

January 1986

# **UNIX\* Word Processing at HAO**

## **(For Use with the 4.2 BSD Version and -me Macro Package)**

Nancy Dawson

---

HIGH ALTITUDE OBSERVATORY

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH  
BOULDER, COLORADO

**\*UNIX is a trademark of Bell Laboratories.**

# Table of Contents

- I: Introduction 1
  - How to Use This Manual 2
  - Overview of UNIX at HAO 4
  - The Three Categories of Commands 7
  - Getting Started 9
- II: The Word Processing Commands 10
  - Alphabetical List of Commands with Page Number 12
  - What Everyone Should Know about the Word Processing Commands 14
  - Section 1: Spacing, Line Spacing, Break 16
  - Section 2: Paragraphs and Begin Page 18
  - Section 3: Indentations for Left Margin 20
  - Section 4: Indentations from Right Margin 22
  - Section 5: Centering 24
  - Section 6: Type Styles (Fonts) 26
  - Section 7: Underlining 28
  - Section 8: Type Size (Point Size) 30
  - Section 9: Headers and Footers 34
  - Section 10: Quotation Marks 36
  - Section 11: Block Quotes 38
  - Section 12: Footnotes 40
  - Section 13: Line Filling 42
  - Section 14: Lists 44
  - Section 15: Section Headings 46
  - Section 16: Section Headings with Text 48
  - Section 17: Indented Section Headings 50
  - Section 18: Line Length 52
  - Section 19: Two-Columned Text 54
  - Section 20: Two-Columned Lists 56
  - Section 21: Multi-Columned Lists (Mark and Return) 59
  - Section 22: Some Special Characters 61
  - Section 23: Blank Horizontal Spaces 62
  - Section 24: Indented Paragraphs 64
  - Section 25: More Indented Paragraphs 66
  - Section 26: Numbered Paragraphs 68
  - Section 27: More Numbered Paragraphs 70
  - Section 28: Final Section of Numbered Paragraphs 72
  - Section 29: Base Indents 74
  - Section 30: Some Backslash Commands 76
  - Section 31: Right Justification 78
  - Section 32: Page Offset 80
  - Section 33: Page Offset with Line Length 82
  - Section 34: Hyphenation 86
  - Section 35: Title Page 88
  - Section 36: Blank Space at the Top of the Page 90
  - Section 37: Top and Bottom Margins 92
  - Section 38: Block Keeps 95

Section 39: Floating Keeps with Horizontal Lines 100  
Section 40: Creating Your Own Commands (Macros) 104  
Section 41: Vertical Spacing 106  
Section 42: Outlines 108  
Section 43: More Outlines 110  
Section 44: Changing the Page Number 112  
Section 45: Delayed Text 114  
Section 46: Table of Contents 116  
Section 47: Boxing Words 118

III: Sample Letters and Memos 121

IV: Additional Information 125

Tables 126  
Equations 127  
Mathematical and Special Characters 130  
Greek Characters 131  
Some UNIX Commands 133  
Additional Resources 136  
vi Crib Sheet 138  
Acknowledgements 140

Index 141

## **Part 1: Introduction**

How to Use This Manual 2

Overview of UNIX at HAO 4

The Three Categories of Commands 7

Getting Started 9

## **Introduction**

### **Why Another Manual for UNIX Word Processing?**

The instructions for word processing on UNIX are scattered over several manuals, and many of the manuals are difficult to understand because of the technical language. This manual is a compilation of the word processing commands you are likely to need while using UNIX at the High Altitude Observatory of the National Center for Atmospheric Research in Boulder, Colorado.

If you've found other UNIX *documentation* (instruction manuals) a bit terse and confusing, don't despair. The manual you are now reading is written in plain English (as opposed to computer jargon) and is full of examples.

### **Who This Manual Is for**

- New secretaries
- Scientists who want to learn more about word processing
- Students writing their theses
- Visiting scientists who need to produce a few letters or memos
- Experienced word processors (as a reference guide)
- Programmers writing documentation

### **What's Here**

The word processing commands and examples of their printed results make up the major part of this manual, part 2. Part 3 contains sample letters and memos. Part 4 contains some tips on tables and equations, the lists of mathematical and Greek characters, UNIX commands that are frequently used with word processing, a brief bibliography, and a vi crib sheet (which contains frequently used editing commands).

### **What Isn't Here and Where to Find it**

- Complete instructions for equations -- see "Typesetting Mathematics -- User's Guide," by Kernighan and Cherry; and "Typing Mathematical Formulas with UNIX," by Koppel.
- Complete instructions for tables -- see "Tbl -- A Program to Format Tables," by Lesk.
- Editing (vi) commands -- use the "learn vi" program.
- UNIX commands -- see "UNIX Command Notebook For Word Processors," by McCune.

### **How to Use This Manual**

Take the time to read "Things Everyone Should Know About the Word Processing Commands," located at the beginning of part 2. That section contains definitions and information every user should know *before* turning to the individual commands.

Generally speaking, "easier" (or at least "more frequently used") commands are in sections 1-30 of part 2. Less frequently used commands are in sections 31-47 of part 2.

Each command is explained on a left-hand page, with cross references to related sections at the bottom of the page. An example of each command is on the right-hand page that faces the explanation. The notation for the command (.sp, for example) is in large, bold

type at the upper corner of the right-hand page to help you find the example you're looking for as you flip through the pages. (There are a few exceptions to this format because of longer explanations or examples.)

Special terms used with UNIX and computers are defined the first time they appear in the text. Use the index to locate those definitions when you need them in a later section.

**If you are a beginner and want to learn UNIX word processing,**

- Read all of part 1 and "What Everyone Should Know about the Word Processing Commands" in part 2.
- Create a file where you can test word processing commands as you learn them. (Creating a file is covered in part 1 under "Getting Started." It is also covered in the "learn files" program, which you will be doing soon if you haven't already completed it.)
- Start with section 1 of part 2 and try out the commands until you can produce the expected printed results. (Get a knowledgeable user to show you how to make printed copies, or read about it in part 1, "Getting Started", and in part 4, "Some UNIX Commands".)
- Continue on through the command sections, or pick out the sections you need.
- Don't let yourself get overwhelmed! *UNIX is complex.* Take a break. Ask an experience word processor for help when you're stuck. You don't have to memorize all the commands right away. After all, you can always refer back to the appropriate manual page.

**If you are a visitor and need to produce a few letters, but don't need to learn the whole word processing system,**

- Read all of part 1.
- Read part 3, "Sample Letters and Memos." Copy the sample you need and type in your own text. Then print your letter or memo.

**If you are an experienced UNIX word processor,**

- Read "What Everyone Should Know about Word the Processing Commands" at the beginning of part 2.
- Use the commands list on pages 12-13 to find a specific command.
- Flip through the examples to find the format you want to reproduce.

This manual is made to be USED. Write notes in it, insert dividers, photocopy pertinent pages and post them by your terminal, or whatever, to adapt the manual to your needs.

From time to time, new commands and corrections are issued in memos from the HAO word processing specialist. The memo will tell you where it should be inserted in this manual. (To be sure you are on the distribution list for word processing memos, check with the HAO word processing specialist.)

**Finding What You Want in This Manual**

- Use the alphabetical list of commands on pages 12-13.
- Use the table of contents.
- Use the index.

## No One Is Perfect...

If you find errors in this manual, or sections that are incomprehensible, please photocopy the offending pages and attach a note that explains the problem. Give it to the HAO word processing specialist for correction. Just think of the grief you may be saving other users by finding and reporting an error.

## Overview of UNIX at HAO

UNIX is an operating system that is composed of many sets of instructions -- instructions that control the computers. UNIX is used at thousands of institutions worldwide.

UNIX has advanced word processing capabilities. It produces typeset documents with a professional look. For example, UNIX word processing reproduces complex mathematical equations and tables, changes type sizes and styles (such as to *italics*) and saves and prints numbered footnotes at the bottom of the appropriate pages.

There are a few things you should already be able to do on UNIX before reading further in this manual.

- Login: Obtain your own login code and password from the system administrator. Know how to turn on the terminal, login, and logout.
- Editing commands: Complete the "learn vi" program.
- File commands: Complete the "learn files" program.

If you don't know these, stop reading now and ask the person training you or the HAO word processing specialist for help.

## Basic Hardware

*Hardware* is the machinery and equipment of the system. A *terminal* is the keyboard (like a typewriter keyboard with additional keys) used to enter *input* (text and commands). Input is displayed on the *terminal screen*, which looks like a small television.

The terminals are connected to *computers* that are controlled by the UNIX system. At HAO, the computers are

- The IS, known as the *i-s*. It is used primarily for secretarial work.
- The HW, known as *the h-w* or one of the *vaxes*. It is used primarily for scientific programming.
- The HV, known as *the h-v*, the other vax. It is used primarily for displaying color images and for scientific analysis of images.
- The HA (11/70), known as *the h-a* or *the eleven-seventy*. It is used for some scientific programming, library work, and other projects.

The computers, in turn, are connected to a printer, currently the Imagin 8/300. It is referred to as *the laser printer*, or just *the laser*. The hardware is often updated; exact machine names may change. However, the overall concept remains the same: a terminal for entering text and commands, which is connected to computers that store and format what you entered, which are connected to a printer that produces the final copy.

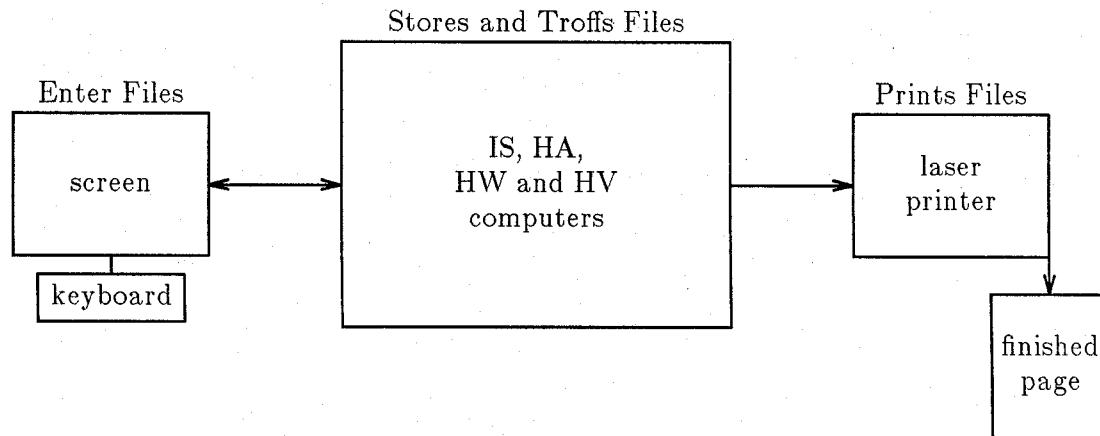
## Basic Software

*Software* is a set of computer programs that make hardware perform various tasks. The UNIX commands are an example of a software package. *Troff* (tee-roff) is the text

formatting software program used at HAO. *Text formatting* controls the layout of the document on the page, the paragraph indents, line length, type size, tables, and so on.

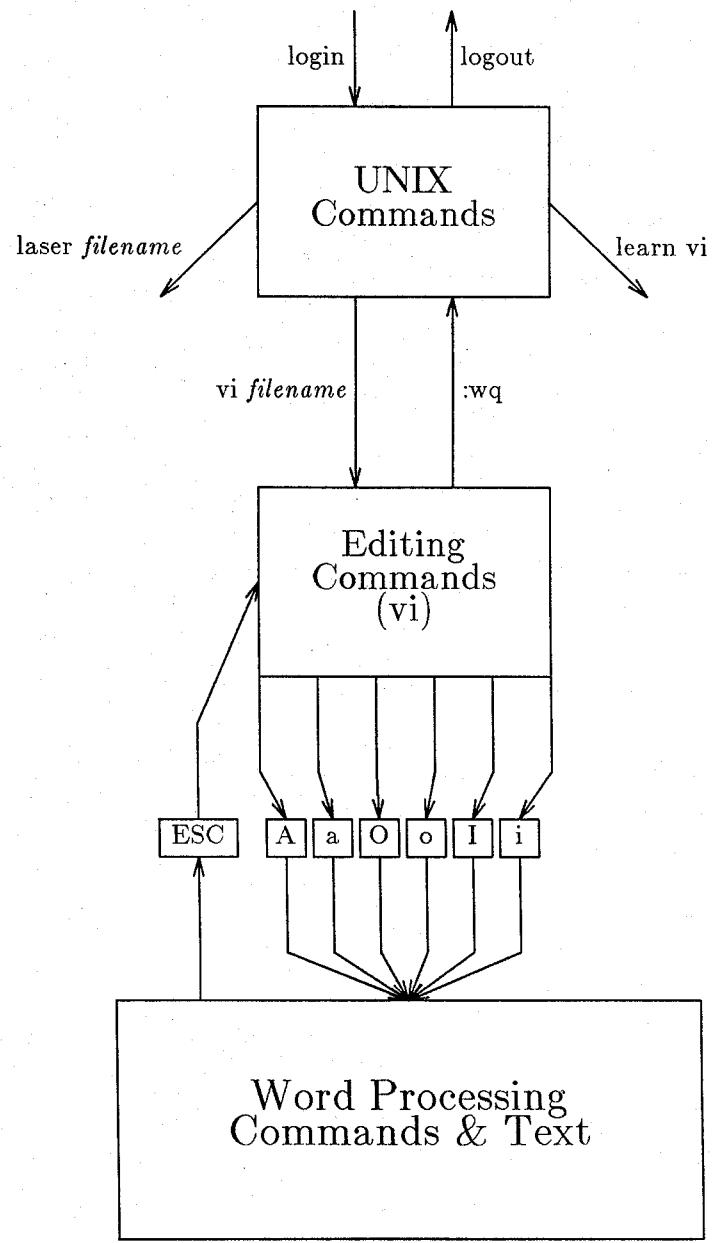
### How a Document Is Created

1. The text of a *document* (letter, report, etc.) and the word processing commands to format it are entered on the terminal. The troff program has *embedded word processing commands*, which means that the actual commands appear on the screen "inside" (embedded in) the text. What you see on the screen is not what you get in the printed copy. You will not see the actual indentations, blank lines, centered titles, and so on.
2. Next, the computer stores the *file* containing the document you just created. The document can be retrieved from the computer for reading or editing at any time.
3. Before printing a document, a computer interprets the word processing commands to create the requested format, that is, the computer *troffs* the document.
4. The computer sends instructions for printing the formatted document to the printer, where the final document is printed.



There are specific commands that execute each step. (See part 1, "Getting Started," when you are ready to complete a document.)

## COMMAND CATEGORIES



## NOTES

laser *filename* - prints the document in *filename* on laser printer  
learn vi - brings the vi section of the learn program to the screen  
vi *filename* - opens the file, *filename*, for reading and writing  
:wq - closes a file after reading and writing  
A, a, O, o, I, i - any of these begins text entering mode  
ESC - ends text entering mode and returns to the vi editor

## The Three Categories of Commands

Each category of commands performs a distinct set of tasks that cannot be performed by any other category of commands. Refer to the diagram on the facing page as you read this section.

### 1. UNIX Commands

UNIX commands communicate between you at your terminal and the computer. UNIX commands perform tasks such as creating or printing a file. They can be entered only when there is a *prompt sign* on the screen (a "\$" or "%" on most HAO terminal screens). UNIX commands cannot be entered at any other level. For example, a UNIX command will not be accepted in the midst of text being entered. Examples of UNIX commands are

- learn vi (request to bring the vi section of the learn program to your screen)
- vi paper (request to bring your file named "paper" to your screen for reading or editing)

### 2. Editing Commands

Editing commands are what you already learned in the "learn vi" program. *Vi* (vee-i) stands for *visual*, the name of the *text editing* system used at HAO. Editing commands manipulate the text, such as deleting and inserting words or lines, moving the *cursor* around on the screen, and so on. (The *cursor* is the symbol that appears on the screen to indicate where you are in the text. At HAO, the cursors are small rectangles.)

Editing commands work only within the editing mode, known as being *in the editor* or being *in vi*. Most editing commands do not appear on the screen when you enter them. The commands simply perform the task, deleting a line or whatever. The exception is editing commands which begin with a colon (:). These commands appear at the bottom of the screen before performing the editing task. Examples of editing commands are

- dd (delete line)
- r (replace the character under the cursor with the next character typed)

### 3. Word Processing Commands

Word processing commands regulate the format of the document, the spacing, indentations, tables, and so on. Word processing commands can be entered only within the editor, vi. Furthermore, word processing commands must be entered within one of the modes that allows text to be entered, such as *insert* (I or i), *append* (A, a), or *open* (O or o). (These are editor commands that are covered in the "learn vi" program.) Examples of word processing commands are

- .pp (leaves a blank line and begins a new, indented paragraph)
- .ce (causes the text on the following line to be centered horizontally)

Word processing commands appear on the terminal screen in the text exactly where you enter them, but the commands themselves are not printed in the output.

**Important Point:** New users of UNIX often try to enter commands when they are in the wrong place. Remember,

- Enter UNIX commands only when there is a prompt sign (\$ or %) on the screen.
- Enter editing commands only within vi.
- Enter word processing commands only within vi *and* within one of the text entering modes.

### **Beeps From the Terminal**

New users often have trouble translating the concept of three distinct categories of commands into practice. When the terminal "beeps," it usually means you tried to enter a command in the wrong category. Hit the escape key (esc) and try again. The terminal might beep again, but that's okay; enter the command again. If you're still having trouble, ask an experienced word processor for help.

## Getting Started

1. Switch on the terminal.
2. Login. There will be a UNIX prompt sign (\$ or %) on the screen when login is completed.
3. Open a file of any name. On HA, the file name must be 14 characters or shorter, with no spaces. Use letters and numbers in file names; avoid using symbols as they often have special command functions. (File names on HV, HW, and IS can be longer.) In this example, the file name will be "test1".

**Opening a file** requires a UNIX command,  
\$ vi test1 (return)

A message like "test1 No such file or directory" or "test1 [New file]" appears at the bottom of the screen.)

Enter an editing command, such as "a" or "i", to get into the text insertion mode. Then hit the return.

If you are having trouble, ask for help or go back to the "learn files" program. Retrieving that program requires a UNIX command:

\$ learn files (return).

1. Within the editor, begin entering text and word processing commands. Use editing commands to move the cursor on the screen and to make changes. This is a good time to start using the "vi Crib Sheet" (a list of frequently used editing commands), which is in part 4 of this manual. If you need more help with editing commands, go back and do the "learn vi" program again. A UNIX command is required to retrieve that program:  
\$ learn vi (return)
2. After all text and word processing commands have been entered, leave the text insertion mode and return to the editor by hitting the escape key (esc). Leave the editor by typing  
:wq (return)  
The "wq" stands for "write quit".
3. Printing the finished document from the file you have just created and stored requires a UNIX level command:  
\$ laser test1 (return)  
The "laser" software program uses the word processing commands to format the document (to regulate the paragraphs, type size, spacing, etc.).
4. Additional commands for printing are covered in part 4, "Some UNIX Commands."

## **Part 2: The Word Processing Commands**

Alphabetical List of Commands with Page Number 12

What Everyone Should Know about the Word Processing Commands 14

Section 1: Spacing, Line Spacing, Break 16

Section 2: Paragraphs and Begin Page 18

Section 3: Indentations for Left Margin 20

Section 4: Indentations from Right Margin 22

Section 5: Centering 24

Section 6: Type Styles (Fonts) 26

Section 7: Underlining 28

Section 8: Type Size (Point Size) 30

Section 9: Headers and Footers 34

Section 10: Quotation Marks 36

Section 11: Block Quotes 38

Section 12: Footnotes 40

Section 13: Line Filling 42

Section 14: Lists 44

Section 15: Section Headings 46

Section 16: Section Headings with Text 48

Section 17: Indented Section Headings 50

Section 18: Line Length 52

Section 19: Two-Columned Text 54

Section 20: Two-Columned Lists 56

Section 21: Multi-Columned Lists (Mark and Return) 59

Section 22: Some Special Characters 61

Section 23: Blank Horizontal Spaces 62

- Section 24: Indented Paragraphs 64
- Section 25: More Indented Paragraphs 66
- Section 26: Numbered Paragraphs 68
- Section 27: More Numbered Paragraphs 70
- Section 28: Final Section of Numbered Paragraphs 72
- Section 29: Base Indents 74
- Section 30: Some Backslash Commands 76
- Section 31: Right Justification 78
- Section 32: Page Offset 80
- Section 33: Page Offset with Line Length 82
- Section 34: Hyphenation 86
- Section 35: Title Page 88
- Section 36: Blank Space at the Top of the Page 90
- Section 37: Top and Bottom Margins 92
- Section 38: Block Keeps 95
- Section 39: Floating Keeps with Horizontal Lines 100
- Section 40: Creating Your Own Commands (Macros) 104
- Section 41: Vertical Spacing 106
- Section 42: Outlines 108
- Section 43: More Outlines 110
- Section 44: Changing the Page Number 112
- Section 45: Delayed Text 114
- Section 46: Table of Contents 116
- Section 47: Boxing Words 118

## Alphabetical List of Commands with Page Number

### Dot Commands

.ad	adjust (justify) 78	.m1	margin one 92
.af	assign numbering format 112	.m2	margin two 92
.(b	open block keep 95	.m3	margin three 92
.)b.	close block keep 95	.m4	margin four 92
.b	bold 26	.mk	mark 59
.ba	base indent 74	.*N	numbered paragraph 70
.bc	begin column 54, 56	.*n	numbered paragraph 72
.bp	begin page 18, 112	.na	no adjust (no justification) 78
.br	break 16	.nf	no fill 42
.bx	box words 118	.nh	no hyphenation 86
.(c	open centered block 24	.np	numbered paragraph 68
.)c	close centered block 24	.nr	number register 50, 70, 72, 108
.2c	two columns 54, 56	.ou	outline 108, 110
.1c	one column 54, 56	.pd	print delayed text 114
.ce	center 24	.pn	page number 112
.(d	open delayed text 114	.po	page offset 80, 84
.)d	close delayed text 114	.pp	paragraph 18
.de	define macro 104	.ps	point size 32
.EN	end equation 127	.(q	open block quote 38
.EQ	start equation 127	.)q	close block quote 38
.(f	open footnote 40	.q	quote 36
.)f	close footnote 40	.r	roman 26
.fi	fill 42	.rt	return 59
.fo	footer 34	.*s	type size 31
.ft	font change 26	.sc	special characters 61
.he	header 34	.sh	section heading 46, 48, 50
.hl	horizontal line 101	.sp	space 16
.hw	hyphenate word 86	.sz	type size 32
.hx	suppress headers, footers 34	.TE	table end 126
.hy	hyphenation 86	.ti	temporary indent 20
.*i	indent from right 22	.tp	title page 88
.i	italics 26	.tr	translate as blank space 62
.in	indent 20	.TS	table start 126
.ip	indented paragraph 64, 66, 74	.u	underline 28
.(l	open list 44	.uh	unnumbered heading 46, 48
.)l	close list 44	.*v	vertical spacing 106, 110
.ll	line length 52	.vs	vertical spacing 106
.lo	local 22, 31, 70, 72, 106	.(x	open table of contents 116
.lp	left paragraph 18	.)x	close table of contents 116
.ls	line spacing 16	.xp	print table of contents 116

.(z open floating keep 100  
.z close floating keep 100  
.\" comment line 77  
. end macro definition 104

### Backslash Commands

\d down 1/2 line 76  
\f font 26  
\'Nc' horizontal line 77  
\s type size 32  
\u up 1/2 line 76  
\\*(td today's date 76  
\& non-printing character 76  
\(space) unpaddable space 76  
\" comment line 77, 104  
\\*\* footnote number 40  
\\*# delayed text number 114  
\% hyphenation 86

## READ THESE TWO PAGES!

### What Everyone Should Know about the Word Processing Commands

The word processing commands used at HAO always begin with either a *dot* (.) or a *backslash* (\), followed by the command. Dot commands must be at the beginning of a line. In most cases, text must be entered on the following line, *not* on the line with the command. Backslash commands can be entered in the middle of a line of text.

Troff reads commands literally. For example, lower case letters in commands must be lower case. Never put a space between the dot and the command.

#### Arguments With Dot Commands

*Arguments* follow commands to modify the meaning of the basic command. For example, in the spacing command of

.sp 2

.sp is the basic command for 1 blank line. The 2 is the argument that modifies the command to mean "2 blank lines." When new commands are explained in this manual, the basic command is in **bold** and the argument is in *italics*.

#### Number Arguments, $\pm N$ :

Number arguments after commands can be in any of the following three formats:

1. The number argument can be the *absolute* value of the number. This means the number has no plus or minus sign. Using the indent command (.in) as an example, a command of  
.in .5i  
will indent .5 inches from the original left margin.
2. A number argument can be preceded by a plus sign. This means the command is increased *relative* to the existing value. For example,  
.in +.5i  
will indent the left margin .5 inches more than what it is currently.
3. A number argument can be preceded by a minus sign. This means the command is decreased *relative* to the existing value. For example,  
.in -.5i  
will subtract .5 inches from the current left margin.

The argument notation of  $\pm N$  means that command can take all three forms of number arguments.

#### Word Arguments

In other cases the argument is words. If the argument is more than one word, it must be surrounded by double quotes so that troff knows that it all stays together. For example, an underlining command of

.u "some words to be underlined".  
produces some words to be underlined.

Both sets of double quote marks surrounding the argument are right quote marks ("), the only set of quote marks available on your keyboard.

Some commands can have more than one argument, but the arguments must be separated by blank spaces. Get in the habit of entering blank spaces before all

arguments. Use the command entry format of

**command argument<sub>1</sub> argument<sub>2</sub>**

### Backslash Commands

Backslash commands (\) are entered within a text line to make a temporary change, such as to make one word *italic*.

### Input/Output

*Input* is the text and commands you enter into the computer. *Output* is the printed copy resulting from the input.

### Default

The troff program has a standard format for margin width, line length, type size, and so on, built into it. This standard format is called the *default*. Documents are printed according to the default values unless other commands are entered to override the defaults.

<u>Default Value</u>	
line length	6.0 inches
left margin	1.375 inches (1 3/8 inches)
right margin	1.375 inches
line spacing	single spacing
type size	10 point
type style	roman
top margin	1 inch
bottom margin	1.25 inches

### A Few More Guidelines

- Troff does all the work of producing the format for you. Entering blank spaces in your file to get indenting, centering, etc. will *not* produce the desired results. A specific command must be entered for each formatting request. Remember, what you see on the screen is *not* what you get in the output!
- Do not start a line with a blank space, as that causes the output to stop and resume on the next line. All lines must start with a character.
- Keep input lines short, as they will be easier to edit. Troff will fill the lines anyway (unless specifically commanded not to). Since editing changes often involve deleting or moving an entire phrase or sentence, get in the habit of starting a new input line after each period or comma to facilitate editing later.
- Troff *right justifies* documents by adding extra spaces between words to make all lines end evenly (unless specifically commanded not to). Enter only one blank space after a word or punctuation mark. No spaces are necessary at the end of the line; the "return" automatically provides a space.
- Do not hyphenate words at the end of lines. Troff does that, too, (unless specifically commanded not to). Words that contain internal hyphens (sister-in-law, for example) should be entered on one line to avoid extra spaces within the set of words when printed.

## Section 1: Spacing, Line Spacing, Break

.sp *N*

The **spacing** command leaves (*N*) amount of blank vertical space.  
For example,

<u>Command</u>	<u>Result</u>
.sp	1 blank line
.sp 2	2 blank lines
.sp 2.25i	2.25 blank inches
.sp 3c	3 blank centimeters

.ls *N*

The **line spacing** command controls the number of blank lines between lines of type, much like line spacing on a typewriter. If no line spacing command is entered, the printing is single-spaced.  
For example,

<u>Command</u>	<u>Result</u>
.ls 2	double spacing
.ls 3	triple spacing
.ls 1.5	1 1/2 spacing
.ls	restores single spacing

.br

The **break** command causes a new line of printing to begin at the current left margin.

## **Example**

.sp, .ls, .br

### **Input\***:

The Mesa Laboratory of the National Center for Atmospheric Research stands on a site of more than 400 acres at the southwestern edge of Boulder, Colorado.

This site was presented to the National Science Foundation, NCAR's sponsor, by the state of Colorado.

.sp

The mesa was named the Walter Orr Roberts Mesa for the first director of NCAR.

It is also popularly known as the NCAR Mesa.

.ls 2

.br

The 120 acres immediately west of the NCAR site were deeded to the city of Boulder by the U.S. government as an addition to Boulder's mountain parks and greenbelt of open space.

The altitude of the mesa is approximately 6,000 feet, 600 feet higher than Boulder.

.br

.ls

The Mesa Nature Trail behind the NCAR building covers the transition from grasslands to foothills to the montane zone (lower mountains) in less than half a mile.

### **Output:**

The Mesa Laboratory of the National Center for Atmospheric Research stands on a site of more than 400 acres at the southwestern edge of Boulder, Colorado. This site was presented to the National Science Foundation, NCAR's sponsor, by the state of Colorado.

The mesa was named the Walter Orr Roberts Mesa for the first director of NCAR. It is also popularly known as the NCAR Mesa.

The 120 acres immediately west of the NCAR site were deeded to the city of Boulder by the U.S. government as an addition to Boulder's mountain parks and greenbelt of open space. The altitude of the mesa is approximately 6,000 feet, 600 feet higher than Boulder.

The Mesa Nature Trail behind the NCAR building covers the transition from grasslands to foothills to the montane zone (lower mountains) in less than half a mile.

---

\*Much of the material for examples was taken from documents published by the Public Information Office of NCAR.

## Section 2: Paragraphs and Begin Page

.pp                  The **paragraph** command leaves a blank line and indents the next line 5 spaces (about 3/8 inch).

.lp                  The **left paragraph** command leaves a blank line. The following line of text starts at the left margin.

**Important Point:** The .pp and the .lp commands leave blank lines of the same height, while the .sp command leaves a slightly larger blank line. Using .sp commands and the paragraph commands in the same document produces varied white space.

.bp                  The **begin page** command begins printing a new page. I'm using the .bp command to start each example page of this manual.

**Caution:**

A .bp command will not work immediately following a .pp or .lp command. In those situations, always enter the .bp command *first*, like this:

.bp  
.lp

**Cross References**    To number paragraphs, see sections 26-28.

For more complicated forms of indented paragraphs, see sections 24 and 25.

To change the page number in conjunction with the .bp command, see section 44.

## Example

.pp, .lp

### Input:

.pp

It was a dark and stormy afternoon on the NCAR Mesa in south Boulder. People in the HAO division sat hunched over their terminals pecking away as usual.

.pp

One question was in the back of everyone's minds.

.lp

Would the lightning cause a loss of power?

The dreaded loss of power.

Even a microsecond could be crucial.

.sp

(Compare the height of the blank line before this paragraph in the output with the blank line above the preceding paragraph, then look at the commands for each.)

The .sp command leaves a larger blank line.)

### Output:

It was a dark and stormy afternoon on the NCAR Mesa in south Boulder. People in the HAO division sat hunched over their terminals pecking away as usual.

One question was in the back of everyone's minds.

Would the lightning cause a loss of power? The dreaded loss of power. Even a microsecond could be crucial.

(Compare the height of the blank line before this paragraph in the output with the blank line above the preceding paragraph, then look at the commands for each. The .sp command leaves a larger blank line.)

### Section 3: Indentation from Left Margin

**Important Point:** If you do not know what  $\pm N$  means, read p. 14 before continuing with this command.

.in  $\pm N$

The **indent** command increases or decreases the indentation of the left margin.

A command of .in  $+N$  adds indentation and a command of .in  $-N$  subtracts indentation from the previous indent. In the example on the facing page, notice that the second command of .in  $+1i$  is added onto the previous indent (.in  $.5i$ ), making the indent a total of 1.5 inches.

Indent commands can be in several different units of horizontal measurement. Here are examples of the options:

Command	Result
.in $+1.256i$	increases indent 1.256 inches
.in $.5c$	increases indent .5 centimeters
.in $-5m$	decreases indent 5 em's
.in $-5$	also decreases indent 5 em's
.in $-5n$	decreases indent 5 en's
.in $20p$	indents 20 points from original left margin

1 em = approximately the width of the letter "m"

1 en = .5 em

72 points = 1 inch

The command .in  $\emptyset$  resets indents to the original left margin. The paragraph commands also reset indents to the original left margin.

.ti  $\pm N$

The **temporary indent** command will change the indent of the next line *only*.

**Important Point:** It can be confusing to decide which measurement units to use with .in and .ti commands. Most users find it easiest to work in inches or centimeters (to any number of decimal places). Using em's, en's, and points makes very small changes possible.

**Cross References**

To produce short lines that are flush with the right margin, see section 4.

To produce a block quote that is indented equal amounts from the left and right margins, see section 11.

For more complex indentation formats, see sections 24, 25, and 29.

## **Example**

.in, .ti

### **Input:**

.in +.5i

It was a dark and stormy afternoon on the NCAR Mesa in south Boulder. People in the HAO division sat hunched over their terminals pecking away as usual.

.in +1i

One question was in the back of everyone's minds.

Would the lightning cause a loss of power? The dreaded loss of power.

.in Ø

Even a microsecond could be crucial.

The loud crack of thunder.

Lights, air conditioning, and screens dim and flicker off for an instant.

But it is enough.

.in 3c

Everyone realizes the consequences.

The computer crashed. It is Down. Gone. Slumbering.

.in -3c

Two questions occupy everyone's thoughts:

When will the computer come back up,

and is the document I was working on irretrievably lost?

.ti .5i

Ah...once again the computer is up and running.

More storm clouds gather.

Lightning. Thunder.

Another crash could come at any minute.

### **Output:**

It was a dark and stormy afternoon on the NCAR Mesa in south Boulder. People in the HAO division sat hunched over their terminals pecking away as usual.

One question was in the back of everyone's minds. Would the lightning cause a loss of power? The dreaded loss of power.

Even a microsecond could be crucial. The loud crack of thunder. Lights, air conditioning, and screens dim and flicker off for an instant. But it is enough.

Everyone realizes the consequences. The computer crashed. It is Down. Gone. Slumbering.

Two questions occupy everyone's thoughts: When will the computer come back up, and is the document I was working on irretrievably lost?

Ah...once again the computer is up and running. More storm clouds gather. Lightning. Thunder. Another crash could come at any minute.

## Section 4: Indentation from Right Margin

.\*i "some words"      The **star indent** command produces short lines that are flush with the right margin.

The argument "*some words*" contains the actual words you want to appear at the right margin, surrounded by double quotes. Enter the actual words again on the line following the command, only this time, don't use quotes. To end the .\*i series, enter the .in Ø command after the last flush right line.

The .\*i command is used to place the inside address of a letter at the right margin.

For another format produced by the .\*i command, look at the closing lines of the letter in the example on the facing page.

**Important Point:** The .\*i command can be used only at HAO. All commands that begin with .\* were created at HAO. When a .\* command is used, the top of the file *must* contain the command line of .lo (which stands for local commands).

## Example

.\*i

### Input:

```
.lo  
.i "1234 Broadway Avenue"  
1234 Broadway Avenue  
.i "Boulder, CO 803XX"  
Boulder, CO 803XX  
.i "July 23, 1985"  
July 23, 1985  
.in Ø
```

.pp

Notice that the three lines above are flush with the right margin.

.pp

Notice that the longest line of the closing below is flush with the right margin.

The other lines of the closing are flush left with the longest line.

Compare the commands for each format.

.i "Executive Director"

Sincerely yours,

.br

.sp 3

Jane J. Doe

.br

Executive Director

.in Ø

### Output:

1234 Broadway Avenue  
Boulder, CO 803XX  
July 23, 1985

Notice that the three lines above are flush with the right margin.

Notice that the longest line of the closing below is flush with the right margin. The other lines of the closing are flush left with the longest line. Compare the commands for each format.

Sincerely yours,

Jane J. Doe  
Executive Director

## Section 5: Centering

.ce *N*

The **centering** command centers any number of lines (*N*) between the current margins. A command of .ce (with no number) centers 1 line.

Enter the command on one line and the text to be centered on the following line(s).

To center several lines without counting the exact number, enter

.ce 1000

the lines to center

more lines to center

.ce Ø

.(c

The **begin centered block** command centers the lines following it in a block style. The longest line in the group is centered, and other lines are flush left with it. Unlike the .ce command, breaks do not occur automatically after each line of input, so enter a no fill (.nf) command before the text to be centered. Be sure to enter a fill (.fi) command after the centered text, before regular text continues.

.)c

The **end centered block** command ends the centered group of lines. Always use this pair of commands together.

**Cross References**

The explanation of no fill/fill is in section 13.

## Example

.ce, .(c, .)c

### Input:

.ce 2

Word Processing Commands  
In Plain English

.sp

.ce 1000

Word Processing  
With UNIX,  
A Challenge  
For Today's Workers

.ce 0

.sp

.nf

.(c

Word Processing  
With UNIX,  
A Challenge  
For Today's Workers

.)c

.fi

### Output:

Word Processing Commands  
In Plain English

Word Processing  
With UNIX,  
A Challenge  
For Today's Workers

Word Processing  
With UNIX,  
A Challenge  
For Today's Workers

## Section 6: Type Styles (Fonts)

A *font* is a style of type. On UNIX at HAO, three fonts are available: times roman, *times italic*, and **times bold**.

**.ft N**      The **font** command changes the type style. The argument *N* may be *B* (bold), *I* (italic), *R* (roman), or *P* (previous font). The font command without an argument returns the following lines to the previous font.

**.b**      Another command that results in bold font. The commands **.r** and **.i** cause changes to roman and italic fonts, respectively. Notice these commands are lower case letters and that there is no command for previous font.

**\fN**      Changes the font in-line for as long as desired. Possible arguments (*N*) are the same as for the **.ft** command: *B*, *I*, *R*, *P*.

When entering a backslash command in the middle of a line, leave one space before the command, no space between the command and the following word, and one space after the last command. Otherwise, the spacing of the words will not come out right. For example, entering

This is the \fIcorrect spacing\fR for making an in-line change.

produces

This is the *correct spacing* for making an in-line change.

All font commands are reset by any other font command or any paragraph command.

**Important Point:** If you're thinking that several of the font commands accomplish the same task, you are right. UNIX has extensive capabilities, and there are often several different commands that accomplish the same task, although some may take less keystrokes. Get in the habit of using a dot command to change more than 1 line and using the backslash command to make in-line changes, since this same pattern is applicable to other commands.

## Example

.ft, .b, .i, .r, \f

### Input:

The \fBUtes\fR and the \fBArapahoes\fP  
were two of the tribes  
that lived in the Boulder Valley  
long \fIbefore\fR Hispanic and Anglo settlers arrived.

.br

.b

The Utes lived in small bands and hunted deer, elk, and buffalo  
in the area.

.pp

The Arapahoes grew corn and farmed in the Minnesota area before  
\fImigrating\fR to the high plains.

.i

On the plains, they became hunters and were dependent on the buffalo.

.r

This brought them into competition and conflict with the Ute.

.ft B

.br

By the 1860s, the Arapahoes dominated the foothills and the plains  
and the Utes occupied the mountains.

.ft I

The Apache, Shoshone, and Cheyennes also lived in the area.

### Output:

The **Utes** and the **Arapahoes** were two of the tribes that lived in the Boulder Valley  
long *before* Hispanic and Anglo settlers arrived.

**The Utes lived in small bands and hunted deer, elk, and buffalo in the area.**

The Arapahoes grew corn and farmed in the Minnesota area before *migrating* to the  
high plains. *On the plains, they became hunters and were dependent on the buffalo.* This  
brought them into competition and conflict with the Ute.

**By the 1860s, the Arapahoes dominated the foothills and the plains and the  
Utes occupied the mountains. The Apache, Shoshone, and Cheyennes also lived in  
the area.**

## Section 7: Underlining

.u "some words"

The **underline** command underlines 1 or more words. Put double quotes around the words that are to be underlined. (Actually, double quote marks are not necessary for word arguments of only 1 word. However, since double quotes are necessary for more than 1 word, it's a good habit to always use double quotes so you don't omit them when they are needed.)

Continuous underlining of more than a few words doesn't work well with troff. Try using bold or italics and/or a larger type size to make text stand out. If you really *must* have underlining, here's the problem and what to do about it.

If the underlined words break between two lines in the output, the underlining will not come out right. To solve this problem, start a new line of input at the point where the new line of output begins. Be sure to start the new line of input with .u and put double quotes around the words to be underlined.

Then print the document again. The results should be better, though perhaps not perfect. Sometimes the only solution is to use a black pen, ruler, and correction fluid for final adjustments.

### Cross References

For type style changes, see section 6.

To change type size, see section 8.

## Example

.u

### First Input:

.u "Early"  
.u "civilizations"  
.u "in"  
.u "the"  
.u "Boulder"  
.u "Valley:"  
.lp

Long before

.u "Hispanic and Anglo"

settlers arrived in the Boulder Valley,

.u "native tribes such as the Utes and Arapahoes lived there."

### First Output:

Early civilizations in the Boulder Valley:

Long before Hispanic and Anglo settlers arrived in the Boulder Valley, native tribes such as the Utes and Arapahoes lived there.

---

### Corrected Input:

.u "Early"  
.u "civilizations"  
.u "in"  
.u "the"  
.u "Boulder"  
.u "Valley:"  
.lp

Long before

.u "Hispanic and Anglo"

settlers arrived in the Boulder Valley,

.u "native tribes such"

.u "as the Utes and Arapahoes lived there."

### Final Output:

Early civilizations in the Boulder Valley:

Long before Hispanic and Anglo settlers arrived in the Boulder Valley, native tribes such as the Utes and Arapahoes lived there.

## **Section 8: Type Size (Point Size)**

Type size is measured in a typesetting unit known as *points*. There are 72 points to the inch. Standard type in troff is 10-point; the text of this manual is in 11-point type. There are 15 point sizes available on UNIX, as illustrated here:

6 point: Marmots like to sunbathe.  
7 point: Marmots like to sunbathe.  
8 point: Marmots like to sunbathe.  
9 point: Marmots like to sunbathe.  
10 point: Marmots like to sunbathe.  
11 point: Marmots like to sunbathe.  
12 point: Marmots like to sunbathe.  
14 point: Marmots like to sunbathe.  
16 point: Marmots like to sunbathe.  
18 point: Marmots like to sunbathe.  
20 point: Marmots like to sunbathe.  
22 point: Marmots like to sunbathe.  
24 point: Marmots like to sunbathe.  
28 point: Marmots like to sunbathe.  
**36 point: Marmots like to sun**

The commands to change type size are on the following pages.

## Section 8: Type Size (Point Size)

.\*s, \s, .sz

### Changing Type Size of a Whole Document

.\*s N

The **star size** command changes the point size, with *N* as any point size number. If you enter a number that is not on the point size list on the preceding page, troff increases the size to the next legitimate point size. For example,

.\*s 17

produces 18 point type, since 17 is not a legitimate point size.

The command automatically adjusts the white space between lines to allow for taller or shorter type.

The .\*s command is local to HAO, as are all .\* commands. Remember to put .lo (local) at the top of the file before entering any .\* command.

The .\*s command is reset by another .\*s command. Other size commands (\s, .sz, .ps) *temporarily* reset .\*s, but only until the next paragraph or section heading command. After any one of the paragraph or section heading commands, the type size returns to that specified by the last .\*s command. For example,

<u>Command</u>	<u>Result</u>
.*s 11	11-point text follows
.ps +3	14-point text follows
.pp	11-point text follows

Usually, it's easiest to change type size with a .\*s command and to reset type size with another .\*s command.

### Changing Type Size Temporarily Within a Document

All 3 of the type size commands listed on the next page also change the type size, but only temporarily. These three type size commands are reset by any of the paragraph commands (.pp, .lp, .ip, .np, .\*N, or .\*n) or the section heading commands (.sh and .uh). (You do not need to know about all those commands at this point, so continue reading.) After one of the paragraph or section heading commands, the type size returns to what it was before the last size command. For example,

<u>Command</u>	<u>Result</u>
no size command entered	10-point type, the normal size
.sz 12	12-point type follows
.lp	returns to 10-point type

**\s±N**

Changes the point size in-line. A command of **\s0** returns the point size to its previous value. There is no space between the command of s and the number argument.

**.sz ±N**

The **size** command increases or decreases point size and automatically adjusts the white space between lines.

**.ps ±N**

The **point size** command also increases or decreases point size. However, this command does *not* adjust the white space between lines. As a result, type in larger sizes may overlap with the lines above and below it.

**Cross References**

For the explanation of the  $\pm N$  argument, see p. 14.

To change the white space between lines, see line spacing in section 1 and vertical spacing in section 41.

## Example

.\*s, \s, .sz

### Input 1:

.lo

.\*s 12

.lp

Marmots are commonly called \s16woodchucks\s0 and \s+4ground hogs\s-4 in North America.

Several live in the open space surrounding \s+2NCAR\s0.

.pp

At least one lives quite close to the building, and can often be spotted sunning on a rock.

.\*s 8

Marmots use a loud alarm bark when they sense danger.

### Output 1:

Marmots are commonly called woodchucks and ground hogs in North America. Several live in the open space surrounding NCAR.

At least one lives quite close to the building, and can often be spotted sunning on a rock. Marmots use a loud alarm bark when they sense danger.

### Input 2:

.lp

.sz 14

This example uses the .sz command instead of the .\*s command. Notice that the new paragraph resets the point size.

.lp

Marmots are commonly called \s16woodchucks\s0 and \s+2ground hogs\s-2 in North America.

Several live in the open space surrounding \s+2NCAR\s0.

.pp

.sz +2

At least one lives quite close to the building, and can often be spotted sunning on a rock.

.pp

Marmots use a loud alarm bark when they sense danger.

### Output 2:

This example uses the .sz command instead of the .\*s command. Notice that the new paragraph resets the point size.

Marmots are commonly called woodchucks and ground hogs in North America. Several live in the open space surrounding NCAR.

At least one lives quite close to the building, and can often be spotted sunning on a rock.

Marmots use a loud alarm bark when they sense danger.

## Section 9: Headers and Footers

**.he 'left'center'right'** The **header** command sets the headings that appear in bold at the top of each page. There are three possible title locations: the left margin, centered, or the right margin. The 'left'center'right' argument parts are replaced with the desired titles.

The three heading titles are separated by *delimiters*, which are single quote marks (') in this example. *Delimiters* are characters that are used to mark the beginning and end of a special unit.

**.fo 'left'center'right'** The **footer** command operates just like the header command, only the footers are at the bottom of each page.

### Important Points:

1. The header and footer commands should be entered at the top of the file, before the text.
2. Notice in the example that a delimiter symbol must be entered on either side of each heading title, even if there is no text for that title location.
3. To number pages consecutively starting with "1", enter the symbol "%" in the title location where the number should appear. Do not try to use % as a delimiter for headers and footers.
4. Pairs of other symbols, such as @, can be used as delimiters to separate parts of the headers and footers. If a single quote is in one of the titles, another symbol *must* be used as delimiters.

### .hx

This command will omit the headers and footers on the next page only. You might want to use this command for a page with a large figure or table. The .hx command is often used with the begin page (.bp) command (see section 2).

**Cross References** To set headers or footers for every even-numbered or odd-numbered page, see the "-me Reference Manual", section 3.

To change the horizontal spacing between header or footer titles, see the "-me Reference Manual", section 3.

To change the page numbering sequence, see section 44.

To move the header or footer up or down the page, see section 37.

## Example

.he, .fo

### Input:

.he "L. Skywalker'Star Wars Revised'  
.fo '1st Edition%'Draft Copy'

Look at the header above.

Notice that the left title location is empty, the center title is  
\fIL. Skywalker\fR, and the right title is

\fIStar Wars Revised.\fR

.lp

Now look at the footer below.

The left title is \fI1st Edition\fR,

the center is \fI%\fR

(which starts page numbering at 1),

and the right title is

\fIDraft Copy\fR.

### Output:

L. Skywalker

Star Wars Revised

Look at the header above. Notice that the left title location is empty, the center title is *L. Skywalker*, and the right title is *Star Wars Revised*.

Now look at the footer below. The left title is *1st Edition*, the center is *%* (which starts page numbering at 1), and the right title is *Draft Copy*.

1st Edition

1

Draft Copy

## Section 10: Quotation Marks

- .q "some words" The **quote** command produces double quotation marks around a group of words that is not longer than a terminal screen line. The words to be quoted must be enclosed in double quotation marks on the command line.
- Enter text that comes after the quoted words on the following line. However, enter punctuation marks on the *same* line as the command to avoid getting an extra space between the quoted words and the punctuation mark. Enter the punctuation mark wherever it belongs -- inside or outside of the quotation marks. See the examples on the facing page.
- \\*(lq The **left quote** command places left quotation marks in-line. Use this command to begin a quotation that is longer than a terminal screen line.
- There is no space between the command and the first letter inside the quotation marks.
- \\*(rq The **right quote** command places right quotation marks in-line. Use this command to end a quotation that is longer than a terminal screen line.
- There is no space between the last letter inside the quotation marks and this command.
- single quotes** Both the left (‘) and right (’) quotation marks are on the keyboards at HAO.
- Cross References** To produce block quotes that are indented from both margins and are not surrounded by quotation marks, see section 11.

## Example

.q, \\*(lq, \\*(rq

### Input:

.lp

During the summer around NCAR,  
you may see

.q "small holes"

of about a one-half inch diameter in the earth.

These are the homes of

.q "wolf spiders".

These large, swift spiders dart out of their burrows to catch  
insects.

.q "The wolf spider is related to the tarantula."

.lp

The female wolf spider \\*(lqwraps her eggs in a silken brood  
pouch;

the eggs (several dozen for each female)

hatch in midsummer.\\*(rq

The wolf spider's sting \\*(lqis not poisonous to human beings.

Its effect is similar to that of a bee sting.\\*(rq

### Output:

During the summer around NCAR, you may see "small holes" of about a one-half inch diameter in the earth. These are the homes of "wolf spiders". These large, swift spiders dart out of their burrows to catch insects. "The wolf spider is related to the tarantula."

The female wolf spider "wraps her eggs in a silken brood pouch; the eggs (several dozen for each female) hatch in midsummer." The wolf spider's sting "is not poisonous to human beings. Its effect is similar to that of a bee sting."

## Section 11: Block Quotes

.(q

The **begin block quote** command opens a block quote that is indented 3/8 inch from the left and right margins. The block quote is single spaced and the type is one point smaller than the main text.

The .(q command automatically leaves a blank line before starting the block quote.

.)q

The **end block quote** command closes the quote and returns the margins, type size, and spacing to their previous values.

The .)q command automatically leaves a blank line after the block quote.

### Cross References

To produce quotation marks, see section 10.

## **Example**

**.(q, .)q**

### **Input:**

.lp

Harold Baynton, who retired from NCAR in 1985, recalled,

.(q

I was a graduate student when university faculties were first discussing the idea of NCAR.

It seemed like such a grand and glorious concept that I and my fellow students didn't even dare to hope that some day we might be there....

I entered meteorology in June 1941 in the Canadian Weather Service.

At that time [World War II]

the weather service was frantically recruiting math and physics majors and training them to be forecasters...

.)q

Baynton joined the NCAR Facilities Laboratory staff in 1965.

His field studies took him to Puerto Rico, Grand Bahama Island, and other sites.

### **Output:**

Harold Baynton, who retired from NCAR in 1985, recalled,

I was a graduate student when university faculties were first discussing the idea of NCAR. It seemed like such a grand and glorious concept that I and my fellow students didn't even dare to hope that some day we might be there.... I entered meteorology in June 1941 in the Canadian Weather Service. At that time [World War II] the weather service was frantically recruiting math and physics majors and training them to be forecasters...

Baynton joined the NCAR Facilities Laboratory staff in 1965. His field studies took him to Puerto Rico, Grand Bahama Island, and other sites.

## Section 12: Footnotes

Footnote commands number footnotes consecutively throughout the file and print them at the bottom of each page in type 2 points smaller than the text.

\\*\*

The **footnote number** command places the footnote number in the body of the text and at the beginning of the footnote itself. The actual number is *superimposed*, which means that it is slightly above the line of text. Enter the command in-line (it's a backslash command) exactly where the number belongs and again before the text of the footnote.

.(f

The **begin footnote** command opens the text of the footnote. The command moves the footnote to the bottom of the page.

.)f

The **end footnote** command closes the footnote text.

**Important Point:** A special character, such as a star (\*) can be used to mark reference notes instead of numbers. Enter the star where it belongs in the text and again at the beginning of the footnote text. The star is automatically superimposed.

### Cross References

To superimpose special characters besides the star, see \u and \d, section 30 and "Typesetting Mathematics -- User's Guide," section 7.

For another style of notes, see section 45.

## Example

\\*\*, .(f, .)f

### Input:

.lp

UNIX is a time-sharing system; this means it can serve several users simultaneously by sharing its time among them.

.(f

\\*\*Ann Nichols Lomuto and Nico Lomuto,  
\fIA UNIX Primer\fR  
(Englewood Cliffs, N.J.: 1983),  
p. 5.

.)f

.sp

Why isn't there a good beginner's book about UNIX?

Surely not for lack of experts!\*

.(f

\*ibid., p. xiv.

.)f

### Output:

UNIX is a time-sharing system; this means it can serve several users simultaneously by sharing its time among them.

Why isn't there a good beginner's book about UNIX? Surely not for lack of experts!\*

---

<sup>1</sup>Ann Nichols Lomuto and Nico Lomuto, *A UNIX Primer* (Englewood Cliffs, N.J.: 1983), p. 5.  
\*ibid., p. xiv.

## Section 13: Line Filling

Unless commanded not to, troff prints lines all the way to the right margin, even though the lines of input that appear on the terminal screen are much shorter.

.nf

The **no fill** command removes line filling. Lines of printed output begin and end exactly where lines of input begin and end on the screen, except the spacing between characters is different. On the screen, each character or space has the same width; in the output, troff assigns different widths to different characters. For example, troff assigns different printing widths to the letters "l" and "m."

.fi

The **fill** command restores line filling.

## **Example**

.nf, .fi

### **Input:**

.lp

.nf

Several different kinds of ants live  
in the area surrounding NCAR.

The Colorado thatching ants  
build nests at the bases  
of trees or shrubs.

.fi

The nests are covered with mounds of twigs and pine needles.

The species of these ants is \fIFormica obscuripes\fR.

If people disturb a thatching ant nest, the ants will abandon  
the area.

### **Output:**

Several different kinds of ants live  
in the area surrounding NCAR.

The Colorado thatching ants  
build nests at the bases  
of trees or shrubs.

The nests are covered with mounds of twigs and pine needles. The species of these ants  
is *Formica obscuripes*. If people disturb a thatching ant nest, the ants will abandon the  
area.

## Section 14: Lists

.(l *m f*

The **begin list** command leaves a blank line and then opens a single-spaced list. A command of .(l with no arguments indents the list about 3/8 inch from the left margin, and the lines are not filled. The first argument (*m*) can be any of the following:

<u>Command</u>	<u>Result</u>
.(l I	Indented from left margin (same as default)
.(l M	At the original left margin
.(l L	At the current left margin
.(l C	Each line centered between current margins

There must be a first argument in order to enter a second argument. Enter a second argument of *F* to fill the lines. For example,

<u>Command</u>	<u>Result</u>
.(l I F	Indented 3/8 inch from left margin, lines filled
.(l M F	At original left margin, lines filled

.)l

The **end list** command leaves a blank line and closes the list.

### Cross References

For the explanation of no fill/fill, see section 13.

## Example

.(1, .)1

### Input:

```
.(1
peaches
pears
plums
.)1
.(1 M F
cantaloupe
honey dew
watermelon
.)1
.(1 C
oranges
lemons
limes
.)1
.sp
.in +1i
```

Listing all these fruits makes me feel hungry.

This text is indented 1 inch from the left margin,  
as is the following list.

```
.(1 L
mangoes
pineapples
coconuts
.)1
.in -1i
```

Here is the text at the original left margin for comparison.

### Output:

```
peaches
pears
plums
cantaloupe honey dew watermelon
```

```
oranges
lemons
limes
```

Listing all these fruits makes me feel hungry. This text is indented 1 inch  
from the left margin, as is the following list.

```
mangoes
pineapples
coconuts
```

Here is text at the original left margin for comparison.

## Section 15: Section Headings

.uh "some words" The **unnumbered heading** command leaves a blank line and produces a section heading of "some words" in bold.

.sh N "some words" The **section heading** command leaves a blank line and produces a numbered section headings of "some words" in bold.

The argument *N* is a number that represents the *depth* of the section heading. Depth is how many numbers, separated by periods, appear in the printed section number. For example, the section number 2.3 has a depth of 2, while the section number 2.3.6 has a depth of 3. Section numbers have a maximum depth of 6. The section number at each depth is automatically increased with each new entry at that depth.

**Important Point:** You must be in the fill (.fi) mode before entering section headings to insure that the section number and its title are printed on the same line.

**Cross References** The explanation of no fill/fill is in section 13.

To enter text with section headings, see section 16.

To change the numbering of section headings, see "Writing Papers With Nroff Using -me," section 5.2.

To indent numbered section headings at each level, see section 17.

For outlines, see sections 42 and 43.

## **Example**

**.uh, .sh**

### **Input:**

```
.uh "The System at HAO"
.sh 1 "Overview of Unix"
.sh 2 "Basic Hardware"
.sh 3 "Terminals, Keyboards, Screens"
.sh 3 "Computers"
.sh 4 "IS"
.sh 4 "HA"
.sh 4 "HW"
.sh 4 "HV"
.sh 3 "Printers"
.sh 2 "Software"
.sh 2 "Job Path"
.uh "When Things Go Wrong"
```

### **Output:**

#### **The System at HAO**

##### **1. Overview of Unix**

###### **1.1. Basic Hardware**

###### **1.1.1. Terminals, Keyboards, Screens**

###### **1.1.2. Computers**

###### **1.1.2.1. IS**

###### **1.1.2.2. HA**

###### **1.1.2.3. HW**

###### **1.1.2.4. HV**

###### **1.2. Software**

###### **1.3. Job Path**

#### **When Things Go Wrong**

## Section 16: Section Headings With Text

**.sh** *N* "some words" To begin text on the *same line* as the heading, enter the text on the next line (with no command between the heading and the text).

**.uh** "some words" To begin text on the *line following* a section heading, enter a command of .br, .lp, .sp, or .pp on the line following the heading, then begin text on the next line.

These command sequences work for both .sh and .uh headings.

Text after section headings is automatically in roman type.

**Cross References** For an introduction to section headings, see section 15.

To indent numbered section headings at each level, see section 17.

For outlines see sections 42 and 43.

## **Example**

.uh, .sh

### **Input:**

.sh 2 "Basic Hardware"

Hardware is the machinery and equipment of the system.

.sh 3 "Terminals and Screens"

.br

A terminal is a keyboard...

.sh 3 "Computers"

.lp

The terminals are connected to computers...

.uh "Software"

.sp

Software is a set of computer programs...

.uh "When Things Go Wrong"

.pp

Sometimes things \fI do \fR go wrong when you are using UNIX.

It could be a hardware or software problem,

or the problem could be the result of human error --yours or  
someone else's.

### **Output:**

**1.4. Basic Hardware** Hardware is the machinery and equipment of the system.

**1.4.1. Terminals and Screens**

A terminal is a keyboard...

**1.4.2. Computers**

The terminals are connected to computers...

## **Software**

Software is a set of computer programs...

## **When Things Go Wrong**

Sometimes things *do* go wrong when you are using UNIX. It could be a hardware or software problem, or the problem could be the result of human error --yours or someone else's.

## **Section 17: Indented Section Headings**

**.nr si N**

If this command is entered on the line before a .sh command, each section depth will be indented *N* units. The argument *N* may be in inches, centimeters, points, en's or em's. Unlike other commands, some type of units (i for inches, for example) *must* be specified after the number.

**Cross References**

For an easier way to make outlines, see sections 42 and 43.

For an introduction to section headings, see sections 15 and 16.

For the explanation of en's and em's, see section 3.

## **Example**

.nr si N, .sh

### **Input:**

.nr si .5i  
.sh 1 "Overview of Unix"  
.sh 2 "Basic Hardware"  
.br

Hardware is the machinery and equipment of the system.

.sh 3 "Terminals"  
.lp

A terminal is a keyboard...

.sh 3 "Computers"

.pp  
The terminals are connected to computers...

.sh 4 "IS"

.sp  
Most secretarial work is done on the IS.

.sh 4 "HW"

.sh 4 "HV"

.sh 3 "Printers"

.sh 2 "Software"

### **Output:**

## **1. Overview of Unix**

### **1.1. Basic Hardware**

Hardware is the machinery and equipment of the system.

#### **1.1.1. Terminals**

A terminal is a keyboard...

#### **1.1.2. Computers**

The terminals are connected to computers...

##### **1.1.2.1. IS**

Most secretarial work is done on the IS.

##### **1.1.2.2. HW**

##### **1.1.2.3. HV**

#### **1.1.3. Printers**

## **1.2. Software**

## Section 18: Line Length

.ll  $\pm N$

The **line length** command changes the length of lines. The normal length is 6.0 inches.

The line length command takes effect *only* after the current line of printing is completed. This usually doesn't cause problems because a line length command almost always precedes or follows a command that starts a new line (such as a .br, .sp, or paragraph command).

Only another line length command will reset (change) the line length.

Like other commands that change horizontal spacing, the argument *N* may be in centimeters, inches, en's, em's, or points. Most users find it easiest to work in inches or centimeters.

**Important Point:** By adding extra line length, the right margin can be pushed to within about .75 inches of the right edge of the paper. If the line length command exceeds this limit, the type will pile up at the end of the line. (See the end of the example on the facing page.)

**Cross References** To push the left margin toward the left edge of the paper, see sections 32 and 33.

For the explanation of  $\pm N$ , see p. 14.

For more on en's, em's, and points, see section 3.

## Example

### Input:

.ll -2i

Most of the trees on the NCAR site are ponderosa pines,  
also called western yellow pines.

In the Boulder area,

the ponderosas do not become very large,  
but in parts of the West they grow quite tall and are used for  
lumber.

.br

.ll +1i

Ponderosa needles grow in bundles of two or three.

The trunks are thick,  
and the bark is coarse and reddish.

.ll -2.5i

The red cedar, a graceful tree with gray-green lacy foliage,  
also grows on the NCAR site.

Red cedars prefer soil with a high lime content.

A group of them has developed along the Niobrara Limestone  
outcrop extending across the west end of the NCAR mesa.

.pp

.ll 8i

A low lying shrub juniper also occurs here.

It can be recognized by the white lower surface of the needles  
and the sharp needle points.

### Output:

Most of the trees on the NCAR site are ponderosa pines,  
also called western yellow pines. In the Boulder area, the  
ponderosas do not become very large, but in parts of the  
West they grow quite tall and are used for lumber.

Ponderosa needles grow in bundles of two or three. The trunks are thick,  
and the bark is coarse and reddish. The red cedar, a graceful tree with  
gray-green lacy foliage, also grows on  
the NCAR site. Red cedars prefer  
soil with a high lime content. A  
group of them has developed along  
the Niobrara Limestone outcrop  
extending across the west end of the  
NCAR mesa.

A low lying shrub juniper also occurs here. It can be recognized by the white lower surface  
the sharp needle points. Berries of this and other junipers provide a flavoring ingredient.

## Section 19: Two-Columned Text

### .2c

The **two columns** command begins two-columned text. The white space between columns is about .25 inches and the lines of each column are filled.

Use this command for journal articles that must be submitted in the two-column format. Troff automatically begins the second column and new pages as needed.

#### Cross References

To change the white space between the columns, see the “-me Reference Manual,” section 6.

For other uses of the .2c command, see section 20.

## **Example**

**.2c**

**Input:**

**.2c**

**.pp**

Some of the large boulders around NCAR have hollows that collect rainwater. Most of the year, the hollows are dry and apparently empty. But even when the hollows are dry, they contain both living materials and food substances.

**.pp**

When a hollow is filled with rainwater, it becomes a miniature world of teeming microscopic life.

The hollow becomes the scene of repeated population explosions, as one species after another increases in numbers and then dies off.

**Output:**

Some of the large boulders around NCAR have hollows that collect rainwater. Most of the year, the hollows are dry and apparently empty. But even when the hollows are dry, they contain both living materials and food substances.

When a hollow is filled with rain, it becomes a miniature world of teeming microscopic life. The hollow becomes the scene of repeated populations explosions, as one species after another increases in numbers and then dies off.

(Assume this is the bottom of the page.)

## Section 20: Two-Columned Lists

- .2c      The **two columns** command begins a two-columned format.
  - .bc      The **begin column** command starts the second column.
  - .1c      The **one column** command restores the text to 1 column, the usual format.

Use a no fill command (.nf) to produce two-columned lists without filled lines. Remember to restore filled lines (.fi) at the end of the columns.

To get a blank line before and after the columns, enter space commands (.sp) before the .2c and after the .1c.

To produce two-columned distribution lists at the end of letters and memos, use the two-column series of commands and a shorter line length. The shorter line length brings the right column closer to the left column. (See the example at the end of this section.)
- Cross References**
- For another use of the .2c command, see section 19.
  - The explanation of no fill/fill is in section 13.
  - The explanation of line length is in section 18.
  - For the easiest ways to change the amount of white space between columns and/or to produce more columns, use either the mark (.mk) and return (.rt) commands explained in section 21, or create a table, as explained in "Tbl -- A Program to Format Tables."

## Example

.2c, .bc, .1c

### Input:

.sp  
.2c  
.nf

.u "Mammals"

Least chipmunk

Abert's squirrel

Marmots

Mule deer

Racoon

Skunk

.bc

.u "Birds"

Flicker

Yellow warbler

Downy woodpecker

Hairy woodpecker

Prairie falcon

Pygmy nuthatch

.1c

.sp

.fi

The mammals and birds listed above frequent the NCAR mesa.

This is, of course, only a partial listing.

### Output:

#### Mammals

Least chipmunk

Abert's squirrel

Marmots

Mule deer

Racoon

Skunk

#### Birds

Flicker

Yellow warbler

Downy woodpecker

Hairy woodpecker

Prairie falcon

Pygmy nuthatch

The mammals and birds listed above frequent the NCAR mesa. This is, of course, only a partial listing.

## **Example**

### **Input:**

Distribution:

.in +1i

.ll 4i

.2c

.nf

Bugs Bunny

Bullwinkle

Cinderella

Daffy Duck

Donald Duck

Goofey

Mickey Mouse

.bc

Minnie Mouse

Peter Pan

Pluto

Roadrunner

Snow White

Tinkerbell

Yogi Bear

.1c

.fi

.in -1i

.ll 6i

### **Output:**

Distribution:

Bugs Bunny  
Bullwinkle  
Cinderella  
Daffy Duck  
Donald Duck  
Goofey  
Mickey Mouse

Minnie Mouse  
Peter Pan  
Pluto  
Roadrunner  
Snow White  
Tinkerbell  
Yogi Bear

.mk, .rt

## Section 21: Multi-Columned Lists (Mark and Return)

The **mark** and **return** commands are used with the indent command to create a number of columns. Sometimes it's easier to use these commands than to create a table. However, if there are many columns that need to have the same amount of white space between them, or if the whole set of columns needs to be centered on the page, making a table is probably easier than using these commands.

The mark and return commands can be used to mark any line and then return later text to that same line. You must be in no fill (.nf) before starting a mark and return series. One problem is that the return command cannot return text to the correct location if a new page of printing has started or if the return command occurs near the bottom of a page.

**.mk**      The **mark** command marks the place of the current line. Enter a no fill command (.nf) on the line before the mark command. When making multi-columned lists, enter the .mk command before the first column.

**.rt**      The **return** command returns text to the beginning of the "marked" line.

**.in +N**      After each .rt command, enter an **indent** command with a +N to move the text over on the page. Otherwise, the "returned" text will be on top of the text on the "marked" line.

When you are finished entering text at the "returned" spot, enter an indent command with a -N to get back out to the original margin. Then enter the fill command (.fi) to restore filled lines of text.

**Cross References**      The explanation of no fill/fill is in section 13.

For additional ways to create columns, see section 20 and "Tbl -- A Program to Format Tables."

(The example of mark and return is on the following page.)

## **Example**

### **Input:**

There have been some telephone and room changes to accommodate our visitors.

```
.sp
.nf
.mk
.in +1.5i
.u Visitor
L. Skywalker
D. Spock
E. Terrestrial
.rt
.in +1.5i
.u Extension
2468
1369
9876
.rt
.in +1i
.u Room
608
954
911
.in -4i
.fi
.sp
```

Please use this information to update your directory.

### **Output:**

There have been some telephone and room changes to accommodate our visitors.

<u>Visitor</u>	<u>Extension</u>	<u>Room</u>
L. Skywalker	2468	608
D. Spock	1369	954
E. Terrestrial	9876	911

Please use this information to update your directory.

## Section 22: Some Special Characters

.sc

.sc

The **special character** command must be near the top of the file in order to produce any of the special characters listed below.

Enter special character commands in-line. For example,  
nin\\*~o produces niño.

Input:	Output:	Character Name: <sup>1</sup>
.sc		
a\*'	á	Acute accent
e\*`	è	Grave accent
u\*:	ü	Umlat
n\*~	ñ	Tilde
e\*^	ê	Caret
c\*,	ç	Cedilla
e\*v	é	Czech
A\*o	Å	Circle
\*(qe	Ǝ	There exists
\*(qa	∀	For all

### Important Points:

1. The acute accent (á) uses the right single quote mark and the grave accent (è) uses the left single quote mark. Each single quote mark has a key on the keyboard.
2. To produce special characters within tables, you must use 2 backslashes. For example, to produce niño inside a table, enter  
nin\\\*~o

**Cross References** To produce many other special characters, including tildes over capital letters, see the list of mathematical and special characters in part 4.

<sup>1</sup> Eric P. Allman, "me Reference Manual," in *Unix Programmer's Manual*, vol. 2c, section 54, (Berkeley: Univ. of California, Dept. of Electrical Engineering and Computer Science, 1980), p. 9.

## Section 23: Blank Horizontal Spaces

.tr c

The **translate** command followed by an argument of *c* translates any *c* in the file into a blank space. The argument *c* can be any character that is not used in elsewhere in the file.

Many people at HAO use a command of .tr ~ at the top of the file and then enter tildes (~) as needed for blank spaces. If the file contains words with real tildes, use some other character that does not appear in the file (@, for example) for the argument.

**Important Point:** There are times when this command is useful, as shown in the example. However, don't get in the habit of using strings of tildes to produce blank spaces when there are more efficient commands that produce the same results (indent, temporary indent, mark and return, and tables, for example).

## **Example**

**.tr**

### **Input:**

**.tr**

Use the translate command (.tr) to leave  
blank ~~~~~ spaces when  
something has to be typed in later.

### **Output:**

Use the translate command (.tr) to leave blank spaces when something has to be typed in later.

## Section 24: Indented Paragraphs

**Important Point:** The indented paragraph (.ip) command is much more complicated than the indent (.in) and temporary indent (.ti) commands. Look at section 3 to see if those commands can produce the results you want before deciding to use the .ip command.

.ip

The **indented paragraph** command without arguments leaves a blank line and indents the following lines about 3/8 inch (5 en's). The .ip command without arguments produces the same results as a command of .in +5n. However, .ip *with* arguments can do much more than the .in command, so keep reading.

.ip "some words" N

The **indented paragraph** command leaves a blank line and produces a label on the first line by putting the argument of "some words" at the left margin.

The text is indented by *N* amount from the left margin. If *N* is not specified, the text is indented 5 en's. Like other commands that create horizontal space, the argument *N* may be in inches, centimeters, en's, em's, or points. For example, entering

.ip "Snakes" 1i

Snakes are not common...

produces

Snakes                    Snakes are not common...

The .ip command is reset by another .ip command, by any of the paragraph commands, .pp, .lp, .np, .\*N, or .\*n, and by the section heading commands of .sh and .uh.

Sometimes the first word of the text in an .ip is indented 1 extra space. There is no remedy for this at the present time.

### Cross References

For more variations with .ip, see sections 25 and 29.

For more on en's, em's, and points, see section 3.

## **Example**

.ip

### **Input:**

.ip

Snakes are not common around NCAR, although bull snakes, garter snakes, and hognosed snakes are likely inhabitants of the foothills region in general.

.ip "one"

Prairie rattlesnakes may also be seen occasionally in the open space surrounding NCAR.

.ip "two"

Fox squirrels and Abert's (tufted-eared) squirrels also live around NCAR.

Abert's are seen in both their brown and black color phases.

.ip "chipmunks" 1.5i

Another small mammal around NCAR is the least chipmunk.

It moves quickly and stops often, rapidly flicking its tail up and down.

### **Output:**

Snakes are not common around NCAR, although bull snakes, garter snakes, and hognosed snakes are likely inhabitants of the foothills region in general.

one Prairie rattlesnakes may also be seen occasionally in the open space surrounding NCAR.

two Fox squirrels and Abert's (tufted-eared) squirrels also live around NCAR. Abert's are seen in both their brown and black color phases.

chipmunks                  Another small mammal around NCAR is the least chipmunk. It moves quickly and stops often, rapidly flicking its tail up and down.

## Section 25: More Indented Paragraphs

**Important Point:** The indented paragraph (.ip) command is much more complicated than the indent (.in) and temporary indent (.ti) commands. Look at section 3 to see if those commands can produce the results you want before deciding to use the .ip command. Read section 24 before reading section 25!

**.ip "long label" N** If the label is longer than the space allocated by the argument of *N*, the text will automatically start on the line following the label.

Use a space command (.sp) to leave a blank line and start a new line of text under the same label.

Use indent (.in) or temporary indent (.ti) commands to further indent text under an indented paragraph.

Use a break command (.br) to start text on the line following a short label.

**.ip "" N** The indented paragraph command without any words (but with two sets of double quote marks) produces a paragraph indented *N* amount, without any label.

The .ip command is reset by any another .ip command, the paragraph commands of .pp, .lp, .np, .\*N, or .\*n, and the section heading commands of .sh and .uh.

**Cross References** For an introduction to .ip, see section 24.

For more uses of .ip, see section 29.

## Example

.ip

### Input:

.ip "Mule Deer"

Mule deer frequent the NCAR mesas.

They are well accustomed to their human neighbors,  
and are often seen grazing around the  
building or parking lot.

.sp

It's not unusual for deer to be on the winding road leading up  
to the mesa.

.ip " Chipmunks and Squirrels" .5i

The least chipmunk inhabits the NCAR area.

It is smaller than a ground squirrel  
and has stripes that run from the head to the hips.

.in +1i

Golden-mantled ground squirrels are  
occasionally cited around NCAR.

Their heads are golden tan,  
and they have two black stripes from the shoulders to the hips.

.ip "" 1.5i

Skunks, raccoons, jackrabbits, coyotes, and foxes are occasionally  
seen in the area.

.lp

Other Wildlife

### Output:

#### Mule Deer

Mule deer frequent the NCAR mesas. They are well accustomed to their human  
neighbors, and are often seen grazing around the building or parking lot.

It's not unusual for deer to be on the winding road leading up to the mesa.

#### Chipmunks and Squirrels

The least chipmunk inhabits the NCAR area. It is smaller than a ground squirrel  
and has stripes that run from the head to the hips.

Golden-mantled ground squirrels are occasionally cited around  
NCAR. Their heads are golden tan, and they have two black  
stripes from the shoulders to the hips.

Skunks, raccoons, jackrabbits, coyotes, and foxes are occasionally  
seen in the area.

#### Other Wildlife

## Section 26: Numbered Paragraphs

The 3 numbered paragraph commands, .np, .\*N, and .\*n, all produce numbered paragraphs in various formats.

For all 3, use a space command (.sp) to leave a blank line and then continue text under the numbered paragraph.

For additional indents under a numbered paragraph, use the indent (.in) or temporary indent (.ti) command.

The .np, .\*N, and .\*n commands are reset by paragraph commands of .pp and .lp and by the section heading command of .sh. This means that if you enter another .np, .\*N, or .\*n command after the .pp, .lp, or .sh, the numbering sequence starts over with 1.

The indented paragraph command (.ip) or the unnumbered heading command (.uh) can be used in combination with numbered paragraphs. Either command returns text to the original left margin and then performs its usual .ip or .uh function. The next numbered paragraph command resumes numbering where it left off, *not* at a new sequence beginning with 1.

### .np

The **numbered paragraph** command produces paragraphs that are numbered sequentially beginning with 1. The number, surrounded by parens, is at the left margin.

### Cross References

For the numbering formats produced by .\*N and .\*n, see sections 27 and 28.

To change the beginning point of the numbering sequence, see sections 27 or 28.

## **Example**

.np

### **Input:**

.np

In the late fall and winter, mule deer frequent the NCAR meadow. The herds are usually made up of about a dozen does and fawns, often escorted by one buck.

.sp

Young bucks appear alone or in small groups.

.np

Tracks in the snow indicate that the deer frequent the mesa tops in winter.

.in +10

Cottontail rabbits also leave tracks in the snow.

.uh "Birds"

.np

The Boulder area is a stopping-off place for most of the bird species known in the Rocky Mountains and Great Plains.

.lp

Many of these species can be recognized on the NCAR site.

.np

Stellar's jays, meadowlarks, magpies, and robins are common. Both the downy and hairy woodpeckers are sometimes seen.

### **Output:**

- (1) In the late fall and winter, mule deer frequent the NCAR meadow. The herds are usually made up of about a dozen does and fawns, often escorted by one buck.

Young bucks appear alone or in small groups.

- (2) Tracks in the snow indicate that the deer frequent the mesa tops in winter.

Cottontail rabbits also leave tracks in the snow.

### **Birds**

- (3) The Boulder area is a stopping-off place for most of the bird species known in the Rocky Mountains and Great Plains.

Many of these species can be recognized on the NCAR site.

- (1) Stellar's jays, meadowlarks, magpies, and robins are common. Both the downy and hairy woodpeckers are sometimes seen.

## Section 27: More Numbered Paragraphs

.\*N f

The **star capital N** command numbers paragraphs in various sequences. The numbering format is controlled by the argument *f*. For example,

<u>Command</u>	<u>Numbered Sequence is</u>
*N	1. 2. 3. 4....
.*N 001	001. 002. 003. 004....
.*N i	i. ii. iii. iv....
.*N I	I. II. III. IV....
.*N a	a. b. c. d....
.*N A	A. B. C. D....

Like other star commands, .\*N is local to HAO. Enter the local command (.lo) near the top of the file that contains .\*N commands.

.nr \$p -1

Use this command to change the beginning number for any of the numbered paragraph commands.

The notation for the argument (-1) means the number entered here is *one less* than the new starting number. For example, to obtain a beginning number of 8, enter:

.nr \$p 7

### Cross References

See section 26 for general rules that apply to all 3 numbered paragraph commands, .\*N, .\*n, and .np.

For other styles of numbered paragraphs, see sections 26 and 28.

**Example**

.\*N, .nr \$p

**Input 1:****.lo****.\*N**

The first point.

**.\*N**

The second point.

**.\*N**

The third point.

**.lp****.\*N 001**

The left paragraph (.lp) command ended the .\*N series.

This is the first point in a new series..

**.\*N 001**

The second point.

**.pp****Output 1:**

1. The first point.
2. The second point.
3. The third point.

001. The left paragraph (.lp) command ended the .\*N series. This is the first point in a new series..

002. The second point.

**Input 2:****.lo****.nr \$p 8****.\*N i**

This is now the ninth point,  
and the sequence is in small roman numerals.

**.\*N i**

This is now the tenth point.

**.sh 2 "Capital Letters" 1.1****.\*N A**

The section head (.sh) command ended the .\*N i series.

This is the first point in a new series.

**Output 2:**

- ix. This is now the ninth point, and the sequence is in small roman numerals.
- x. This is now the tenth point.

**1.1. Capital Letters**

A. The section head (.sh) command ended the .\*N i series. This is the first point in a new series.

## Section 28: Final Section of Numbered Paragraphs

.\*n f

The **star small n** command numbers paragraphs and right justifies the numbers.

Aligned like this

9  
10  
100

Not like this:

9  
10  
100

The numbers are indented about .25 inches from the margin.

The argument *f* is any character that is to follow the number, such as a right paren or a period.

Since the .\*n command is local to HAO, enter the local command (.lo) near the top of the file that contains .\*n commands.

.nr \$p #-1

Use this command to change the beginning number for any of the numbered paragraph commands.

The notation for the argument (#-1) means the number entered here is *1 less* than the new starting number. For example, to obtain a beginning number of 8, enter:

.nr \$p 7

### Cross References

See section 26 for general rules that apply to all 3 numbered paragraph commands, .\*N, .\*n, and .np.

For other styles of numbered paragraphs, see sections 26 and 27.

## Example

.\*n, .nr \$p

### Input:

.lo  
.nr \$p 8  
.n

A number of lichens grow on the rocks and trees on the NCAR site.

.sp

Among these are the granite lichen  
(light green with orange dots), the chip lichen (lemon yellow),  
flat gray lichens, and shaggy gray lichens.

.\*n .

Lichens are the first organisms to colonize bare rock.

.\*n )

Each lichen is a partnership of two different kinds of plants.  
Most of the visible portion of the lichen is a fungus, which  
cannot create its own food.

.\*n ).

Just below the surface of the lichen, held in a meshwork  
of fungal cells, are myriads of green algae cells  
which manufacture food,  
with the aid of sunlight.

.nr \$p 99

.\*n ----

Some of this food is used by the fungus, while the fungal  
cell network provides the moist living space needed by the algae.

### Output:

9 A number of lichens grow on the rocks and trees on the NCAR site.

Among these are the granite lichen (light green with orange dots), the chip lichen  
(lemon yellow), flat gray lichens, and shaggy gray lichens.

10. Lichens are the first organisms to colonize bare rock.

11) Each lichen is a partnership of two different kinds of plants. Most of the visible  
portion of the lichen is a fungus, which cannot create its own food.

12.) Just below the surface of the lichen, held in a meshwork of fungal cells, are  
myriads of green algae cells which manufacture food, with the aid of sunlight.

100---- Some of this food is used by the fungus, while the fungal cell network provides  
the moist living space needed by the algae.

## Section 29: Base Indents

.ba  $\pm N$

The **base indent** command produces several complex indentation formats when combined with various paragraph commands. In fact, it takes two commands to begin a base indent format, and it also takes two commands to end a base indent format. In both cases, the sequence must be a .ba command followed on the next line by a paragraph command of .pp, .lp, .ip, .np, .\*N, or .\*n, or by the section heading command of .sh.

The argument  $N$  may be in inches, centimeters, en's, em's, or points. When no unit is specified, the default unit is en's.

The base indent command is used most often to further indent numbered paragraphs or indented paragraphs.

The base indent command is reset only by another .ba command.

### The Cumulative Effect of .ba Commands

If a base indent command is later followed by another base indent command, the arguments ( $N$ ) are added together for a cumulative effect. For example, if a .ba +.5i is followed by a numbered paragraph, and then followed by another .ba +.5i and numbered paragraph, that last paragraph will be indented a total of 1 inch.

After this sequence, there are 2 ways to get back to the original margin. Enter a command of .ba -1i (which counters the indents from the 2 commands of +.5i), followed by a paragraph or section heading command. Or, enter a command of .ba Ø, followed by a paragraph or section heading command. Use the .ba Ø command to avoid figuring out the cumulative amount of several different indent commands.

One way to avoid the confusion of the cumulative effect is to use absolute arguments, such as .ba .5i (no plus or minus).

**Important Point:** To begin and end a base indent section, always use a base indent command followed by one of the paragraph commands or a section head command.

**Cross References** For the explanation of  $\pm N$ , see p. 14.

For more on en's, em's, and points, see section 3.

For explanations of indented or numbered paragraphs that are often used with the .ba command, see sections 24-28.

## Example

.ba

### Input:

.ba +.5i

.np

One of the most prominent plants on the NCAR site is the yucca.  
Yucca remains green all year and produces a stalk of white flowers.

.np

The yucca blooms in early summer.

Dozen of tiny moths flutter around the blossoms.

.ba -.5i

.lp

These are the yucca moths,  
and the life cycle of the yucca depends on their activities.

.ba 1i

.ip "Pollination"

As soon as the flowers open,  
the moths flutter inside, scraping pollen grains  
from the anthers of the flowers and depositing  
the pollen grains on the pistils, thus insuring pollination.

.ba 2i

.ip "Larvae"

The moths lay their eggs in the flower cup;  
the eggs develop into larvae inside the yucca seed pod.

.ba Ø

.pp

After eating much of the plant tissue inside the pod,  
the larvae bore their way out.

They then lower themselves to the ground by silken threads and  
burrow into the ground beneath the plant.

### Output:

- (A) One of the most prominent plants on the NCAR site is the yucca. Yucca remains green all year and produces a stalk of white flowers.
- (B) The yucca blooms in early summer. Dozen of tiny moths flutter around the blossoms.

These are the yucca moths, and the life cycle of the yucca depends on their activities.

#### Pollination

As soon as the flowers open, the moths flutter inside, scraping pollen grains from the anthers of the flowers and depositing the pollen grains on the pistils, thus insuring pollination.

#### Larvae

The moths lay their eggs in the flower cup; the eggs develop into larvae inside the yucca seed pod.

After eating much of the plant tissue inside the pod, the larvae bore their way out. They then lower themselves to the ground by silken threads and burrow into the ground beneath the plant.

## Section 30: Some Backslash Commands

Backslash commands (\) can be entered anywhere in-line or at the beginning of a new line.

\\*(td Prints today's date.

\(space) The **unpaddable space** command leaves 1 blank space that cannot be added to when troff justifies the lines. Further, the unpaddable space leaves 1 blank space but does not allow a break for a new line to occur at that point. If, for example, you typed in

Dr. L. S. Skywalker

and troff inserted extra spaces to right justify the line, you might get a spaced-out result like this:

Dr. L. S. Skywalker

To correct it, enter

Dr.\ L.\ S.\ Skywalker

For another way to leave blank spaces, see the .tr command, section 23.

\&

Use the **non-printing character** if you want a line to begin with a dot. (Without the non-printing character, troff tries to turn any line that begins with a dot into a command, which can create havoc.) For example,

<u>Input</u>	<u>Output</u>
\&....'Tis nobler to give	....'Tis nobler to give

The non-printing character can be used to create blank space at the top of the page. See section 36 for details.

\u

The **up** command moves the type after it up about half a line. Use this command to superimpose a special symbol like  $\dagger\dagger$  to mark a note.

For footnotes, use the footnote commands to superimpose the numbers. See section 12 for details. For superscripts and subscripts in mathematical equations, see "Typesetting Mathematical Equations -- User's Guide."

\d

The **down** command moves the type down about half a line. If you use the \u command to make a superscript, use the \d command to return to the normal line of type. For example,

<u>Input</u>	<u>Output</u>
See Table C\u++\d for...	See Table C <sup>++</sup> for...

\l'Nc'

The **horizontal line** command makes horizontal lines of the length (*N*) you specify with the character (*c*) you specify. For example,

<u>Input</u>	<u>Output</u>
\l'6iX'	6 inch line of XXX
\l'4i+'	4 inch line of +++
\l'3i_'	3 inch line of __

**Caution:** To create lines of stars (\*\*\*) or dots (...), use the non-printing character of \& before the star or dot, since a star or dot have special meanings in commands. For example,

<u>Input</u>	<u>Output</u>
\l'6i\&*	6 inch line of ***
\l'4i\&.	4 inch line of ...

If you forget the \&, your job will get caught in an *infinite loop* on the computer and no other jobs will be processed!

\"

The in-line **comment command** allows you to write a comment to yourself that will never be printed. The comment only appears on the screen as part of the file. This command can be used to remind yourself what a command does. For example,

.\*N \"numbers paragraphs, no parens

After entering the comment, hit the return and continue entering regular text or commands on the next line.

.\"

This **comment command** also allows you to write a comment to yourself that will never be printed. The comment appears on the screen only. Since this is a dot command, it must go at the beginning of a line. For example, if you want to remind yourself to check on a diagram before printing the file, you could enter

.\" Decide about diagram size before printing file

After entering the comment, hit the return and continue entering regular text or a command on the next line.

## Section 31: Right Justification

*Right justification* means the lines end evenly at the right margin. The term *line adjustment* means the same thing. The lines are right justified unless you command them not to be.

- .na              The **no adjust** command removes right justification. The right margin is slightly ragged (called *ragged right*).
- .ad              The **adjust** command restores right justification.

## **Example**

.na, .ad

### **Input:**

.pp

.na

After the yucca moth larvae burrow into the ground,  
they construct cocoons in which to spend the winter.

In late spring they emerge as adult yucca moths.

Even though some of the yucca pod was destroyed by the larvae  
the year before, enough seeds survive to insure that the plant  
can reproduce.

.pp

The yucca supports another interacting group of organisms.  
At flowering time plant lice (aphids) congregate in great numbers  
on yucca plants to suck the nutrient juices.

.pp

.ad

After the yucca moth larvae burrow into the ground,  
they construct cocoons in which to spend the winter.

In late spring they emerge as adult yucca moths.

Even though some of the yucca pod was destroyed by the larvae  
the year before, enough seeds survive to insure that the plant  
can reproduce.

.pp

The yucca supports another interacting group of organisms.  
At flowering time plant lice (aphids) congregate in great numbers  
on yucca plants to suck the nutrient juices.

### **Output:**

After the yucca moth larvae burrow into the ground, they construct cocoons in  
which to spend the winter. In late spring they emerge as adult yucca moths. Even  
though some of the yucca pod was destroyed by the larvae the year before, enough seeds  
survive to insure that the plant can reproduce.

The yucca supports another interacting group of organisms. At flowering time  
plant lice (aphids) congregate in great numbers on yucca plants to suck the nutrient  
juices.

After the yucca moth larvae burrow into the ground, they construct cocoons in  
which to spend the winter. In late spring they emerge as adult yucca moths. Even  
though some of the yucca pod was destroyed by the larvae the year before, enough seeds  
survive to insure that the plant can reproduce.

The yucca supports another interacting group of organisms. At flowering time  
plant lice (aphids) congregate in great numbers on yucca plants to suck the nutrient  
juices.

## Section 32: Page Offset

*Page offset* means the unused white space on a printed document. The term is used in UNIX word processing to refer *only* to the unused white space at the left margin of the page.

.po  $\pm N$

The **page offset** command resets the left margin. Normally, there is a left margin of about 1.13 inches (about 1 1/8 inches). There will always be a minimum left margin of about .13 inches. The page offset command can move the margin to the left on the page up to that minimum boundary of .13 inches. Therefore, you need to mentally add about .13 inches to any page offset command to predict how far type will be indented from the left edge of the paper. For example,

Command

.po -.5i

.po 2i

.po +.5i

Result

decreases left margin .5i

left margin is 2.13 inches  
from edge of paper

increases left margin .5i

Like other horizontal spacing commands, the argument  $N$  may be in inches, centimeters, em's, en's, or points. Inches or centimeters are generally the most convenient units to use.

To change the page offset of a whole document, enter the .po command near the top of the file before any text is entered.

The page offset command is reset only by another page offset command.

**Caution:** To change the page offset within the text, the page offset command must immediately *follow* a command that causes a break for a new line, such as a break (.br) or one of the paragraph commands. If the command that causes a break is omitted, the page offset command will not work.

**Cross References** To combine the .po command with a different line length, see section 33.

For the explanation of  $\pm N$ , see p. 14.

For more on em's, en's, and points, see section 3.

To change the top and bottom margins of the page, see section 37.

## Example

.po

### Input:

Fallen trees and dead shrubs are as much a part of the ecology as are the living trees and plants.

.br

.po -3c

They decompose slowly through the action of fungi, ants, beetles, and other woodboring insects.

Exposed inner portions of dead logs often reveal that they are already breaking up into rectangular chips -- a sure sign of fungal action.

.lp

.po +4c

Rain and sunlight, which contributed to the vitality of the trees when they were alive, now become part of the decomposing process.

.pp

Eventually -- perhaps in 50 to 75 years -- the materials of a fallen tree will be returned to the earth to be reused by other plants and animals.

.pp

.po .5i

It took at least as many years for these trees to grow to maturity as it takes for them to decompose.

.lp

.po 1.25i

Fallen trees...

### Output:

Fallen trees and dead shrubs are as much a part of the ecology as are the living trees and plants.

They decompose slowly through the action of fungi, ants, beetles, and other woodboring insects. Exposed inner portions of dead logs often reveal that they are already breaking up into rectangular chips -- a sure sign of fungal action.

Rain and sunlight, which contributed to the vitality of the trees when they were alive, now become part of the decomposing process.

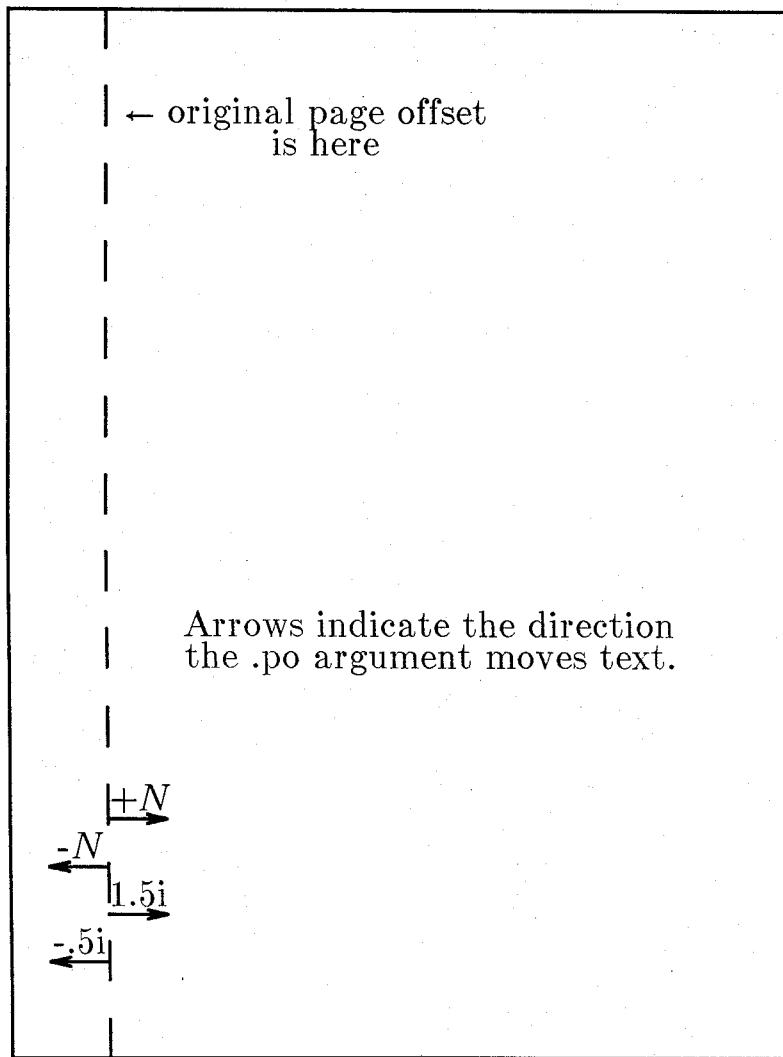
Eventually -- perhaps in 50 to 75 years -- the materials of a fallen tree will be returned to the earth to be reused by other plants and animals.

It took at least as many years for these trees to grow to maturity as it takes for them to decompose.

Fallen trees...



## The Effects of Page Offset Arguments



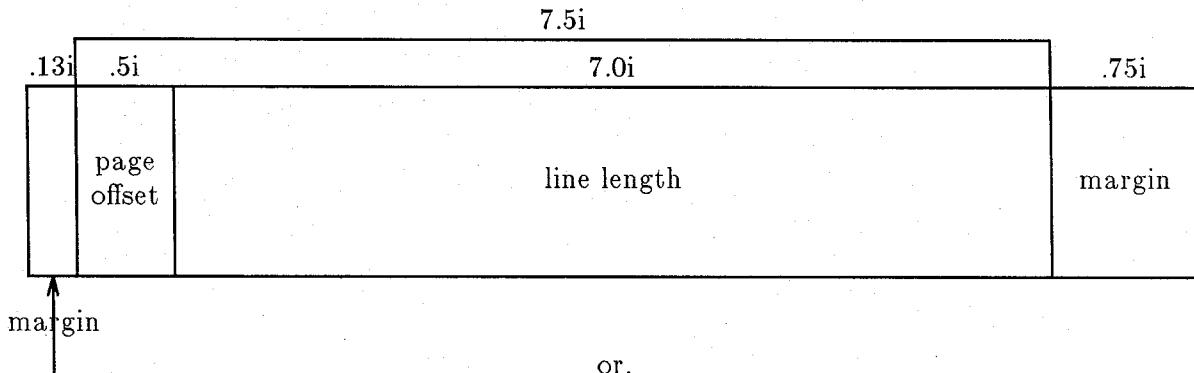
## Section 33: Page Offset with Line Length

The page offset (.po) command is usually used in combination with the line length (.ll) command. To change the page offset and line length of a whole file, enter both commands near the top of the file before any text is entered.

Changing the page offset and line length in the middle of a file first requires a command that causes a break for a new line (such as a .br or .pp command). The order of the commands must be as follows:

.br (or any command that starts a new line)  
.po  $\pm N$   
.ll  $\pm N$

The maximum horizontal space for type is just over 7.5 inches. The page offset plus the line length may not exceed this maximum. For example,



**Cross References** For more on line length, see section 18.

For more on page offset, see section 32.

For another way to get more words on a line, use a smaller type size (see section 8).

## **Example**

.po, .ll

### **Input:**

.br  
.po -1i  
.ll +1.5i

This is an example of the maximum amount of type that may be entered on one line of output.

The right margin may not be less than about .75 inches.

.sp  
.po +1i  
.ll -1.5i

Dead plant materials,  
ranging in size from tree trunks to grass blades,  
are a necessary part of the economy of any biologic community.  
As they decompose, substances essential to life are released  
for use by future generations of plants and animals.

.sp  
.po +1i  
.ll -2i

Much of the work of decomposition is carried out by fungi.  
The fungi rhizomes grow and spread in dark, damp places,  
digesting dead organic materials.

### **Output:**

is an example of the maximum amount of type that may be entered on one line of output. The right margin may not be less than about .75 inches.

Dead plant materials, ranging in size from tree trunks to grass blades, are a necessary part of the economy of any biologic community. As they decompose, substances essential to life are released for use by future generations of plants and animals.

Much of the work of decomposition is carried out by fungi.  
The fungi rhizomes grow and spread in dark, damp places,  
digesting dead organic materials.

## Section 34: Hyphenation

Words at the end of lines are hyphenated automatically as needed.

.nh                  The **no hyphenation** command removes automatic hyphenation.

.hy                  The **hyphenation** command restores automatic hyphenation.

.hw *word(s)*      The **hyphenate word** command uses minus signs embedded within a particular word to create hyphenation that is different than the hyphenation in the default. The .hw command affects the hyphenation of that particular word whenever it appears in the file after the .hw command.

The list of words following the .hw commands can contain about 128 characters in 1 file.

### The Main Uses of the .hw Command

Use the .hw command to correct hyphenation errors made by troff. Troff follows a set of rules when hyphenating, but since there are many exceptions to the "rules" in the English language, troff makes mistakes now and then. For example, "rearrange" is hyphenated as "rear-range" by troff. To change it, enter the following command on a separate line before the word appears in the text:

<u>Command</u>	<u>Result</u>
.hw re-arrange	re- arrange (only hyphenation allowed)

Use the .hw command to prevent a word from being hyphenated. For example,

<u>Command</u>	<u>Result</u>
.hw rearrange	rearrange (no hyphenation possible)

\%*word*            Use this command to change hyphenation of a word *one time only*, not throughout the whole document.

**Important Point:** When typing word groups that already contain hyphens (brother-in-law, for example), enter the whole word group on the same line to avoid extra spaces between the words in the output.

## **Example**

**Input:**

**.hw frus-tration**

**.hw frustration**

**.hw frustration patience humor**

the **\%frustration** level

the **frus\%tration** level

**.hw, \%**

**Output at End of Line:**

frus-  
tration (only possible hyphenation)

frustration (no hyphenation allowed)

frustration patience humor  
(no hyphenation allowed in these words)

the frustration level  
(frustration not hyphenated this time only)

the frus-  
tration level  
(only possible hyphenation, this time only)

## Section 35: Title Page

.tp

The **title page** command allows for spacing at the top of the page. Headers and footers are suppressed and the title page does not increase the number of the next page.

This command does not work correctly in the middle of a file. Use the title page command at the beginning of a new file, or put the title page in a separate file.

When deciding how much blank space to leave at the top of the page, remember that there is already a top margin of about 1 inch. Any spacing command you enter will be *in addition* to the existing 1 inch margin.

**Cross References** To produce blank space at the top of a page without suppressing the headers, footers, and numbering system, see section 36.

### Input for the Following Page:

.lo

.he "Section 35>Title Page"

.fo "%January, 1986 DRAFT"

.tp

.ce 5

.\*s 14

.sp 3i

Marmots and Sunbathing:

.sp 2

Observations and Research

.sp 2i

.\*s 11

a paper presented to  
the faculty of the University

.sp 2

January, 1986

Marmots and Sunbathing:  
Observations and Research

a paper presented to  
the faculty of the University

January, 1986

## Section 36: Blank Space at the Top of the Page

A spacing (.sp) command does not work at the top of a page before any text is entered.

\&

Enter the **non-printing character** to create blank space at the top of a page. On the next line, enter a spacing command (.sp) with an argument that is the amount of blank space you want.

The normal top margin is about 1 inch. The spacing command you enter is added onto the existing 1 inch margin.

Unlike the title page command (.tp), headers and footers are printed in their usual locations. The .sp command affects where the first line of the text, not the header, appears on the page. Using the .sp command allows the page number to increase as usual.

### Cross Reference

To eliminate headers and footers and not increase the page numbering, use the title page command to produce blank space at the top of the page. See section 35.

### Input for the Following Page:

\&

.sp 2i

.\*s 14

.b

Example

.sp

.ce

Blank Space at the Top of the Page

.\*s 11

.r

.pp

Notice that this page starts lower on the paper than usual.

The non-printing character can be used to leave blank space at the top of the page.

## **Example**

### **Blank Space at the Top of the Page**

Notice that this page starts lower on the paper than usual. The non-printing character can be used to leave blank space at the top of the page.

## Section 37: Top and Bottom Margins

The margin between the edge of the paper and the header or footer is normally about .5 inches. The space between the edge of the paper and the first and last line of type is about 1 inch. Use the commands below to change these margins.

.m1 N

The **margin one** command changes the space between the top of the page and the header. The argument *N* can be any number, *but not Ø*. The default number is 4. Numbers smaller than 4 *decrease* the white space; numbers larger than 4 *increase* the white space.

.m2 N

The **margin two** command changes the space between the header and the first line of text. The argument *N* can be any number, including Ø. The default number is 2. Numbers smaller than 2 *decrease* the white space; numbers larger than 2 *increase* the white space.

.m3 N

The **margin three** command changes the space between the bottom line of text and the footer. The argument *N* can be any number, including Ø. The default number is 2. Numbers smaller than 2 *decrease* the white space; numbers larger than 2 *increase* the white space.

.m4 N

The **margin four** command changes the space between the footer and the bottom of the page. The argument *N* can be any number, including Ø. The default number is 4. Numbers smaller than 4 *decrease* the white space; numbers larger than 4 *increase* the white space.

### Squeezing Extra Lines onto the Page

Entering a margin command at the top of a file affects the margins of the whole document. Entering a margin one, two, or three command (.m1, .m2, .m3) in the middle of a file affects the margins of the *next* page and all pages thereafter. Entering a margin four (.m4) command in the middle of a file affects the margin of the *current* page and all pages thereafter. Here are some examples:

<u>Command</u>	<u>Result</u>
.m3 1	adds 1 line of text at bottom of next page
.m3 Ø	adds 2 lines of text at bottom of next page
.m2 1	adds 1 line of text at top of next page
.m2 Ø	adds 2 lines of text at top of next page
.m3 Ø	moves header up 2 lines on next page
.m4 Ø	moves footer down 2 lines on <i>current</i> page

## .m1, .m2, .m3, .m4

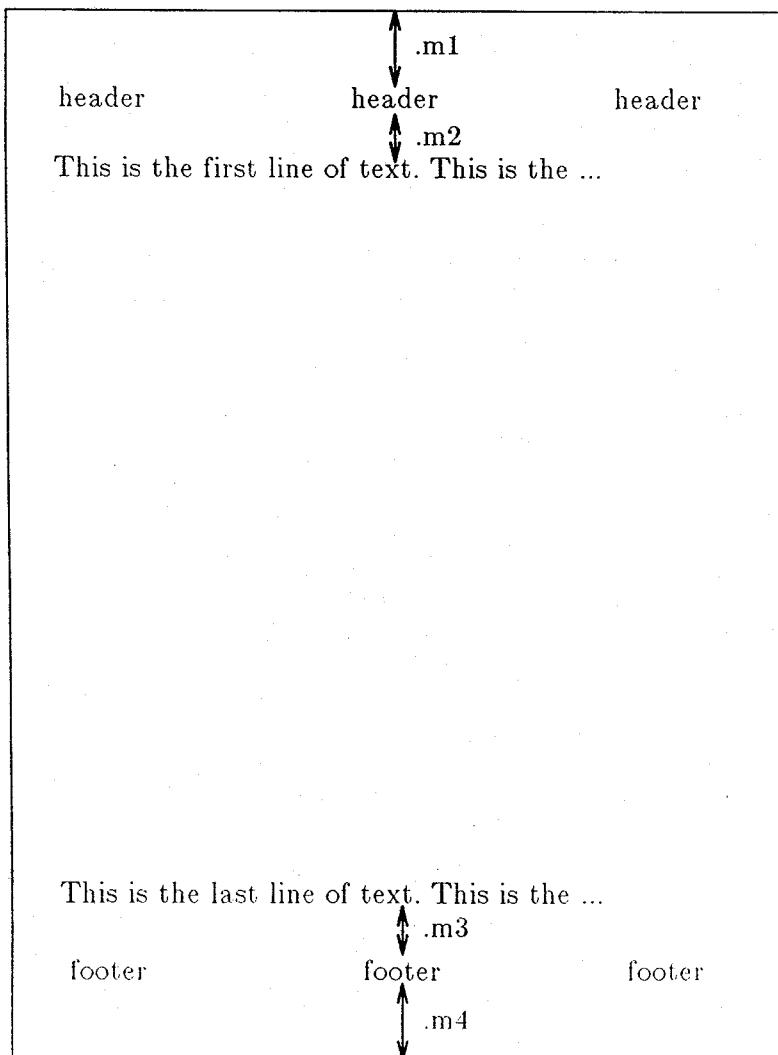
This command does not always work in the middle of a file. If you have trouble, change the location of the .m commands in the file.

It is possible to push the header to the top edge of the paper, but there will always be a bottom margin of .25 inches. To run text the maximum page length, use the following set of margin commands:

```
.m1 1  
.m2 Ø  
.m3 Ø  
.m4 Ø
```

**Caution:** The .m1 command is the only one that does not take an argument of Ø. Never try any margin command with a negative argument (-1, for example). A command such as .m2 -1 catches the document in an *infinite loop* on the computer and no other jobs can be processed. This is also known as *getting hung*.

### Margin Locations on the Page



**Input:**

```
.m1 1
.m2 ø
.m3 ø
.m4 ø
```

```
.he 'left header'center header"
 fo 'left footer'%right footer'
```

```
.ps +3
```

```
.b
```

Example

```
.lp
```

This page is an example of getting the most lines possible on the page by using the .m commands.

Compare the top and bottom margins of this page with other pages.

**Output:**

This page is an example of getting the most lines possible on the page by using the .m commands. Compare the top and bottom margins of this page with other pages.

## Section 38: Block Keeps

.(b, .)b

A *keep* is a body of text that must be "kept" together on one page, such as a list, table or a diagram.

.(b m f

The **begin block keep** command opens a keep in the text exactly where the command is entered. If the entire keep does not fit on the current page, the command begins a new page, leaving white space at the bottom of the first page.

It is possible to place commands for lists (see section 14) or centered lines (see section 5) within a block command to keep all the text on the same page.

.)b

The **end block keep** command follows the last line of the text that is inside the keep.

### Arguments with Block Keeps

The first argument (*m*) controls the left margin of the keep. Without a second argument (*f*), the lines are not filled. For example,

<u>Command</u>	<u>Result</u>
.(b	Indented 5 en's (3/8 inch) from left margin
.(b I	Indented 5 en's left margin (same as the default)
.(b M	At the original left margin
.(b L	At the current left margin
.(b C	Centered between the current margins

A second argument of *F* fills the lines. Remember, you must enter a first argument in order to enter a second argument. For example,

<u>Command</u>	<u>Result</u>
.(b I F	Indented 5 en's, lines filled
.(b M F	At original left margin, lines filled

### **Sequence of Commands for Block Keeps**

Treat the text of the keep as if it is a separate document. Enter commands to change the keep line length, point size, or type style after the opening command of .(b. Enter the commands to reset the line length, point size, or type style before the closing command of .)b. For example,

```
.(b  
.ll 4i  
.i  
Text of the keep goes here.  
.ll 6i  
.r  
.)b
```

produces a block keep with 4 inch lines in italic type. The text after the keep is restored to 6 inch lines and roman type.

### **Block Keep to Leave Extra White Space**

You may want to leave a specified amount of white space where a diagram will be added later and the diagram must be kept on 1 page. To do this, use block keep commands along with space (.sp) commands. For example,

<u>Command</u>	<u>Result</u>
.(b	leaves a blank space
.sp 3i	of 3 inches,
.)b	all on the same page

**Cross References** For another keep format, see section 39.

## **Example**

**.(b, .)b**

### **Input:**

**.lp**

The commands for block keeps are used to keep certain text together on 1 page.

If the whole keep does not fit on the current page,  
the keep is

forced onto the top of the next page,  
leaving white space at the bottom of the current page.

A block keep can be used to keep a list on 1 page, as in this example.

(The list and text entered after it in this example are forced onto the following page by the block keep command.)

**.(b**

Mule deer

Skunk

Coyote

Raccoon

Marmot

Abert's squirrel

Golden-mantled ground squirrel

Least chipmunk

Fox

Jackrabbit

Red-lipped rock lizard

Hognosed snake

Prairie rattlesnake

Bull snake

Garter snake

Wolf spider

Thatching ants

**.sp**

Mule deer

Skunk

Coyote

Raccoon

Marmot

Abert's squirrel

Golden-mantled ground squirrel

Least chipmunk

Fox

Jackrabbit

Red-lipped rock lizard

Hognosed snake

Prairie rattlesnake

Bull snake

Garter snake

**[Continuation of Input:]**

Wolf spider  
Thatching ants

.sp

Mule deer  
Skunk  
Coyote  
Raccoon  
Marmot  
Abert's squirrel  
Golden-mantled ground squirrel

.)b

.lp

All text entered after the block keep continues here.

**Output:**

The commands for block keeps are used to keep certain text together on 1 page. If the whole keep does not fit on the current page, the keep is forced onto the top of the next page, leaving white space at the bottom of the current page. A block keep can be used to keep a list on 1 page, as in this example. (The list and the text entered after it in this example are forced onto the following page by the block keep command.)

[Continuation of Output:]

.(b, .)b

Mule deer  
Skunk  
Coyote  
Raccoon  
Marmot  
Abert's squirrel  
Golden-mantled ground squirrel  
Least chipmunk  
Fox  
Jackrabbit  
Red-lipped rock lizard  
Hognosed snake  
Prairie rattlesnake  
Bull snake  
Garter snake  
Wolf spider  
Thatching ants

Mule deer  
Skunk  
Coyote  
Raccoon  
Marmot  
Abert's squirrel  
Golden-mantled ground squirrel  
Least chipmunk  
Fox  
Jackrabbit  
Red-lipped rock lizard  
Hognosed snake  
Prairie rattlesnake  
Bull snake  
Garter snake  
Wolf spider  
Thatching ants

Mule deer  
Skunk  
Coyote  
Raccoon  
Marmot  
Abert's squirrel  
Golden-mantled ground squirrel

All text entered after the block keep continues here.

## Section 39: Floating Keeps With Horizontal Lines

A floating keep is a body of text (such as a table or list) that must be kept together and placed at the bottom or top of a page.

.(z m f

The **begin floating keep** command floats the text after it to the *bottom* of the current page or the *top* of the next page, depending on available space.

.)z

The **end floating keep** command follows the last line of keep text.

### Arguments With Floating Keeps

The first argument (*m*) controls the left margin of the keep. Without a second argument (*f*), the lines are not filled. For example,

<u>Command</u>	<u>Result</u>
.(z	At the original left margin
.(z M	At the original left margin (same as default)
.(z I	Indented 5 en's (3/8 inch) from left margin
.(b L	At the current left margin
.(b C	Centered between the current margins

A second argument of *F* fills the lines. Remember, you must enter a first argument in order to enter a second argument. For example,

<u>Command</u>	<u>Result</u>
.(z I F	Indented 5 en's, lines filled
.(z M F	At original left margin, lines filled

### Sequence of Commands

Treat the text of the keep as if it is a separate document. Enter commands to change the keep line length, point size, or type style after the opening command of .(z. Enter the commands to reset the line length, point size, or type style before the closing command of .)z. For example,

```
.(z
.ll 4i
.sz 8
Text of the keep goes here.
.ll 6i
.sz 10
.)z
```

produces a floating keep with 4 inch lines and 8-point type. The text after the keep is restored to 6 inch lines and 10-point type.

## Floating Keep to Leave Extra White Space

.(z, .)z, .hl

You may want to create a floating keep that contains only white space where a diagram will be inserted later. To do this, use floating keep commands along with space (.sp) commands. For example,

<u>Command</u>	<u>Result</u>
.(z	Leaves a blank space
.sp 4i	of 4 inches at the
.)z	bottom or top of a page

**Caution:** If a keep is floated to the top of a new page, and there is not enough text following the keep to fill the original page, *the floating keep command will not work*. One remedy: replace the floating keep command with a begin page command (.bp, see section 2) before the material to be kept together on the new page.

.hl

The **horizontal line** command draws a horizontal line between the current margins. It is often used to separate a keep from the text.

**Cross References** For another keep format, see section 38.

## **Example**

### **Input:**

.pp

A floating keep contains text that must be kept together and placed at the top or bottom of a page.

A floating keep is used for a figure or diagram that is not part of the running text, but is referred to in the text.

Figure 1 at the bottom of the page is the result of floating keep commands.

.(z I F

.ll 5.5i

.hl

.ce

Figure 1

All text inside the keep commands is floated to the bottom of the current page or the top of the next page, depending on available space.

To separate the keep from the text, use the .hl command to draw horizontal lines.

.hl

.ll 6i

.)z

.pp

You may be wondering how floating keeps differ from the block keeps that were introduced in the previous section.

Floating keeps always send the keep to the bottom of the current page or the top of the next page, depending on available space.

Block keeps appear in the text exactly where the opening command is entered.

If there is not enough space for the block keep on the current page, the whole keep is jumped to the top of a new page, leaving white space at the bottom of the first page.

**Output:**

.(z, .)z, .hl

A floating keep contains text that must be kept together and placed at the top or bottom of a page. A floating keep is used for a figure or diagram that is not part of the running text, but is referred to in the text. Figure 1 at the bottom of the page is the result of floating keep commands.

You may be wondering how floating keeps differ from the block keeps that were introduced in the previous section. Floating keeps always send the keep to the bottom of the current page or the top of the next page, depending on available space. Block keeps appear in the text exactly where the opening command is entered. If there is not enough space for the block keep on the current page, the whole keep is jumped to the top of a new page, leaving white space at the bottom of the first page.

---

Figure 1

All text inside the keep commands is floated to the bottom of the current page or the top of the next page, depending on available space. To separate the keep from the text, use the .hl command to draw horizontal lines.

---

## Section 40: Creating Your Own Commands (Macros)

*Macros* are a series of frequently used commands that are combined into 1 command for the sake of efficiency. You've probably been using some macros all along. For example, the .pp command is a macro. It combines the command for a blank line with the command to indent the next line 5 spaces, so that you only need to enter 1 command, the .pp, to perform 2 tasks.

Creating your own macro is the way to cut out repeating the same series of commands several times in 1 file. For example, the following macro is created for a publications list with the format of indenting all lines after the first one of each entry.

<u>Command</u>	<u>What it does</u>
.de PL	defines macro by naming it PL
.sp	leaves a blank line
.in 5n	indents the text 5 en's (about 3/8 inch)
.ti -5n	doesn't indent first line 5 en's
..	ends the macro definition

(The last line of defining a macro is 2 dots.)

Now that the PL macro is defined, whenever you want the publication list series of commands in the file, just enter the dot command of .PL.

Enter the lines that define the macro at the top of the file to avoid having to tangle with some problems over line breaks. It's possible to define any number of macros for 1 file. Use different sets of letters to name each macro (.SS, .TL, and so on). People at HAO name macros with 2 capital letters to keep them separate from the regular commands, which are nearly all lower case letters.

### Writing a Comment to Yourself

It's a good practice to write yourself a comment in the file, telling you what the macro does, so that when you or someone else returns to the file weeks or months later, you can see what the macro produces. The comment appears on screen, but is never printed. Use the command of \" at the beginning of an in-line comment. Hit the return at the end of the comment to continue entering regular text or command lines. For example,

<u>Command</u>
.de PL \"publication list macro
.sp
.in +5n
.ti -5n
..

**Cross References** See section 30 for more on writing comments.

For more on creating reference lists at HAO, see the "Reference Handout."

## Example

.de

Input:

```
.de PL\" publication list macro
.sp
.in 5n
.ti -5n
..
.de BH \"begin heading macro
.sz +2
.b
.ce
..
.de EH \"end heading macro
.sz -2
.r
..
.BH
Publications
.EH
.PL
Strong, I. B., J. R. Asbridge, S. J. Bame, H. H. Heckman, and
A. J. Hundhausen.
Measurements of Proton Temperatures in the Solar Wind.
\&Physical Review Letters\&R,
16, 631-633, 1966.
.PL
Hundhausen, A. J., J. R. Asbridge, S. J. Bame, H. E. Gilbert,
and I. B. Strong. Vela 3 Satellite Observations of Solar Wind
Ions, a Preliminary Report.
\&J. Geo. Res.\&R, 72, 87-100, 1967.
```

Output:

## Publications

Strong, I. B., J. R. Asbridge, S. J. Bame, H. H. Heckman, and A. J. Hundhausen. Measurements of Proton Temperatures in the Solar Wind. *Physical Review Letters*, 16, 631-633, 1966.

Hundhausen, A. J., J. R. Asbridge, S. J. Bame, H. E. Gilbert, and I. B. Strong. Vela 3 Satellite Observations of Solar Wind Ions, a Preliminary Report. *J. Geo. Res.*, 72, 87-100, 1967.

## Section 41: Vertical Spacing

*Vertical spacing* regulates the amount of white space between lines of type. When type size is changed with the commands of .\*s or .sz, the vertical spacing between lines is automatically adjusted to accommodate the new point size.

No vertical spacing command is needed unless you want to change the normal spacing. If you do want to change the vertical spacing, read about the commands below.

.\*v N

The **star vertical spacing** command changes the amount of white space between lines of type. The default is 10-point type and vertical spacing of 12 points. (Remember, there are 72 points to the inch.) The argument *N* is measured in points and can be 6, 7, 8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 28, or 36.

The .\*v command is local to HAO. Enter a .lo (local) command near the top of the file before the .\*v command.

Enter the .\*v command at the top of a file to change the vertical spacing of the entire document. For example, use the .\*v command to improve the vertical spacing of outlines.

The .\*v command is only reset by another .\*v command. A .vs command *temporarily* resets .\*v, but after the next paragraph or section heading command, the .\*v command is in effect again. For example,

<u>Command</u>	<u>Result</u>
.*v 14	14-point vertical spacing follows
.vs 18	18-point vertical spacing follows
.ip	14-point vertical spacing follows

If you are using this manual outside of HAO, use the following standard vertical spacing command.

.vs N

The **vertical spacing** command works like the .\*v command, but it has one disadvantage. The .vs command is reset by any of the paragraph commands of .pp, .lp, .np, .\*n, .\*N, or .ip and the section heading commands of .sh and .uh.

**Cross References**

For the explanation of points, see section 3.

To change type size, see section 8.

For an example of .\*v used with an outline, see section 43.

For another way to change vertical spacing, see the .ls command in section 1.

## **Example**

.\*v

### **Input:**

.lo  
.v 18  
.pp

The last events of the geologic history of the NCAR site are associated with the Ice Age glaciation.

As the world's climate became cooler, sizable glaciers developed along the Front Range.

There is no evidence that the glaciers reached as far east as the mountains.

However, the runoff from the glaciers surely had a role in shaping the present landscape visible from the NCAR Laboratory.

.pp  
.v 10

Boulder Creek tapped the waters of glaciers that extended from the Continental Divide to about where Nederland now stands.

Bear Canyon and Skunk Canyon were cut by running water.

Glacier erosion occurs at and above Nederland and in the upper valley of South Boulder Creek.

### **Output:**

The last events of the geologic history of the NCAR site are associated with the Ice Age glaciation. As the world's climate became cooler, sizable glaciers developed along the Front Range. There is no evidence that the glaciers reached as far east as the mountains. However, the runoff from the glaciers surely had a role in shaping the present landscape visible from the NCAR Laboratory.

Boulder Creek tapped the waters of glaciers that extended from the Continental Divide to about where Nederland now stands. Bear Canyon and Skunk Canyon were cut by running water. Glacier erosion occurs at and above Nederland and in the upper valley of South Boulder Creek.

## Section 42: Outlines

.ou *N* "some words" The **outline command** produces an outline in the traditional format with roman numerals, upper and lower case letters, and so on, each at the appropriate indentation level.

The .ou command is local to HAO. Unlike other local commands, the .ou command *does not* require the local command (.lo) at the top of the file.

The argument *N* is a number that represents the level of outline indentation. There are six outline levels, as shown here:

<u>Command</u>	<u>Result</u>
.ou 1	I. II. III. IV....
.ou 2	A. B. C. D....
.ou 3	1. 2. 3. 4....
.ou 4	a. b. c. d....
.ou 5	i. ii. iii. iv....
.ou 6	1. 2. 3. 4....

The "some words" argument contains the actual words of the outline level title, and must be surrounded by quotes. The outline title is printed in bold in the default.

### .nr sf 1

To produce an outline in roman type, enter this command on the line before the .ou command.

This command also changes section headings (produced by .sh or .uh commands) to roman type. The command of

### .nr sf 3

returns the headings following it to bold type.

To exit from the outline format, enter a base indent command of .ba Ø. On the next line, enter a paragraph command.

### Cross References

To enter text under outline titles, see section 43.

To change the white space between lines, see section 41 and the example in section 43.

## Example

.ou, .nr sf 1

### Input:

```
.ou 1 "Introduction"
.ou 2 "What you should already be able to do"
.ou 3 "Login"
.ou 3 "learn vi"
.ou 3 "learn files"
.nr sf 1
.ou 2 "Overview of Unix at HAO"
.ou 3 "Basic hardware"
.ou 4 "Terminals"
.ou 4 "Computers"
.ou 5 "HA"
.ba ø
.lp
```

There are other computers in use at HAO as well.

### Output:

## I. Introduction

### A. What you should already be able to do

1. Login
2. learn vi
3. learn files

### B. Overview of Unix at HAO

#### 1. Basic hardware

##### a. Terminals

##### b. Computers

##### i. HA

There are other computers in use at HAO as well.

## **Section 43: More on Outlines**

If the outline title is longer than one screen line, enter all of it on the line following the .ou command. In this case, do not put quotes around the title. The resulting title is in roman type unless you enter a command to make it bold.

To produce text under an outline title, enter a space (.sp) or break (.br) command. Then enter the text on the next line. Text under outline titles is usually slightly indented. There is no good remedy for this at the present time.

There is a lot of white space between outline titles in the default. To change the amount of white space, use a vertical spacing command (\*.v).

**Cross References** For an introduction to outlines, see section 42.

For the vertical spacing command (\*.v), see section 41.

## **Example**

.ou, .\*v

**Input:**

.ou 1 "Mammals"

.ou 2

Grooming Habits of Mule Deer

.ou 2 "Marmots"

.sp

Marmots seem to enjoy eating large quantities of food and relaxing  
in the sun.

.ou 2 "Squirrels"

.\*v 7

.ou 1 "Mammals"

.ou 2

Grooming Habits of Mule Deer

.ou 2 "Marmots"

.br

Marmots seem to enjoy eating large quantities of food and relaxing  
in the sun.

.ou 2 "Squirrels"

.ba Ø

.lp

**Output:**

### **I. Mammals**

#### **A. Grooming Habits of Mule Deer**

#### **B. Marmots**

Marmots seem to enjoy eating large quantities of food and relaxing in the sun.

#### **C. Squirrels**

### **II. Mammals**

#### **A. Grooming Habits of Mule Deer**

#### **B. Marmots**

Marmots seem to enjoy eating large quantities of food and relaxing in the sun.

#### **C. Squirrels**

## Section 44: Changing the Page Numbering

To establish sequential page numbering beginning with 1, enter "%” in one of the header or footer locations. (See section 9.)

.pn ±N

The **page numbering** command changes the page number of the next page, whenever the next page of output occurs. Subsequent pages are numbered sequentially from the new number.

Assume the current page number is 10 for each example below:

<u>Command</u>	<u>Next page number is</u>
.pn +6	16
.pn -6	4
.pn 6	6

To change the number starting with the first page of a document, enter the .pn command at the top of the file.

This command is useful when working with long documents. The long document can be divided into several files to make editing and printing easier. Then when you are ready to print the whole document, use the .pn command at the top of each file after the first one to establish the correct page number. For example, if the first file contains pages 1-10, then enter a command of

.pn 11

at the beginning of the second file.

.af % f

The **assign format for numbering** command changes the numbering format through the use of different arguments. The possibilities are as follows:

<u>Command</u>	<u>Numbering Sequence is</u>
.af % 001	001. 002. 003. 004....
.af % i	i. ii. iii. iv....
.af % I	I. II. III. IV....
.af % a	a. b. c. d....
.af % A	A. B. C. D....

To change the numbering format starting with the first page, enter the .af % command at the top of the file.

To change the number within an .af sequence, use the .pn command. For example,

.af % i

Any amount of text entered here

.pn +4

produces a page number of "v" on the next page.

## .pn, .af %, .bp

### .bp $\pm N$

The **begin page** command stops the printing and begins a new page at the point where the command is entered. Add an argument to the .bp command to change the number of the next page.

Assume the current page number is 10 for each example below:

<u>Command</u>	<u>Next page number is</u>
.bp	11
.bp +3	13
.bp -3	7
.bp 3	3

#### **Caution:**

A .bp command will not work immediately following a paragraph (.pp) or left paragraph (.lp) command. In those situations, always enter the .bp command first, like this:

```
.bp  
.lp
```

**Cross References** For placement of the page number on the page, see section 9.

For the explanation of  $\pm N$ , see p. 14.

## Section 45: Delayed Text

*Delayed text* is text that is saved and printed separately from the rest of the file. Use the command for notes that come at the end of a paper.

.(d           The **begin delayed text** command opens the text that is saved to be printed later.

.)d           The **end delayed text** command closes the text that is saved to be printed later.

.pd           The **print delayed text** command causes all entries between the .(d and .)d commands to be printed when the file is sent to the laser for printing.

Enter this command at the end of the file to print all the delayed text at the end of the document. Enter a begin page command (.bp) before the print command to print the delayed text on a separate page.

\\*#           This command places a note number in the text and at the beginning of the note that is saved for separate printing. The numbering is automatically increased. The number is inside brackets.

**Cross References**   To produce footnotes, see section 12.

## Example

.(d, .)d, .pd, \\*#

### Input:

At the beginning of Paleozoic time, the sea gradually invaded Colorado from the west.\\*#

.(d

Notes For Chapter 1

.sp

\\*#*The Natural Features of the NCAR Site,*  
p. 4.

.)d

The sea deposited beach sands and then limestone.\\*#

.(d

\\*#ibid., p. 4.

.)d

.sp

(Text entered here until end of chapter.)

.sp 4

[The printed results of the .pd command appear next.]

.sp

.pd

### Output:

At the beginning of Paleozoic time, the sea gradually invaded Colorado from the west.[1]  
The sea deposited beach sands and then limestone.[2]

(Text entered here until end of chapter.)

[The printed results of the .pd command appear next.]

Notes For Chapter 1

[1] *The Natural Features of the NCAR Site*, p. 4.

[2]ibid., p. 4.

## Section 46: Table of Contents

The table of contents commands save text and print it separately after a special print command is entered.

.(x

The **begin table of contents** command opens the entry.

.)x P

The **end table of contents** command closes the entry. The argument *P* controls what follows each entry. Here are the possibilities:

<u>Command</u>	<u>Result</u>
.)x	.....current page number
.)x ""	.....no page number
.)x _	no line of dots, no page number
.)x 6	.....6 (or any number specified)

Note that the second command above has an argument of 2 sets of double quote marks (not 4 single quote marks).

Unlike other commands, the table of contents arguments *follow the closing*, not the opening, command.

.xp

The **print table of contents** command causes all entries between sets of .(x and .)x commands to be printed when the document is sent to the laser for printing.

Enter this command at the end of the file to print the table of contents separately from the main text. Enter a command of begin page (.bp) before the print table of contents command to produce a table of contents on a separate page.

### Cross References

For additional variations of this command, see the “-me Reference Manual,” section 5.

## Example

.(x, .)x, .xp

### Input:

.lp

Software is a set of computer programs that make hardware perform various tasks.

.(x

software

.)x

The line spacing command controls the number of blank lines between lines of type.

.(x

line spacing

.)x ""

Troff is a specific text formatting program.

.(x

troff

.)x 8

The paragraph command leaves a blank line and indents the next line about 3/8 of an inch.

.(x

paragraph

.)x \_

.sp 2

[Output resulting from the

.q ".xp "

command.]

.xp

### Output:

Software is a set of computer programs that make hardware perform various tasks. The line spacing command controls the number of blank lines between lines of type. Troff is a specific text formatting program. The paragraph command leaves a blank line and indents the next line about 3/8 of an inch.

[Output resulting from the ".xp " command.]

software .....	117
line spacing .....	
troff .....	8
paragraph	

## Section 47: Boxing Words

**.bx "some words"** The **box** command draws a box around a small group of words. However, the boxes don't always meet at the corners, particularly when the boxed area is long.

**Cross References** To make larger boxes, such as the ones used throughout this manual for important points, see "Tables" in part 4.

## Example

.bx

Input:

```
.bx "some words"  
.sp  
.bx "Several words in a much longer box"  
.sp  
.bx "Here is an attempt at a box that is particularly long"
```

Output:

some words

Several words in a much longer box

Here is an attempt at a box that is particularly long



## **Part 3: Sample Letters and Memos Available at HAO**

Copying the Basic Letter and Memo Formats 122

Writing Your Own Letter or Memo 122

Printing Your Letter or Memo 123

## **Sample Letters and Memos Available at HAO**

Before you write your first letter or memo, read part 1 of this manual. Also complete the "learn vi" program to learn the editing commands.

Basic letter and memo formats are set up already and are stored in public files that are available for your use. These files may have been copied into your own directory when you received your login code and password. The 4 basic format files are as follows:

<u>File Name</u>	<u>Contains</u>
basic.letter	basic letter format, no letterhead
letter.head	basic letter format with HAO letterhead
basic.memo	basic memo format, no letterhead
memo.head	basic memo format with HAO letterhead

After you login, the UNIX prompt sign (\$ or %) appears on the screen. To see if the basic letter and memo file are already in your directory, enter

ls (return)

which is the UNIX command that lists all the files in your current directory. If the names of the files listed above appear on your screen, skip reading "Copying the Basic Letter and Memo Formats into Your Directory," and go on to "Writing Your Own Letter or Memo."

### **Copying the Basic Letter and Memo Formats into Your Directory**

Use the UNIX command, cp, to copy the basic format files into your directory. After the prompt sign, enter

cp /usr/pub/format/basic.letter . (return)

(Be sure to enter a space and then the final dot in the command line shown above.) The prompt sign appears again after you hit return. Copy the next format file by entering

cp /usr/pub/format/letter.head . (return)

Repeat for the last two files:

cp /usr/pub/format/basic.memo . (return)

cp /usr/pub/format/memo.head . (return)

### **Writing Your Own Letter or Memo**

1. Make a copy of the file you want to use for your letter or memo. For example, if your memo is to Dr. Luke Skywalker, you might name the file that contains that memo "skymemo." After the prompt sign, enter the UNIX command of

cp basic.memo skymemo (return)

After you hit the return, the UNIX prompt sign appears again, which means that the copying is done.

2. Next, call your new file, skymemo, to your screen by entering the UNIX command of

vi skymemo (return)

The format for your memo appears on the screen, and looks like the left column on the following page. (The right column is a *brief* explanation of the command.)

### **Example of basic.memo**

```
.lo                                -- calls local macros
.tr ~                             -- translates ~ into space
.he "-%-"                         -- numbers pages at top in center
.hx                                -- doesn't print number on first page
.*i " \ *(td"                      -- indents from right margin
\ *(td                            -- today's date
.in                                -- returns to normal indent
.sp 2                             -- spaces 2 lines
MEMO TO:~~put who to here
.sp                                -- spaces 1 line
FROM:~~~~~ put who from here
.sp                                -- spaces 1 line
SUBJECT:~~~ put subject here
.sp                                -- spaces 1 line
.pp                                -- begins paragraph
replace this line with body of memo
.sp 2                             -- spaces 2 lines
.ce                                -- centers the next line
-END OF MEMO-
```

3. Move the cursor down to the "MEMO TO:" line, and then move the cursor over to the first letter of "put." Delete the rest of the line. You are ready to start writing now.
4. Get into the text entering mode by entering one of the commands that opens text entering. For example, enter  
    i  
which is the editing command for "insert." (Nothing appears on the screen after an editing command.) After "MEMO TO:" type in  
    Dr. Luke Skywalker
5. Hit the escape key (esc) to return to the editor.
6. Repeat steps 3 and 4 to enter text for FROM:, SUBJECT:, and the body of the memo. Remember to hit the escape key each time you need to return to the editor in order to move the cursor. Then get back into a text entering mode by entering  
    i  
which is the editing command for "insert."
7. When you have finished entering all the text, hit the escape key (esc) to return to the editor. Then enter  
    :wq (return)  
(stands for "write quit") to leave the editor. When the prompt sign appears on screen again, the file "skymemo" has been stored on the computer.

### **Printing Your Letter or Memo**

When the prompt sign is on the screen, enter a UNIX command of  
    laser skymemo (return)

and your skymemo file is sent to the printer. If the first copy doesn't come out quite the

way you wanted, don't be discouraged. Bring the file back to your screen with a command of

vi skymemo

Repeat steps 3 through 7 to make changes, then print again.

### **Changing the Basic Format Files Stored in Your Directory**

The format files can be changed to meet your needs after you copy them into your directory. For example, you might need to change the phone number in the two letterhead files. Follow steps 2 through 7 to change the format files before you copy the files over for each use.

**Important Point:** If you mess up one of the format files stored in your directory, just make a new copy of it. (You can't mess up the master copy that is in the public user file.) To copy a file again, follow the directions under "Copying the Basic Letter and Memo Formats into Your Directory" above.

**Cross References** For more help with UNIX and editing commands, see the "UNIX Command Notebook For Word Processors" (an HAO publication), the "learn vi" program, and the "vi Crib Sheet," (which is in part 4 of this manual).

For more help with word processing commands, see part 2 of this manual.

## **Part 4: Useful Information**

Tables 126

Equations 127

Mathematical and Special Characters 130

Greek Characters 131

Some UNIX Commands 133

Additional Resources 136

vi Crib Sheet 138

## Tables

Learn to create tables from "Tbl -- A Program to Format Tables," by Lesk. It presents lots of sample tables that will help you get started.

Producing tables can be tricky. Here are 2 obscure sources of errors.

1. If first column in a table has no text entry, the table may not print correctly. To avoid this problem, enter the non-printing character command of \& in the empty first column. See section 30 in part 2 for other uses of the non-printing character command.
2. Normal special character commands, such as \\* to produce an accent mark, do not always work within tables. In tables, use 2 backslashes to produce the special character (\\\\*, for example). See section 22 in part 2 for the complete list of special characters.

When one of your tables doesn't come out right and you can't figure out why, don't just sit there in frustration. Go ask the HAO word processing specialist or someone else experienced with tables for help.

### Using Table Commands to Create Boxed Text

Perhaps you wonder how to make boxes around some text the way I've been doing for the "Caution" and "Important Point" boxes throughout this manual. Here is the sequence of commands:

.TS	starts table
box expand;	boxes table; width is current line length
lw(5.5i).	left justifies, text width of 5.5 inches
T{	begin table text
.fi	fill lines (necessary when text is short)
Text entered here	body of text
T}	end table text
.TE	end table

### Printing a File That Contains Tables

Tables are processed before the file is printed. The command for processing tables (-tbl) always precedes the file name. When there is a prompt sign on the screen enter the following command to print a file that contains tables:

laser -tbl filename

The command for processing tables *always* goes before the command for processing equations (-eqn). For example,

laser -tbl -eqn filename

prints a file that has both tables and equations. There is a space before the -tbl and the -eqn.

## Equations

Learn to create equations from "Typesetting Mathematics -- User's Guide," by Kernighan and Cherry, and "Typing Mathematical Formulas with UNIX," by Koppel (a document produced for HAO).

This section on equations covers only a few additional things that are helpful *after* you know the basics of equations.

### Creating Equations That Appear Within the Line of Text

*In-line equations* appear in the middle of a line of text. Before entering in-line equations, enter the following 3 lines at the top of your file:

```
.EQ  
delim $$ (or another symbol)  
.EN
```

The \$ sign is known as a *delimiter* because it sets the limits of where the equation begins and ends. Put the delimiter (\$) before and after each in-line equation.

**Caution:** If you use \$ signs for delimiters and also have actual \$ signs in the text, your job will be messed up, since troff will be treating everything after the \$ sign in the text as an equation. It's easy to spot such a problem; the \$ sign will not be printed and the text suddenly switches to italics, just as it does for an equation. To correct it, you have two choices:

1. Use a symbol that is not used in the file for the delimiters, such as @ or ^.  
The opening lines of the file would look like

```
.EQ  
delim @@  
.EN
```

and each equation would be surrounded by @ signs.

2. Use the \$ sign for the delimiters, but each time the \$ appears in the text, enter the following instead of a single \$ sign:

\$"\$\$"

Avoid using # for the delimiter; the # symbol has special functions for tables and using the # as an equation delimiter can cause strange table errors if there are tables in the same file.

Special characters, such as bullets at the beginning of listed items or zeros with a slash through them, are created with in-line equation commands. For example,

#### Input

```
.EQ  
delim $$  
.EN  
$bullet$ The first point...  
The number $empty$ is...
```

#### Output

- The first point...

The number Ø is..

## **Creating Equations That Appear on a Separate Line**

Use the following sequence of commands for each equation that will be printed on a separate line:

.EQ

equation entered here for any number of lines

.EN

When it is printed, the equation is centered automatically and a blank line appears before and after the equation.

## **Checking Equations Before Printing (checkeq)**

Common equation errors are not entering the closing delimiter sign after an in-line equation or not entering the closing command of .EN after a separate equation. Use the UNIX command, **checkeq**, to see if you have left out a delimiter before sending a file with equations to the printer. In this example, the file is named "math" and the delimiter is the @ sign. When the prompt sign appears on your screen, enter

checkeq math (return)

If you have not left out a delimiter, the checkeq report looks something like

New delims @@, line 2 (can be any line number)  
@@ in EQ, line 2 (can be any line number)

If you have left out a delimiter, the checkeq report looks like

New delims @@, line 4 (can be any line number)  
@@ in EQ, line 4 (can be any line number)  
7 line @@, line 15-22 (can be any line numbers)  
Unfinished @@

The last line, "Unfinished @@" means you may have left out a delimiter. The line above it, "7 line @@, line 15-22" means that you have an in-line equation 7 lines long, and that it starts at line 15. In the unlikely case that you intended to have an in-line equation that is 7 lines long, then there is no error. Checkeq reports on any equation of 2 or more lines, so sometimes there is not really a missing delimiter, just a long equation.

If you left out an even number of delimiters, say 2 or 4, the checkeq report looks like the one above, but it does not have the "Unfinished @@" line because there are no "unfinished" equations, since the number of delimiters is still evenly matched. But the tipoff to a missing delimiter is the line that says you have a "7 line" equation, probably not what you intended.

In either case, get back into the file named "math" by entering the command of

vi math (return)

(The prompt sign automatically appears at the end of the checkeq report.) Look at line 15 to see if you left out a delimiter. If you don't find an error there, look at the surrounding lines.

If you left out a delimiter early in the file, the checkeq report on your screen might be very long because the first error mismatches the delimiters of the rest of the equations in the file. In fact, the checkeq report might be so long that the first part of it (which is the part you need the most) goes zipping off the screen before you can read it. You can

get the checkeq report for the "math" file back at a more readable pace by entering the following command after the prompt sign returns to the screen:

checkeq math | more (return)  
(The symbol | is on your keyboard.)

To see the next section of the checkeq report, hit the space bar.

When you have a long checkeq report, the place to begin looking for missing delimiters is the first several lines mentioned at the top of the checkeq report. Adding the missing delimiter in the early part of the file will "fix" all the equations later in the file, unless you left out another delimiter somewhere.

If you left out the .EN of an equation that appears on a separate line, the checkeq report looks like

@@ in EQ, line 4 (can be any line number)  
@@ in EQ, line 10 (can be any line number)  
Unfinished EQ

The last line, "Unfinished EQ," means you have an equation that is not "finished" with a command of .EN. There is an .EQ at line 4 and an .EQ at line 10, but there is no .EN between them.

If you left out an .EQ, the checkeq report will look like the one above, with EN substituted for EQ.

### **Printing a File That Contains Equations**

Equations are processed before the file is printed. The command for processing equations (-eqn) always precedes the file name. When there is a prompt sign on the screen, enter the following command to print a file that contains equations:

laser -eqn filename (return)  
(Do not omit the minus sign (-) in front of the eqn.)

If your file contains any lines of .EQ or .EN, you *must* include the -eqn when giving directions to print the file. If you forget the -eqn, the top of your printed document will have something like "delim @@" and the equations will not be printed correctly.

If the file also contains tables, use the following print command:

laser -tbl -eqn filename (return)  
The command for processing tables (-tbl) *always goes before* the command for processing equations. There is a space before -tbl and -eqn.

### **Special Characters**

Lists of the mathematical and special characters and the Greek characters that can be created within equations are on the following 2 pages. If you need to use a character that is not on the lists, check "Some Special Characters," section 22 of part 2 of this manual. If it's not there, ask the HAO word processing specialist, who might be able to create a new character for you. (The mathematical and special character list is updated occasionally; be sure to insert the updated list in your manual.)

## Mathematical and Special Characters

Char	Character Name	Char	Character Name	Char	Character Name
$\oplus$	ciplus	'	acute	*	star
$\otimes$	cetimes	'	grave	*	bigstar
$\hbar$	hbar	$\approx$	approx	*	mstar
$\cdot$	=dot	$\leq$	$\leq$	$\S$	section
$\text{Å}$	angstrom	$\geq$	$\geq$	$\dot{x}$	x dot
$\wedge$	andsign	$\ll$	$\ll$	$\ddot{X}$	X dot
$\vee$	orsign	$\gg$	$\gg$	$\ddot{x}$	x dotdot
$\parallel$		.	degree	$\ddot{X}$	X dotdot
$\Delta$	=del	$\rangle$	rangle	$\hat{x}$	x hat
$\sim$	wig	$\langle$	langle	$\hat{X}$	X hat
$\negsim$	-wig	$\lrcorner$	ang	$\bar{x}$	x bar
$\equiv$	=wig	$\lrcorner$	rang	$\bar{X}$	X bar
$\lessgtr$	<wig	$\times$	times	$\tilde{x}$	x tilde
$\gtrless$	>wig	$\rightarrow$	->	$\tilde{X}$	X tilde
$\forall$	oppA	$\leftarrow$	<-	$\vec{x}$	x vec
$\exists$	oppE	$\equiv$	==	$\vec{X}$	X vec
$\sqsubset$	incl	$\neq$	!=	$\vec{\pi}$	x dyad
$\notin$	nomem	$\neq$	noteq	$\vec{X}$	X dyad
$\pm$	ppd	$\pm$	+-	$\underline{x}$	x under
$\infty$	inf	$\Leftrightarrow$	$\Leftrightarrow$	$\underline{X}$	X under
$\dots$	3dot	$\leftrightarrow$	$\leftrightarrow$		nothing
$\cdots$	cdot	$\ntriangleright$	>	$\phi$	lint
$\bullet$	bullet	$\ntriangleleft$	<	$\sqrt{x}$	sqrt x
$\frac{1}{2}$	half	$\nabla$	del	$\lessgtr$	$\langle o \rangle$
$\frac{1}{4}$	quarter	$\nabla$	grad	$\gtrless$	$\rangle o \langle$
$\frac{3}{4}$	3quarter	$\therefore$	thf	$\//$	para
$\odot$	solar	$\ldots$	...	$\square$	box
$\circ$	circle	$\ldots$	....	$\wp$	blank
$\emptyset$	empty	$\approx$	prop	$\phi$	slasho
$\in$	member	'	prime	"	arcsec2
$\subset$	subset	$\cup$	union	$\mp$	-+
$\supset$	!subset	$\cap$	inter	$\neg$	not
$\supseteq$	supset	$\cap$	cap	-	hyphen
$\supseteqq$	!supset	$\cup$	cup	-	-
$\uparrow$	uparrow	$\Sigma$	sum	-	dash
$\downarrow$	darrow	$\prod$	prod	$\circledcirc$	copyright
$\rightarrow$	x rarrow	$\int$	int	$f_i$	fi
$\leftarrow$	x larrown	$\partial$	partial	$f_l$	fl
$\wedge$	arcmin	sec	sec	$ff$	ff
$\wedge$	arcsec	cm	cm	$ffi$	ffi
$\wedge$	arc	$\div$	divide	$ffl$	ffl
$\Rightarrow$	rdaro	$\ddagger$	ddagger	$m$	mdot
$\Leftarrow$	ldaro	$\dagger$	dagger		

## Greek Characters

Char	Character Name	Char	Character Name
$\alpha$	alpha		
$\beta$	beta		
$\chi$	chi		
$\delta$	delta	$\Delta$	DELTA
$\epsilon$	epsilon	E	EPSILON
$\eta$	eta		
$\gamma$	gamma	$\Gamma$	GAMMA
$\iota$	iota		
$\kappa$	kappa	$\Lambda$	LAMBDA
$\lambda$	lambda		
$\mu$	mu	$\Omega$	OMEGA
$\nu$	nu		
$\omega$	omega	$\Phi$	PHI
$\circ$	omicron	$\Pi$	PI
$\phi$	phi	$\Psi$	PSI
$\pi$	pi		
$\psi$	psi	$\Sigma$	SIGMA
$\rho$	rho		
$\sigma$	sigma	$\Theta$	THETA
$\tau$	tau	$\Upsilon$	UPSILON
$\theta$	theta	$\Xi$	XI
$\vartheta$	thet		
$\upsilon$	upsilon		
$\xi$	xi		
$\zeta$	zeta		



## Some UNIX Commands

The UNIX commands that are used most frequently are covered on this page. If you need more details on these or other UNIX commands, look in the *UNIX Command Notebook for Word Processors*, by McCune (a publication produced for HAO).

Remember, UNIX commands can be entered only after a prompt sign on the screen.

**laser file** This is the basic print command. Substitute the name of the file you want printed for *file*.

**laser -tbl file** Use this command when there is a table in the file to be printed. Remember to enter a space before -tbl and again before the file name.

**laser -eqn file** Use this command when there is an equation in the file to be printed.

**laser -tbl -eqn file** Use this print command when the file contains both tables and equations. The -tbl *always* precedes the -eqn.

**laser -nN file** Use this command if you want the page numbering to start with something other than 1. The capital *N* is the number of the first page to be printed. For example,

laser -n16 math

prints the file named math, and starts the page numbering with 16. There is no space between -n and the number for the first page.

If the file named math contains tables and equations, and you want to start page numbering with 12, use the following command:

laser -tbl -eqn -n12 math

**laser -opages file** Use this command to print only certain pages of a file. Separate individual page numbers with a comma or use a dash to print a series of pages. For example,

<u>Command</u>	<u>Result</u>
laser -o6,9,15 math	prints only pages 6, 9, 15 of the math file
laser -o12-20 math	prints pages 12 through 20 of the math file
laser -o6,13-17 math	prints page 6, then pages 13 through 17 of the math file

There is no space between the -o and the numbers. (This command uses the lower case letter o, *not* the numerical figure Ø.)

If you are printing some pages from a file that contains tables and equations, use a command based on this model:

laser -tbl -eqn -o6,15-19 math

**Important Point:** To get the prompt sign back quicker after entering a laser command for printing, add the & sign to the end of the command. For example,  
laser -eqn math &

The & sign puts processing your job on "background" and returns the prompt sign to your screen so that you can continue working.

#### **laserq**

After you have sent a document to the laser for printing, use this command if you want to see where your job is in the *queue* (line-up) waiting to be printed. If you are working on the IS, HW, or HV computers, the laserq report appears on your screen and looks something like this:

Jobs waiting to print:

haoimagin is ready and printing

Rank	Owner	Job Files	Total Size
active	linda	403 (standard input)	15795 bytes
1st	nancy	404 (standard input)	97753 bytes

Jobs waiting to run:

no match

The "Jobs waiting to print" section reports the queue waiting to be printed. The "Jobs waiting to run" section reports the queue waiting to be processed by troff before entering the queue to be printed.

If either queue has "no match" or "no entries", it means there are no jobs waiting there.

If you are working on the HA computer, the laserq command has little value, since the report does not appear on the screen, but is sent by electronic mail to your mailbox. By the time it gets to your mailbox and you remember to look in your mailbox to read it, the report information is no longer current.

#### **checkeq file**

If the file has equations, use this command to check for errors before printing the file. The full explanation of checkeq is in "Equations" in part 4 of this manual.

#### **spell file**

This command checks for spelling errors and lists the words that don't match entries in the spell dictionary. Spell finds some errors, but the program does not find all misspellings. For example, if you make an error such as using "affect" for "effect", the

error will not be reported, since both words appear as correct spellings in the spell dictionary. Also, some of the words listed in the report are not misspelled, but they are included in the spell report because they are not in the spell dictionary.

To add more words to the spell dictionary, see the "spell" pages of the *UNIX Command Notebook for Word Processors*.

If you are working on HA, the spell report zips off the screen before you can read it. To bring it back at a readable pace, enter  
spell file | more

(The | symbol is on the keyboard.)

## **Additional Resources**

The resources listed here are additional sources of information for people word processing on UNIX at HAO. The list is by no means a complete bibliography of UNIX books.

### **Word Processing Commands**

Allman, Eric P. "-me Reference Manual," in *UNIX Programmer's Manual*, vol. 2c, section 53. Berkeley: Univ. of California, Dept. of Electrical Engineering and Computer Science, 1979. Difficult reading for beginning computer users, with no examples. However, it contains a few less-frequently-used commands that are not in this manual. Available from the HAO word processing specialist.

\_\_\_\_\_. "Writing Papers with NROFF Using -me," in *UNIX Programmer's Manual*, vol. 2c, section 54. Berkeley: Univ. of California, Dept. of Electrical Engineering and Computer Science, 1979. Good examples, but does not contain all the most-used commands. Available from the HAO word processing specialist.

Kernighan, Brian and Lorinda L. Cherry. "Typesetting Mathematics -- User's Guide," (Second Edition), in *UNIX Programmer's Manual*, seventh edition, vol. 2A, section 9. Berkeley: Univ. of California, Dept. of Electrical Engineering and Computer Science, 1979. A good source on creating equations. Available from the HAO word processing specialist.

\*Koppel, Karla. "Typing Mathematical Formulas with UNIX." 1981. Another good source on creating equations. Has practice exercises. Available from the HAO word processing specialist.

Lesk, M. E. "Tbl -- A Program to Format Tables," in *UNIX Programmer's Manual*, seventh edition, vol. 2A, section 10. Berkeley: Univ. of California, Dept. of Electrical Engineering and Computer Science, 1979. A good source on creating tables, with lots of examples. Available from the HAO word processing specialist.

### **Editing Commands**

"learn vi," an on-line tutorial program.

Lozy, Mohamed el. *Editing in a UNIX Environment: The vi/ex Editor*. Englewood Cliff, N.J.: Prentice-Hall, Inc., 1985. Good explanations of editing commands written in a readable style. Doesn't cover word processing commands at all. In the NCAR Library.

### **UNIX Commands**

\*McCune, Linda. "Reference Handout." 1985. Information on producing bibliographies that can be printed in various bibliographic styles. Available from the HAO word processing specialist.

---

\*Local document available at HAO only.

\*McCune, Linda. *UNIX Command Notebook for Word Processors*. 1985. UNIX commands that are frequently used in conjunction with word processing. Contains pages from the *UNIX Programmer's Manual* and supplemental pages that have examples and other information. Available from the HAO word processing specialist.

## General

\*Alves, Betsy. "A Beginning Manual for Word Processing on UNIX." 1983. Created as a training manual for secretaries. Covers some basic aspects of editing and word processing commands, the files system, mailing labels, and electronic mail. Available from the HAO word processing specialist.

Birns, Peter M., Patrick Brown, and John C. C. Muster. *UNIX for People: A Modular Guide to the UNIX Operating System, Visual Editing, Document Preparation, and Other Resources*. Englewood Cliffs, N.J.: Prentice-Hall, 1985. A readable, sometimes humorous book. However, some of the word processing commands are from the -ms package, and HAO uses the -me package. The -ms commands are in capital letters. Don't try using the -ms commands, since they will be ignored when the file is printed at HAO. Instead, use the manual you are now reading to find the -me command that performs the same task as the -ms command. In the HAO Library.

Foster, Diana and Kathryn Hemness. *ms Dictionary*. Davis, Cal.: Univ. of California, Computer Center, 1983. Revised by Kelli Gant-Errek. Good, short explanations with examples. However, the commands are from the -ms package, and HAO uses the -me package. The -ms commands are in capital letters. Don't try using the -ms commands, since they will be ignored when the file is printed at HAO. Instead, use the manual you are now reading to find the -me command that performs the same task as the -ms command. In the HAO Library.

Lomuto, Ann Nichols and Nico Lomuto. *A UNIX Primer*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1983. A good book for beginners, truely a "primer." Word processing commands are from the -me package, the package used at HAO. Doesn't cover some of the more complex word processing commands. Has sections on editing and files, too. In the NCAR Library.

---

\*Local document available at HAO only.

## vi Crib Sheet

### Make a File

vi *name* creates file called *name*  
(Type when \$ is on screen)

### Insert and Escape

a	append after cursor
A	append at end of line
i	insert before cursor
I	insert at beginning of line
o	open line below cursor
O	open line above cursor
DEL	deletes chars during insertion (DEL key)
ESC	exits any insertion (ESC key)

### Delete and Replace

x	deletes (erases) one character
nx	deletes <i>n</i> characters
dd	deletes one line (current)
ndd	deletes <i>n</i> lines
dw	deletes one word
ndw	deletes <i>n</i> words
D	deletes rest of line
rx	replace single character with <i>x</i>

### Change and Substitute

cc	changes one line (current)
ncc	changes <i>n</i> lines
cw	changes one word
ncw	changes <i>n</i> words
C	changes rest of line
ns	substitutes <i>n</i> chars
S	substitutes one lines
nS	substitutes <i>n</i> lines
DEL	deletes chars during change
ESC	exits any change or substitute

### Line Positioning of Cursor

H	first line of screen (home)
M	middle line of screen
L	last line of screen
RETURN	next line, at beginning
+	next line, at beginning
-	previous line, at beginning
j	next line, same column
k	previous line, same column

### Character Positioning of Cursor

0	beginning of line (zero)
^	First character on line
\$	end of line
space	forward (right)
l	forward (right)
h	backwards (left)
fx	move to <i>x</i> forwards (find)
Fx	move to <i>x</i> backwards (find)
tx	move up to <i>x</i> forwards (to)
Tx	move up to <i>x</i> backwards (to)
;	repeat the last f F t or T command
,	inverse of ;

### Positioning Within File

^F	move forward screenful
^B	move backward screenful
^D	move down half screen
^U	move up half screen
^G	show current file and line number
G	go to last line of file
nG	go to line number <i>n</i>
/chars	go to next occurrence of <i>chars</i>
?chars	go to previous occurrence of <i>chars</i>
n	repeat last / or ? (forward)
N	repeat last / or ? (backward)

**Note:** The notation '^F' means control-F (or whichever letter is used instead of 'F'). This is generated by holding down the CTRL key and typing the letter (in lower case).

### Adjusting the Screen

^L	redraw screen, use when a message appears - (or ^R)
^R	redraw screen - (or ^L)
z (RTN)	redraw, current line at top
z-	redraw, current line at bottom
z.	redraw, current line in middle
"	return to previous location (single quotes)
mx	mark position with letter <i>x</i>
'x	return to mark <i>x</i>

**Note:** ^L and ^R are terminal and/or computer dependent

### Move by Words (cont.)

w	move forward one word
nw	move forward <i>n</i> words
b	move backward one word
nb	move backward <i>n</i> words
e	move forward to end of word
ne	move forward to end of <i>n</i> 'th word
W	like w, but disregards punctuation and special characters
B	like b, but disregards punctuation and special characters
E	like e, but disregards punctuation and special characters

**Note:** W, B, and E can be used with *n* indicating number of 'words' in the same manner as w, b, and e.

### Move by Sentences, Paragraphs

)	move to next sentence
(	move to previous sentence
}	move to next paragraph (.pp)
{	move to previous paragraph

### Undo, Redo, Retrieve

u	undo last change
U	restore current line
"np	retrieve <i>n</i> 'th last delete
.	repeat last command which caused a change

### Miscellaneous

J	join next line to current line
nJ	join <i>n</i> next lines to current line
<<	shift line to left
n<<	shift <i>n</i> lines to left
>>	shift line to right
n>>	shift <i>n</i> lines to right

### File Manipulation

:w	writes (stores) changes
:wq	writes changes and quits
:q	quits (no changes)
:q!	quits, discards changes
:w <i>name</i>	writes current file to file <i>name</i> (you are still in original file)

:e <i>name</i>	edits file called <i>name</i>
:e #	edits previous file
:n	edits next file, assuming you typed vi <i>file1 file2 ...</i>
:!cmd	performs UNIX command, <i>cmd</i> , and returns to editor (vi)

### Yank and Put

yy	yank one line
Y	yank one line
nyy	yank <i>n</i> lines
nY	yank <i>n</i> lines
yw	yank one word
nyw	yank <i>n</i> words
p	puts most recent yank after cursor (yw) or line (yy or Y)
P	puts most recent yank before cursor (yw) or line (yy or Y)

### Stored Yanks and Puts

"xyy	yank one line
"xY	store in location x
"xnyy	yank one line
"xnY	store in location x
"xyw	yank one word
"xnyw	yank <i>n</i> words
"xp	store in location x
"xP	puts yank stored in x after cursor (yw) or line (yy or Y)
"xP	puts yank stored in x before cursor (yw) or line (yy or Y)

**Note:** The storage locations remain intact until the file is quit (:q, :wq). You can do a stored yank ("a23yy), edit a new file (:e *newfile*), and put the stored yank in *newfile* ("ap).

The symbol " is a double quote.

## **Acknowledgements**

Many people at HAO, particularly the secretaries, contributed to this manual by making suggestions based on their UNIX expertise and by editing earlier drafts.

Special thanks goes to programmer Linda McCune who edited the manual for technical accuracy and created the diagrams (using the PIC program), mathematical and Greek characters lists, and the vi crib sheet.

# Index

## A

Accent marks, 61  
.ad, 78-79  
Adjust, 78-79  
.af, 112  
Allman, Eric, 136  
Alves, Betsy, 137  
Arguments, 14, 28, 74, 116-117  
Assign numbering format, 112

## B

.b, 26-27  
.b, 95-99  
.b, 95-99  
.ba, 74-75  
Backslash commands, 14, 26, 76-77  
Base indents, 74-75  
.bc, 56-58  
Beeps from terminal, 8  
Begin page, 18-19, 113  
Begin second column, 56-58  
*Beginning Manual for Word Processing on UNIX*, 137  
Bibliography, annotated, 136-137  
Birns, Peter, 137  
Blank space, (*see* Space and Spacing)  
Block keep, 95-99  
Block quotes, 38-39  
Bold font, 26-27  
Boxing text, 126  
Boxing words, 118-119  
.bp, 18-19, 113  
.br, 16-17, 52-53, 80-85  
Break, 16-17, 52-53, 80-85  
Brown, Patrick, 137  
Bullet, 127  
.bx, 118-119

## C

.(c, 24-25  
.c, 24-25  
.1c, 56-58  
.2c, 54-58  
Caret symbol, 61  
.ce, 24-25  
Cedilla symbol, 61  
Centered lines, 24-25, 95

Centimeters, 20  
Character width in troff, 42  
checkeq, 128-129  
Cherry, Lorinda, 136  
Circle symbol, 61  
Commands  
  backslash, 76-77  
  categories of, 6-9  
  creating, 104-105  
  in-line, 26  
    (*see also* Editing commands)  
    (*see also* Local commands)  
    (*see also* UNIX commands)  
    (*see also* Word processing commands)

Comment line, 77, 104-105  
Computers at HAO, 4  
Copying letter & memo samples, 122  
Corrections to manual, 3-4  
cp, 122  
Cursor, 7  
Czech symbol, 61

## D

.(d, 114-115  
.d, 114-115  
.de, 104-105  
Default, 15  
Delayed text, 114-115  
Delimiters, 34, 127  
Directory, 122  
Distribution lists, 56-58  
Document, 5  
Documentation, 2  
Dot command, 14  
Down command, 76

## E

Editing commands, 6-9  
  other manuals on, 136  
  vi crib sheet, 138-139  
*Editing in a UNIX Environment*, 136  
Editor, the, 7, 9  
11/70, 4  
Em's, 20  
.EN, 128-129  
En's, 20  
.EQ, 128-129  
eqn, 129, 133

Equations, 127-129, 133  
other manuals on, 136  
Escape key, 6, 9  
Extra lines on page, 92-94

## F

.(f, 40-41  
.f, 40-41  
.fi, 42-43  
File, 5, 9  
Fill, 42-43  
Floating keeps, 100-103  
.fo, 34-35  
Font, 26-27  
Footers, 34-35  
Footnotes, 40-41  
For all symbol, 61  
Formatting, 5, 7, 15  
Foster, Diana, 137  
.ft, 26-27

## G

Gant-Errek, Kelli, 137  
Getting hung, 93  
Getting started, 9  
Greek characters, 131

## H

HA computer, 4  
Hardware, 4  
.he, 34-35  
Headers, 34-35  
Hemness, Kathryn, 137  
.hl, 101-103  
Horizontal lines, 77, 101-103  
Horizontal space, 84  
HV computer, 4  
.hw, 86-87  
HW computer, 4  
.hx, 34-35  
.hy, 86-87  
Hyphenation, 15, 86-87

## I

.i, 26-27  
.\*i, 22-23  
Imagin 8/300, 4  
.in, 20-21, 59-60  
Inches, 20

Indentation  
left, 20-21  
right, 22-23  
temporary, 20-21  
with mark and return, 59-60

Indented paragraphs, 64-67, 74

Infinite loop, 77, 93

Input, 4, 15

Inside address of letter, 22-23

.ip, 64-67

IS computer, 4

Italic font, 26-27

## J

Justification, right, 15, 78-79

## K

Keeps, 95-103  
Kernighan, Brian, 136  
Koppel, Karla, 136

## L

.(l, 44-45  
.)l, 44-45  
Label on indented paragraph, 64-67  
laser command, 9, 123, 126, 129, 133  
Laser printer, 4-6  
laserq, 134  
learn files, 3, 9  
learn vi, 6, 9  
Left paragraph, 18-19  
Left quote mark, 36-37  
Lesk, M. E., 136  
Letters, 122-124  
Library books at NCAR and HAO, 136

Lines  
adjustment, 78  
blank, 18  
centered, 24-25  
extra on page, 92-94  
filling, 15, 42-43  
horizontal, 77, 101-103  
length, 52-53  
length, with page offset, 84-85  
of characters, 77  
of dots, 77  
of stars, 77  
spacing, 16-17  
Lists, 44-45  
distribution, 56-58

in keeps, 95  
multi-columned, 59-60  
two-columned, 56-58  
.ll, 52-53, 84-85  
.lo, 22-23, 31-33, 70-73, 106-107  
Local commands, 22-23, 31-33, 70-73,  
106-107, 108-111  
Login/logout, 4  
Lomuto, Ann and Nico, 137  
Lozy, Mohamed el, 136  
.lp, 18-19, 113  
.ls, 16-17

## M

.m1, 92-94  
.m2, 92-94  
.m3, 92-94  
.m4, 92-94  
Macros, 104-105  
Manual, how to use, 2-3  
Margins  
    bottom, 92-94  
    four, 92-94  
    left, 20-21, 80  
    one, 92-94  
    right, 52  
    three, 92-94  
    top, 88-94  
    two, 92-94  
Mark, 59-60  
Mathematical characters, 130  
McCune, Linda, 136, 137  
-me Reference Manual, 136  
Memos, 122-124  
Minus sign, 86, 129  
.mk, 59-60  
More, 129, 135  
ms Dictionary, 137  
-ms package, 137  
Multi-columned lists, 59-60  
Muster, John, 137

## N

.na, 78-79  
.\*n, 68-69, 72-73  
.\*N, 68-71  
.nf, 42-43  
.nh, 86-87  
No adjust, 78-79  
No fill, 42-43

Non-printing character, 76, 90-91, 126  
Notes at end of paper, 114-115  
.np, 68-69  
.nr, 50-51, 70-73, 108-109  
Number register command, 50-51, 70-73,  
108-109  
Numbered paragraphs, 68-73, 74  
Numbering  
    assigning format, 112  
    end notes, 114-115  
    pages, 34-35, 112-113, 133

## O

.ou, 108-111  
Outlines, 108-111  
Output, 15

## P

Page numbering, 34-35, 112-113, 133  
Page offset, 80-85  
Page offset with line length, 84-85  
Paragraphs, 18-19  
    indented, 64-67  
    left, 18-19  
    numbered, 68-73  
    with base indents, 74-75  
.pd, 114-115  
.pn, 112  
.po, 80-85  
Points, 20, 30-33, 106  
.pp, 18-19, 113  
Previous font, 26-27  
Printer at HAO, 4  
Printing, 9  
    delayed text, 114-115  
    equations and tables, 126, 129  
    jobs waiting, 134  
    long documents, 112  
    part of a file, 133  
    steps in, 123  
    table of contents, 116-117  
    tables, 126  
Prompt sign, 7, 134  
.ps, 32-33  
Public user file, 124  
Publications list macro, 104-105  
Punctuation marks, 15, 36

## **Q**

.q, 36-37  
.jq, 38-39  
.q, 38-39  
Queue, 134  
Quotation marks, 28, 36-37  
Quotes, block style, 38-39

## **R**

.r, 26-27  
Ragged right, 78  
*Reference Handout*, 136  
Resources, additional, 136-137  
Return, 59-60  
Right justification, 78-79  
Right quote mark, 36-37  
Roman font, 26-27  
Roman numerals in page numbering, 112  
.rt, 59-60

## **S**

.\*s, 31-33  
Sample letters & memos, 122-124  
.sc, 61  
Section headings  
    indented, 50-51  
    numbered, 46-49  
    text with, 48-49  
    unnumbered, 46-49  
.sh, 46-51  
Single quotation marks, 36  
Software, 4  
.sp, 16-17, 90-91  
Space  
    at beginning of line, 15  
    blank horizontal, 62-63  
    horizontal, 84  
    left by keep, 96, 101  
    top of page, 88-91  
    unpaddable, 76  
Spacing  
    double, 16-17  
    line, 16-17  
    triple, 16  
    vertical, 16, 106-107  
Special characters, 61, 126, 129, 130-131  
spell command, 134-135  
Spelling errors, 134-135  
Star commands, (see Local commands)  
Star to mark references, 40

Subscripts, 76  
Superimpose, 40, 76  
Superscripts, 76  
Symbols, 61, 130-131  
.sz, 32-33

## **T**

Table of contents command, 116-117  
Tables, 126, 133  
    other manuals on, 136  
tbl, 126, 129, 133  
*Tbl -- A Program to Format Tables*, 136  
.TE, 126  
Temporary indent, 20-21  
Terminal & screen, 4  
Terminal beeps, 8  
Text editing, 7  
Text formatting, 5  
Text, boxed, 126  
There exists symbol, 61  
.ti, 20-21  
Tilde symbol, 61  
    for blank space, 62-63  
Title page, 88-90  
Today's date, 76  
.tp, 88-90  
.tr, 62-63  
Translate space, 62-63  
Troff, 4  
.TS, 126  
Two-columned lists, 56-58  
Two-columned text, 54-55  
Type size, 30-33  
*Typesetting Mathematics--User's Guide*,  
    136  
*Typing Mathematical Formulas with  
    UNIX*, 136

## **U**

.u, 28-29  
.uh, 46-49  
Umlat symbol, 61  
Underlining, 28-29  
UNIX at HAO, 4  
*UNIX Command Notebook for Word Pro-  
cessors*, 137  
UNIX commands, 6-9, 133-135  
    other manuals on, 136  
*UNIX for People*, 137

*UNIX Primer, A*, 137

Unpaddable space, 76

Up command, 76

## V

.\*v, 106-107, 110-111

vax computers, 4

Vertical spacing, 16, 31, 106-107

    in outlines, 110-111

vi, 7

vi crib sheet, 138-139

.vs, 106-107

## W

Word processing commands, 6-9, 16-120

    backslash commands, 14-15, 26, 76-77

    dot commands, 14

    embedded, 5

    other manuals on, 136

Words, boxed, 118-119

Write quit (wq), 6, 9, 123

*Writing Papers with NROFF Using -me*,  
    136

## X

.(x, 116-117

.)x, 116-117

.xp, 116-117

## Z

.(z, 100-103

.)z, 100-103

Zeros, 127

## Other Characters

.., 104-105

\", 77

\\", 77

\%, 86-87

\&, 76, 90-91

\(space), 76

\\*#, 114-115

\\*(lq, 36-37

\\*(rq, 36-37

\\*(td, 76

\\*\*, 40-41

\d, 76

\f, 26-27

\'Nc', 77

\s, 32-33

\u, 76

\, 129, 135