Name

groff_ms – GNU *roff* manuscript macro package for formatting documents

Synopsis

groff –ms [*option* ...] [*file* ...]
groff –m ms [*option* ...] [*file* ...]

Description

The GNU implementation of the *ms* macro package is part of the *groff* document formatting system. The *ms* package is suitable for the composition of letters, memoranda, reports, and books.

These *groff* macros support cover page and table of contents generation, automatically numbered headings, several paragraph styles, a variety of text styling options, footnotes, and multi-column page layouts. *ms* supports the *tbl*(1), *eqn*(1), *pic*(1), and *refer*(1) preprocessors for inclusion of tables, mathematical equations, diagrams, and standardized bibliographic citations.

This implementation is mostly compatible with the documented interface and behavior of AT&T Unix Version 7 *ms*. Many extensions from 4.2BSD (Berkeley) and Tenth Edition Research Unix have been recreated.

Usage

The *ms* macro package expects a certain amount of structure: a well-formed document contains at least one paragraphing or heading macro call. To compose a simple document from scratch, begin it by calling **.LP** or **.PP**. Longer documents have a structure as follows.

Document type

Calling the **RP** macro at the beginning of your document puts the document description (see below) on a cover page. Otherwise, *ms* places this information on the first page, followed immediately by the body text. Some document types found in other *ms* implementations are specific to AT&T or Berkeley, and are not supported in *groff ms*.

Format and layout

By setting registers and strings, you can configure your document's typeface, margins, spacing, headers and footers, and footnote arrangement. See subsection "Document control settings" below.

Document description

A document description consists of any of: a title, one or more authors' names and affiliated institutions, an abstract, and a date or other identifier. See subsection "Document description macros" below.

Body text

The main matter of your document follows its description (if any). *ms* supports highly structured text consisting of paragraphs interspersed with multi-level headings (chapters, sections, subsections, and so forth) and augmented by lists, footnotes, tables, diagrams, and similar material. The preponderance of subsections below covers these matters.

Table of contents

Macros enable the collection of entries for a table of contents (or index) as the material they discuss appears in the document. You then call a macro to emit the table of contents at the end of your document. The table of contents must necessarily follow the rest of the text since GNU *troff* is a single-pass formatter; it thus cannot determine the page number of a division of the text until it has been set and output. Since *ms* output was designed for the production of hard copy, the traditional procedure was to manually relocate the pages containing the table of contents between the cover page and the body text. Today, page resequencing is more often done in the digital domain. An index works similarly, but because it typically needs to be sorted after collection, its preparation requires separate processing.

Document control settings

The following tables list the document control registers, strings, and special characters. For any parameter whose default is unsatisfactory, define it before calling any *ms* macro other than **RP**.

Margin settings

Parameter	Definition	Effective	Default
\n[PO]	Page offset (left margin)	next page	1i (0)
\n[LL]	Line length	next paragraph	6.5i (65n)
\n[LT]	Title line length	next paragraph	6.5i (65n)
\n[HM]	Top (header) margin	next page	1i
\n[FM]	Bottom (footer) margin	next page	1i

Titles (headers, footers)

Parameter	Definition	Effective	Default
*[LH]	Left header text	next header	*empty*
*[CH]	Center header text	next header	−\n[%]−
*[RH]	Right header text	next header	*empty*
*[LF]	Left footer text	next footer	*empty*
*[CF]	Center footer text	next footer	*empty*
*[RF]	Right footer text	next footer	*empty*

Text settings

Parameter	Definition	Effective	Default
\n[PS]	Point size	next paragraph	10p
\n[VS]	Vertical spacing (leading)	next paragraph	12p
\n[HY]	Hyphenation mode	next paragraph	6
*[FAM]	Font family	next paragraph	T

Paragraph settings

Parameter	Definition	Effective	Default
\n[PI]	Indentation	next paragraph	5n
\n[PD]	Paragraph distance (spacing)	next paragraph	0.3v (1v)
\n[QI]	Quotation indentation	next paragraph	5n
\n[PORPHANS]	# of initial lines kept	next paragraph	1

Heading settings

Parameter	Definition	Effective	Default
\n[PSINCR]	Point size increment	next heading	1p
\n[GROWPS]	Size increase depth limit	next heading	0
\n[HORPHANS]	# of following lines kept	next heading	1
*[SN−STYLE]	Numbering style (alias)	next heading	*[SN−DOT]

***[SN−STYLE]** can alternatively be made an alias of ***[SN−NO−DOT]** with the **als** request.

Footnote settings

Parameter	Definition	Effective	Default
\n[FI]	Indentation	next footnote	2n
\n[FF]	Format	next footnote	0
\n[FPS]	Point size	next footnote	\n[PS]−2p
\n[FVS]	Vertical spacing (leading)	next footnote	\n[FPS]+2p
\n[FPD]	Paragraph distance (spacing)	next footnote	\n[PD]/2
*[FR]	Line length ratio	*special*	11/12

Display settings

Parameter	Definition	Effective	Default
\n[DD]	Display distance (spacing)	*special*	0.5v (1v)
\n[DI]	Display indentation	*special*	0.5i

Other settings

Parameter	**Definition**	**Effective**	**Default**
`\n[MINGW]`	Minimum gutter width	next page	`2n`
`\n[TC-MARGIN]`	TOC page number margin width	next **PX** call	`\w'000'`
`\[TC-LEADER]`	TOC leader character	next **PX** call	`.\h'1m'`

For entries marked "*special*" in the "Effective" column, see the discussion in the applicable section below. The **PO**, **LL**, and **LT** register defaults vary by output device and paper format; the values shown are for typesetters using U.S. letter paper, and then terminals. See section "Paper format" of *groff*(1). The **PD** and **DD** registers use the larger value if the vertical motion quantum of the output device is too coarse for the smaller one; usually, this is the case only for output to terminals and emulators thereof. The "gutter" affected by **\n[MINGW]** is the gap between columns in multiple-column page arrangements. The **TC−MARGIN** register and **TC−LEADER** special character affect the formatting of tables of contents assembled by the **XS**, **XA**, and **XE** macros.

Document description macros

Define information describing the document by calling the macros below in the order shown; **.DA** or **.ND** can be called to set the document date (or other identifier) at any time before (a) the abstract, if present, or (b) its information is required in a header or footer. Use of these macros is optional, except that **.TL** is mandatory if any of **.RP**, **.AU**, **.AI**, or **.AB** is called, and **.AE** is mandatory if **.AB** is called.

.RP [**no−repeat−info**] [**no−renumber**]

Use the "report" (AT&T: "released paper") format for your document, creating a separate cover page. The default arrangement is to place most of the document description (title, author names and institutions, and abstract, but not the date) at the top of the first page. If the optional **no−repeat−info** argument is given, *ms* produces a cover page but does not repeat any of its information on subsequently (but see the **DA** macro below regarding the date). Normally, **.RP** sets the page number following the cover page to 1. Specifying the optional **no−renumber** argument suppresses this alteration. Optional arguments can occur in any order. "**no**" is recognized as a synonym of **no−repeat−info** for AT&T compatibility.

.TL Specify the document title. *ms* collects text on input lines following this call into the title until reaching **.AU**, **.AB**, or a heading or paragraphing macro call.

.AU Specify an author's name. *ms* collects text on input lines following this call into the author's name until reaching **.AI**, **.AB**, another **.AU**, or a heading or paragraphing macro call. Call it repeatedly to specify multiple authors.

.AI Specify the preceding author's institution. An **.AU** call is usefully followed by at most one **.AI** call; if there are more, the last **.AI** call controls. *ms* collects text on input lines following this call into the author's institution until reaching **.AU**, **.AB**, or a heading or paragraphing macro call.

.DA [*x* ...]

Typeset the current date, or any arguments *x*, in the center footer, and, if **.RP** is also called, left-aligned at the end of the document description on the cover page.

.ND [*x* ...]

Typeset the current date, or any arguments *x*, if **.RP** is also called, left-aligned at the end of the document description on the cover page. This is *groff ms*'s default.

.AB [**no**]

Begin the abstract. *ms* collects text on input lines following this call into the abstract until reaching an **.AE** call. By default, *ms* places the word "ABSTRACT" centered and in italics above the text of the abstract. The optional argument "**no**" suppresses this heading.

.AE End the abstract.

Text settings

The **FAM** string, a GNU extension, sets the font family for body text; the default is "**T**". The **PS** and **VS** registers set the type size and vertical spacing (distance between text baselines), respectively. The font family and type size are ignored on terminal devices. Setting these parameters before the first call of a heading, paragraphing, or (non-date) document description macro also applies them to headers, footers, and (for **FAM**) footnotes.

The **HY** register defines the automatic hyphenation mode used with the **hy** request. Setting **\n[HY]** to **0** is equivalent to using the **nh** request. This is a Tenth Edition Research Unix extension.

Typographical symbols

ms provides a few strings to obtain typographical symbols not easily entered with the keyboard. These and many others are available as special character escape sequences—see *groff_char*(7).

***[−]** Interpolate an em dash.

***[Q]**
***[U]** Interpolate typographer's quotation marks where available, and neutral double quotes otherwise. ***[Q]** is the left quote and ***[U]** the right.

Paragraphs

Paragraphing macros *break*, or terminate, any pending output line so that a new paragraph can begin. Several paragraph types are available, differing in how indentation applies to them: to left, right, or both margins; to the first output line of the paragraph, all output lines, or all but the first. All paragraphing macro calls cause the insertion of vertical space in the amount stored in the **PD** register, except at page or column breaks, or adjacent to displays.

The **PORPHANS** register defines the minimum number of initial lines of any paragraph that must be kept together to avoid isolated lines at the bottom of a page. If a new paragraph is started close to the bottom of a page, and there is insufficient space to accommodate **\n[PORPHANS]** lines before an automatic page break, then a page break is forced before the start of the paragraph. This is a GNU extension.

.LP Set a paragraph without any (additional) indentation.

.PP Set a paragraph with a first-line left indentation in the amount stored in the **PI** register.

.IP [*marker* [*width*]]
Set a paragraph with a left indentation. The optional *marker* is not indented and is empty by default. *width* overrides the indentation amount in **\n[PI]**; its default unit is "**n**". Once specified, *width* applies to further **.IP** calls until specified again or a heading or different paragraphing macro is called.

.QP Set a paragraph indented from both left and right margins by **\n[QI]**.

.QS
.QE Begin (**QS**) and end (**QE**) a region where each paragraph is indented from both margins by **\n[QI]**. The text between **.QS** and **.QE** can be structured further by use of other paragraphing macros.

.XP Set an "exdented" paragraph—one with a left indentation of **\n[PI]** on every line *except* the first (also known as a hanging indent). This is a Berkeley extension.

Headings

Use headings to create a hierarchical structure for your document. The *ms* macros print headings in **bold** using the same font family and, by default, type size as the body text. Headings are available with and without automatic numbering. Text on input lines following the macro call becomes the heading's title. Call a paragraphing macro to end the heading text and start the section's content.

.NH [*depth*]
Set an automatically numbered heading. *ms* produces a numbered heading in the form *a.b.c*..., to any level desired, with the numbering of each depth increasing automatically and being reset to zero when a more significant depth is increased. "**1**" is the most significant or coarsest division of the document. Only non-zero values are output. If *depth* is omitted, it is taken to be **1**. If you specify *depth* such that an ascending gap occurs relative to the previous **NH** call—that is, you "skip a depth", as by "**.NH 1**" and then "**.NH 3**", *groff ms* emits a warning on the standard error stream.

.NH S *heading-depth-index . . .*
Alternatively, you can give **NH** a first argument of "**S**", followed by integers to number the heading depths explicitly. Further automatic numbering, if used, resumes using the specified indices as their predecessors. This feature is a Berkeley extension.

After **.NH** is called, the assigned number is made available in the strings **SN−DOT** (as it appears in a printed heading with default formatting, followed by a terminating period) and **SN−NO−DOT** (with the terminating period omitted). These are GNU extensions.

You can control the style used to print numbered headings by defining an appropriate alias for the string **SN−STYLE**. By default, ***[SN−STYLE]** is aliased to ***[SN−DOT]**. If you prefer to omit the terminating period from numbers appearing in numbered headings, you may alias it to ***[SN−NO−DOT]**. Any such change in numbering style becomes effective from the next use of **.NH** following redefinition of the alias for ***[SN−STYLE]**. The formatted number of the current heading is available in ***[SN]** (a feature first documented by Berkeley); this string facilitates its inclusion in, for example, table captions, equation labels, and **.XS**/**.XA**/**.XE** table of contents entries.

.SH [*depth*]
Set an unnumbered heading. The optional *depth* argument is a GNU extension indicating the heading depth corresponding to the *depth* argument of **.NH**. It matches the type size at which the heading is set to that of a numbered heading at the same depth when the **\n[GROWPS]** and **\n[PSINCR]** heading size adjustment mechanism is in effect.

The **PSINCR** register defines an increment in type size to be applied to a heading at a lesser depth than that specified in **\n[GROWPS]**. The value of **\n[PSINCR]** should be specified in points with the "**p**" scaling unit and may include a fractional component.

The **GROWPS** register defines the heading depth above which the type size increment set by **\n[PSINCR]** becomes effective. For each heading depth less than the value of **\n[GROWPS]**, the type size is increased by **\n[PSINCR]**. Setting **\n[GROWPS]** to a value less than 2 disables the incremental heading size feature.

In other words, if the value of **GROWPS** register is greater than the *depth* argument to a **.NH** or **.SH** call, the type size of a heading produced by these macros increases by **\n[PSINCR]** units over **\n[PS]** multiplied by the difference of **\n[GROWPS]** and *depth*.

The **\n[HORPHANS]** register operates in conjunction with the **NH** and **SH** macros to inhibit the printing of isolated headings at the bottom of a page; it specifies the minimum number of lines of the subsequent paragraph that 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, a page break is forced before the heading is printed. Any display macro call or *tbl*, *pic*, or *eqn* region between the heading and the subsequent paragraph suppresses this grouping.

Typeface and decoration

The *ms* macros provide a variety of ways to style text. Attend closely to the ordering of arguments labeled *pre* and *post,* which is not intuitive. Support for *pre* arguments is a GNU extension.

.B [*text* [*post* [*pre*]]]
Style *text* in bold, followed by *post* in the previous font style without intervening space, and preceded by *pre* similarly. Without arguments, *ms* styles subsequent text in bold until the next paragraphing, heading, or no-argument typeface macro call.

.R [*text* [*post* [*pre*]]]
As **.B**, but use the roman style (upright text of normal weight) instead of bold. Argument recognition is a GNU extension.

.I [*text* [*post* [*pre*]]]
As **.B**, but use an italic or oblique style instead of bold.

.BI [*text* [*post* [*pre*]]]
As **.B**, but use a bold italic or bold oblique style instead of upright bold. This is a Tenth Edition Research Unix extension.

.CW [*text* [*post* [*pre*]]]
As **.B**, but use a constant-width (monospaced) roman typeface instead of bold. This is a Tenth Edition Research Unix extension.

.BX [*text*]
Typeset *text* and draw a box around it. On terminal devices, reverse video is used instead. If you want *text* to contain space, use unbreakable space or horizontal motion escape sequences (**\~**, *****space*, **\^**, **\|**, **\0**, or **\h**).

.UL [*text* [*post*]]
Typeset *text* with an underline. *post,* if present, is set after *text* with no intervening space.

.LG Set subsequent text in larger type (2 points larger than the current size) until the next type size, paragraphing, or heading macro call. You can specify this macro multiple times to enlarge the type size as needed.

.SM Set subsequent text in smaller type (2 points smaller than the current size) until the next type size, paragraphing, or heading macro call. You can specify this macro multiple times to reduce the type size as needed.

.NL Set subsequent text at the normal type size (**\n[PS]**).

When *pre* is used, a hyphenation control escape sequence **\%** that would ordinarily start *text* must start *pre* instead.

groff ms also offers strings to begin and end super- and subscripting. These are GNU extensions.

***{**
***}** Begin and end superscripting, respectively.

***<**
***>** Begin and end subscripting, respectively.

Indented regions

You may need to indent a region of text while otherwise formatting it normally. Indented regions can be nested.

.RS Begin a region where headings, paragraphs, and displays are indented (further) by **\n[PI]**.

.RE End the (next) most recent indented region.

Keeps, boxed keeps, and displays

On occasion, you may want to *keep* several lines of text, or a region of a document, together on a single page, preventing an automatic page break within certain boundaries. This can cause a page break to occur earlier than it normally would.

You can alternatively specify a *floating keep:* if a keep cannot fit on the current page, *ms* holds its contents and allows text following the keep (in the source document) to fill in the remainder of the current page. When the page breaks, whether by reaching the end or **bp** request, *ms* puts the floating keep at the beginning of the next page.

.KS Begin a keep.

.KF Begin a floating keep.

.KE End (floating) keep.

As an alternative to the keep mechanism, the **ne** request forces a page break if there is not at least the amount of vertical space specified in its argument remaining on the page.

A *boxed keep* has a frame drawn around it.

.B1 Begin a keep with a box drawn around it.

.B2 End boxed keep.

Boxed keep macros cause breaks; if you need to box a word or phrase within a line, see the **BX** macro in section “Highlighting” above. Box lines are drawn as close as possible to the text they enclose so that they are usable within paragraphs. If you wish to place one or more paragraphs in a boxed keep, you may improve their appearance by calling **.B1** after the first paragraphing macro, and by adding a small amount of vertical space before calling **.B2**.

If you want a boxed keep to float, you will need to enclose the **.B1** and **.B2** calls within a pair of **.KF** and **.KE** calls.

Displays turn off filling; lines of verse or program code are shown with their lines broken as in the source document without requiring **br** requests between lines. Displays can be kept on a single page or allowed to break across pages. The **DS** macro begins a kept display of the layout specified in its first argument; non-kept displays are begun with dedicated macros corresponding to their layout.

.DS L
.LD Begin (**DS**: kept) left-aligned display.

.DS [**I** [*indent*]]
.ID [*indent*]
Begin (**DS**: kept) display indented by *indent* if specified, **\n[DI]** otherwise.

.DS B
.BD Begin (**DS**: kept) block display: the entire display is left-aligned, but indented such that the longest line in the display is centered on the page.

.DS C
.CD Begin (**DS**: kept) centered display: each line in the display is centered.

.DS R
.RD Begin (**DS**: kept) right-aligned display. This is a GNU extension.

.DE End any display.

The distance stored in **\n[DD]** is inserted before and after each pair of display macros; this is a Berkeley extension. In *groff ms*, this distance replaces any adjacent inter-paragraph distance or subsequent spacing prior to a section heading. The **DI** register is a GNU extension; its value is an indentation applied to displays created with **.DS** and **.ID** without arguments, to "**.DS I**" without an indentation argument, and to equations set with "**.EQ I**". Changes to either register take effect at the next display boundary.

Tables, figures, equations, and references

The *ms* package is often used with the *tbl*, *pic*, *eqn*, and *refer* preprocessors. The **\n[DD]** distance is also applied to regions of the document preprocessed with *eqn*, *pic*, and *tbl*. Mark text meant for preprocessors by enclosing it in pairs of tokens as follows, with nothing between the dot and the macro name. The preprocessors match these tokens only at the start of an input line.

.TS [**H**]
.TE Demarcate a table to be processed by the *tbl* preprocessor. The optional **H** argument instructs *ms* to repeat table rows (often column headings) at the top of each new page the table spans, if applicable; calling the **TH** macro marks the end of such rows. *tbl*(1) provides a comprehensive reference to the preprocessor and offers examples of its use.

.PS
.PE
.PF **.PS** begins a picture to be processed by the *pic* preprocessor; either of **.PE** or **.PF** ends it, the latter with "flyback" to the vertical position at its top.

.EQ [*align* [*label*]]
.EN Demarcate an equation to be processed by the *eqn* preprocessor. The equation is centered by default; *align* can be **C**, **L**, or **I** to (explicitly) center, left-align, or indent it by **\n[DI]**, respectively. If specified, *label* is set right-aligned.

.[
.] Demarcate a bibliographic citation to be processed by the *refer* preprocessor. *refer*(1) provides a comprehensive reference to the preprocessor and the format of its bibliographic database.

When *refer* emits collected references (as might be done on a "Works Cited" page), it interpolates the string ***[REFERENCES]** as an unnumbered heading (**.SH**).

Attempting to place a multi-page table inside a keep can lead to unpleasant results, particularly if the *tbl* "**allbox**" option is used.

Footnotes

A footnote is typically anchored to a place in the text with a *marker,* which is a small integer, a symbol, or arbitrary user-specified text.

\** Place an *automatic number,* an automatically generated numeric footnote marker, in the text. Each time this string is interpolated, the number it produces increments by one. Automatic numbers start at 1. This is a Berkeley extension.

Enclose the footnote text in **FS** and **FE** macro calls to set it at the nearest available "foot", or bottom, of a text column or page.

.FS [*marker*]
Begin a footnote. The **.FS−MARK** hook (see below) is called with any supplied *marker* argument, which is then also placed at the beginning of the footnote text. If *marker* is omitted, the next pending automatic number enqueued by interpolation of the ***** string is used, and if none exists, nothing is prefixed.

.FE End footnote text.

groff ms provides a hook macro, **FS−MARK**, for user-determined operations to be performed when the **FS** macro is called. It is passed the same arguments as **.FS** itself. By default, this macro has an empty definition. **.FS−MARK** is a GNU extension.

Footnote text is formatted as paragraphs are, using analogous parameters. The registers **FI**, **FPD**, **FPS**, and **FVS** correspond to **PI**, **PD**, **PS**, and **VS**, respectively; **FPD**, **FPS**, and **FVS** are GNU extensions.

The **FF** register controls the formatting of automatically numbered footnote paragraphs, and those for which **.FS** is given a *marker* argument, at the bottom of a column or page as follows.

0 Set an automatic number, or a specified **FS** *marker* argument, as a superscript (on typesetter devices) or surrounded by square brackets (on terminals). The footnote paragraph is indented as with **.PP** if there is an **.FS** argument or an automatic number, and as with **.LP** otherwise. This is the default.

1 As **0**, but set the marker as regular text, and follow an automatic number with a period.

2 As **1**, but without indentation (like **.LP**).

3 As **1**, but set the footnote paragraph with the marker hanging (like **.IP**).

Language and localization

groff ms provides several strings that you can customize for your own purposes, or redefine to adapt the macro package to languages other than English. It is already localized for Czech, German, French, Italian, and Swedish. Load the desired localization macro package after *ms*; see *groff_tmac*(5).

String	Default
`\*[REFERENCES]`	`References`
`\*[ABSTRACT]`	`\f[I]ABSTRACT\f[]`
`\*[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 default for **ABSTRACT** includes font selection escape sequences to set the word in italics.

Headers and footers

There are multiple ways to produce headers and footers. One is to define the strings **LH**, **CH**, and **RH** to set the left, center, and right headers, respectively; and **LF**, **CF**, and **RF** to set the left, center, and right footers. This approach suffices for documents that do not distinguish odd- and even-numbered pages.

Another method is to call macros that set headers or footers for odd- or even-numbered pages. Each such macro takes a delimited argument separating the left, center, and right header or footer texts from each other. You can replace the neutral apostrophes (') shown below with any character not appearing in the header or footer text. These macros are Berkeley extensions.

.OH '*left*'*center*'*right*'
.OF '*left*'*center*'*right*'
.EH '*left*'*center*'*right*'
.EF '*left*'*center*'*right*'

The **OH** and **EH** macros define headers for odd- (recto) and even-numbered (verso) pages, respectively; the **OF** and **EF** macros define footers for them.

With either method, a percent sign **%** in header or footer text is replaced by the current page number. By default, *ms* places no header on a page numbered "1" (regardless of its number format).

.P1 Typeset the header even on page 1. To be effective, this macro must be called before the header trap is sprung on any page numbered "1". This is a Berkeley extension.

For even greater flexibility, *ms* permits redefinition of the macros called when the page header and footer traps are sprung. **PT** ("page trap") is called by *ms* when the header is to be written, and **BT** ("bottom trap") when the footer is to be. The *groff* page location trap that *ms* sets up to format the header also calls the (normally undefined) **HD** macro after **.PT**; you can define **.HD** if you need additional processing after setting the header. The **HD** hook is a Berkeley extension. Any such macros you (re)define must implement any desired specialization for odd-, even-, or first numbered pages.

Tab stops

Use the **ta** request to set tab stops as needed.

.TA Reset the tab stops to the *ms* default (every 5 ens). Redefine this macro to create a different set of default tab stops.

Margins

Control margins using the registers summarized in the "Margins" portion of the table in section "Document control settings" above. There is no setting for the right margin; the combination of page offset **\n[PO]** and line length **\n[LL]** determines it.

Multiple columns

ms can set text in as many columns as reasonably fit on the page. The following macros force a page break if a multi-column layout is active when they are called. **\n[MINGW]** is the default minimum gutter width; it is a GNU extension. When multiple columns are in use, keeps and the **HORPHANS** and **PORPHANS** registers work with respect to column breaks instead of page breaks.

.1C Arrange page text in a single column (the default).

.2C Arrange page text in two columns.

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

Arrange page text in multiple columns. 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 minimum distance between columns.

Creating a table of contents

Define an entry to appear in the table of contents by bracketing its text between calls to the **XS** and **XE** macros. A typical application is to call them immediately after **NH** or **SH** and repeat the heading text within them. The **XA** macro, used within **.XS/.XE** pairs, supplements an entry—for instance, when it requires multiple output lines, whether because a heading is too long to fit or because style dictates that page numbers not be repeated. You may wish to indent the text thus wrapped to correspond to its heading depth; this can be done in the entry text by prefixing it with tabs or horizontal motion escape sequences, or by providing a second argument to the **XA** macro. **.XS** and **.XA** automatically associate the page number where they are called with the text following them, but they accept arguments to override this behavior. At the end of the document, call **TC** or **PX** to emit the table of contents; **.TC** resets the page number to **i** (Roman numeral one), and then calls **PX**. All of these macros are Berkeley extensions.

.XS [*page-number*]
.XA [*page-number* [*indentation*]]
.XE Begin, supplement, and end a table of contents entry. Each entry is associated with *page-number* (otherwise the current page number); a *page-number* of "**no**" prevents a leader and page number from being emitted for that entry. Use of **.XA** within **.XS/.XE** is optional; it can be repeated. If *indentation* is present, a supplemental entry is indented by that amount; ens are assumed if no unit is indicated. Text on input lines between **.XS** and **.XE** is stored for later recall by **.PX**.

.PX [**no**]
Switch to single-column layout. Unless "**no**" is specified, center and interpolate ***[TOC]** in bold and two points larger than the body text. Emit the table of contents entries.

.TC [**no**]
Set the page number to 1, the page number format to lowercase Roman numerals, and call **PX** (with a "**no**" argument, if present).

The remaining features in this subsection are GNU extensions. *groff ms* obviates the need to repeat heading text after **.XS** calls. Call **.XN** and **.XH** after **.NH** and **.SH**, respectively. Text to be appended to the formatted section heading, but not to appear in the table of contents entry, can follow these calls.

.XN *heading-text*
Format *heading-text* and create a corresponding table of contents entry; the indentation is computed from the *depth* argument of the preceding **NH** call.

.XH *depth heading-text*
As **.XN**, but use *depth* to determine the indentation.

groff ms encourages customization of table of contents entry production. (Re-)define any of the following macros as desired.

.XN−REPLACEMENT *heading-text*
.XH−REPLACEMENT *depth heading-text*
These hook macros implement **.XN** and **.XH**, and call **XN−INIT** and **XH−INIT**, respectively, then call **XH−UPDATE−TOC** with the arguments given them.

.XH−INIT
.XN−INIT
These hook macros do nothing by default.

.XH−UPDATE−TOC *depth heading-text*
Bracket *heading-text* with **XS** and **XE** calls, indenting it by 2 ens per level of *depth* beyond the first.

You can customize the style of the leader that bridges each table of contents entry with its page number; define the **TC−LEADER** special character by using the **char** request. A typical leader combines the dot glyph "**.**" with a horizontal motion escape sequence to spread the dots. The width of the page number field is stored in the **TC−MARGIN** register.

Differences from AT&T *ms*

The *groff ms* macros are an independent reimplementation, using no AT&T code. Since they take advantage of the extended features of *groff*, they cannot be used with AT&T *troff*. *groff ms* supports features described above as Berkeley and Tenth Edition Research Unix extensions, and adds several of its own.

- The internals of *groff ms* differ from the internals of AT&T *ms*. Documents that depend upon implementation details of AT&T *ms* may not format properly with *groff ms*. Such details include macros whose function was not documented in the AT&T *ms* manual ("Typing Documents on the UNIX System: Using the –ms Macros with Troff and Nroff", M. E. Lesk, Bell Laboratories, 1978).
- The error-handling policy of *groff ms* is to detect and report errors, rather than to ignore them silently.
- Tenth Edition Research Unix supported **P1/P2** macros to bracket code examples; *groff ms* does not.
- *groff ms* does not work in GNU *troff*'s AT&T compatibility mode. If loaded when that mode is enabled, it aborts processing with a diagnostic message.
- Multiple line spacing is not supported. Use a larger vertical spacing instead.
- *groff ms* uses the same header and footer defaults in both *nroff* and *troff* modes as AT&T *ms* does in *troff* mode; AT&T's default in *nroff* mode is to put the date, in U.S. traditional format (e.g., "January 1, 2021"), in the center footer (the **CF** string).
- Many *groff ms* macros, including those for paragraphs, headings, and displays, cause a reset of paragraph rendering parameters, and may change the indentation; they do so not by incrementing or decrementing it, but by setting it absolutely. This can cause problems for documents that define additional macros of their own that try to manipulate indentation. Use **.RS** and **.RE** instead of the **in** request.
- AT&T *ms* interpreted the values of the registers **PS** and **VS** in points, and did not support the use of scaling units with them. *groff ms* interprets values of the registers **PS**, **VS**, **FPS**, and **FVS**, equal to or larger than 1,000 (one thousand) as decimal fractions multiplied by 1,000. (Register values are converted to and stored as basic units. See "Measurements" in the *groff* Texinfo manual or in *groff*(7)). This threshold makes use of a scaling unit with these parameters practical for high-resolution devices while preserving backward compatibility. It also permits expression of non-integral type sizes. For example, "**groff –rPS=10.5p**" at the shell prompt is equivalent to placing "**.nr PS 10.5p**" at the beginning of the document.
- AT&T *ms*'s **AU** macro supported arguments used with some document types; *groff ms* does not.
- Right-aligned displays are available. The AT&T *ms* manual observes that "it is tempting to assume that "**.DS R**" will right adjust lines, but it doesn't work". In *groff ms*, it does.
- To make *groff ms* use the default page offset (which also specifies the left margin), the **PO** register must stay undefined until the first *ms* macro is called. This implies that **\n[PO]** should not be used early in the document, unless it is changed also: accessing an undefined register automatically defines it.
- *groff ms* supports the **PN** register, but it is not necessary; you can access the page number via the usual **%** register and invoke the **af** request to assign a different format to it if desired. (If you redefine the *ms* **PT** macro and desire special treatment of certain page numbers—like "**1**"—you may need to handle a non-Arabic page number format, as *groff ms*'s **.PT** does; see the macro package source. *groff ms* aliases the **PN** register to **%**.)
- The AT&T *ms* manual documents registers **CW** and **GW** as setting the default column width and "intercolumn gap", respectively, and which applied when **.MC** was called with fewer than two arguments. *groff ms* instead treats **.MC** without arguments as synonymous with **.2C**; there is thus no occasion for a default column width register. Further, the **MINGW** register and the second argument to **.MC** specify a *minimum* space between columns, not the fixed gutter width of AT&T *ms*.
- The AT&T *ms* manual did not document the **QI** register; Berkeley and *groff ms* do.
- The register **GS** is set to 1 by the *groff ms* macros, but is not used by the AT&T *ms* package. Documents that need to determine whether they are being formatted with *groff ms* or another implementation should test this register.

Unix Version 7 macros not implemented by *groff ms*

Several macros described in the Unix Version 7 *ms* documentation are unimplemented by *groff ms* because they are specific to the requirements of documents produced internally by Bell Laboratories, some of which also require a glyph for the Bell System logo that *groff* does not support. These macros implemented several document type formats (**EG**, **IM**, **MF**, **MR**, **TM**, **TR**), were meaningful only in conjunction with the use of certain document types (**AT**, **CS**, **CT**, **OK**, **SG**), stored the postal addresses of Bell Labs sites (**HO**, **IH**, **MH**, **PY**, **WH**), or lacked a stable definition over time (**UX**).

Legacy features

groff ms retains some legacy features solely to support formatting of historical documents; contemporary ones should not use them because they can render poorly. See *groff_char*(7) instead.

AT&T *ms* accent mark strings

AT&T *ms* defined accent mark strings as follows.

String	Description
`\*[']`	Apply acute accent to subsequent glyph.
``\*[`]``	Apply grave accent to subsequent glyph.
`\*[:]`	Apply dieresis (umlaut) to subsequent glyph.
`\*[^]`	Apply circumflex accent to subsequent glyph.
`\*[~]`	Apply tilde accent to subsequent glyph.
`\*[C]`	Apply caron to subsequent glyph.
`\*[,]`	Apply cedilla to subsequent glyph.

Berkeley *ms* accent mark and glyph strings

Berkeley *ms* offered an **AM** macro; calling it redefined the AT&T accent mark strings (except for ***C**), applied them to the *preceding* glyph, and defined additional strings, some for spacing glyphs.

.AM Enable alternative accent mark and glyph-producing strings.

String	Description
`\*[']`	Apply acute accent to preceding glyph.
``\*[`]``	Apply grave accent to preceding glyph.
`\*[:]`	Apply dieresis (umlaut) to preceding glyph.
`\*[^]`	Apply circumflex accent to preceding glyph.
`\*[~]`	Apply tilde accent to preceding glyph.
`\*[,]`	Apply cedilla to preceding glyph.
`\*[/]`	Apply stroke (slash) to preceding glyph.
`\*[v]`	Apply caron to preceding glyph.
`\*[_]`	Apply macron to preceding glyph.
`\*[.]`	Apply underdot to preceding glyph.
`\*[o]`	Apply ring accent to preceding glyph.
`\*[?]`	Interpolate inverted question mark.
`\*[!]`	Interpolate inverted exclamation mark.
`\*[8]`	Interpolate small letter sharp s.
`\*[q]`	Interpolate small letter o with hook accent (ogonek).
`\*[3]`	Interpolate small letter yogh.
`\*[d-]`	Interpolate small letter eth.
`\*[D-]`	Interpolate capital letter eth.
`\*[th]`	Interpolate small letter thorn.
`\*[TH]`	Interpolate capital letter thorn.
`\*[ae]`	Interpolate small ae ligature.
`\*[AE]`	Interpolate capital ae ligature.
`\*[oe]`	Interpolate small oe ligature.
`\*[OE]`	Interpolate capital oe ligature.

Naming conventions

The following conventions are used for names of macros, strings, and 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. Conventions for identifier names 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/local/share/groff/1.23.0/tmac/s.tmac
implements the package.

/usr/local/share/groff/1.23.0/tmac/refer−ms.tmac
implements *refer*(1) support for *ms*.

/usr/local/share/groff/1.23.0/tmac/ms.tmac
is a wrapper enabling the package to be loaded with "**groff −m ms**".

Authors

The GNU version of the *ms* macro package was written by James Clark and contributors. This document was written by Clark, Larry Kollar ⟨lkollar@despammed.com⟩, and G. Branden Robinson ⟨g.branden.robinson@gmail.com⟩.

See also

A manual is available in source and rendered form. On your system, it may be compressed and/or available in additional formats.

/usr/local/share/doc/groff−1.23.0/ms.ms
/usr/local/share/doc/groff−1.23.0/ms.ps
"Using *groff* with the *ms* Macro Package"; Larry Kollar and G. Branden Robinson.

/usr/local/share/doc/groff−1.23.0/msboxes.ms
/usr/local/share/doc/groff−1.23.0/msboxes.pdf
"Using PDF boxes with *groff* and the *ms* macros"; Deri James. **BOXSTART** and **BOXSTOP** macros are available via the *sboxes* extension package, enabling colored, bordered boxes when the **pdf** output device is used.

Groff: The GNU Implementation of troff, by Trent A. Fisher and Werner Lemberg, is the primary *groff* manual. You can browse it interactively with "info groff".

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