Abstract

This document extends the base definition of ABNF (Augmented Backus-Naur Form) to include comprehensive support for certain symbols related to ASCII, and defines an import syntax.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any 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 April 12, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
1. Comprehensive Core Rule Update and Import Syntax

Augmented Backus-Naur Form (ABNF) [RFC5234] is a formal syntax that is popular among many Internet specifications. Many Internet documents employ this syntax along with the Core Rules defined in Appendix B.1 of [RFC5234]. However, the Core Rules do not specify many symbols in the ASCII range that are also needed by these relying documents, forcing document authors to define them as local rules. Sometimes different documents define these common symbols in different ways, resulting in confusion or incompatibility when the rules are misread or are combined with other sets of rules. Furthermore, [RFC5234] does not clarify whether referencing [RFC5234] for ABNF automatically defines its Core Rules.

[RFC5234] also lacks a syntax for importing rules from other specifications. Instead, authors have been required to name the rules and sources in the specification prose. While this method has served authors well, it has hampered machine-readable ABNF efforts for services such as syntax highlighting, automatic grammar checking, and compiling into target computer languages.

This document provides Core Rules that include comprehensive support for certain symbols, namely DELETE (DEL) and the C0 control characters in [ASCII86], which for purposes of this document is equivalent to [RFC0020].

To import a rule, define the rule with a local rule name, and put the reference to the rule in a prose-val. The rule syntax is:

"<" [ rulename "@" ] (import-ref / import-uri) ">"

The form import-ref is a document reference. In IETF-related publications, import-ref will be enclosed in square brackets, such as "[RFC1605]".

The form import-uri is supposed to be a Uniform Resource Indicator [RFC3986], but a machine implementation is not required to validate conformance to the URI production of [RFC3986]. Fragment components might be present, but only if the resource defines the fragment to mean a range of text (i.e., not just a point in the text).

When the 'rulename "@"' syntax is present, the rulename production preceding the "@" specifies the name of the rule in the reference. When the 'rulename "@"' syntax is absent, the name of the rule in the reference is the same as the name of the rule in the rule definition preceding the ".".
[[DISCUSS: Alternative delimiters? Right now this syntax shares < > with prose-val; this is intentional for compatibility and to reduce symbol proliferation.]]

[[DISCUSS: ABNF for this ABNF? The author considers it very undesirable to import URI normatively from RFC 3986. URI is very complicated and RFC 3986 predates RFC 5234 anyway. Need clean break with the past. import-uri = VCHAR could work since VCHAR does not include spaces, and most free-form prose will include at least one space.]]

Formally, this document does not make changes to [RFC5234]. Authors need to reference this document if they want to include these enhancements; bare references to [RFC5234] do not include this specification (or, for that matter, [RFC7405]). This directive follows a model whereby document authors can choose whether to invoke particular enhancements to ABNF. As time goes on, the IETF can determine how often these enhancements are invoked, and decide whether to include them as part of a revision to the base [RFC5234].

A reference to this document invokes the import syntax enhancement, as well as all of the Core Rules of Appendix A (i.e., the Core Rules do not have to be imported).

Appendix A of this document is meant to mirror Appendix B.1 of [RFC5234]. Document authors who reference this document should use the rules of Appendix A, and should not attempt to redefine or augment them (except for backwards compatibility with prior documents).

2. IANA Considerations

This document implies no IANA considerations.

3. Security Considerations

Security is truly believed to be irrelevant to this document.

4. References

4.1. Normative References


4.2. Informative References


Appendix A. Comprehensive Core Rules

Certain basic rules are in uppercase, such as SP, HTAB, CRLF, DIGIT, ALPHA, etc.

\[
\begin{align*}
\text{ALPHA} &= \%x41-5A / \%x61-7A \quad \text{A-Z / a-z} \\
\text{BIT} &= "0" / "1" \\
\text{CHAR} &= \%x01-7F \\
&\quad \text{; any 7-bit US-ASCII character,} \\
&\quad \text{excluding NUL} \\
\text{CR} &= \%x0D \\
&\quad \text{; carriage return} \\
\text{CRLF} &= \text{CR LF} \\
&\quad \text{; Internet standard newline} \\
\text{CTL} &= \%x00-1F / \%x7F \\
&\quad \text{; controls} \\
\text{DIGIT} &= \%x30-39 \\
&\quad \text{; 0-9} \\
\text{DQUOTE} &= \%x22 \\
&\quad \text{; " (Double Quote)} \\
\text{HEXDIG} &= \text{DIGIT / "A" / "B" / "C" / "D" / "E" / "F"} \\
\text{HTAB} &= \%x09 \\
&\quad \text{; horizontal tab} \\
\text{LF} &= \%x0A \\
&\quad \text{; linefeed} \\
\text{LWSP} &= *(\text{WSP / CRLF WSP}) \\
&\quad \text{; Use of this linear-white-space rule} \\
&\quad \text{permits lines containing only white} \\
&\quad \text{space that are no longer legal in} \\
&\quad \text{mail headers and have caused} \\
&\quad \text{interoperability problems in other} \\
&\quad \text{contexts.} \\
&\quad \text{Do not use when defining mail} \\
&\quad \text{headers and use with caution in} \\
&\quad \text{other contexts.} \\
\text{OCTET} &= \%x00-FF
\end{align*}
\]
; 8 bits of data

SP    =  %x20
VCHAR =  %x21-7E
        ; visible (printing) characters
WSP   =  SP / HTAB
        ; white space

NUL   =  %d0
SOH   =  %d1
STX   =  %d2
ETX   =  %d3
EOT   =  %d4
ENQ   =  %d5
ACK   =  %d6
BEL   =  %d7
BS    =  %d8
HT    =  %d9 ; also defined as HTAB

VT    =  %d11
FF    =  %d12 ; (literally used in every RFC)

SO    =  %d14
SI    =  %d15
DLE   =  %d16
DC1   =  %d17
DC2   =  %d18
DC3   =  %d19
DC4   =  %d20
NAK   =  %d21
SYN   =  %d22
ETB   =  %d23
CAN   =  %d24
EM    =  %d25
SUB   =  %d26
ESC   =  %d27
FS    =  %d28
GS    =  %d29
RS    =  %d30
US    =  %d31

DEL   =  %d127
Appendix B. Guidance for Rule Names for C1 Controls and Other Desiderata

Internet protocols have been migrating to Unicode and specifically UTF-8 for general text encoding. Authors need to consider the presence and possible effects of characters and code points beyond ASCII. See [RFC5198]. Therefore, the following rule names MAY take on special meanings. This document does not formally define these rule names, nor does this document prohibit other specifications from using them. However, authors ought only to use these rule names in their normal and natural senses. For the underlying sources, consult [UNICODE] and [RFC1345].

ABNF rules resolve into a string of terminal values. Such a value "is merely a non-negative integer"; only context can furnish a specific mapping of values into a character set. [RFC5234] Therefore, even if Unicode is specified, mappings between terminal values beyond %x7F may be encoded to different bit combinations depending on the encoding method.

This document does not purport to change the character set of ABNF itself, which remains [ASCII86]. (See [RFC5234].)

[[DISCUSS: what if you include ABNF in a UTF-8 document and you really want to use characters beyond ASCII in literals? Foreseeable? Dangerous?]]

<table>
<thead>
<tr>
<th>ASCII</th>
<th>terminal values between 0 - 7F (cf. CHAR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C0</td>
<td>synonym for CTL</td>
</tr>
<tr>
<td></td>
<td>[[DISCUSS: migrate to Appendix A?]]</td>
</tr>
<tr>
<td>UNICODE</td>
<td>terminal values representing 0 - 10FFFF</td>
</tr>
<tr>
<td>BEYONDASCII</td>
<td>terminal values representing 80 - 10FFFFF</td>
</tr>
<tr>
<td></td>
<td>[[DISCUSS: these definitions include all code points, including surrogate code points, which are not valid or encodable in UTF-8.]]</td>
</tr>
<tr>
<td>C1</td>
<td>terminal values representing 80 - 9F</td>
</tr>
<tr>
<td>PAD</td>
<td>terminal value representing 80</td>
</tr>
<tr>
<td>HOP</td>
<td>terminal value representing 81</td>
</tr>
<tr>
<td>BPH</td>
<td>terminal value representing 82</td>
</tr>
<tr>
<td>NBH</td>
<td>terminal value representing 83</td>
</tr>
<tr>
<td>IND</td>
<td>terminal value representing 84</td>
</tr>
<tr>
<td>NEL</td>
<td>terminal value representing 85</td>
</tr>
<tr>
<td>NL</td>
<td>terminal value possibly representing CRLF, CR, LF, NEL, or any combination thereof (but not LS or PS)</td>
</tr>
<tr>
<td>SSA</td>
<td>terminal value representing 86</td>
</tr>
<tr>
<td>ESA</td>
<td>terminal value representing 87</td>
</tr>
</tbody>
</table>
HTS  terminal value representing 88
HTJ  terminal value representing 89
VTS  terminal value representing 8A
PLD  terminal value representing 8B
PLU  terminal value representing 8C
RI   terminal value representing 8D
SS2  terminal value representing 8E
SS3  terminal value representing 8F
DCS  terminal value representing 90
PU1  terminal value representing 91
PU2  terminal value representing 92
STS  terminal value representing 93
CCH  terminal value representing 94
MW   terminal value representing 95
SPA  terminal value representing 96
EPA  terminal value representing 97
SOS  terminal value representing 98
SGCI terminal value representing 99
SCI  terminal value representing 9A
CSI  terminal value representing 9B
ST   terminal value representing 9C
OSC  terminal value representing 9D
PM   terminal value representing 9E
APC  terminal value representing 9F
NBSP terminal value representing A0
SHY  terminal value representing AD
LS   terminal value representing 2028
PS   terminal value representing 2029

Author's Address

Sean Leonard
Penango, Inc.
5900 Wilshire Boulevard
21st Floor
Los Angeles, CA  90036
USA

EMail: dev+ietf@seantek.com
URI:  http://www.penango.com/