.

[15] IBM 3270 Information Display System, Component Description, GA27-2749-10, 1980.

[16] IBM 3164 ASCII Color Display Station Description, GA18-2317-1, 1986.

[17] ISO International Standard 2022, Information processing -- ISO 7-bit and 8-bit coded character sets -- Code extension techniques, Third Edition, Geneva, 1986.

[18] ISO/IEC International Standard 6429, Information technology -- Control functions for coded character sets, Third Edition, Geneva, 1992.

[19] ISO/IEC 10646-1, International Standard 10646, Information Processing -- Multiple-Octet Coded Character Set,

1993-now.

[20] Perkin Elmer Model 1100 User's Manual, Randolph, NJ, 1978.

[21] Siemens Nixdorf, Bildschirmeinheit 97801-5xx Schnittstellen, Benutzerhandbuch, München, 1991.

[22] Televideo 922 Video Terminal Display Operator's Manual, Sunnyvale, CA, 1984.

[23] Televideo 965 Video Terminal Display Operator's Manual, Sunnyvale, CA, 1988.

[24] The Unicode Standard, Version 2.0, Addison-Wesley Developers Press, 1996.

[25] Wyse WY-60 Programmer's Guide, Wyse Technology, San Jose, CA, 1987.

[26] Wyse WY-370 Programmer's Guide, Wyse Technology, San Jose, CA, 1990.

[27] IBM 3270 Information Display System, Data Stream Programmer's Reference, GA23-0059-06, 1991.

[28] ISO International Register of Coded Characters to Be Used with Escape Sequences, European Computer Manufacturers Association (ECMA), Geneva, 1985-present.

[29] IBM Character Data Representation Architecture, Level 1 Registry, IBM Canada Ltd., National Language Technical Centre, Ontario, SC09-1391-00, 1990 (superseded by: IBM Character Data Representation Architecture, Registration and Registry, IBM Canada Ltd., Toronto, SC09-2190-00, 1995).

[30] Knuth, Donald, "TeX and METAFONT, New Directions in Typesetting", American Mathematical Society / Digital Press, Bedford MA, 1979.

[31] Apple Computer Corporation, Inside Macintosh, 1984.

[32] HDS-3200 Terminal Series Owner's Manual, Philadelphia PA, 1987.

[33] Zenith Data Systems Video Terminal Z-19-CN Operation Manual, Saint Joseph, MI, 1981.

[34] Interview 30A/40A Operator's Field Reference Guide, Atlantic Research Corporation, ATLC-107-919-101, Alexandria, VA, 1982.

4. EXHIBITS

The following exhibits, available only on paper, are reproduced from the terminal manuals indicated by the numeric reference number. Each exhibit is 1 page unless otherwise indicated.

[A1] VT220 Display Controls Font (Left Half) [5].

[A2] VT220 Display Controls Font (Right Half) [5].

[A3] VT220 DEC Special Graphics Character Set [5].

[B1] VT320 Display Controls Font (Left Half) [7].

[B2] VT320 Display Controls Font (Right Half) [7].

[C1] VT420 Display Controls Font (Both Halves) [8].

[C2] VT420 DEC Technical Character Set [8].

[C3] HDS-3200 DEC Technical Character Set [32].

[D1] Data General US ASCII Character Set [2].

[D2] Data General Word-Processing, Greek, and Math Character Set [2].

[D3] Data General Line Drawing Character Set [2].

[D4] Data General Special Graphics Character Set [2].

[D5] Data General VT Multinational Character Set [2].

[D6] Data General VT Special Graphics Character Set [2].

[D7] Data General ISO 8859/1.2 Character Set [2].

[E1] Siemens Nixdorf 97801 ISO 8859-1 Character Set [21].

[E2] Siemens Nixdorf 97801 Klammern (Brackets) Character Set [21].

[E3] Siemens Nixdorf 97801 Facet Character Set [21].

[E4] Siemens Nixdorf 97801 IBM Character Set [21].

[E5] Siemens Nixdorf 97801 Math Character Set [21].

[E6] Siemens Nixdorf 97801 Character Generator (8 pages) [21].

[F1] Wyse 60 Native, Multinational, PC, and ASCII Character Sets [25].

[F2] Wyse 60 Graphics 1, 2, and 3 Character Sets [25].

[F3] Wyse 60 Standard ANSI, ANSI Graphics, and UK ANSI Character Sets [25].

[G1] Wyse 370 Controls Display Mode (74Hz) [26].

[G2] Wyse 370 Controls Display Mode (60Hz) [26].

[G3] Wyse 370 C0, ASCII, and Special Graphics Character Sets [26].

[G4] Wyse 370 C1, Multinational, and Latin-1 Character Sets [26].

[H1] IBM 3270 Operator Information Area Symbols (10 pages) [15].

[I1] TeX Standard Extension Font [30].

[J1] Apple Symbol Font (2 pages) [31].

[K1] Hewlett Packard 2621A/P National Terminal Character Set [11].

[L1] Heath/Zenith-19 Graphic Symbols (2 pages) [33].

[M1] Televideo 922 ASCII, Supplemental, Special Character Sets (4 pages) [22].

[N1] Sample screen from a data analyzer showing hex display [34].

(End)