

The Text/Plain Format and DelSp Parameters

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

This specification establishes two parameters (Format and DelSP) to be used with the Text/Plain media type. In the presence of these parameters, trailing whitespace is used to indicate flowed lines and a canonical quote indicator is used to indicate quoted lines. This results in an encoding which appears as normal Text/Plain in older implementations, since it is in fact normal Text/Plain, yet provides for superior wrapping/flowing, and quoting.

This document supersedes the one specified in RFC 2646, "The Text/Plain Format Parameter", and adds the DelSp parameter to accommodate languages/coded character sets in which ASCII spaces are not used or appear rarely.

Table of Contents

1.	Introduction.	2
2.	Conventions Used in this Document	2
3.	The Problem	3
3.1.	Paragraph Text.	3
3.2.	Embarrassing Line Wrap	3
3.3.	New Media Types	4
4.	The Format and DelSp Parameters	5
4.1.	Interpreting Format=Flowed.	6
4.2.	Generating Format=Flowed	7
4.3.	Usenet Signature Convention	9
4.4.	Space-Stuffing	9

4.5. Quoting	9
4.6. Digital Signatures and Encryption	11
4.7. Examples.	12
5. Interoperability.	12
6. ABNF.	13
7. Failure Modes	14
7.1. Trailing White Space Corruption	14
8. Security Considerations	15
9. IANA Considerations	15
10. Internationalization Considerations	15
11. Acknowledgments	15
12. Normative References.	16
13. Informative References.	16
Appendix A: Changes from RFC 2646	18
Author's Address.	19
Full Copyright Statement.	20

1. Introduction

Interoperability problems have been observed with erroneous labelling of paragraph text as Text/Plain, and with various forms of "embarrassing line wrap". (See Section 3.)

Attempts to deploy new media types, such as Text/Enriched [Rich] and Text/HTML [HTML] have suffered from a lack of backwards compatibility and an often hostile user reaction at the receiving end.

What is required is a format which is in all significant ways Text/Plain, and therefore is quite suitable for display as Text/Plain, and yet allows the sender to express to the receiver which lines are quoted and which lines are considered a logical paragraph, and thus eligible to be flowed (wrapped and joined) as appropriate.

2. Conventions Used in this Document

The key words "REQUIRED", "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [KEYWORDS].

The term "paragraph" is used here to mean a series of lines which are logically to be treated as a unit for display purposes and eligible to be flowed (wrapped and joined) as appropriate to fit in the display window and when creating text for replies, forwarding, etc.

3. The Problem

The Text/Plain media type is the lowest common denominator of Internet email, with lines of no more than 998 characters (by convention usually no more than 78), and where the carriage-return and line-feed (CRLF) sequence represents a line break (see [MIME-INT] and [MSG-FMT]).

Text/Plain is usually displayed as preformatted text, often in a fixed font. That is, the characters start at the left margin of the display window, and advance to the right until a CRLF sequence is seen, at which point a new line is started, again at the left margin. When a line length exceeds the display window, some clients will wrap the line, while others invoke a horizontal scroll bar.

Text which meets this description is defined by this memo as "fixed".

Some interoperability problems have been observed with this format:

3.1. Paragraph Text

Many modern programs use a proportional-spaced font, and use CRLF to represent paragraph breaks. Line breaks are "soft", occurring as needed on display. That is, characters are grouped into a paragraph until a CRLF sequence is seen, at which point a new paragraph is started. Each paragraph is displayed, starting at the left margin (or paragraph indent), and continuing to the right until a word is encountered which does not fit in the remaining display width. This word is displayed at the left margin of the next line. This continues until the paragraph ends (a CRLF is seen). Extra vertical space is left between paragraphs.

Text which meets this description is defined by this memo as "flowed".

Numerous software products erroneously label this format as Text/Plain, resulting in much user discomfort.

3.2. Embarrassing Line Wrap

As Text/Plain messages are quoted in replies or forwarded messages, each line gradually increases in length, eventually being arbitrarily hard wrapped, resulting in "embarrassing line wrap". This produces text which is, at best, hard to read, and often confuses attributions.

Example:

```
>>>>>This is a comment from the first message to show a
>quoting example.
>>>>>This is a comment from the second message to show a
>quoting example.
>>>>This is a comment from the third message.
>>>This is a comment from the fourth message.
```

It can be confusing to assign attribution to lines 2 and 4 above.

In addition, as devices with display widths smaller than 79 or 80 characters become more popular, embarrassing line wrap has become even more prevalent, even with unquoted text.

Example:

```
This is paragraph text that is
meant to be flowed across
several lines.
However, the sending mailer is
converting it to fixed text at
a width of 72
characters, which causes it to
look like this when shown on a
PDA with only
30 character lines.
```

3.3. New Media Types

Attempts to deploy new media types, such as Text/Enriched [Rich] and Text/HTML [HTML] have suffered from a lack of backwards compatibility and an often hostile user reaction at the receiving end.

In particular, Text/Enriched requires that open angle brackets ("<") and hard line breaks be doubled, with resulting user unhappiness when viewed as Text/Plain. Text/HTML requires even more alteration of text, with a corresponding increase in user complaints.

A proposal to define a new media type to explicitly represent the paragraph form suffered from a lack of interoperability with currently deployed software. Some programs treat unknown subtypes of TEXT as an attachment.

What is desired is a format which is in all significant ways Text/Plain, and therefore is quite suitable for display as Text/Plain, and yet allows the sender to express to the receiver which lines can be considered a logical paragraph, and thus flowed (wrapped and joined) as appropriate.

4. The Format and DelSp Parameters

This specification defines two MIME parameters for use with Text/Plain:

Name: Format
Value: Fixed, Flowed

Name: DelSp
Value: Yes, No

(Neither the parameter names nor values are case sensitive.)

If Format is not specified, or if the value is not recognized, a value of Fixed is assumed. The semantics of the Fixed value are the usual associated with Text/Plain [MIME-INT].

A Format value of Flowed indicates that the definition of flowed text (as specified in this memo) was used on generation, and MAY be used on reception.

Note that because Format is a parameter of the Text/Plain content-type, any content-transfer-encoding used is irrelevant to the processing of flowed text.

If DelSp is not specified, or if its value is not recognized, a value of No is assumed. The use of DelSp without a Format value of Flowed is undefined. When creating messages, DelSp SHOULD NOT be specified in Text content types other than Text/Plain with Format = Flowed. When receiving messages, DelSp SHOULD be ignored if used in a Text content type other than Text/Plain with Format = Flowed.

This section discusses flowed text; section 6 provides a formal definition.

Section 5 discusses interoperability.

Note that this memo describes an on-the-wire format. It does not address formats for local file storage.

4.1. Interpreting Format=Flowed

If the first character of a line is a quote mark (">"), the line is considered to be quoted (see Section 4.5). Logically, all quote marks are counted and deleted, resulting in a line with a non-zero quote depth, and content. (The agent is of course free to display the content with quote marks or excerpt bars or anything else.) Logically, this test for quoted lines is done before any other tests (that is, before checking for space-stuffed and flowed).

If the first character of a line is a space, the line has been space-stuffed (see Section 4.4). Logically, this leading space is deleted before examining the line further (that is, before checking for flowed).

If the line ends in a space, the line is flowed. Otherwise it is fixed. The exception to this rule is a signature separator line, described in Section 4.3. Such lines end in a space but are neither flowed nor fixed.

If the line is flowed and DelSp is "yes", the trailing space immediately prior to the line's CRLF is logically deleted. If the DelSp parameter is "no" (or not specified, or set to an unrecognized value), the trailing space is not deleted.

Any remaining trailing spaces are part of the line's content, but the CRLF of a soft line break is not.

A series of one or more flowed lines followed by one fixed line is considered a paragraph, and MAY be flowed (wrapped and unwrapped) as appropriate on display and in the construction of new messages (see Section 4.5).

An interpreting agent SHOULD allow for three exceptions to the rule that paragraphs end with a fixed line. These exceptions are improperly constructed messages: a flowed line SHOULD be considered to end the paragraph if it is followed by a line of a different quote depth (see 4.5) or by a signature separator (see 4.3); the end of the body also ends the paragraph.

A line consisting of one or more spaces (after deleting a space acting as stuffing) is considered a flowed line.

An empty line (just a CRLF) is a fixed line.

Note that, for Unicode text, [Annex-14] provides guidance for choosing at which characters to wrap a line.

4.2. Generating Format=Flowed

When generating Format=Flowed text, lines SHOULD be 78 characters or shorter, including any trailing white space and also including any space added as part of stuffing (see Section 4.4). As suggested values, any paragraph longer than 78 characters in total length could be wrapped using lines of 72 or fewer characters. While the specific line length used is a matter of aesthetics and preference, longer lines are more likely to require rewrapping and to encounter difficulties with older mailers. (It has been suggested that 66 character lines are the most readable.)

The restriction to 78 or fewer characters between CRLFs on the wire is to conform to [MSG-FMT].

(In addition to conformance to [MSG-FMT], there is a historical need that all lines, even when displayed by a non-flowed-aware program, will fit in a standard 79- or 80-column screen without having to be wrapped. The limit is 78, not 79 or 80, because while 79 or 80 fit on a line, the last column is often reserved for a line-wrap indicator.)

When creating flowed text, the generating agent wraps, that is, inserts 'soft' line breaks as needed. Soft line breaks are added at natural wrapping points, such as between words. A soft line break is a SP CRLF sequence.

There are two techniques for inserting soft line breaks. The older technique, established by RFC 2646, creates a soft line break by inserting a CRLF after the occurrence of a space. With this technique, soft line breaks are only possible where spaces already occur. When this technique is used, the DelSp parameter SHOULD be used; if used it MUST be set to "no".

The newer technique, suitable for use even with languages/coded character sets in which the ASCII space character is rare or not used, creates a soft line break by inserting a SP CRLF sequence. When this technique is used, the DelSp parameter MUST be used and MUST be set to "yes". Note that because of space-stuffing (see Section 4.4), when this technique is used and a soft line break is inserted at a point where a SP already exists (such as between words), if the SP CRLF sequence is added immediately before the SP, the pre-existing SP becomes leading and thus requires stuffing. It is RECOMMENDED that agents avoid this by inserting the SP CRLF sequence following the existing SP.

Generating agents MAY use either method within each Text/Plain body part.

Regardless of which technique is used, a generating agent SHOULD NOT insert a space in an unnatural location, such as into a word (a sequence of printable characters, not containing spaces, in a language/coded character set in which spaces are common). If faced with such a word which exceeds 78 characters (but less than 998 characters, the [SMTP] limit on line length), the agent SHOULD send the word as is and exceed the 78-character limit on line length.

A generating agent SHOULD:

- o Ensure all lines (fixed and flowed) are 78 characters or fewer in length, counting any trailing space as well as a space added as stuffing, but not counting the CRLF, unless a word by itself exceeds 78 characters.
- o Trim spaces before user-inserted hard line breaks.

A generating agent MUST:

- o Space-stuff lines which start with a space, "From ", or ">".

In order to create messages which do not require space-stuffing, and are thus more aesthetically pleasing when viewed as Format=Fixed, a generating agent MAY avoid wrapping immediately before ">", "From ", or space.

(See Sections 4.4 and 4.5 for more information on space-stuffing and quoting, respectively.)

A Format=Flowed message consists of zero or more paragraphs, each containing one or more flowed lines followed by one fixed line. The usual case is a series of flowed text lines with blank (empty) fixed lines between them.

Any number of fixed lines can appear between paragraphs.

When placing soft line breaks in a paragraph, generating agents MUST NOT place them in a way that causes any line of the paragraph to be a signature separator line, because paragraphs cannot contain signature separator lines (see Sections 4.3 and 6).

[Quoted-Printable] encoding SHOULD NOT be used with Format=Flowed unless absolutely necessary (for example, non-US-ASCII (8-bit) characters over a strictly 7-bit transport such as unextended [SMTP]). In particular, a message SHOULD NOT be encoded in Quoted-Printable for the sole purpose of protecting the trailing space on flowed lines unless the body part is cryptographically signed or encrypted (see Section 4.6).

The intent of Format=Flowed is to allow user agents to generate flowed text which is non-obnoxious when viewed as pure, raw Text/Plain (without any decoding); use of Quoted-Printable hinders this and may cause Format=Flowed to be rejected by end users.

4.3. Usenet Signature Convention

There is a long-standing convention in Usenet news which also commonly appears in Internet mail of using "-- " as the separator line between the body and the signature of a message. When generating a Format=Flowed message containing a Usenet-style separator before the signature, the separator line is sent as-is. This is a special case; an (optionally quoted or quoted and stuffed) line consisting of DASH DASH SP is neither fixed nor flowed.

Generating agents MUST NOT end a paragraph with such a signature line.

A receiving agent needs to test for a signature line both before the test for a quoted line (see Section 4.5) and also after logically counting and deleting quote marks and stuffing (see Section 4.4) from a quoted line.

4.4. Space-Stuffing

In order to allow for unquoted lines which start with ">", and to protect against systems which "From-munge" in-transit messages (modifying any line which starts with "From " to ">From "), Format=Flowed provides for space-stuffing.

Space-stuffing adds a single space to the start of any line which needs protection when the message is generated. On reception, if the first character of a line is a space, it is logically deleted. This occurs after the test for a quoted line (which logically counts and deletes any quote marks), and before the test for a flowed line.

On generation, any unquoted lines which start with ">", and any lines which start with a space or "From " MUST be space-stuffed. Other lines MAY be space-stuffed as desired.

(Note that space-stuffing is conceptually similar to dot-stuffing as specified in [SMTP].)

4.5. Quoting

In Format=Flowed, the canonical quote indicator (or quote mark) is one or more close angle bracket (">") characters. Lines which start with the quote indicator are considered quoted. The number of ">"

characters at the start of the line specifies the quote depth. Flowed lines which are also quoted may require special handling on display and when copied to new messages.

When creating quoted flowed lines, each such line starts with the quote indicator.

Note that because of space-stuffing, the lines

```
>> Exit, Stage Left
```

and

```
>>Exit, Stage Left
```

are semantically identical; both have a quote-depth of two, and a content of "Exit, Stage Left".

However, the line

```
> > Exit, Stage Left
```

is different. It has a quote-depth of one, and a content of "> Exit, Stage Left".

When generating quoted flowed lines, an agent needs to pay attention to changes in quote depth. All lines of a paragraph MUST be unquoted, or else they MUST all be quoted and have the same quote depth. Therefore, whenever there is a change in quote depth, or a change from quoted to unquoted, or change from unquoted to quoted, the line immediately preceding the change MUST NOT be a flowed line.

If a receiving agent wishes to reformat flowed quoted lines (joining and/or wrapping them) on display or when generating new messages, the lines SHOULD be de-quoted, reformatted, and then re-quoted. To de-quote, the number of close angle brackets in the quote indicator at the start of each line is counted. To re-quote after reformatting, a quote indicator containing the same number of close angle brackets originally present are prefixed to each line.

On reception, if a change in quote depth occurs on a flowed line, this is an improperly formatted message. The receiver SHOULD handle this error by using the 'quote-depth-wins' rule, which is to consider the paragraph to end with the flowed line immediately preceding the change in quote depth.

In other words, whenever two adjacent lines have different quote depths, senders MUST ensure that the earlier line is not flowed (does not end in a space), and receivers finding a flowed line there SHOULD treat it as the last line of a paragraph.

For example, consider the following sequence of lines (using '*' to indicate a soft line break, i.e., SP CRLF, and '#' to indicate a hard line break, i.e., CRLF):

```
> Thou villainous ill-breeding spongy dizzy-eyed*
> reeky elf-skinned pigeon-egg!*      <--- problem ---<
>> Thou artless swag-bellied milk-livered*
>> dismal-dreaming idle-headed scut!#
>>> Thou errant folly-fallen spleeny reeling-ripe*
>>> unmuzzled ratsbane!#
>>>> Henceforth, the coding style is to be strictly*
>>>> enforced, including the use of only upper case.#
>>>>> I've noticed a lack of adherence to the coding*
>>>>> styles, of late.#
>>>>> Any complaints?#
```

The second line ends in a soft line break, even though it is the last line of the one-deep quote block. The question then arises as to how this line is to be interpreted, considering that the next line is the first line of the two-deep quote block.

The example text above, when processed according to quote-depth wins, results in the first two lines being considered as one quoted, flowed section, with a quote depth of 1; the third and fourth lines become a quoted, flowed section, with a quote depth of 2.

A generating agent **MUST NOT** create this situation; a receiving agent **SHOULD** handle it by giving preference to the quote depth.

4.6. Digital Signatures and Encryption

If a message is digitally signed or encrypted it is important that cryptographic processing use the same text for signature verification and/or decryption as was used for signature generation and/or encryption. Since the use of format=flowed allows text to be altered (by adding or removing line breaks and trailing spaces) between composition and transmission, and between reception and display, interoperability problems or security vulnerabilities may arise if originator and recipient do not both use the on-the-wire format for cryptographic processing.

The implications of the interaction between format=flowed and any specific cryptographic process depend on the details of the cryptographic processing and should be understood before using format=flowed in conjunction with signed and/or encrypted messages.

Note that [OpenPGP] specifies (in Section 7.1) that "any trailing whitespace (spaces, and tabs, 0x09) at the end of any line is ignored when the cleartext signature is calculated."

Thus it would be possible to add, in transit, a format=flowed header to a regular, format=fixed vanilla PGP (not [OpenPGP-MIME]) signed message and add arbitrary trailing space characters without this addition being detected. This would change the rendering of the article by a client which supported format=flowed.

Therefore, the use of [OpenPGP] with format=flowed messages is strongly discouraged. [OpenPGP-MIME] is recommended instead.

4.7. Examples

The following example contains three paragraphs:

'Take some more tea,' the March Hare said to Alice, very earnestly.

'I've had nothing yet,' Alice replied in an offended tone, 'so I can't take more.'

'You mean you can't take LESS,' said the Hatter: 'it's very easy to take MORE than nothing.'

This could be encoded as follows (using '*' to indicate a soft line break, that is, SP CRLF sequence, and '#' to indicate a hard line break, that is, CRLF):

```
'Take some more tea,' the March Hare said to Alice, very*
earnestly.#
#
'I've had nothing yet,' Alice replied in an offended tone, 'so*
I can't take more.'#
#
'You mean you can't take LESS,' said the Hatter: 'it's very*
easy to take MORE than nothing.'#
```

To show an example of quoting, here we have the same exchange, presented as a series of direct quotes:

```
>>>Take some more tea.#
>>I've had nothing yet, so I can't take more.#
>You mean you can't take LESS, it's very easy to take*
>MORE than nothing.#
```

5. Interoperability

Because flowed lines are all-but-indistinguishable from fixed lines, software which does not recognize Format=Flowed treats flowed lines as normal Text/Plain (which is what they are). Thus, Format=Flowed

interoperates with older clients, although flowed lines will have trailing white space inserted.

If a space-stuffed message is received by an agent which handles Format=Flowed, the space-stuffing is reversed and thus the message appears unchanged. An agent which is not aware of Format=Flowed will of course not undo any space-stuffing; thus Format=Flowed messages may appear with a leading space on some lines (those which start with a space, ">" which is not a quote indicator, or "From "). Since lines which require space-stuffing rarely occur, and the aesthetic consequences of unreversed space-stuffing are minimal, this is not expected to be a significant problem.

If some lines begin with one or more spaces, the generating agent MAY space-stuff all lines, to maintain the relative indentation of the lines when viewed by clients which are not aware of Format=Flowed.

Messages generated with DelSp=yes and received by clients which are aware of Format=Flowed but are not aware of the DelSp parameter will have an extra space remaining after removal of soft line breaks. Thus, when generating text in languages/coded character sets in which spaces are common, the generating agent MAY always use the DelSp=no method.

Hand-aligned text, such as ASCII tables or art, source code, etc., SHOULD be sent as fixed, not flowed lines.

6. ABNF

The constructs used in Text/Plain; Format=Flowed body parts are described using Augmented Backus-Naur Form [ABNF], including the core rules defined in Appendix A.

Note that the SP (space) and ">" characters are encoded according to the charset parameter.

```

flowed-body      = *( paragraph / fixed-line / sig-sep )
paragraph        = 1*flowed-line fixed-line
                  ; all lines in paragraph MUST be unquoted or
                  ; have same quote depth
flowed-line      = ( flowed-line-qt / flowed-line-unqt ) flow CRLF
flowed-line-qt   = quote ( ( stuffing stuffed-flowed ) /
                          unstuffed-flowed )
flowed-line-unqt = ( stuffing stuffed-flowed ) / unstuffed-flowed
stuffed-flowed  = *text-char
unstuffed-flowed = non-sp-quote *text-char
fixed-line      = fixed-line-qt / fixed-line-unqt
fixed-line-qt   = quote ( ( stuffing stuffed-fixed ) /

```

```

                                unstuffed-fixed ) CRLF
fixed-line-unqt  = ( stuffed-fixed / unstuffed-fixed ) CRLF
stuffed-fixed   = *text-char non-sp
unstuffed-fixed = non-sp-quote [ *text-char non-sp ]
sig-sep        = [ quote [stuffing] ] "--" SP CRLF
quote-mark     = ">"
quote          = 1*quote-mark
stuffing       = SP ; space-stuffed, added on generation if
                  ; needed, deleted on reception
flow          = SP ; space before CRLF indicates flowed line,
                  ; if DelSp=yes, space was added on generation
                  ; and is deleted on reception
non-sp-quote   = < any character except NUL, CR, LF, SP, quote-mark >
non-sp         = non-sp-quote / quote-mark
text-char      = non-sp / SP

```

That is, a Format=Flowed message body consists of any number of paragraphs and/or fixed lines and/or signature separator lines; paragraphs need at least one flowed line and are terminated by a fixed line; the fixed line terminating the paragraph is part of the paragraph. (There are some exceptions to this described in the text.)

Without at least one flowed line, there is a series of fixed lines, each independent. There is no paragraph.

With at least one flowed line, there is a paragraph, and the received lines can be reformed and flowed to fit the display window size. This can only be done if the lines are part of a logical grouping, the paragraph.

Note that the definitions of flowed-line and sig-sep are potentially ambiguous: a signature separator line matches both, but is treated as a signature separator line and not a flowed line.

7. Failure Modes

7.1. Trailing White Space Corruption

There are systems in existence which alter trailing whitespace on messages which pass through them. Such systems may strip, or in rarer cases, add trailing whitespace, in violation of RFC 2821 [SMTP] Section 4.5.2.

Stripping trailing whitespace has the effect of converting flowed lines to fixed lines, which results in a message no worse than if Format=Flowed had not been used.

Adding trailing whitespace to a Format=Flowed message may result in a malformed display or reply.

Since most systems which add trailing white space do so to create a line which fills an internal record format, the result is almost always a line which contains an even number of characters (counting the added trailing white space).

One possible avoidance, therefore, would be to define Format=Flowed lines to use either one or two trailing space characters to indicate a flowed line, such that the total line length is odd. However, considering the scarcity of such systems today, it is not worth the added complexity.

8. Security Considerations

Any security considerations which apply to Text/Plain also apply to Text/Plain with Format=Flowed.

Section 4.6 discusses the interaction between Format=Flowed and digital signatures or encryption.

9. IANA Considerations

IANA has added a reference to this specification in the Text/Plain Media Type registration.

10. Internationalization Considerations

The line wrap and quoting specifications of Format=Flowed may not be suitable for certain charsets, such as for Arabic and Hebrew characters that read from right to left. Care needs to be taken in applying format=flowed in these cases, as format=fixed combined with [quoted-printable] encoding may be more suitable.

The DelSp parameter was added specifically to permit Format=Flowed to be used with languages/coded character sets in which the ASCII space character is rarely used, or not used at all.

11. Acknowledgments

The DelSp parameter was developed during a series of discussions among a number of people, including Harald Alvestrand, Grant Baillie, Ian Bell, Steve Dorner, Patrik Faltstrom, Eric Fischer, Ned Freed, Alexey Melnikov, John Myers, and Pete Resnick.

Corrections and clarifications to RFC 2646 and early versions of this document were pointed out by several people, including Adam Costello, Jutta Degener, Tony Hansen, Simon Josefsson, Dan Kohn, Ragho Mahalingam, Keith Moore, Greg Troxel, and Dan Wing.

I'm told that NeXT's mail application used a very similar mechanism (without support for non-Western languages) in 1992.

12. Normative References

- [ABNF] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.
- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [MIME-IMT] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.
- [Quoted-Printable] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

13. Informative References

- [Annex-14] Unicode Standard Annex #14, "Line Breaking Properties"
<URL:<http://www.unicode.org/unicode/reports/tr14/>>
- [MSG-FMT] Resnick, P., Ed., "Internet Message Format", RFC 2822, April 2001.
- [OpenPGP] Callas, J., Donnerhacke, L., Finney, H. and R. Thayer, "OpenPGP Message Format", RFC 2440, November 1998.
- [OpenPGP-MIME] Elkins, M., "MIME Security with Pretty Good Privacy (PGP)", RFC 2015, October 1996.

Elkins, M., Del Torto, D., Levien, R. and J. Roessler, "MIME Security with OpenPGP", RFC 3156, August 2001.

- [Rich] Resnick, P. and A. Walker, "The text/enriched MIME Content-type", RFC 1896, February 1996.
- [SMTP] Klensin, J., Ed., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

Appendix A: Changes from RFC 2646

Substantive:

- o Added DelSp parameter to handle languages and coded character sets in which space is less common or not used.
- o Updated text on generating and interpreting to accommodate the DelSp parameter.
- o Changed the limits of 79 or 80 to be 78 in conformance with RFC 2822.
- o Added text on generating to clarify that the 78-character limit includes trailing white space and stuffing.
- o Changed sig-sep in ABNF to allow stuffing.
- o Changed fixed-line to allow empty lines in ABNF.
- o Added explanatory text following ABNF.
- o Moved text from Abstract to new Introduction; rewrote Abstract.
- o Moved interoperability text to new section, and updated.
- o Clarified Security Considerations.
- o Text on digital signatures now discusses that OpenPGP ignores trailing white space.
- o Mention Unicode Annex 14.
- o Added mention of quoting to Abstract and Introduction.
- o Deleted line analysis table.
- o Added recommendations for OpenPGP and OpenPGP-MIME.
- o Rewrote ABNF rules to remove most ambiguity and note remaining case.
- o Added note that c-t-e is irrelevant to flowed text processing.
- o Added text indicating that end of data terminates a paragraph.
- o Moved sig-sep out of fixed-line ABNF.
- o Changed some SHOULDs to MUSTs (space-stuffing, quoted paragraphs).
- o Added note to ABNF that space and ">" are encoded according to charset.
- o Mentioned exceptions in section on interpreting.
- o Clarified and made consistent treatment of signature separator lines.

Editorial:

- o Added mention of NeXT's mail application to Acknowledgments.
- o Updated Acknowledgments.
- o Updated [SMTP] reference to 2821.
- o Added Notices.
- o Split References into Normative and Informative.
- o Improved text wording in some areas.
- o Standardize on "quote depth", not "quoting depth".
- o Moved section on interpreting before section on generating.
- o Reworded non-normative "should"s.
- o Noted meaning of "paragraph".

The DelSp parameter was added specifically to permit Format=Flowed to be used with languages/coded character sets in which the ASCII space character is rarely used, or not used at all. The DelSp mechanism was selected despite having been initially rejected as too much of a kludge, because among the many different techniques proposed, it allows for maximum interoperability among clients which support neither this specification nor RFC 2646, those which do support RFC 2646 but not this specification, and those that do support this specification; this set is multiplied by those that handle languages/coded character sets in which spaces are common, and in which they are uncommon or not used.

Author's Address

Randall Gellens
QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121
USA

Phone: +1 858 651 5115
EMail: randy@qualcomm.com

Full Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78 and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

