NAME

groff_ms – GNU roff manuscript macro package for formatting documents

SYNOPSIS

groff -ms [*option* . . .] [*input-file* . . .]
groff -m ms [*option* . . .] [*input-file* . . .]

DESCRIPTION

This manual page describes the GNU version of the *ms* macros, part of the *groff* typesetting system. The *ms* macros are mostly compatible with the documented behavior of the 4.3 BSD Unix *ms* macros (see *Differences from troff ms* below for details). The *ms* macros are suitable for reports, letters, books, and technical documentation.

USAGE

The *ms* macro package expects files to have a certain amount of structure. The simplest documents can begin with a paragraph macro and consist of text separated by paragraph macros or even blank lines. Longer documents have a structure as follows:

Document type
If you use the **RP** (report) macro at the beginning 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 ms*.

Format and layout
By setting number registers, you can change your document's margins, spacing, headers and footers, footnotes, and the base point size for the text. See *Document control registers* below for more details.

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

Body Following the cover page is your document. It consists of paragraphs, headings, and lists.

Table of contents
Longer documents usually include a table of contents, which you can add by placing the **TC** macro at the end of your document.

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.

Margin settings

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

Text settings

Reg.	Definition	Effective	Default
PS	Point size	next paragraph	10p
VS	Line spacing (leading)	next paragraph	12p
PSINCR	Point size increment for section headings of increasing importance	next heading	1p
GROWPS	Heading level beyond which PSINCR is ignored	next heading	0

Paragraph settings

Reg.	Definition	Effective	Default
PI	Initial indent	next paragraph	5n
PD	Space between paragraphs	next paragraph	0.3v
QI	Quoted paragraph indent	next paragraph	5n
PORPHANS	Number of initial lines to be kept together	next paragraph	1
HORPHANS	Number of initial lines to be kept with heading	next heading	1

Footnote settings

Reg.	Definition	Effective	Default
FL	Footnote length	next footnote	\n[LL]*5/6
FI	Footnote indent	next footnote	2n
FF	Footnote format	next footnote	0
FPS	Point size	next footnote	\n[PS]−2
FVS	Vert. spacing	next footnote	\n[FPS]+2
FPD	Para. spacing	next footnote	\n[PD]/2

Other settings

Reg.	Definition	Effective	Default
DD	Display, table, eqn, pic spacing	next para.	0.5v
MINGW	Minimum width between columns	next page	2n

Cover page macros

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

.RP [**no**]

Specifies the report format for your document. The report format creates a separate cover page. With no **RP** macro, *groff* prints a subset of the cover page on page 1 of your document.

If you use the optional **no** argument, *groff* prints a title page but does not repeat any of the title page information (title, author, abstract, etc.) on page 1 of the document.

.P1 (P-one) Prints the header on page 1. The default is to suppress the header.

.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 *nroff*.

.ND [*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 *troff*.

.TL Specifies the document title. *Groff* collects text following the **TL** macro into the title, until reaching the author name or abstract.

.AU Specifies the author's name. You can specify multiple authors by using an **AU** macro for each author.

.AI Specifies the author's institution. You can specify multiple institutions.

.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.

Paragraphs

Use the **PP** macro to create indented paragraphs, and the **LP** macro to create paragraphs with no initial indent.

The **QP** macro indents all text at both left and right margins by the amount of the register **QI**. The effect is reminiscent of the HTML **<BLOCKQUOTE>** tag. The next paragraph or heading returns the margins to normal. **QP** inserts the vertical space specified in register **PD** as inter-paragraph spacing.

A paragraph bracketed between the macros **QS** and **QE** has the same appearance as a paragraph started with **QP** and a following paragraph started with **LP**. Both **QS** and **QE** insert the inter-paragraph spacing specified in **PD** and the text is indented on both sides by the amount of register **QI**. The text between **QS** and **QE** can be split into further paragraphs by using **.LP** or **.PP**.

The **XP** macro produces an "exdented" paragraph; that is, one with a hanging indent. The first line of the paragraph begins at the left margin, and subsequent lines are indented (the opposite of **PP**).

For each of the above paragraph types, and also for any list entry introduced by the **IP** macro (described later), the document control register **PORPHANS**, sets the *minimum* number of lines which must be printed, after the start of the paragraph, and before any page break occurs. If there is insufficient space remaining on the current page to accommodate this number of lines, then a page break is forced *before* the first line of the paragraph is printed.

Similarly, when a section heading (see subsection "Headings" below) precedes any of these paragraph types, the **HORPHANS** document control register specifies the *minimum* number of lines of the paragraph which must be kept on the same page as the heading. If insufficient space remains on the current page to accommodate the heading and this number of lines of paragraph text, then a page break is forced *before* the heading is printed.

Headings

Use headings to create a hierarchical structure for your document. By default, the *ms* macros print headings in **bold** using the same font family and point size as the body text. For output devices which support scalable fonts, this behaviour may be modified by defining the document control registers **GROWPS** and **PSINCR**.

The following heading macros are available:

.NH *xx* Numbered heading. The argument *xx* is either a numeric argument to indicate the level of the heading, or **S** *xx xx* ... to set the section number explicitly. If you specify heading levels out of sequence, such as invoking **.NH 3** after **.NH 1**, *groff* prints a warning on standard error.

If the **GROWPS** register is set to a value greater than the level of the heading, then the point size of the heading will be increased by **PSINCR** units over the text size specified by the **PS** register, for each level by which the heading level is less than the value of **GROWPS**. For example, the sequence:

```
.nr PS 10
.nr GROWPS 3
.nr PSINCR 1.5p
.
.NH 1
Top Level Heading
.
.NH 2
Second Level Heading
.
.NH 3
Third Level Heading
```

will cause "*1. Top Level Heading*" to be printed in 13pt **bold** text, followed by "*1.1. Second Level Heading*" in 11.5pt **bold** text, while "*1.1.1. Third Level Heading*", and all more deeply nested heading levels, will remain in the 10pt **bold** text which is specified by the **PS** register.

Note that the value stored in **PSINCR** is interpreted in *groff* basic units; the *p* scaling factor should be employed when assigning a value specified in points.

The style used to represent the section number, within a numbered heading, is controlled by the **SN−STYLE** string; this may be set to either the **SN−DOT** or the **SN−NO−DOT** style, (described below), by aliasing **SN−STYLE** accordingly. By default, **SN−STYLE** is initialised by defining the alias

```
.als SN−STYLE SN−DOT
```

it may be changed to the **SN−NO−DOT** style, if preferred, by defining the alternative alias

```
.als SN−STYLE SN−NO−DOT
```

Any such change becomes effective with the first use of **.NH**, *after* the new alias is defined.

After invoking **.NH**, the assigned heading number is available in the strings **SN−DOT** (as it appears in the default formatting style for numbered headings, with a terminating period following the number), and **SN−NO−DOT** (with this terminating period omitted). The string **SN** is also defined, as an alias for **SN−DOT**; if preferred, the user may redefine it as an alias for **SN−NO−DOT**, by including the initialisation:

```
.als SN SN−NO−DOT
```

at any time; the change becomes effective with the next use of **.NH**, *after* the new alias is defined.

.SH [*xx*]

Unnumbered subheading. The use of the optional *xx* argument is a GNU extension, which adjusts the point size of the unnumbered subheading to match that of a numbered heading, introduced using **.NH** *xx* with the same value of *xx*. For example, given the same settings for **PS**, **GROWPS** and **PSINCR**, as used in the preceding **.NH** example, the sequence:

```
.SH 2
An Unnumbered Subheading
```

will print "*An Unnumbered Subheading*" in 11.5pt **bold** text.

Highlighting

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

.B [*txt* [*post* [*pre*]]]

Sets its first argument in **bold type**. If you specify a second argument, *groff* 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,

```
.B foo ) (
```

prints "(**foo**)".

If you give this macro no arguments, *groff* prints all text following in bold until the next highlighting, paragraph, or heading macro.

.R [*txt* [*post* [*pre*]]]
: Sets its first argument in roman (or regular) type. It operates similarly to the **B** macro otherwise.

.I [*txt* [*post* [*pre*]]]
: Sets its first argument in *italic type*. It operates similarly to the **B** macro otherwise.

.CW [*txt* [*post* [*pre*]]]
: Sets its first argument in a constant-width face. It operates similarly to the **B** macro otherwise.

.BI [*txt* [*post* [*pre*]]]
: Sets its first argument in bold italic type. It operates similarly to the **B** macro otherwise.

.BX [*txt*]
: Prints its argument and draws a box around it. If you want to box a string that contains spaces, use a digit-width space (\0).

.UL [*txt* [*post*]]
: Prints its first argument with an underline. If you specify a second argument, *groff* prints it in the previous font after the underlined text, with no intervening space.

.LG
: 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.

.SM
: 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.

.NL
: Prints all text following in the normal point size (that is, the value of the **PS** register).

{text****}**
: Print the enclosed *text* as a superscript.

Indents

You may need to indent sections of text. A typical use for indents is to create nested lists and sublists.

Use the **RS** and **RE** macros to start and end a section of indented text, respectively. The **PI** register controls the amount of indent.

You can nest indented sections as deeply as needed by using multiple, nested pairs of **RS** and **RE**.

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.

Tab stops

Use the **ta** request to set tab stops as needed. Use the **TA** macro to reset tabs to the default (every 5n). You can redefine the **TA** macro to create a different set of default tab stops.

Displays and keeps

Use displays to show text-based examples or figures (such as code listings). 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		Type of display
With keep	**No keep**	
`.DS L`	`.LD`	Left-justified.
`.DS I` [*indent*]	`.ID`	Indented (default indent in the **DI** register).
`.DS B`	`.BD`	Block-centered (left-justified, longest line centered).
`.DS C`	`.CD`	Centered.
`.DS R`	`.RD`	Right-justified.

Use the **DE** macro to end any display type. The macros **Ds** and **De** were formerly provided as aliases for **DS** and **DE**, respectively, but they have been removed, and should no longer be used. X11 documents which actually use **Ds** and **De** always load a specific macro file from the X11 distribution (*macros.t*) which provides proper definitions for the two macros.

To *keep* text together on a page, such as a paragraph that refers to a table (or list, or other item) immediately following, use the **KS** and **KE** macros. 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* using the **KF** and **KE** macros. 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.

The macros **B1** and **B2** can be used to enclose a text within a box; **.B1** begins the box, and **.B2** ends it. Text in the box is automatically placed in a diversion (keep).

Tables, figures, equations, and references

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

.TS [**H**] and **.TE**

Denote a table to be processed by the *tbl* preprocessor. The optional **H** argument instructs *groff* to create a running header with the information up to the **TH** macro. *Groff* prints the header at the beginning of the table; if the table runs onto another page, *groff* prints the header on the next page as well.

.PS and **.PE**

Denote a graphic to be processed by the *pic* preprocessor. You can create a *pic* file by hand, using the AT&T *pic* manual available on the Web as a reference, or by using a graphics program such as *xfig*.

.EQ [*align*] and **.EN**

Denote an equation to be processed by the *eqn* preprocessor. The optional *align* argument can be **C**, **L**, or **I** to center (the default), left-justify, or indent the equation, respectively.

.[and **.]**

Denote a reference to be processed by the *refer* preprocessor. The GNU *refer*(1) manual page provides a comprehensive reference to the preprocessor and the format of the bibliographic database.

Footnotes

The *ms* macros provide a flexible footnote system. You can specify a numbered footnote 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 mark character (such as **\(dg** for the dagger character) in the body text, followed by the text of the footnote enclosed by **FS \(dg** and **FE** macros.

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

0 Prints the footnote number as a superscript; indents the footnote (default).

1 Prints the number followed by a period (that is, "1.") and indents the footnote.

2 Like 1, without an indent.

3 Like 1, but prints the footnote number as a paragraph with a hanging indent.

You can use footnotes safely within keeps and displays, but avoid using numbered footnotes within floating keeps. You can set a second ****** between a ****** and its corresponding **.FS**; as long as each **.FS** occurs *after* the corresponding ****** and the occurrences of **.FS** are in the same order as the corresponding occurrences of ******.

Headers and footers

There are three ways to define headers and footers:

- Use the strings **LH**, **CH**, and **RH** to set the left, center, and right headers. Use **LF**, **CF**, and **RF** to set the left, center, and right footers. The string-setting approach 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:

  .*XX* '*left*'*center*'*right*'

  where *XX* is one of the foregoing four macros and each of *left*, *center*, and *right* is text of your choice. You can replace the quote (') marks with any character not appearing in the header or footer text.
- You can redefine the **PT** and **BT** macros to change the behavior of the header and footer, respectively. The header process also calls the (undefined) **HD** macro after **PT**; you can define this macro if you need additional processing after printing the header (for example, to draw a line below the header).

Margins

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

Reg.	Definition	Effective	Default
PO	Page offset (left margin)	next page	1i
LL	Line length	next paragraph	6i
LT	Header/footer length	next paragraph	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.

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.

.1C Single-column mode.

.2C Two-column mode.

.MC [*column-width* [*gutter-width*]]

Multi-column mode. If you specify no arguments, it is equivalent to the **2C** macro. Otherwise, *column-width* is the width of each column and *gutter-width* is the space between columns. The **MINGW** number register is the default gutter width.

Creating a table of contents

Wrap text that you want to appear in the table of contents in **XS** and **XE** macros. Use the **TC** macro to print the table of contents at the end of the document, resetting the page number to **i** (Roman numeral 1).

You can manually create a table of contents by specifying a page number as the first argument to **XS**. Add subsequent entries using the **XA** macro. For example:

```
.XS 1
Introduction
.XA 2
A Brief History of the Universe
.XA 729
Details of Galactic Formation
. . .
.XE
```

Use the **PX** macro to print a manually-generated table of contents without resetting the page number.

If you give the argument **no** to either **PX** or **TC**, *groff* suppresses printing the title specified by the ***[TOC]** string.

Fractional point sizes

Traditionally, the *ms* macros only support integer values for the document's font size and vertical spacing. To overcome this restriction, values larger than or equal to 1000 are taken as fractional values, multiplied by 1000. For example, '.nr PS 10250' sets the font size to 10.25 points.

The following four registers accept fractional point sizes: **PS**, **VS**, **FPS**, and **FVS**.

Due to backwards compatibility, the value of **VS** must be smaller than 40000 (this is 40.0 points).

DIFFERENCES FROM troff ms

The *groff ms* macros are a complete re-implementation, using no original AT&T code. Since they take advantage of the extended features in *groff*, they cannot be used with AT&T *troff*. Other differences include:

- The internals of *groff ms* differ from the internals of Unix *ms*. Documents that depend upon implementation details of Unix *ms* may not format properly with *groff ms*.
- The error-handling policy of *groff ms* is to detect and report errors, rather than silently to ignore them.
- Some Bell Labs localisms are not implemented by default. However, if you call the otherwise undocumented **SC** section-header macro, you will enable implementations of three other archaic Bell Labs macros: **UC**, **P1**, and **P2**. These are not enabled by default because (a) they were not documented, in the original *ms manual*, and (b) the **P1** and **UC** macros both collide with different macros in the Berkeley version of *ms*.

  These emulations are sufficient to give back the 1976 Kernighan & Cherry paper *Typesetting Mathematics – User's Guide* its section headings, and restore some text that had gone missing as arguments of undefined macros. No warranty express or implied is given as to how well the typographic details these produce match the original Bell Labs macros.
- Berkeley localisms, in particular the **TM** and **CT** macros, are not implemented.
- *Groff ms* does not work in compatibility mode (e.g., with the **−C** option).
- There is no support for typewriter-like devices.
- *Groff ms* does not provide cut marks.

- Multiple line spacing is not supported (use a larger vertical spacing instead).
- Some Unix *ms* documentation says that the **CW** and **GW** number registers can be used to control the column width and gutter width, respectively. These number registers are not used in *groff ms*.
- Macros that cause a reset (paragraphs, headings, etc.) may change the indent. Macros that change the indent do not increment or decrement the indent, but rather set it absolutely. This can cause problems for documents that define additional macros of their own. The solution is to use not the **in** request but instead the **RS** and **RE** macros.
- The number register **GS** is set to 1 by the *groff ms* macros, but is not used by the Unix *ms* macros. Documents that need to determine whether they are being formatted with Unix *ms* or *groff ms* should use this number register.
- To make *groff ms* use the default page offset (which also specifies the left margin), the **PO** number register must stay undefined until the first **ms** macro is evaluated. This implies that **PO** should not be used early in the document, unless it is changed also: remember that accessing an undefined register automatically defines it.

Strings

You can redefine the following strings to adapt the *groff ms* macros to languages other than English:

String	Default Value
REFERENCES	References
ABSTRACT	ABSTRACT
TOC	Table of Contents
MONTH1	January
MONTH2	February
MONTH3	March
MONTH4	April
MONTH5	May
MONTH6	June
MONTH7	July
MONTH8	August
MONTH9	September
MONTH10	October
MONTH11	November
MONTH12	December

The ***–** string produces an em dash—like this.

Use ***Q** and ***U** to get a left and right typographer's quote, respectively, in *troff* (and plain quotes in *nroff*).

Text Settings

The **FAM** string sets the default font family. If this string is undefined at initialization, it is set to Times.

The point size, vertical spacing, and inter-paragraph spacing for footnotes are controlled by the number registers **FPS**, **FVS**, and **FPD**; at initialization these are set to **\n(PS–2**, **\n[FPS]+2**, and **\n(PD/2**, respectively. If any of these registers are defined before initialization, the initialization macro does not change them.

The hyphenation flags (as set by the **hy** request) are set from the **HY** register; the default is 6.

Improved accent marks (as originally defined in Berkeley's *ms* version) are available by specifying the **AM** macro at the beginning of your document. You can place an accent over most characters by specifying the string defining the accent directly after the character. For example, **n*~** produces an n with a tilde over it.

NAMING CONVENTIONS

The following conventions are used for names of macros, strings, and number registers. External names available to documents that use the *groff ms* macros contain only uppercase letters and digits.

Internally the macros are divided into modules; naming conventions are as follows:

- Names used only within one module are of the form *module***name*.
- Names used outside the module in which they are defined are of the form *module*@*name*.
- Names associated with a particular environment are of the form *environment*:*name*; these are used only within the **par** module.
- *name* does not have a module prefix.
- Constructed names used to implement arrays are of the form *array*!*index*.

Thus the groff ms macros reserve the following names:

- Names containing the characters *, @, and :.
- Names containing only uppercase letters and digits.

FILES

/usr/share/groff/1.22.4/tmac/ms.tmac (a wrapper file for *s.tmac*)
/usr/share/groff/1.22.4/tmac/s.tmac

AUTHORS

The GNU version of the *ms* macro package was written by James Clark and contributors. This document was (re-)written by Larry Kollar ⟨lkollar@despammed.com⟩.

SEE ALSO

groff(1), **troff**(1), **tbl**(1), **pic**(1), **eqn**(1), **refer**(1)

Groff: The GNU Implementation of troff , by Trent A. Fisher and Werner Lemberg