Network Working Group B. Lilly
Internet-Draft May 2005
Updates: 2223 (if approved)
Intended status: Informational
Expires: November 6, 2005

Writing Internet-Drafts and Requests For Comments using troff and nroff
draft-lilly-using-troff-04

Status of this Memo

Copyright Notice

Abstract

Internet-Drafts and RFCs have traditionally been produced using a variety of tools, including nroff. This memo describes production of Internet-Drafts and RFCs using a set of troff/nroff macros suitable for that purpose. Public comments regarding this memo may be sent to the rfc-interest mailing list.

Table of Contents

1. Introduction

The Instructions to RFC Authors RFC [N1.RFC2223] describes the editorial format of RFCs and outlines editorial production of Internet-Drafts and RFCs using basic troff directives [I1.CSTR54] with the ms macro package [I2.ms] and a postprocessor which is necessitated by use of the ms macro package. However, use of basic directives and the ms macro package is not ideal.

1.1. Basic Directives Are at Too Low a Level

Use of basic troff directives is not ideal for authors or editors who are not familiar with troff arcana. On the one hand, the example in [N1.RFC2223] uses a larger variety of troff directives than some casual users might be familiar with. On the other hand, instead of using directives such as .tl to produce first page headings, it suggests that authors or editors should count characters to manually justify the first page heading.

1.2. Authors Specify Content Rather Than Issue Directives

Another problem with low-level directives is that authors are not primarily concerned with formatting details. An author needs to indicate things like "this is the document title" or "a new section starts here". Authors rarely think in low-level formatting terms such as are provided by basic troff directives, e.g. "center the next 3 lines".

1.3. ms Macros Are Not Designed For RFCs

Macros provide a mechanism for authors to specify type of content rather than formatting details. However, use of the ms macro package is not ideal for production of RFCs and Internet-Drafts as that macro package was not designed for the specific document format described in [N1.RFC2223]. Use of some high-level ms macros results in text which violates some provisions of the [N1.RFC2223] format, while other ms macros are unusable due to differences between RFC document format and the types of documents that the ms package was designed to produce. There are also implementation differences in different versions of "ms" macro packages. Finally, it is the ms macro package which has stymied the RFC Editor's attempts to get a form feed character at the end of each page [N1.RFC2223], [I3.RFC1543].

1.4. RFC-specific Macros

This memo describes use of a macro package which is designed specifically for production of RFCs and Internet-Drafts, which provides a high-level abstraction to insulate authors and editors from many troff details and frees authors and editors from tedium such as counting characters.

2. Requirement Levels

The key words "SHOULD", "SHOULD NOT" and "RECOMMENDED" in this document are to be interpreted as described in [N2.BCP14].

3. Processing

Production of a document using troff and a macro package proceeds in several stages:

1. A mixture of free-form text, processing directives, and material describing non-textual content (tables, diagrams, etc.) is generated by the author.

2. Preprocessors [I4.CSTR49], [I5.CSTR114], [I6.CSTR116], [I7.CSTR122], [I8.eqn], [I9.CSTR142] may be used to transform descriptions of tables, diagrams, etc.

3. A formatter transforms instructions and other input into a formatted document according to rules corresponding to the desired document format.

4. Some postprocessing may take place to repaginate output and to make final adjustments to the output format.

Tools, e.g. [I10.make], are available to automate the preprocessor, formatter, and postprocessing portions of document processing, freeing the author to concentrate on the document content.

4. Input Format

Input consists of text to be formatted, directives that specify formatting parameters and actions, and special sequences that provide access to saved strings and numbers. Directives consist of a line beginning with a period (also known as "dot"). The dot may be followed by space characters, and then a named directive. The name consists of one or more printing characters. Many directives also take arguments, which are space-separated strings of characters. Unless directed otherwise, input is collected and formed into paragraphs without regard to line breaks in the input. Input text consisting of the lines:

```
.IP
All bad precedents
begin with
justifiable measures. \" Julius Caesar
.\" This is a comment.
A strong conviction
that something must be done
is the parent of
many bad measures. \" Daniel Webster
Any excuse
will serve
a tyrant. \" Aesop
```

is formatted into an indented paragraph which looks like:

All bad precedents begin with justifiable measures. A strong conviction that something must be done is the parent of many bad measures. Any excuse will serve a tyrant.

The character sequence \" introduces a comment which continues to the end of the line. If an entire line is a comment, starting the line with a dot followed by the comment ensures that extra space does not appear in the output. Authors should be careful about use of the backslash, which is an escape character that introduces special sequences. If it is necessary to have a backslash appear in the formatted output, the sequence \e, i.e. a backslash followed by a lower-case letter e, will produce it. Unescaped backslashes followed by double quotes have been known to cause important text to vanish from published RFCs.

5. Specifics

5.1. Loading the RFC Macros

The first directive might be the line:

```
.so tmac.rfc
```

That directs the processor to read the contents of a file named "tmac.rfc", which contains the macros that describe the RFC format, holds boilerplate text, and provides convenience functions.

Alternate methods of using the macros might be available:

1. the file tmac.rfc can be specified on the command line before the file with source content

2. the rfc macros may have been locally installed in a manner that permits use of a command-line -mrfc argument.

5.2. Specifying the Document Particulars

Each RFC or Internet-Draft has a title, one or more authors, etc. These are specified with directives.

5.2.1. Titles, Long and Short

The TL directive, as in

```
.TL "Writing Internet\-Drafts and Requests For Comments using troff and\
\& nroff
```

specifies the document title. The double-quote causes the remainder of the (logical) line to be taken as a single argument to the TL macro; otherwise each space character would be taken as an argument separator. Note also that a backslash character appears before the hyphen; that causes the hyphen to be interpreted as a normal text hyphen rather than as an indication of a possible hyphenation (line-breaking) point. The backslash at the end of the first line escapes the end-of-line, allowing the line to effectively continue with the contents of the second line. Escape sequences are discussed in detail in [I1.CSTR54]. There is a short title (ST) macro which is used to specify a short version of the document title for use in page headings. For this document, that was specified as:

```
.ST "Writing I\-Ds and RFCs using troff
```

5.2.2. RFC or Draft Version Number

RFCs have an RFC number assigned by the RFC Editor; Internet-Drafts have a version number. The number is specified with the NU macro or by setting a register with a command-line argument as described in Appendix A: a negative number is interpreted as an incremental version number for a draft; -1 for draft version -00, -2 for draft version -01, etc. A positive number is interpreted as an RFC number and SHOULD only be assigned by the RFC Editor. A zero value is treated like an RFC, but prints XXXX where an RFC number would appear.

5.2.3. Document Category (RFCs)

RFCs have a category; one of Experimental, Informational, Standards Track, or Best Current Practice [I11.BCP9]. The category is specified as the argument to the CA macro; that macro directive need only be supplied for RFCs (if supplied in drafts it shows as Intended status in the first page heading) [N3.ID].

5.2.4. Updated and Obsoleted RFCs

Some RFCs update or obsolete earlier RFCs. If an RFC is updated by the document, the number of the updated RFC should be specified as the argument to an UP macro. Similarly, an obsoleted RFC is indicated by an argument to an OB macro. Multiple UP and OB macros may be used to indicate that multiple RFCs are updated or obsoleted.

5.2.5. Internet-Draft File Name

An Internet-Draft has a file name used for retrieval; the base file name (without the trailing hyphen and version number suffix) is specified as the argument to the FN macro; the FN macro is not necessary (is ignored) for RFCs. The file name may contain only lower-case letters, digits, and hyphens [N3.ID].

5.2.6. Authors or Editors

An Internet-Draft or RFC may have up to five authors specified. Information for each author may be specified via the AU macro. The macro takes 9 arguments:

Argument	Description
1	First initial and surname for first page heading
2	First name and/or initials for Author's Address section
3	Surname for page footer and Author's Address section
4	Gender; one of m (male), f (female), or o (other or indeterminate)
5	Internet mailbox
6	Telephone Number
7	Fax Number
8	URL
9	Affiliation

The author's gender is used to customize boilerplate verbiage to accommodate enforcers of political correctness, at least until individual male or female authors are prohibited from submitting Internet-Drafts with sensible boilerplate. Arguments containing space characters must be enclosed in double quotes. Arguments that are not used must have an empty double quoted string or other zero-width argument as a placeholder. Optional text following the AU macro line is collected verbatim (without flowing into a paragraph) as the postal address for the author. Similar information can be provided for additional authors by additional AU macro calls.

Some documents, notably those produced by IETF Working Groups, are prepared by a Document Editor [I12.BCP25]. To distinguish an Editor from an Author, use the ED macro instead of the AU macro. There may be multiple Editors, but one cannot have a mix of Authors and Editors (if one is specified as an Editor, all are treated as Editors).

5.2.7. Date

The date used for the document heading is normally the current date when formatted. It can be explicitly specified via the DT macro which takes three numeric arguments (4-digit year, month, and day respectively), or the values may be specified via command-line arguments as described in Appendix A.

5.2.8. Section Headings and the Table of Contents

An RFC or Internet-Draft may contain numbered sections. Sections may have subsections, and the section level represents the number of dot-separated levels. This section of this document has level 3. A long document benefits from a Table of Contents. As sections are specified, the rfc macro package collects information for a Table of Contents. The TC macro takes two arguments. The first specifies the number of pages to reserve for the Table of Contents (zero to omit the table), and the second argument specifies the maximum section level to include in the Table of Contents entries.

These parameters may alternatively be specified via command-line arguments. See Appendix A for details.

5.2.9. Independent vs. IETF Submissions

Internet-Drafts and RFCs may be products of IETF Working Groups or other IETF entities, or they may be independent submissions. The macro IS is used to identify independent submissions, as there is a slight difference in the boilerplate text which accompanies them. The IS macro takes no arguments.

Unfortunately, the IETF Secretariat rejects independent Internet-Draft submissions that have the independent submission boilerplate required by the RFC Editor [I13.Instruct] (bottom of page 33), probably because it is not recognized by Henrik Levkowetz' "idnits" program [I14.idnits]. Therefore, until the IETF Secretariat and the RFC Editor resolve their differences the IS macro SHOULD NOT be used except by the RFC Editor.

5.2.10. Optional Notices

Certain optional notices may be placed at the front of Internet-Drafts. The ON macro is used to specify one of the following notices via a numeric argument:

Type	Text
1	This document may not be modified, and derivative works of it may not be created. This document may only be posted in an Internet-Draft.
2	This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

If a second argument is given to the ON macro, it is interpreted as the number of a section that may be extracted. The text is modified to reflect that. For example:

```
.ON 1 1.2.3
```

produces the text:

This document may not be modified, and derivative works of it may not be created other than to extract section 1.2.3 as-is for separate use. This document may only be posted in an Internet-Draft.

5.3. Document Content

Having specified preliminary information as described above, the document content may be specified.

5.3.1. Abstract

The first item that an author might want to provide is the Abstract for the document. It is initiated with the AB macro, which takes no arguments. It collects text until the next macro directive into a paragraph for the document abstract. An abstract is required for an Internet-Draft; it is optional for an RFC.

5.3.2. Numbered and Unnumbered Sections

Additional document content is usually organized into sections. There are two types of sections: numbered and unnumbered. For the purpose of collecting headings for the Table of Contents, unnumbered sections have a section "level" even though no number appears. A section is initiated with an SH macro for unnumbered sections, or an NH macro for numbered sections. Each macro takes two arguments: the section level and the section heading text. Numbering for numbered sections is computed automatically.

5.3.3. Paragraphs

Within sections, text is organized into paragraphs. RFCs and Internet-Drafts usually use indented block paragraphs like this one; these are introduced with the IP macro. That macro can take two optional arguments: initial unindented text and an indent distance. The defaults are no unindented text and an indent distance of 3 character spaces (0.3 inch). Various units may be used in troff; it is strongly RECOMMENDED that horizontal distances, such as the indent distance, be specified in multiples of 0.1 inch so that nroff- and troff-formatted versions of documents maintain the same line and page breaks.

Unindented paragraphs are possible via the LP macro.

5.3.4. Special Indentation

Indentation may be controlled with RS and RE macros. The RS macro will increase the indent level by the paragraph indentation increment of 3 character spaces, while the RE macro will reduce indentation by that amount.

5.3.5. Reserving Vertical Space

Sometimes it is necessary to ensure that there is sufficient vertical space before the bottom of the current page for some complex output that cannot easily be handled with a keep or display (see below). The NS macro takes one argument specifying the number of text lines needed. If fewer lines remain on the current page, output will continue at the top of the next page.

5.3.6. Keeps and Displays

Sometimes one has lines of text or non-textual content which one would like to appear as a group, uninterrupted by page breaks. Such a group is called a "keep" and is specified by preceding the group with the KS macro and following it with a KE macro. Keeps are typically used to fine-tune formatting; they are rarely required.

Keeps will not work with multiple paragraphs (or list items) between KS and KE directives. To force a page break if there is less than a certain amount of vertical space remaining on the current page, use the NS macro directive to reserve space.

More complex groupings can be obtained by "displays". Displays are started like keeps, using the DS macro. Display content is not filled into paragraphs; output appears like the input lines except for indentation. The DS macro takes an argument consisting of a single letter which identifies the type of display:

Type	Description
C	Each line of text is centered
B	The input lines are read as a block, and the block is centered
L	The display is left-aligned
I	The display is indented by the amount indicated by an optional second argument (default is 3 character spaces)

Displays are typically used for block quoted material and for preventing page breaks within non-textual material such as tables and diagrams. For example, the input:

```
.DS C
War is Peace
Freedom is Slavery
Ignorance is Strength
.\" Eric Arthur Blair, a.k.a. George Orwell
.DE
```

produces output in which each line is centered:

```
                           War is Peace
                        Freedom is Slavery
                       Ignorance is Strength
```

whereas the input:

```
.DS B
War can still settle problems,
but it can only settle them the wrong way. \" Bertrand Russell
.DE
```

produces the text lines as a centered block:

```
               War can still settle problems,
               but it can only settle them the wrong way.
```

Sometimes placement of a display, such as one containing a diagram, is not critical. A floating display allows text and non-floating displays which follow it in the source document to be output before it if it will not fit on the current output page. A floating display begins with a DF macro call, taking the same arguments as DS and ending with a DE macro call. Floating displays are output before the end of the section and subsection in which they are defined in the input, so they cannot appear in the wrong section of the formatted document.

Text following a display or keep should be organized as a paragraph, to ensure that the text appears where expected on output and so that page breaks can be adjusted to prevent widows and orphans.

5.3.7. Lists

5.3.7.1. Numbered Lists

The NL macro initializes a numbered list. It takes three optional arguments:

1. a prefix string. The default is an empty string (no prefix).

2. a one-character numbering format; one of

   1 number using Arabic numerals (the default)

   I number using Roman numbers expressed as upper-case letters

   i number using Roman numbers expressed as lower-case letters

   A letter with upper-case letters

   a letter with lower-case letters

3. a suffix string. The default is a period.

Each list item after the initialization is introduced by the LI (list item) macro. Text following the LI macro line is processed as an indented paragraph with the first line numbered.

The most recently started list is terminated with the LE (list end) macro.

Lists may be nested up to 9 levels deep. Each level of nesting is indented by the paragraph indent value (3 characters). The list above has a variable list (see description below) inside a numbered list.

5.3.7.2. Bulleted Lists

A bulleted list is initialized with the BL macro. List items also use the LI macro and the LE macro ends a bulleted list.

The input lines:

```
.BL
.LI
first bulleted item
.LI
second
bulleted item
.LE
```

produce the output:

- first bulleted item

- second bulleted item

The BL macro will accept an optional argument, which is treated as a string to use in place of the default bullet character.

5.3.7.3. Variable Lists

The VL macro initializes a variable list. Each list item is labeled with a string supplied as an argument on that item's LI macro line. The LE macro terminates a variable list.

5.3.8. References

Internet-Drafts and RFCs contain two types of references: normative references and informative references. Citations for the reference material appear near the end of the document and are indicated by a bracketed tag within the document text like this: [N1.RFC2223]. Macros are provided to automatically number and format normative and informative references in the style used in this document. The reference tags consist of the letter N (for normative references) or I (for informative references), followed by a sequential number within each series, optionally followed by a dot and a string supplied as an argument to the NR macro for normative references, IR for informative references. A second argument may be supplied to save the generated reference tag in a named string; the second argument must be 2 characters suitable as the name of a string. Internal string names used by the rfc macro package always begin with an upper-case letter; it is generally safe to use a two-character name that begins with a lower-case letter which is followed by an upper-case letter or a digit (troff internally has some strings which are named with two lower-case letters). The first normative reference in this document could have been specified by the lines:

```
.NR RFC2223 r1
Postel, J. and J. Reynolds, "Instructions to RFC Authors", RFC\ 2223,
October 1997.
.NE
```

For references to RFCs, one can simply copy and paste the text made available by the RFC Editor in the file rfc-ref.txt, as has been done in this example (the actual reference text was on a single input line, though that is not required, and the backslash preceding the space after "RFC" prevents lines being broken at that space, i.e. within the RFC document number). The reference text follows the NR or IR macro line and will be suitably formatted and placed in the Normative References or Informative References section of the generated document. The reference text is ended with an NE or IE macro.

The bracketed reference string appears in the text at the place where the NR/NE or IR/IE macros are placed, while the actual reference citation text is saved for output in unnumbered Normative References and Informative References sections. No space appears after the bracketed reference, so a list of such references may appear with comma separators by simply placing a line containing a comma between reference macro groups. If a bracketed reference appears at the end of a sentence, the period marking the end of the sentence must be provided in the input after the NE or IE macro line. However, a line beginning with a dot is normally interpreted as a directive; to prevent that, the line should begin with the two character sequence \& preceding the period. That sequence represents a zero-width non-printing space which prevents interpretation of the line as a directive without affecting the appearance of the output.

The reference tag (including the brackets) generated by the example above is saved in the troff string named r1, which can be used to refer to the same reference. That reference tag can be made to subsequently appear in the document text via a reference to the saved troff string, viz. the character sequence *(r1.

References to RFCs by number or via numeric reference to the corresponding BCP, STD, or FYI series number can be generated by using the rfcref preprocessor. That preprocessor takes input like:

```
.RR N 2223 r1
```

and generates the corresponding reference text and NR/NE or IR/IE macro directives. That involves a little less manual work for authors; there is no need to copy reference citation text and there is no end-of-reference directive.

Four preprocessor directives are recognized:

Directive	Document Series
BR	BCP
FR	FYI
SR	STD
RR	RFC

Each directive takes two or three arguments:

1. a single letter, I or N, to indicate an informative or normative reference, respectively

2. the document series number

3. (optional) a string name for saving the reference tag

Note that some STD documents map to multiple RFC numbers, and that some STD document series numbers are either reserved for future use or are no longer in use.

The rfcref preprocessor also checks whether documents have been updated or obsoleted, and issues suitable warnings.

5.3.9. MIB modules, etc.

When a MIB module or similar entity is defined and there is a limitation notice (via the ON macro), there may need to be a separate copyright notice placed in the part to be extracted per [N4.BCP78]. The IB macro may be called at the appropriate section to generate the properly-formatted boilerplate text. The mandatory first argument to the IB macro selects the type of statement from the ones permitted by [N4.BCP78], and the optional second argument permits specifying something other than "MIB module". An optional third argument may be used to specify a prefix to the statement, an optional fourth argument may be used to specify an indent other than paragraph indent, and an optional fifth argument may be used for a suffix. Unused arguments may be specified as empty strings or other zero-width items. The types are:

Type	Text
1	Copyright © The Internet Society (2005). This version of this MIB module is part of RFC XXXX; see the RFC itself for full legal notices.
2	Copyright © The Internet Society (2005). The initial version of this MIB module was published in RFC XXXX; for full legal notices see the RFC itself. Supplementary information may be available on http://www.ietf.org/copyrights/ianamib.html.

The correct year is used, and when an RFC number is assigned by the RFC Editor, the RFC number will appear in place of XXXX. As noted above, alternate text may be substituted for "MIB module".

5.3.10. Document Appendices

Appendices appear like numbered sections, except that the first-level consists of upper-case letters rather than numbers. Each appendix to appear in the document should be introduced with the AP macro, which causes such numbering to take place. The AP macro takes a single argument, the appendix heading text. Subdivided sections within an appendix can be introduced with second-, third-level, etc. numbering via NH macros as with document body sections.

5.4. End Matter

At the very end of the input file describing the document, the author should place an EM macro directive. That causes output of references, author's address (or authors' addresses), and IPR boilerplate sections. If a TC macro was provided at the beginning of the document input with a non-zero first argument, or if the document is more than 15 pages in length, a Table of Contents is also produced.

6. Acknowledgments

The author gratefully acknowledges the discussions of RFC formatting on the rfc-interest mailing list.

The format for references was inspired by Keith Moore's draft subsequently published as [I15.RFC3834].

7. Security Considerations

No security considerations are addressed by this memo.

8. Internationalization Considerations

This memo raises no new internationalization considerations.

9. IANA Considerations

This memo adds no new IANA considerations.

Appendix A. Esoterica

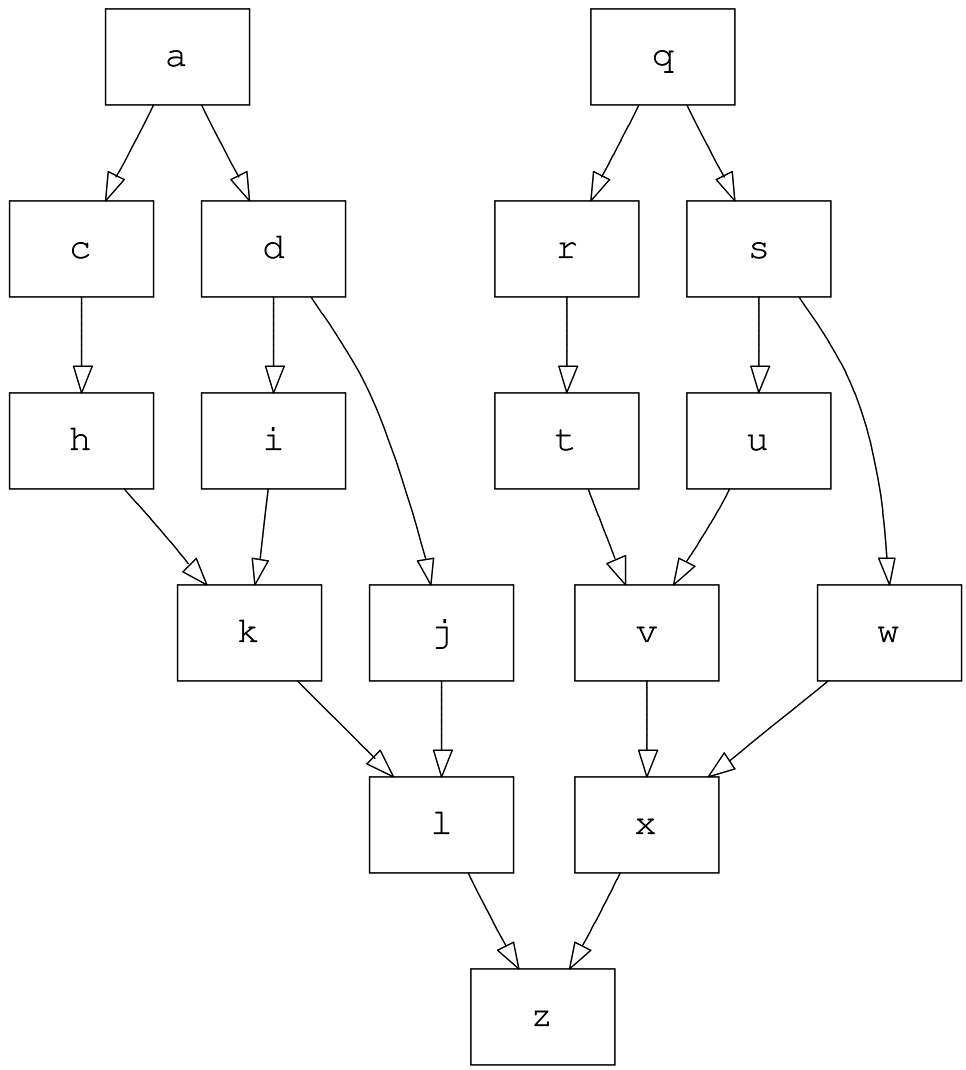
A.1. Troff vs. Nroff

RFCs and Internet-Drafts are produced as text documents. It is sometimes desirable to produce a formatted version for improved rendition of diagrams, etc. The rfc macro package has been designed so that troff formatted output resembles the text version as closely as possible; it utilizes a monospaced font for text, does not embolden section headings (unlike the ms macro package), and centers each page within 8.5 by 11 inch page boundaries. Other than improved appearance of non-textual matter, troff and nroff versions of documents produced with the rfc macro package should appear nearly identical after postprocessing, including page and line breaks. One subtlety is that centered text with an odd number of characters is offset left or right in nroff output, whereas it is truly centered in troff output. Output processed with nroff for text production includes a form feed character after each page (the "fix.pl" script specified in [N1.RFC2223] is not necessary).

Separate processing can be performed by troff and nroff by using nroff/troff conditionals, e.g.:

```
.ie t \{ \
<troff input goes here>
.\}
.el \{ \
<nroff input goes here>
.\}
```

For a more complete example, consider:



Diagrams imported from some sources which generate pic may require a reset, and in some cases setting of pic variables such as maxpsht and maxpswid.

To conform page breaks between troff- and nroff-generated documents, the height of troff- and nroff-produced output should be the same. In the example above, that is implemented with an invisible pic box in the nroff-processed version which contains the text referring to the troff-formatted versions.

Some implementations fake text formatting with "troff -a" or "troff -N" rather than a true nroff. That generally works poorly, particularly if tables, diagrams, etc. are included in the document.

Some troff implementations can produce PostScript directly; others require a separate postprocessor. PostScript produced by some troff implementations is not amenable to postprocessing to repaginate (relocate the table of contents).

A.2. Output Adjustments

Formatting may be adjusted by setting registers using command-line arguments [I1.CSTR54]. Register names and default values are:

Name	Default Value	Function
C	0 (zero)	Table of Contents length (pages)
D	Current day of month	Day of month used as publication date. Overridden by DT macro.
H	2	Highest section heading level to include in Table of Contents
I	0.3i	Paragraph indent increment. Each character is 0.1i wide.
L	7.2i	Text line length. Each character is 0.1i wide. The default value provides 72 characters per line and may not be exceeded.
M	Current month	Month number (January = 1) used as publication date. Overridden by DT macro.
N	0 (zero)	RFC or Draft version number (negative for drafts). Overrides NU macro if non-zero.
O	2	Minimum number of continued running text lines at bottom of page (anything less is considered an orphan).
P	12	Text point size (troff).
Q	58	Printed lines per page. The default value is the maximum permitted.
R	for centered text (troff)	Page offset for odd-numbered (recto) pages. Always zero for nroff (text output).
S	1P	Interparagraph vertical space. The default is equivalent to one text line. For nroff/troff consistency, the value should be an integral multiple of 1P.
T	for centered text (troff)	Top margin. Always zero for nroff (text output).
V	for centered text (troff)	Page offset for even-numbered (verso) pages. Always zero for nroff (text output).
W	2	Minimum number of continued running text lines at top of page (anything less is considered a widow).
X	8.5i	Physical page width (troff).
Y	Current year	4-digit year number used as publication date. Overridden by DT macro.
Z	11i	Physical page height (troff).

A.2.1. Formatting for Specific Paper Sizes

By default, output processed by troff (as distinct from text output produced by nroff) is formatted for economical ANSI standard A (8.5 x 11 inches) [I16.ANSI151] paper, with the printed content centered on each page. Output may instead be formatted for other paper sizes by setting troff registers listed above (as these all have one-letter names, they can be set by command-line arguments; they SHOULD NOT be set in the source file as that would adversely affect output produced from that source file, e.g. by the RFC Editor). For example, to produce output formatted for ISO [I17.ISO216] A4 paper, with a 20mm margin for hole punches for ring binder filing, registers X, Z, R, and V would be set with command-line arguments -rX21c -rZ29.7c -rR2c -rV0.712c.

A.3. Tips and Tricks

Handling non-textual material is simplified by using preprocessors. However, there are some subtleties that can cause frustration for authors. Some tips regarding preprocessors and a few tricks to ensure satisfactory formatting follow.

A.3.1. Mathematics and Chemistry

While eqn [I8.eqn] is capable of setting complex equations, many symbols cannot be adequately represented in plain text. so, while

$$x = f(y/2) + y/2$$

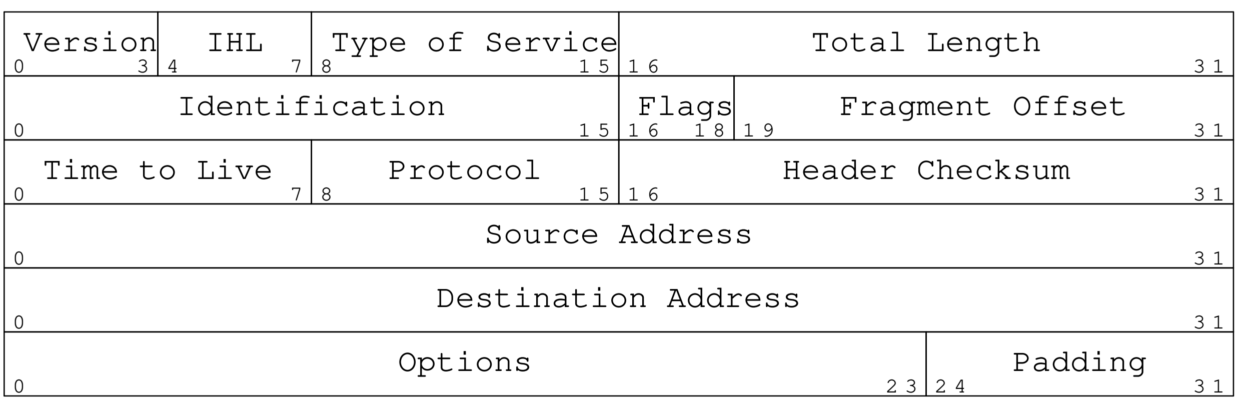
can be approximated, complex mathematics cannot be represented well in plain text. Subscripts and superscripts pose problems, and for that reason chemical formulae cannot be represented well except for trivial cases. Chem [I7.CSTR122] also tends to emit excess vertical space before a formula not containing a carbon ring and tends to screw up subsequent diagrams unless at least the first diagram after a chemical formula contains a pic "reset" directive. That can be accomplished with an otherwise empty pic diagram.

A.3.2. Data Formats

The DFORMAT program [I9.CSTR142] produces output suitable for processing by pic [I6.CSTR116] to generate data format layouts. For example:

```
.DS B
.begin dformat
style fill off
style bitwid 0.20
style recspread 0
style recht 0.33333
noname
 0-3 \0Version
 4-7 IHL
 8-15 \0Type of Service
 16-31 Total Length
noname
 0-15 Identification
 16-18 \0Flags
 19-31 Fragment Offset
noname
 0-7 Time to Live
 8-15 Protocol
 16-31 Header Checksum
noname
 0-31 Source Address
noname
 0-31 Destination Address
noname
 0-23 Options
 24-31 Padding
.end
.DE
```

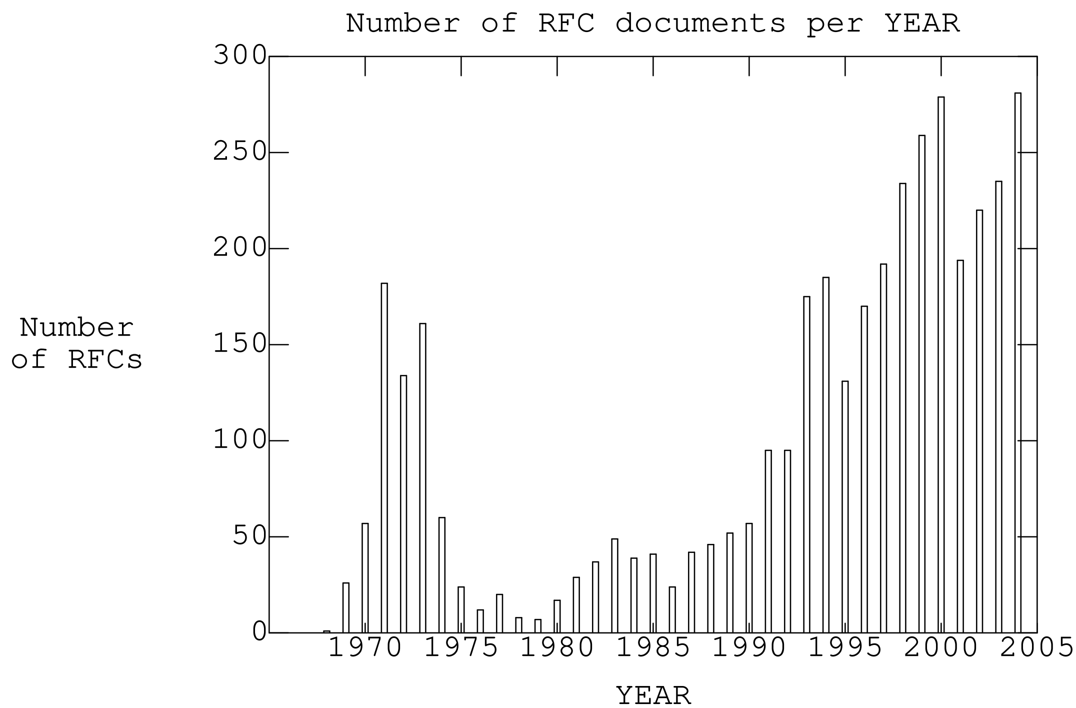
Output looks like:



Note that dimensions are specified to dformat in inches, and in the example suitable dimensions were chosen to match text format character width (multiples of 0.1 inch) and line height (1/6 inch) so that text and troff-formatted versions are reasonably consistent. The \0 sequences are invisible character-width spaces which improve the text (nroff) output.

A.3.3. Graphs and charts

Graphs and charts may be generated from data using the grap preprocessor [I5.CSTR114]. For example, the following graph:



was produced by the source lines:

```
.DS B
.G1
# Source: http://www.rfc-editor.org/num_rfc_year.html
frame ht 3 wid 4 ; coord x 1965,2005 y 0,300
label left "Number" "of RFCs" left 0.6 ; label bot "YEAR" up 0.066667
label top "Number of RFC documents per YEAR" down 0.233333
ticks left in 0.1 from 0 to 300 by 50
ticks right in 0.1 from 0 to 300 by 50 ""
ticks top in 0.1 from 1970 to 2005 by 5 ""
ticks bot in 0.05 from 1970 to 2005 by 5
copy thru { line from $1-0.15,0 to $1-0.15,$2 ; \
 line from $1-0.15,$2 to $1+0.15,$2 ; \
 line from $1+0.15,$2 to $1+0.15,0 } until "EOF"
1968 1
1969 26
1970 57
1971 182
1972 134
1973 161
1974 60
1975 24
1976 12
1977 20
1978 8
1979 7
1980 17
1981 29
1982 37
1983 49
1984 39
1985 41
1986 24
1987 42
1988 46
1989 52
1990 57
1991 95
1992 95
1993 175
1994 185
1995 131
1996 170
1997 192
1998 234
1999 259
2000 279
2001 194
2002 220
2003 235
2004 281
EOF
.G2
.DE
```

Grap produces output in pic input format; it is subsequently processed by pic to generate a graphical representation. There are multiple implementations of both grap and pic, and there is also some interaction between pic-produced output and formatters. In particular, some implementations of grap produce pic with dimensions unsuitable for generating plain text in Internet-Draft/RFC format. Authors who include graphs should be aware of these issues; they can be safely ignored if no graphs are included.

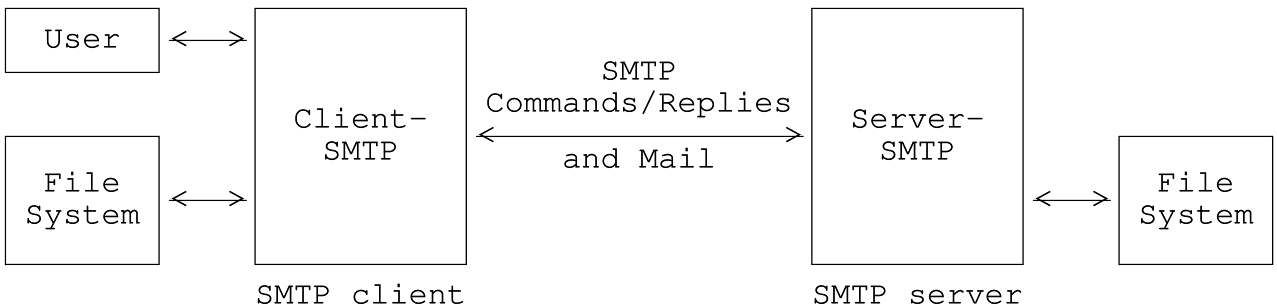
For consistency, the output produced by grap and pic should be examined to confirm that the height is specified as a multiple of 1/6 inch and the width is a multiple of 0.1 inch, not exceeding 7.2 inches. Placement of labels should also be checked between text (nroff) and PostScript (troff) versions, and adjustments to the source graph specification made as required.

A.3.4. Simple Diagrams

Pic [I6.CSTR116] can produce simple diagrams. Text output approximation is limited to rectilinear lines, so one should avoid ellipses, splines, etc. Box, line, etc. dimensions should be multiples of character width (0.1 inch) and text line height (1/6 inch) for consistency between text and troff-formatted documents. For example:

```
.DS B
.PS
boxwid = 0.8
# arrow approximation that looks acceptable in troff and nroff
define myarrow X A: [ move right 0.055;\
 "<" ljust;line right ($1 - 0.1);">" rjust;\
 move right 0.045 ]\
X
User: box ht 0.333333 "User"
FS: box ht 0.666667 "File" "System" with .n at User.s -0, 0.333333
Client: box ht 1.333333 wid 1.1 "Client\-" "SMTP" \
 with .sw at FS.se +0.5, 0
"SMTP client" rjust at Client.se -0, 0.166667
move to User.e ; myarrow(0.5)
move to FS.e ; myarrow(0.5)
move to Client.e ; SMTP: myarrow(1.8)
Server: box ht 1.333333 wid 1.1 "Server\-" "SMTP" \
 with .sw at Here.x, Client.s.y
box invis ht 0.5 "SMTP" "Commands/Replies" with .s at SMTP.c
box invis ht 0.25 "and Mail" with .n at SMTP.c
"SMTP server" ljust at Server.sw -0, 0.166667
move to Server.e.x, FS.e.y ; myarrow(0.5)
FS2: box ht 0.666667 "File" "System" \
 with .sw at Server.se.x +0.5, FS.s.y
.PE
.DE
```

The diagram looks like this:



Some versions of pic are able to approximate arrows well; others are not. The example above emulates arrows with a mix of lines and text characters to work around poor approximations of pic arrows. Some implementations of pic generate output for multi-line text labels which does not work well with some implementations of nroff. Authors who include graphs and diagrams should be aware of these issues; they can be safely ignored if no graphs or diagrams are included.

A.3.5. Tables

Tables may be formatted using the tbl program [I4.CSTR49]. Horizontal rules produced in troff-formatted documents take less vertical space than those formatted by nroff, throwing off page breaks unless special precautions are taken. Consistent vertical space may be forced by using the following sequence of lines to produce full-width horizontal rules:

```
.mk vP
.if t .sp 0.33P
_
.if t .sp |\n(vPu+1P
```

Those lines perform the following steps to ensure consistency:

1. Mark the current vertical place in register vP.

2. For troff (but not nroff), move down approximately one-third of a text line. This approximately centers the horizontal rule in the vertical space in troff.

3. Emit the horizontal rule (see [I4.CSTR49] for details); an equals sign instead of an underscore produces a double rule.

4. In troff, move to 1 text line below the previously-marked vertical position. This corresponds to the vertical position produced by nroff after a horizontal rule.

Different implementations of tbl produce slightly different output. For example, horizontal rules in text output are sometimes produced with a series of underscore characters; other implementations may use dash (hyphen) characters. While underscoring looks more attractive in heading rules than hyphens, it can interact poorly with text when used within a table.

Some implementations of tbl require extensions specific to a particular implementation of formatters.

A.3.6. ABNF

ABNF as described in [I18.RFC2234] can be difficult to format. ABNF could be formatted manually as one or more displays, or tbl could be used to align related sets of ABNF rules. Both of those options require significant manual effort.

Another option is to use a preprocessor that reads free-form ABNF and formats it for presentation. A tool is available to parse and format ABNF-like grammar descriptions (it requires comments not associated with any rule to begin without whitespace before the semicolon (unlike RFC 2234) and does not require an old-fashioned teletype "carriage return" control character before the end-of-line (also unlike RFC 2234)). ABNF is provided between macro lines AS and AE (ABNF start and ABNF end). Within that ABNF section, another macro, BC, is recognized. It causes the current (or subsequent) rule to be formatted with block comments, that is, the comments appear in a block to the right of the rule definition with running comment text formatted in a block approximately the same height as the ABNF definition.

For example, the input:

```
.AS 6 72 cgsntr
FWS = ( [ *WSP CRLF ] 1*WSP ) / obs-FWS  ; Folding white space


.BC
ctext  =   NO-WS-CTL /     ; Non white space controls.
                       %d33-39 /      ; The rest of the US-ASCII
     %d42-91 /       ;  characters not including "(",
                       %d93-126        ;  ")", or "

ccontent   =       ctext/quoted-pair/COMMENT
comment         =  "(" *([FWS] ccontent) [FWS] ")"
CFWS=*([FwS] comment) (([fws] comment) / fWs)
.AE
```

produces formatted ABNF which looks like:

```
CFWS     = *([FWS] comment) (([FWS] comment) / FWS)

comment  = "(" *([FWS] ccontent) [FWS] ")"

ccontent = ctext / quoted-pair / comment
```

```
   ctext    = NO-WS-CTL /         ; Non white space controls. The rest
              %d33-39 / %d42-91  ; of the US-ASCII characters not
              / %d93-126         ; including "(", ")", or "

   FWS      = ([*WSP CRLF] 1*WSP) / obs-FWS ; Folding white space
```

Each rule is placed in a separate display by the ABNF preprocessor so that an ABNF rule won't be split across a page break. Optional arguments to the AS macro are numbers (not troff register references, etc.) which indicate the indent and line length in characters, and one-character option flags which control processing:

Flag	Description
c	canonicalize case of character literals; type (letter after '%') is lower-case, hexadecimal digit letters are upper-case. Rulename references use the same case as the defining ABNF line.
d	omit display wrappers around output rules, instead separating rules with an empty line. This may be useful if the entire ABNF is placed within a display wrapper.
g	group concatenated elements in parentheses if in an alternation expression
n	normalize repetition specifications: 0*1foo becomes [foo], M[foo bar] becomes *M(foo bar), etc.
s	Combine adjacent quoted strings into a single quoted string where possible
t	topologically sort rules in a ruleset so that productions are defined before use
r	reverse the sense of topological sorting; use precedes definition (ineffective if the t flag is not specified)

A.3.7. Client-Server Protocol Exchanges

Some documents illustrate client server exchanges such as:

S: 220 foo.example.net

C: helo bar.example.com

S: 250 blah blah blah

C: quit

S: 221 Go in peace

Depending on the specific protocol and how it is desired to be formatted and grouped, displays, paragraphs, a list, or a table might be used to format such an exchange. The example above uses a single indented display, specified as:

```
.DS I
S: 220 foo.example.net

C: helo bar.example.com

S: 250 blah blah blah

C: quit

S: 221 Go in peace
.DE
```

Here is the same example as a variable list:

S: 220 foo.example.net

C: helo bar.example.com

S: 250 blah blah blah

C: quit

S: 221 Go in peace

Lists items are implemented as paragraphs, so formatting as paragraphs will look the same. The list was specified as:

```
.VL
.LI S:
220 foo.example.net
.LI C:
helo bar.example.com
.LI S:
250 blah blah blah
.LI C:
quit
.LI S:
221 Go in peace
.LE
```

Below is the same example formatted as a table:

S: 220 foo.example.net

C: helo bar.example.com

S: 250 blah blah blah

C: quit

S: 221 Go in peace

The table was specified by:

```
.TS
tab (~);
l1 l.
S:~220 foo.example.net

C:~helo bar.example.com

S:~250 blah blah blah

C:~quit

S:~221 Go in peace
.TE
```

A.4. Postprocessing

Scripts are available for use with the awk [I19.awk] interpreter for postprocessing of text and PostScript formatter output (primarily to reorder pages so that the Table of Contents appears in the correct location, although the postprocessor for text also removes overstriking). A sed [I20.sed] script is available to assist in preparation of HTML formatting.

A.5. Proofreading and Checking Output

It is so easy to incorporate references that it is easy to unintentionally include a given reference more than once. While authors can generally be nonchalant about formatting of references, references should be proofread carefully to check for duplicates, references categorized as normative which should instead be informative or vice versa, etc.

Henrik Levkowetz' "idnits" program [I14.idnits] is useful for catching diagrams and tables that exceed the maximum allowed line length.

Appendix B. Macros Specifically for the RFC Editor

In addition to the NU, FN, UD, OB, IS, and CA macros discussed in the body of this memo, there are several macros intended for RFC Editor (rather than author) use. They are described in this Appendix.

B.1. IESG Notes

The IN macro takes a single numeric argument indicating the type of IESG note to be included; some types take a second argument:

Type	2nd Argument	Text
1	none	The IESG has not found any conflict between this document and IETF work.

Type	2nd Argument	Text
2	Working Group	The IESG thinks that this work is related to IETF work done in WG Working Group, but this does not prevent publishing.
3	Working Group	The IESG thinks that publication is harmful to the IETF work done in WG Working Group and recommends not publishing the document at this time.
4	Procedure	The IESG thinks that this document violates IETF procedures for Procedure and should therefore not be published without IETF review and IESG approval.
5	none	The IESG thinks that this document extends an IETF protocol in a way that requires IETF review and should therefore not be published without IETF review and IESG approval.
6	none	The content of this RFC was at one time considered by the IETF, and therefore it may resemble a current IETF work in progress or a published IETF work. This RFC is not a candidate for any level of Internet Standard. The IETF disclaims any knowledge of the fitness of this RFC for any purpose and in particular notes that the decision to publish is not based on IETF review for such things as security, congestion control, or inappropriate interaction with deployed protocols. The RFC Editor has chosen to publish this document at its discretion. Readers of this RFC should exercise caution in evaluating its value for implementation and deployment. See RFC 3932 for more information.
7	none	This RFC is not a candidate for any level of Internet Standard. The IETF disclaims any knowledge of the fitness of this RFC for any purpose and in particular notes that the decision to publish is not based on IETF review for such things as security, congestion control, or inappropriate interaction with deployed protocols. The RFC Editor has chosen to publish this document at its discretion. Readers of this document should exercise caution in evaluating its value for implementation and deployment. See RFC 3932 for more information.

Type	2nd Argument	Text
8	none	This RFC is not a candidate for any level of Internet Standard. The IETF disclaims any knowledge of the fitness of this RFC for any purpose and notes that the decision to publish is not based on IETF review apart from IESG review for conflict with IETF work. The RFC Editor has chosen to publish this document at its discretion. See RFC 3932 for more information.

If a type of 0 (zero) is specified, text after the IN macro line is collected. The RFC Editor can use this feature to incorporate an IESG notice other then the ones prescribed in [N4.BCP78] as listed in the table above.

The IN macro should be placed in the front matter (before the first SH or NH macro). The corresponding note will be formatted as an unnumbered section appearing before the Abstract. If no IN macro call is placed in the front matter section, no IESG note will be included in the document.

B.2. Alternate Document Type and Number

Some RFCs are also BCP, STD, or FYI documents. The RFC Editor may indicate such a document type and number via the AN macro. It takes two arguments; the type (which must be one of BCP, STD, or FYI) and the number corresponding to that type. The information is placed in the first page heading, properly formatted. The AN macro should also be placed in the front matter.

Appendix C. Document Progression Using the rfc Macros

C.1. Initial Internet-Draft

An initial draft using the rfc macros should specify a draft number version of 00 by supplying a -1 argument to the NU macro or via command-line specification of register N. The latter may be accomplished by editing a line in the Makefile. Document category (CA macro) can be ignored, or can be specified to indicate intended status. The draft file name (see [N3.ID]) should be specified via the FN macro. Long (TL) and short (ST) titles need to be specified, as well as author(s) or editor(s) (AU or ED macro(s)) and Table of Contents parameters (TC macro or command-line register specifications). An Abstract (AB macro) is required. The remainder of the document is specified by defining sections, paragraphs, lists, and references, plus any diagrams or other non-textual material.

The document is then formatted and the text version submitted as an Internet-Draft [N3.ID].

During processing, the rfc macro package checks for certain types of errors and issues warnings and other messages. If such a message appears, it is advisable to consider a revision to the document or other appropriate action.

C.2. Subsequent Draft Revisions

Each revision involves updating the argument to the NU macro or command-line register N specification (sequencing to -2, then -3, etc.) and making desired changes to the content. Note that if the file name is changed, the numbering must be restarted at 00. If a Makefile is used, it too should be updated to reflect the revised version numbering.

The document is reformatted and resubmitted as a revised Draft [N3.ID].

C.3. Publication as an RFC

The Abstract may be retained (possibly edited) or removed. Document category should be specified in consultation with the IETF or RFC Editor. The document number should be left unspecified (0 (zero) argument to the NU macro or command-line register N specification); the RFC Editor will assign an RFC number at the time of publication. The document is reformatted and sent to the RFC Editor. By prior arrangement with the RFC Editor, the document source can be sent, as that may simplify final formatting by the RFC Editor.

Appendix D. Obtaining the rfc Macros and Related Tools

The rfc macros and related tools are currently available at http://users.erols.com/blilly/formatting. Individual files are:

File	Description
tmac.rfc	the rfc macros
Makefile	a makefile used to automate preprocessing, formatting, and postprocessing. It can be modified as needed for specific documents.
repaginate.awk	an awk script used for postprocessing text output. It reorders pages so that the Table of Contents appears in the correct location.
psrenumber.awk	an awk script for postprocessing PostScript output. It reorders pages so that the Table of Contents appears in the correct location.
htmlize.sed	a sed script for HTML markup
template.rfc	Template suitable as a draft starting point
draft-lilly-using-troff-04.rfc	the source for this document

File	Description
draft-lilly-using-troff-04.txt	text output for this document
draft-lilly-using-troff-04.ps	PostScript output for this document
draft-lilly-using-troff-04.pdf	PDF output for this document
draft-lilly-using-troff-04.psa4	PostScript output for this document, formatted for ISO A4 paper
draft-lilly-using-troff-04.pdfa4	PDF output for this document, formatted for ISO A4 paper
draft-lilly-using-troff-04.html	HTML 4.01 Strict compliant version of this document preserving page layout and numbering

There are also subdirectories containing source code and related files for the ABNF formatter and reference formatter preprocessors.

If the macros and related tools are deemed useful, it is hoped that they might be relocated to the RFC Editor site.

Appendix E. Summary of Directives

The following table summarizes rfc macro directives in alphabetical order (macros specific to preprocessors and internal use macros are omitted). The section number containing detailed information about each macro is listed.

Name	Section	Description
AB	5.3.1	Abstract
AN	B.2	Alternate document series number
AP	5.3.10	Appendix
AU	5.2.6	Author
BL	5.3.7.2	Bullet list
CA	5.2.3	Category or Intended status
DE	5.3.6	Display end
DF	5.3.6	Floating display start
DS	5.3.6	Display start
DT	5.2.7	Date
ED	5.2.6	Editor
EM	5.4	End matter
FN	5.2.5	(Internet-Draft) file name (w/o version)
IB	5.3.9	MIB (or MIB-like) boilerplate
IE	5.3.8	Informative reference end
IN	B.1	IESG note
IP	5.3.3	Indented paragraph
IR	5.3.8	Informative reference start
IS	5.2.9	Independent submission (to RFC Editor)
KE	5.3.6	Keep end
KF	5.3.6	Floating keep start
KS	5.3.6	Keep start
LE	5.3.7	List end

Name	Section	Description
LI	5.3.7	List item
LP	5.3.3	Left-aligned paragraph
NE	5.3.8	Normative reference end
NH	5.3.2	Numbered section heading
NL	5.3.7.1	Numbered list
NR	5.3.8	Normative reference start
NS	5.3.5	Needed (vertical) space
NU	5.2.2	RFC number or draft version
OB	5.2.4	Obsoleted RFCs
ON	5.2.10	Optional notice
RE	5.3.4	Decrease indent
RS	5.3.4	Increase indent
SH	5.3.2	(unnumbered) Section heading
ST	5.2.1	Short title for page heading
TC	5.2.8	Table of contents specification
TL	5.2.1	(Long) document title
UP	5.2.4	Updated RFCs
VL	5.3.7.3	Variable list

Appendix F. Source File Directive Order

Some directives appear in certain parts of a source document as summarized in the following diagram. For content ordering, see [I13.Instruct].

Part	Directives	Comments
Front Matter (before Abstract and first section heading)	NU	RFC or draft version number
	TL	(long) Title
	ST	Short title
	DT	Date override
	AU	Author
	ED	Editor
	FN	Internet-Draft file name
	TC	Table of contents parameters
	CA	category or Intended status
	IS	Independent submission
	IN	IESG note
	ON	Optional notice
	AN	Alternate series number
	UP	Updated RFCs
	OB	Obsoleted RFCs
Abstract	AB	(required for Internet-Draft)
	IP	(additional Abstract paragraphs)

Part	Directives	Comments
	NH	Numbered section
	SH	Unnumbered section
	NS	Need (vertical) space
	IP, LP	paragraphs
	RS, RE	Indentation adjustment
	IR, IE BR, FR, RR, SR	Informative references
	NR, NE BR, FR, RR, SR	Normative references
	DS, DF, DE	displays
	KS, KF, KE	keeps
Body and Appendices	BL, NL, VL LI, LE	Lists
	IB	MIB (or MIB-like) copyright notice
	EQ, EN	Mathematical equations
	TS, TH, T&, TE	Tables
	AS, BC, AE	ABNF
	PS, PE	Diagrams
	G1, G2	Graphs
	.begin, .end	Data formats
	.cstart, .cend	Chemical formulae
	AP	Appendices (after bulk of body)
End Matter	EM	last macro directive in source document

Appendix G. Disclaimers

This document has exactly one (1) author.

In spite of the fact that the author's given name may also be the surname of other individuals, and the fact that the author's surname may also be a given name for some females, the author is, and has always been, male.

The presence of "or she", "/SHE", "each", "their", and "authors" (plural) in the boilerplate sections of this document is irrelevant.

The author of this document is not responsible for the boilerplate text.

Comments regarding the silliness, lack of accuracy, and lack of precision of the boilerplate text should be directed to the IESG, not to the author.

Appendix H. Change History

[[This change history will not be part of a published RFC]]

-03 to -04

- Added information about implementation (preprocessor, formatter, etc.) issues
- [updated Makefile, macros, scripts to improve compatibility]
- [updated htmlize.sed to guess at section number references, URIs, and mailboxes]
- Corrected typos and errors in text

-02 to -03

- Added ED macro directive description for document Editor (vs. Author)
- fixed file list to include draft version suffix
- added tip regarding duplicate references
- added mention and reference to idnits
- [added HTML generation to Makefile]
- added client-server exchange examples
- [revised macros to enforce strict IETF Secretariat boilerplate rules which appear to require authors to be hermaphroditic, transsexual, (individuals) or multi-gender teams effective 6 May 2005]
- lowered page count limit for TOC in accordance with ID-Checklist Revision 1.2
- separate paragraphs in RFC 3978 boilerplate (idnits bug fixed)
- updated reference to March 2005 version of 1id-guidelines (also reported typo, so there'll be yet another version)
- clarified 0 as zero in default value column of register table in Appendix A

- added summary of directives
- added guide to directives in source document sections
- revised some wording
- added grap example
- rearranged directives in front matter to correspond to table in appendix
- added template document to files
- [revised Makefile to better handle preprocessors]
- [revised Makefile to incorporate rfcref preprocessor]
- added discussion of rfcref preprocessor
- [revised HTML generation to improve rendition of copyright symbols and bullets]
- [revised HTML generation to provide reference anchors and links]
- [revised HTML generation to provide table of contents anchors and links]
- [split out much of HTML manipulation to htmlize.sed to facilitate maintenance]
- added additional optional arguments to IB macro directive to accommodate use within MIB syntax
- added disclaimers of boilerplate silliness, inaccuracy, and imprecision

-01 to -02

- added this change history
- added acknowledgment re. reference style
- removed extraneous commas from preprocessor reference list
- corrected typos
- revised pic for SMTP diagram
- added section regarding ABNF formatting
- added caveat regarding multiple paragraphs in a keep
- added .NS macro to reserve vertical space

- updated boilerplate for revised BCP 78 and BCP 79

-00 to -01

- clarified issue w.r.t. low-level directives (as distinguished from macro call directives)
- updated Requirement levels to correspond to changed text
- some extraneous commas unintentionally crept into the list of preprocessor references
- added description of alternative methods for loading macros
- updated descriptions to incorporate setting registers from the command line
- added text regarding draft file name lower-case letters requirement
- improved usage
- rearranged some text under Document Content heading
- added recommendations regarding troff units
- improved differentiation between keeps and displays
- improved table formatting
- added floating keeps and displays
- added pointer regarding text following a display
- added optional argument to bulleted list macro
- added table of single-character registers
- added description of formatting for paper sizes other than ANSI A
- added tips and tricks for eqn, chem, dformat, pic, and tbl
- added description of document progression
- added ISO A4 versions of formatted documents

Normative References

[N1.RFC2223] Postel, J. and J. Reynolds, "Instructions to RFC Authors", RFC 2223, October 1997.

[N2.BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[N3.ID] "Guidelines to Authors of Internet-Drafts", March 2005, ftp://ftp.ietf.org/ietf/1id-guidelines.txt

[N4.BCP78] Bradner, S., "IETF Rights in Contributions", BCP 78, RFC 3978, March 2005.

Informative References

[I1.CSTR54] Ossanna, Joseph F., "NROFF/TROFF User's Manual", Computing Science Technical Report No.54, Bell Laboratories, Murray Hill, New Jersey, 1976.

[I2.ms] Lesk, M. E., "Typing Documents on the UNIX System: Using the -ms Macros with Troff and Nroff", 1978, in "UNIX TIME-SHARING SYSTEM (VOLUME 2): UNIX Programmer's Manual", Holt, Rinehart, & Winston, 1979

[I3.RFC1543] Postel, J., "Instructions to RFC Authors", RFC 1543, October 1993.

[I4.CSTR49] Lesk, M. E., "TBL - A Program for Setting Tables", Bell Laboratories Computing Science Technical Report #49, Murray Hill, New Jersey, 1976.

[I5.CSTR114] Bentley, Jon L. and Kernighan, Brian W., "Grap - A Language for Typesetting Graphs Tutorial and User Manual", Computing Science Technical Report No.114, AT&T Bell Laboratories, Murray Hill, New Jersey, 1991.

[I6.CSTR116] Kernighan, Brian W., "Pic - A Graphics Language for Typesetting User Manual", Computing Science Technical Report No.116, AT&T Bell Laboratories, Murray Hill, New Jersey, 1991.

[I7.CSTR122] Bentley, Jon L., Jelinski, Lynn W., and Kernighan, Brian W., "Chem - A Program for Typesetting Chemical Diagrams: User Manual", Computing Science Technical Report No.122, AT&T Bell Laboratories, Murray Hill, New Jersey, 1992.

[I8.eqn] Kernighan, Brian W, and Cherry, Lorinda L., "A System for Typesetting Mathematics", Communications of the ACM 18, 182-193, 1975.

[I9.CSTR142] Bentley, Jon L. "DFORMAT - A Program for Typesetting Data Formats", Computing Science Technical Report No.142, AT&T Bell Laboratories, Murray Hill, New Jersey, 1988.

[I10.make] Feldman, S. I., "Make — A Program for Maintaining Computer Programs", 1978, in "UNIX TIME-SHARING SYSTEM (VOLUME 2) : UNIX Programmer's Manual", Holt, Rinehart, & Winston, 1979

[I11.BCP9]   Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[I12.BCP25]   Bradner, S., "IETF Working Group Guidelines and Procedures", BCP 25, RFC 2418, September 1998.

[I13.Instruct] Reynolds, J., and R. Braden, "Instructions to Request for Comments (RFC) Authors", Work in progress (August 2004).

[I14.idnits]   http://ietf.levkowetz.com/tools/idnits/

[I15.RFC3834] Moore, K., "Recommendations for Automatic Responses to Electronic Mail", RFC 3834, August 2004.

[I16.ANSI151] American National Standards Institute (ANSI), "Bond Papers and Index Bristols - Common Sheet Sizes", ANSI INCITS 151-1987 (R2002)

[I17.ISO216]   International Organization for Standardization (ISO), "Writing paper and certain classes of printed matter -- Trimmed sizes -- A and B series", ISO 216:1975

[I18.RFC2234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[I19.awk]   Aho, Alfred V., Wienberger, Peter J., and Kernighan, Brian W., "Awk — A Pattern Scanning and Processing Language", 1978, in "UNIX TIME-SHARING SYSTEM (VOLUME 2): UNIX Programmer’s Manual", Holt, Rinehart, & Winston, 1979

[I20.sed]   McMahon, Lee E., "SED — A Non-interactive Text Editor", 1978, "UNIX TIME-SHARING SYSTEM (VOLUME 2): UNIX Programmer’s Manual", Holt, Rinehart, & Winston, 1979

Author’s Address

Bruce Lilly

Email: blilly@erols.com

Full Copyright Statement

Intellectual Property Statement

Acknowledgment

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