

Using *groff* with the *-ms* Macro Package

Larry Kollar

kollar@alltel.net

This document describes the GNU re-implementation of the popular *groff -ms* macro package.

The *-ms* macros are suitable for reports, letters, books, user manuals, and so forth. The package provides macros for cover pages, section headings, paragraphs, lists, footnotes, pagination, and a table of contents.

This document provides both tutorial and reference material. It also discusses differences from the AT&T *-ms* macros, both omissions and additions.

1. Introduction

The original *-ms* macros were included with AT&T *troff* as well as the *-man* macros. While the *-man* package is intended for brief documents that can be read on-line as well as printed, the *-ms* macros are suitable for longer documents that are meant to be printed rather than read on-line.

The *-ms* macro package included with *-groff* is a complete, bottom-up re-implementation. Several macros (specific to AT&T or Berkeley) are not included, while several new commands are. See *Differences from AT&T -ms*, later in this document, for more information.

1.1. Basic information

Most *number registers* have a default unit of measurement. Specifying a unit explicitly does not cause any problems, and can avoid problems in complex situations. The following are typical units:

Unit	Description
i	inches
c	centimeters
p	points (about 1/72")
v	line height
n	width of a letter "n" in the current font
m	width of a letter "M" in the current font
P	pica (1/6")

You set number registers by using the `.nr` request; for example:

```
.nr PS 12
```

2. General structure of an *-ms* document

The *-ms* macro package expects a certain amount of structure, but not as much as packages such as *-man* or *-mdoc*.

The simplest documents can begin with a paragraph macro (such as `.LP` or `.PP`), and consist of text separated by paragraph macros or even blank lines. Longer documents have a structure as follows:

Document type

If you invoke the `.RP` (report) macro on the first line of the document, *groff* prints the cover page information on its own page; otherwise it prints the information on the first page with your document text immediately following. Other document formats found in AT&T *troff* are specific to AT&T or Berkeley, and are not supported in *groff*.

Format and layout

By setting number registers, you can change your document's type (font and size), margins, spacing, headers and footers, and footnotes. See *Document control registers* below for more details.

Cover page

A cover page consists of a title, the author's name and institution, an abstract, and the date.¹ See *Cover page macros* below for more details.

Body

Following the cover page is your document. You can use the *-ms* macros to write reports, letters, books, and so forth. The package is designed for structured documents, consisting of paragraphs interspersed with headings and augmented by lists, footnotes, tables, and other common constructs. See *Body text* below for more details.

Table of contents

Longer documents include a table of contents, which you can invoke by placing the `.TC` macro at the end of your document.² Printing the table of contents at the end is necessary since *groff* is a single-pass text formatter, thus it cannot determine the page number of each section until that section has actually been set and printed. Since *-ms* output was intended for hardcopy, the standard procedure was to manually relocate the pages containing the table of contents between the cover page and the body text.

¹ Actually, only the title is required.

² The *-ms* macros have minimal indexing facilities, consisting of the `.IX` macro, which prints an entry on standard error.

3. Document control registers

The following table lists the document control number registers. For the sake of consistency, set registers related to margins at the beginning of your document, or just after the `.RP` macro. You can set other registers later in your document, but you should keep them together at the beginning to make them easy to find and edit as necessary.

Type	Register	Definition	Effective	Default
Margins	PO	Page offset (left margin)	next page	1i
	LL	Line length	next ¶	6i
	LT	Header/footer (title) length	next ¶	6i
	HM	Top (header) margin	next page	1i
	FM	Bottom (footer) margin	next page	1i
Text	PS	Point size	next ¶	10p
	VS	Line spacing (leading)	next ¶	12p
Paragraphs	PI	Initial indent	next ¶	5n
	PD	Space between paragraphs	next ¶	0.3v
	QI	Quoted paragraph indent	next ¶	5n
Footnotes	FL	Footnote length	next footnote	LL*5/6
	FI	Footnote indent	next footnote	2n
	FF	Footnote format	next footnote	0
Other	MINGW	Minimum width between columns	next page	2n

4. Cover page macros

Use the following macros to create a cover page for your document in the order shown.

Macro	Description
.RP [no]	<p>Specifies the report format for your document. The report format creates a separate cover page. The default action (no .RP macro) is to print a subset of the cover page on page 1 of your document.</p> <p>If you use the optional no argument, <i>groff</i> prints a title page but does not repeat any of the title page information (title, author, abstract, etc.) on page 1 of the document.</p>
.DA [xxx]	(optional) Print the current date, or the arguments to the macro if any, on the title page (if specified) and in the footers. This is the default for <i>nroff</i> .
.ND xxx [xxx]	(optional) Print the current date, or the arguments to the macro if any, on the title page (if specified) but not in the footers. This is the default for <i>troff</i> .
.TL	Specifies the document title. <i>groff</i> collects text following the .TL macro into the title, until reaching the author name or abstract.
.AU	<p>Specifies the author's name, which appears on the line (or lines) immediately following. You can specify multiple authors as follows:</p> <pre> .AU John Doe .AI University of West Bumblefuzz .AU Martha Buck .AI Monolithic Corporation ... </pre>
.AI	Specifies the author's institution. You can specify multiple institutions in the same way that you specify multiple authors.
.AB [no]	Begins the abstract. The default is to print the word ABSTRACT, centered and in italics, above the text of the abstract. The option no suppresses this heading.
.AE	End the abstract.

The following is example mark-up for a title page.

```
.RP
.TL
The Inevitability of Code Bloat
in Commercial and Free Software
.AU
J. Random Luser
.AI
University of West Bumblefuzz
.AB
This report examines the long-term growth
of the code bases in two large, popular software
packages; the free Emacs and the commercial
Microsoft Word.
While differences appear in the type or order
of features added, due to the different
methodologies used, the results are the same
in the end.
.PP
The free software approach is shown to be
superior in that while free software can
become as bloated as commercial offerings,
free software tends to have fewer serious
bugs and the added features are in line with
user demand.
.AE
... the rest of the paper follows ...
```

5. Body text

This section describes macros used to mark up the body of your document. Examples include paragraphs, sections, and other groups.

5.1. Paragraphs

Use the `.PP` macro to create indented paragraphs (like the next paragraph), and the `.LP` macro to create paragraphs with no initial indent (like this one).

The `.QP` macro indents its text at both left and right margins. The effect is identical to the HTML `<BLOCKQUOTE>` element. The next paragraph or heading returns margins to normal.

The following markup uses all three paragraph macros.

```
.NH 2
Cases used in the study
.LP
The following software and versions were
considered for this report.
.PP
For commercial software, we chose
.B "Microsoft Word for Windows" ,
starting with version 1.0 through the
current version (Word 2000).
.PP
For free software, we chose
.B Emacs ,
from its first appearance as a standalone
editor through the current version (v20).
.QP
Franklin's Law applied to software:
software expands to outgrow both
RAM and disk space over time.
```

5.2. Headings

Use headings to create a hierarchical structure for your document. The `-ms` macros print headings in **bold**, using the same font family and point size as the body text.

The following table describes the heading macros:

Macro	Description
<code>.NH xx</code>	Numbered heading. The argument <i>xx</i> is either a numeric argument to indicate the level of the heading, or <i>S xx xx...</i> to set the heading level explicitly. The section headings in this document use the <code>.NH</code> macro to show the level of each section. If you specify heading levels out of sequence, such as invoking <code>.NH 3</code> after <code>.NH 1</code> , <i>groff</i> prints a warning on standard error.
<code>.SH</code>	Unnumbered subheading.

5.3. Highlighting

The `-ms` macros provide a variety of methods to highlight or emphasize text:

Macro	Description
<code>.B [txt [post [pre]]]</code>	<p>Sets its first argument in bold type. If you specify a second argument, <i>groff</i> prints it in the previous font after the bold text, with no intervening space (this allows you to set punctuation after the highlighted text without highlighting the punctuation). Similarly, it prints the third argument (if any) in the previous font before the first argument. For example,</p> <pre>.B foo) (</pre> <p>prints (foo).</p> <p>If you give this macro no arguments, <i>groff</i> prints all text following in bold until the next highlighting, paragraph, or heading macro.</p>
<code>.R [txt [post [pre]]]</code>	<p>Sets its first argument in roman (or regular) type. It operates similarly to the <code>.B</code> macro otherwise.</p>
<code>.I [txt [post [pre]]]</code>	<p>Sets its first argument in <i>italic type</i>. It operates similarly to the <code>.B</code> macro otherwise.</p>
<code>.CW [txt [post [pre]]]</code>	<p>Sets its first argument in a constant width face. It operates similarly to the <code>.B</code> macro otherwise.</p>
<code>.BI [txt [post [pre]]]</code>	<p>Sets its first argument in <i>bold italic type</i>. It operates similarly to the <code>.B</code> macro otherwise.</p>
<code>.BX [txt]</code>	<p>Prints its argument and draws a <code>box</code> around it. If you want to box a string that contains spaces, use a digit-width space (<code>\0</code>).</p>
<code>.UL [txt [post]]</code>	<p>Prints its first argument with an underline. If you specify a second argument, <i>groff</i> prints it in the previous font after the underlined text, with no intervening space.</p>
<code>.LG</code>	<p>Prints all text following in larger type (2 points larger than the current point size) until the next font size, highlighting, paragraph, or heading macro. You can specify this macro multiple times to enlarge the point size as needed.</p>
<code>.SM</code>	<p>Prints all text following in smaller type (2 points smaller than the current point size) until the next type size, highlighting, paragraph, or heading macro. You can specify this macro multiple times to reduce the point size as needed.</p>
<code>.NL</code>	<p>Prints all text following in the normal point size (that is, the value of the PS register).</p>

5.4. Lists

The `.IP` macro handles duties for all lists. Its syntax is as follows:

```
.IP [marker [width]]
```

The *marker* is usually a bullet character (`\(bu)` for unordered lists, a number (or auto-incrementing number register) for numbered lists, or a word or phrase for indented (glossary-style) lists.

The *width* specifies the indent for the body of each list item. Once specified, the indent remains the same for all list items in the document until specified again.

The following are examples of each type of list:

Source	Result
<pre>A bulleted list: .IP \(\bu 2 lawyers .IP \(\bu guns .IP \(\bu money</pre>	<pre>A bulleted list: • lawyers • guns • money</pre>
<pre>.nr step 1 1 A numbered list: .IP \n[step] 3 lawyers .IP \n+[step] guns .IP \n+[step] money</pre>	<pre>A numbered list: 1. lawyers 2. guns 3. money Note the use of the auto-incrementing number register in this example.</pre>
<pre>A glossary-style list: .IP lawyers 0.4i Two or more attorneys. .IP guns Firearms, preferably large-caliber. .IP money Gotta pay for those lawyers and guns!</pre>	<pre>A glossary-style list: lawyers Two or more attorneys. guns Firearms, preferably large-caliber. money Gotta pay for those lawyers and guns!</pre>

In the last example, note how the `.IP` macro places the definition on the same line as the term if it has enough space. This may or may not be the effect you want. The following example shows two possible workarounds:

Code	Result
<pre>A glossary-style list: .IP lawyers 0.4i Two or more attorneys. .IP guns .br Firearms, preferably large-caliber. .IP money Gotta pay for those lawyers and guns!</pre>	<pre>A glossary-style list: lawyers Two or more attorneys. guns Firearms, preferably large-caliber. money Gotta pay for those lawyers and guns!</pre>
<pre>A glossary-style list: .IP lawyers 0.4i Two or more attorneys. .IP guns \p Firearms, preferably large-caliber. .IP money Gotta pay for those lawyers and guns!</pre>	<pre>A glossary-style list: lawyers Two or more attorneys. guns Firearms, preferably large-caliber. money Gotta pay for those lawyers and guns!</pre>

The first example uses the `.br` request to force a break after printing the term or label. The second example uses the `\p` escape to do the same thing. Note the space following the escape; this is important. If you omit the space, *groff* prints the first word on the same line as the term or label (if it fits) **then** breaks the line.

To set nested lists, use the `.RS` and `.RE` macros. These macros begin and end a section indented to line up with the body of an `.IP` macro. For example:

<code>.IP \ (bu 2</code>	• Lawyers:
Lawyers:	• Dewey,
<code>.RS</code>	• Cheatham,
<code>.IP \ (bu</code>	• and Howe.
Dewey,	• Guns ...
<code>.IP \ (bu</code>	
Cheatham,	
<code>.IP \ (bu</code>	
and Howe.	
<code>.RE</code>	
<code>.IP \ (bu</code>	
Guns	
...	

5.5. Displays and keeps

Use displays to show text-based examples or figures (such as code listings). This document shows *groff* code examples inside displays, for example.

Displays turn off filling, so lines of code can be displayed as-is without inserting `.br` requests in between each line. Displays can be *kept* on a single page, or allowed to break across pages. The following table shows the display types available.

Display macro		Description
With keep	No keep	
.DS L	.LD	Left-justified display.
.DS I [<i>indent</i>]	.ID	Indented display (default is the DI register).
.DS B	.BD	Block-centered display (left-justified, longest line centered).
.DS C	.CD	Centers all lines in the display.
.DS R	.RD	Right-justifies all lines in the display.

Use the `.DE` macro to end any display type.

On occasion, you may want to *keep* other text together on a page. For example, you may want to keep two paragraphs together, or a paragraph that refers to a table (or list, or other item) immediately following. The *-ms* macros provide the `.KS` and `.KE` macros for this purpose. The `.KS` macro begins a block of text to be kept on a single page, and the `.KE` macro ends the block.

You can specify a *floating keep*; if the keep cannot fit on the current page, *groff* holds the contents of the keep and allows text following the keep (in the source file) to fill in the remainder of the current page. When the page breaks, whether by an explicit `.bp` request or by reaching the end of the page, *groff* prints the floating keep at the top of the new page. This is useful for printing large graphics or tables that do not need to appear exactly where specified. Use the `.KF` and `.KE` macros to specify a floating keep.

You can also use the `.ne` request to force a page break if there is not enough vertical space remaining on the page.

5.6. Tables, figures, equations, and references

The *-ms* macros support the standard *groff* preprocessors: *tbl*, *pic*, *eqn*, and *refer*. You mark text meant for preprocessors by enclosing it in pairs of tags as follows:

Tag Pair	Description
.TS [H] .TE	Denotes a table, to be processed by the <i>tbl</i> preprocessor. The optional H argument to <code>.TS</code> instructs <i>groff</i> to create a running header with the information up to the <code>.TH</code> macro. <i>Groff</i> prints the header at the beginning of the table; if the table runs onto another page, <i>groff</i> prints the header on the next page as well.
.PS .PE	Denotes a graphic, to be processed by the <i>pic</i> preprocessor. You can create a <i>pic</i> file by hand, using the AT&T <i>pic</i> manual available on the Web as a reference, or by using a graphics program such as <i>xfig</i> .
.EQ [<i>align</i>] .EN	Denotes an equation, to be processed by the <i>eqn</i> preprocessor. The optional <i>align</i> argument can be C , L , or I to center (the default), left-justify, or indent the equation.
. [.]	Denotes a reference, to be processed by the <i>refer</i> preprocessor. The GNU <i>refer</i> (1) manpage provides a comprehensive reference to the preprocessor and the format of the bibliographic database.

5.6.1. An example multi-page table

The following is an example of how to set up a table that may print across two or more pages.

```
.TS H
allbox expand;
cb | cb .
Text      ...of heading...
—
.TH
.T&
l | l .
... the rest of the table follows...
.TE
```

5.7. Footnotes

The *-ms* macro package has a flexible footnote system. You can specify a numbered footnote³ by using the `**` escape, followed by the text of the footnote enclosed by `.FS` and `.FE` macros.

You can specify symbolic footnotes† by placing the character (such as `\(dagger` for the dagger character used here), followed by the symbol and the text of the footnote enclosed by `.FS` and `.FE` macros.

You can control how *groff* prints footnote numbers by changing the value of the `FF` register as follows:

Value	Description
0	Prints the footnote number as a superscript; indents the footnote (default).
1	Prints the number followed by a period (like 1.) and indents the footnote.
2	Like 1, without an indent.
3	Like 1, but prints the footnote number as a hanging paragraph.

6. Page layout

The default output from the *-ms* macros provides a minimalist page layout: it prints a single column, with the page number centered at the top of each page. It prints no footers.

You can change the layout by setting the proper number registers and strings.

6.1. Headers and footers

There are two ways to define headers and footers:

- Set the strings `LH`, `CH`, and `RH`, to set the left, center, and right headers; and `LF`, `CF`, and `RF` to set the left, center, and right footers. This works best for documents that do not distinguish between odd and even pages.
- Use the `.OH` and `.EH` macros to define headers for the odd and even pages; and `.OF` and `.EF` macros to define footers for the odd and even pages. This is more flexible than defining the individual strings. The syntax for these macros is as follows:

```
.OH 'left' center' right'
```

³ This is a numbered footnote.

†This is a symbolic footnote.

You can replace the quote (') marks with any character not appearing in the header or footer text.

6.2. Margins

You control margins using a set of number registers. The following table lists the register names and defaults:

Register	Definition	Effective	Default
PO	Page offset (left margin)	next page	1i
LL	Line length	next ¶	6i
LT	Header/footer (title) length	next ¶	6i
HM	Top (header) margin	next page	1i
FM	Bottom (footer) margin	next page	1i

Note that there is no right margin setting. The combination of page offset and line length provide the information necessary to derive the right margin.

6.3. Multiple columns

The *-ms* macros can set text in as many columns as will reasonably fit on the page. The following macros are available. All of them force a page break if a multi-column mode is already set. However, if the current mode is single-column, starting a multi-column mode does **not** force a page break.

Macro	Description
.1C	Single-column mode.
.2C	Two-column mode.
.MC [<i>width</i> [<i>gutter</i>]]	Multi-column mode. If you specify no arguments, it is equivalent to the .2C macro. Otherwise, <i>width</i> is the width of each column and <i>gutter</i> is the space between columns. The MINGW number register is the default gutter width.

6.4. Creating a table of contents

The facilities in the *-ms* macro package for creating a table of contents are semi-automated at best. Assuming that you want the table of contents to consist of the document's headings, you need to repeat those headings wrapped in .XS and .XE macros.

In addition, the .XS macro does not know to indent a heading based on its level. The easiest way to work around this is to add tabs to the table of contents string. The following is an example:

.NH 1	
Introduction	
.XS	
Introduction	
.XE	
...	
.NH 2	
Methodology	
.XS	
Methodology	
.XE	
...	

The *Groff and Friends HOWTO* includes a *sed* script that automatically inserts *.XS* and *.XE* entries after each heading in a document.

Altering the *.NH* macro to automatically build the table of contents is perhaps initially more difficult, but would save a great deal of time in the long run if you use *-ms* regularly.

7. Differences from AT&T *-ms*

This section lists the (minor) differences between the *groff -ms* macros and AT&T *troff -ms* macros.

7.1. *troff* macros not appearing in *groff*

Macros missing from *groff -ms* are cover page macros specific to Bell Labs. The macros known to be missing are:

- .TM* Technical memorandum; a cover sheet style
- .IM* Internal memorandum; a cover sheet style
- .MR* Memo for record; a cover sheet style
- .MF* Memo for file; a cover sheet style
- .EG* Engineer's notes; a cover sheet style
- .TR* Computing Science Tech Report; a cover sheet style
- .OK* Other keywords
- .CS* Cover sheet information
- .MH* A cover sheet macro

7.2. *groff -ms* macros not appearing in AT&T *troff*

The *groff -ms* macros have a few minor extensions compared to the AT&T *troff -ms* macros.

- .AM* Improved accent marks.
- .DS I* Indented display. The default behavior of AT&T *troff -ms* was to indent; the *groff* default prints displays flush left with the body text.
- .CW* Print text in `constant width` (Courier) font.
- .IX* Indexing term (printed on standard error).

The `MINGW` number register specifies a minimum space between columns (for multi-column output); this takes the place of the `GW` register that was documented but apparently not implemented in AT&T *troff*. Several new string registers are available as well. You can change these to handle (for example) the local language.

REFERENCES

Contains the string printed at the beginning of the references (bibliography) page.

ABSTRACT

Contains the string printed at the beginning of the abstract.

TOC

Contains the string printed at the beginning of the table of contents.

8. Acknowledgements

Two documents provided essential reference material:

- The *Groff and Friends HOWTO*, by Dean Allen Provins.
- *Using the -ms Macros with Troff and Nroff*, the original AT&T document by M. E. Lesk.

Without these documents close at hand, writing this document would have been a much more difficult task.

Table of Contents

Introduction	1
Basic information	1
General structure of an <i>-ms</i> document	2
Document control registers	3
Cover page macros	4
Body text	6
Paragraphs	6
Headings	6
Highlighting	6
Lists	7
Displays and keeps	9
Tables, figures, equations, and references	10
An example multi-page table	10
Footnotes	11
Page layout	11
Headers and footers	11
Margins	12
Multiple columns	12
Creating a table of contents	12
Differences from AT&T <i>-ms</i>	13
<i>troff</i> macros not appearing in <i>groff</i>	13
<i>groff</i> macros not appearing in AT&T <i>troff</i>	13
Acknowledgements	14