DISPATCH Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

C. Bormann
Universitaet Bremen TZI

July 08, 2019

Modern Network Unicode
draft-bormann-dispatch-modern-network-unicode-02

Abstract

RFC 5198 both defines common conventions for the use of Unicode in network protocols and caters for the specific requirements of the legacy protocol Telnet. In applications that do not need Telnet compatibility, some of the decisions of RFC 5198 are cumbersome.

The present specification defines "Modern Network Unicode" (MNU), which is a form of RFC 5198 Network Unicode that can be used in specifications that require the exchange of plain text over networks and where just mandating UTF-8 (RFC 3629) may not be sufficient, but there is also no desire to import all of the baggage of RFC 5198.

In addition to a basic "Clean Modern Network Unicode" (CMNU), this specification defines a number of variances that can be used to tailor MNU to specific areas of application. In particular, "Modern Network Unicode with lines" can be used in applications that require line-structured text such as plain text documents or markdown format.

Status

The present version of this document represents the author's reaction to initial exposure on the art@ietf.org mailing list. Some more editorial cleanup is probably desirable, but could not be achieved in time for the IETF105 Internet-Draft deadline.

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2020.

Table of Contents

1. Introduction

(Insert embellished copy of abstract here.)

Complex specifications that use Unicode often come with detailed information on their Unicode usage; this level of detail generally is necessary to support some legacy applications. New, simple protocol specifications generally do not have such a legacy or need such details, but can instead simply use common practice, informed by decades of using Unicode. The present specification attempts to serve as a convenient reference for such protocol specifications, reducing their need for discussing Unicode to just pointing to the present specification and making a few simple choices.

There is no intention that henceforth all new protocols "must" use the present specification. It is offered as a standards-track specification simply so it can be normatively referenced from other standards-track specifications.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Characters in this specification are named with their Unicode name notated in the usual form U+NNNN or with their ASCII names (such as CR, LF, HT, RS, NUL) [RFC0020].

2. Clean Modern Network Unicode

Clean Modern Network Unicode (CMNU) is the form of Modern Network Unicode that does not make use of any of the variances defined below. It requires conformance to [RFC3629], as well as to the following four mandates:

- Control characters (U+0000 to U+001F and U+007F to U+009F) MUST NOT be used. (Note that this also excludes line endings, so a CMNU text string cannot extend beyond a single line. See Section 3.1 below if line structure is needed.)

- The characters U+2028 and U+2029 MUST NOT be used. (In case future Unicode versions add to the Unicode character categories Zl or Zp, any characters in these categories MUST NOT be used.)

- Modern Network Unicode requires that, except in very unusual circumstances, all text is transmitted in normalization form NFC.

- As per the Unicode specification, the code points U+FFFE and U+FFFF MUST NOT be used. Also, Byte Order Marks (leading U+FEFF characters) MUST NOT be used.

3. Variances

In addition to CMNU, this specification describes a number of variances that can be used in the form "Modern Network Unicode with VVV", or "Modern Network Unicode with VVV, WWW, and ZZZ" for multiple variances used. Specifications that cannot directly use CMNU may be able to use MNU with one or more of these variances added.

3.1. With lines

While Clean Modern Network Unicode rules out line endings completely, line-structured text is often required. The variance "with lines" allows the use of line endings, represented by a single LF character (which is then the only control character allowed).

3.2. With CR-tolerant lines

The variance "with CR-tolerant lines" allows the sequence CR LF as well as a single LF character as a line ending. This may enable existing texts to be used as MNU without processing at the sender side (substituting that by processing at the receiver side). Note that, with this variance, a CR character cannot be used anywhere else but immediately preceding an LF character.

3.3. With HT Characters

In some cases, the use of HT characters ("TABs") cannot be completely excluded. The variance "with HT characters" allows their use, without attempting to define their meaning (e.g., equivalence with spaces, column definitions, etc.).

3.4. With CCC Characters

Some applications of MNU may need to add specific control characters, such as RS [RFC7464] or FF characters. This variance is spelled with the ASCII name of the control character for CCC, e.g., "with RS characters".

3.5. With NFKC

Some applications require a stronger form of normalization than NFC. The variance "with NFKC" swaps out NFC and uses NFKC instead. This is probably best used in conjunction with "with Unicode version NNN".

3.6. With Unicode Version NNN

Some applications need to be sure that a certain Unicode version is used. The variance "with Unicode version NNN" (where nnn is a Unicode version number) defines the Unicode version in use as NNN. Also, it requires that only characters assigned in that Unicode version are being used.

4. Discussion

At the time of writing, RFCs are formatted in "Modern Network Unicode with CR-tolerant lines and FF characters".

4.1. Relationship to RFC 5198

The third and fourth requirement listed above are also posed by [RFC5198], while the first two remove further legacy compatibility considerations.

[RFC5198] contains some discussion and background material that the present document does not attempt to repeat; the interested reader may therefore want to consult it as an informative reference. See also Section 4 below.

Mandates of [RFC5198] that are specific to a version of Unicode are not picked up in this specification, e.g., there is no check for unassigned code points. Note that this means that a CMNU implementation may not be able to handle the normalization of a character not yet assigned in the version of Unicode that it uses. (See also Section 3.6 below.)

4.2. Going beyond RFC 5198

The handling of line endings (not being part of CMNU, providing LF-only and LF/CRLF line endings as variances) may be controversial. In particular, calling out CR-tolerance as an extra (and often undesirable) feature may seem novel to some readers. The handling as specified here is much closer to the way line endings are handled on the software side than the cumbersome rules of [RFC5198]. More generally speaking, one could say that the present specification is intended to be used by state of the art protocols going forward, maybe less so by existing protocols that have legacy baggage.

Even in the "with CR-tolerant lines" variance, the CR character is only allowed as an embellishment of an immediately following LF character. This reflects the fact that overprinting has only seen niche usage for quite a number of decades now.

Unicode Line and Paragraph separators probably seemed like a good idea at the time, but have not taken hold. Today, their occurrence is more likely to trigger a bug or even serve as an attack.

HT characters ("TABs") were needed on ASR33 terminals to speed up whitespace processing at 110 bit/s line speed. Unless some legacy applications require compatibility with this ancient and frequently varied convention, HT characters are no longer appropriate in Modern Network Unicode. In support of legacy compatibility cases that do require tolerating their use, the "with HT characters" variance is defined.

The version-nonspecific nature of CMNU creates some fuzziness that may be undesirable but is more realistic in environments where applications choose the Unicode version with the Unicode library that happens to be available to them.

With respect to Normalization (NFC), the unusual circumstances alluded to above can come from the the fact that some implementations of applications may rely on operating system libraries over which they have little control. Adherence to the robustness principle suggests that receivers of Modern Network Unicode should be prepared to receive unnormalized text and should not react to that in excessive ways; however, there also is no expectation for receivers to go out of their way doing so.

Some background on the prohibition of byte order marks: The 16-bit and 32-bit encodings for Unicode are available in multiple byte orders. The byte order in use in a specific piece of text can be provided by metadata (such as a media type) or by prefixing the text with a "Byte Order Mark", U+FEFF. Since code point U+FFFE is never used in Unicode, this unambiguously identifies the byte order.

For UTF-8, there is no ambiguity and thus no need for a byte order mark. However, some systems have made regular of a leading U+FEFF character in UTF-8 files, anyway, often in order to mark the file as UTF-8 in case other character codings are also in use and metadata is not available. This can wreak havoc with the ASCII compatibility of UTF-8; it also creates problems when systems then start to expect a BOM in UTF-8 input and none is provided. Section 6 of [RFC3629] also recommends not using Byte Order Marks with UTF-8, but does not phrase this as an unambiguous mandate, so we add that here.

5. Using ABNF with Unicode

Internet STD 68, [RFC5234], defines Augmented BNF for Syntax Specifications: ABNF. Since the late 1970s, ABNF has often been used to formally describe the pieces of text that are meant to be used in an Internet protocol. ABNF was developed at a time when character coding grew more and more complicated, and even in its current form, discusses encoding of characters only briefly (Section 2.4 of [RFC5234]). This discussion offers no information about how this should be used today (it actually still refers to 16-bit Unicode!).

The best current practice of using ABNF for Unicode-based protocols is as follows: ABNF is used as a grammar for describing sequences of Unicode code points, valued from 0x0 to 0x10FFFF. The actual encoding (as UTF-8) is never seen on the ABNF level; see Section 9.4 of [RFC6020] for a recent example of this. Approaches such as representing the rules of UTF-8 encoding in ABNF (see Section 3.5 of [RFC5255] as an example) add complexity without benefit and are NOT RECOMMENDED.

ABNF features such as case-insensitivity in literal text strings essentially do not work for general Unicode; text string literals therefore (and by the definition in Section 2.3 of [RFC5234]) are limited to ASCII characters. That is often not actually a problem in text-based protocol definitions. Still, characters beyond ASCII need to be allowed in many productions. ABNF does not have access to Unicode character categories and thus will be limited in its expressiveness here. The core rules defines in Appendix B of [RFC5234] are limited to ASCII as well; new rules will therefore need to be defined in any protocol employing modern Unicode.

The present specification recommends defining the following rules:

```
; modern unicode character:
uchar = %x20-7E / %xA0-2027 / %x202A-D7FF
      / %xE000-FFFD / %x10000-10FFFD
; modern unicode newline:
unl = %x0A
; alternatively, modern unicode CR-tolerant newline:
utnl = [%x0D] %x0A
; if really needed, HT-tolerant unicode character:
utchar = %x09 / uchar
```

6. IANA considerations

This specification places no requirements on IANA.

7. Security considerations

The security considerations of [RFC5198] apply.

A variance "with NUL characters" would create specific security considerations as discussed in the security considerations of [RFC5198] and should therefore only be used in circumstances that absolutely do require it.

8. References

8.1. Normative References

[RFC0020] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <https://www.rfc-editor.org/info/rfc20>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/info/rfc3629>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, DOI 10.17487/RFC5198, March 2008, <https://www.rfc-editor.org/info/rfc5198>.

[RFC5255] Newman, C., Gulbrandsen, A., and A. Melnikov, "Internet Message Access Protocol Internationalization", RFC 5255, DOI 10.17487/RFC5255, June 2008, <https://www.rfc-editor.org/info/rfc5255>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC7464] Williams, N., "JavaScript Object Notation (JSON) Text Sequences", RFC 7464, DOI 10.17487/RFC7464, February 2015, <https://www.rfc-editor.org/info/rfc7464>.

Acknowledgements

Klaus Hartke and Henk Birkholz drove the author out of his mind enough to make him finally write this up. James Manger, Tim Bray and Martin Thomson provided comments on an early version of this draft.

Author's Address

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org