

Copyright © 1972, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**RUNOFF**  
**A PROGRAM FOR THE PREPARATION OF DOCUMENTS**

by

**Richard Herzog and Nagib Badre**

**Memorandum No. ERL-M331**

**March 1972**

**ELECTRONICS RESEARCH LABORATORY**  
**College of Engineering**  
**University of California, Berkeley**  
**94720**

Table of Contents

Acknowledgments	p 2
1.0 Introduction	p 2
1.1 Input Text Format	p 4
1.2 Format Control Characters	p 5
2.0 General Information about Commands	p 7
2.1 General Actions	p 8
2.2 Margin Controls	p 9
2.3 Paragraph Formatting	p 10
2.4 Special Line Justification and Control	p 11
2.5 Heading and Paging	p 11
2.6 Lines and Spacing	p 14
2.7 Miscellaneous	p 14
Summary of RUNOFF commands	p 17
Default Conditions and Editing Conventions	p 18
Character Set	p 20
Punching	p 21
RUNOFF Index	p 22

## Acknowledgements

Most of the specifications for this implementation of RUNOFF and a good part of this manual come from project GENIE. Implementation took place during the month of December 1971 under support of NIH Grant USPHS-GM-17523.

## 1.0 Introduction

RUNOFF is a document preparation program inspired by the program of the same name written by Larry Barnes at the GENIE project. Input for RUNOFF consists of the text itself together with various editing and layout commands: e.g., horizontal and vertical margins can be set, paragraph positioning can be defined, etc. In short, one can obtain neat documents (such as this one) with relatively little effort. RUNOFF is most useful when it is expected that a document may be rewritten with new corrections several times. The first copy could probably be produced more easily using a typewriter. But correcting a typewritten document might involve retyping an entire page because of one or two lines, or even retyping the whole document because margin settings were not appropriate. In most cases, RUNOFF will require repunching a few cards only. Moreover, the corrected version will be as neat as the original: no page will have too many or too few lines, page numbering will be automatically updated, etc. This rather large programming effort was undertaken to provide a suitable tool for the documentation of the CCSEM.

Control over the output is exercised by the user in two ways :

- (1) By the use of special control character combinations inserted anywhere within text input lines. See Section 1.2.

e.g., "°CTHIS IS A °I/SENTENCE."  
becomes "THIS is a Sentence."

- (2) By the use of command lines interspersed in between text lines. These lines begin with a period '.'. See Section 2.

e.g., ".DS" means "start double-space mode."

This version differs considerably in its implementation and its motive from the XDS 940 version, but most of the commands have remained the same, so that people who have used extensively one system (like the authors) will have few problems switching from one version to another. This version is implemented on an IBM 1800, with 16K of core, and a very

simple operating system. It is designed around card input and works with only one pass over the data, printing while formatting the next line. When RUNOFF is started, a certain number of default conditions are preset as explained in the following sections (see also the summary). On the other hand, sense switch 0 being up indicates the special printing ball, with the complete upper-lower case set. If sense switch 0 is down, then the normal ball is assumed, and all letters are upper case. Sense switch 2 must be down for normal operation, since it controls trace print outs. Trace calls are placed in non fully debugged parts of RUNOFF to collect information on weird conditions.

1.1 Input Text Format

Text input to RUNOFF consists of a deck of cards containing both the document to be printed and the control information for formatting. A physical line is a string of characters on one card. After reading in a card, trailing blanks are removed from the right end. In "FILL" mode, one blank is then added back to the right (a physical line may have up to 81 characters). A word is not allowed to overlap on two consecutive cards. All physical lines which do not begin with "." are considered lines of the document to be output.

Initially RUNOFF is set to "FILL" mode. All physical lines are collected until a long enough line is ready for output, then the string is broken at a blank character to fit within the line margins. Since the "adjustment" switch is initially turned on, the string is then filled with extra blanks, uniformly and starting alternatively from the left or the right, so that the printed line will be adjusted against the right margin as well as the left margin. Printing will then occur. The remaining part of the input string will be compared against the line length, and if too short, cards will be appended to it until the input string grows enough to be printed. For example, this paragraph itself may result from the following input cards:

```

                                /INITIALLY °CRUNOFF
IS SET TO °C"FILL" MODE. /ALL
PHYSICAL LINES
ARE COLLECTED UNTIL A LONG ENOUGH LINE IS READY FOR OUTPUT,
THEN
THE STRING

```

etc.

The end of a logical line, or line break, occurs when one of the following conditions are met:

- (1) There are one or more blank lines between two lines of text.  
Extra blank lines in the text give in fact a line break and a line skip as one might expect. The user can have the same effect as a blank line by using the command ".sp 1" if in single space mode, or ".sp 2" in double space.
- (2) The next line begins a new paragraph. (RUNOFF recognizes paragraphs by seeing one or more blanks at the beginning of a line of text.)

- (3) There is a logical line break. This is produced by the BREAK command ".BR" (see Section 2.6) and certain other commands (see summary for a full listing).

## 1.2 Format Control Characters

Any character in an actual text line is interpreted as text except for the escape and shift characters: "°" and "/" respectively. Normally, these are not printed but interpreted as in-line commands. (Thus, to output them as such, one must use "°°", or "°/".)

The escape and the character immediately following have special meanings as defined in the table below.

Note: In the following, "word" designates a sequence of characters normally ending with a space or an end-of-field character. These are any of the following: ".,;:)=&?!" but not including the delimiting double quotes "". (Also, a word can be made to cross over these boundaries using "°G" - see below.)

- °B Backspace-Overprint the preceding character with the next character.  
e.g., "l°B-" yields "1".  
(Using °B at the beginning of a card may give unexpected results.)
- °C capitalizes the next "word" up to a space, end-of-field, or °E.  
e.g., "°CTHIS-IS,ONLY AN °CEXAM°EPLE."  
yields "THIS-IS,only an EXAMple."
- °E is used to force the end of underlining and capitalization before the normal boundary occurs, or defines the end of a group.
- °G defines the beginning of a group. The group ends with °E or the end of the card. The group feature forces several characters to form one "word": a group will always appear on the same line; it will not be "filled" (the number of blanks within the group is preserved); a group will be entirely capitalized or underlined if under the scope of °C or °I. (This command can be

useful for in-line formulas.)

- °I underlines the next word up to but not including a space, end-of-field, °E, or the end of the card.
- °K switches to black ribbon. This is the initial mode.
- °L sets lower-case mode. This is the initial mode. Upper case letters are converted to lower case. Only letters A to Z are affected by this command. Depending upon the typing ball used, this command may or may not have any result.
- °R switches to red ribbon.
- °U sets upper-case mode. Upper case letters A to Z are not converted to lower case. This command remains in effect until a "°L" is found (see above).  
e.g., "°UBLA, BL°LA." yields "BLA, BLA."
- °- underlines all characters up to an end of card, or °E. Note that, if left to the end of the card, this command will include an underlined space at the end of the line.
- °x Other characters following "°" are not recognized as commands, and are not re-coded. That means that °N will have the same effect as /N, since the characters are normally in upper case EBCDIC. The only way to print the escape, or the shift characters is to have them preceded by a "°".
- (underscore) is re-coded so as to be followed by a backspace (as if it was a dead key on a typewriter). This can be avoided by using "°\_".
- / the shift character converts the next letter (A to Z) to upper case. Other characters are not affected, nor does it have any effect if under upper case mode (°U).  
e.g., "G/ H/I J" yields "g hI j"  
It must be emphasized that "/" only affects the next letter if it immediately follows. Otherwise it is ignored.

Note 1: Commands can be juxtaposed freely and will behave as expected. Thus:

"°L°C°IBL/A°GBLA, BLA°EBLA /BLA."

yields:

"BLABLA, BLAbla Bla."

Note 2: A command with no effect ("/" not followed by a letter, "°L", "°U, if the same mode is already set, etc.) appearing at the end of a physical line may be used to force the preceding blanks as part of the input string.

This is mostly useful in some commands such as paging mode prefix (see 2.5 below):

```
.PM PREFIX /PAGE /
```

will define the prefix to be "Page "

## 2.0 General Information about Commands

A command line is recognized by the first character being a ".". This is usually so, but can be over-riden by the preceding card being ".li", for LITERAL. A command card is read independently of the current upper/lower case mode setting. Following the dot, RUNOFF expects an abbreviated command name, followed by parameters for that command, followed optionally by a comment. Lines beginning with a dot but having unrecognizable format are treated as null, and perform no action whatsoever. No error message is output.

All commands are described in the abbreviation lists. Commands which should cause a logical line break do. Several commands may specify an additional argument requiring that a number of blank lines be skipped. In general, this is equivalent to the same command followed by ".sp n" where n is the additional parameter. This is indicated by "(sp)" with the command explanations in the following sections. It is also indicated in the Summary. Note that the same restrictions as for ".sp n" apply: if one of these commands is used towards the end of an output page, it may happen that less than n blank lines are actually skipped.

In a command line, a numeric parameter may start with any number of blanks and terminate with a comma, or a blank if no other parameter follows. If a parameter is optional, the preceding comma may be omitted. Also, if it is to be followed by still another parameter, then a comma must be used to indicate the missing parameter (also if a comment is to be used). String parameters, used for defining a header, a prefix or a suffix, start with the first non-blank character after the command abbreviation and terminate with the rightmost non-blank character on the same card (see Note 2 in section 1.2). Thus a string parameter cannot be followed by a comment. A string parameter may specify a null or empty string in which case the command abbreviation must appear by itself on the card. On the other hand, the string may include in-line control characters (see Section 1.2) to specify upper or lower case, etc.

## 2.1 General Actions

.fi            for .fill     (sp)  
.nf            for .nofill  (sp)

These commands determine which line-processing routine should be used. ".fi" causes RUNOFF to either lengthen short lines by filling with words from the following line, or shorten long lines by breaking between words. This may be performed in one of two modes, depending on the setting of the adjustment switch (see next set of commands). Initially, RUNOFF is in "adjusted fill" mode as explained above in Section 1.1. Note that filling never takes place across a logical line break.

In the no-fill mode each card starts a new line. This mode is used when it is desired to keep strict control over the printed lines. No two cards get mixed together, no extra blanks are inserted within output lines. Each card is printed as it is read if possible. If it does not fit in the margins, then it produces two lines. Of course, in-line controls are processed as usual.

Both commands may optionally specify a number of blank lines to be skipped.

.ad            for .adjust  (sp)  
.nj            for .nojust  (sp)

These commands turn an "adjustment" switch on and off respectively. This switch is only effective in "fill" mode. When off, filling will simply consist of putting enough words on each output line to stay within margin limits. The number of blanks between words is preserved. When the switch is on, each output line is prepared as before but blanks are added between words to insure that all lines are right justified. This is the initial state of RUNOFF.

Both commands may optionally specify a number of blank lines to be skipped.

2.2 Margin Controls

There are two types of margins involved in RUNOFF:

- 1- The physical margins. These are determined by the nature of the printing device. In our particular implementation, this is now an IBM 1816 typewriter (Selectric), but provision has been made for upgrading to a line printer. The margins are built into the code, and there is no command for changing them.
- 2- The logical margins. These can be set by the user as he wishes. (limits are imposed by the physical margins.) They are initialized for standard "8.5 by 11" printing.

Commands concerning vertical and horizontal margins are:

`.pl`            `.page layout ipst,ipend,itol`

This sets the vertical logical margins and vertical tolerance. Parameters are top margin, bottom margin, and tolerance. The printed text (not the header) starts at line IPST, and ends between IPEND-ITOL and IPEND. The tolerance is used to determine where to break pages. If there is a line break within ITOL lines of the bottom, RUNOFF will break the page there; otherwise, it will fill the page completely. (default values are: 7, 61, 4)

`.ll`            `.line layout ilm,irm,ncol,ksp`

This sets the logical left and right margins, the number of index columns and their spacing (only 1 or 2 columns). These margins are used for the page headings, and for restoring the margins when ".er" is found. (default values are: 12, 72, 1, 4.) To adjust the relative text position, use the subsequent commands.

`.em`            `.expand margin lme,rme (sp)`  
`.rm`            `.reduce margin lmr,rmr (sp)`  
`.er`            `.end reduction (sp)`

These commands enable the user to indent a certain portion of his text using the first command, or "undent" his text using the second command. In either case, the original margins are restored by the third command, they are restored to the margins defined by ".ll". The action is cumulative, to indent more and more portions of text, use several ".em" before using an ".er" to restore the margins. Logical margins can be conceived of as being temporarily represented by lm and rm, and somewhat more permanently by ilm and irm. ".ll"

sets `ilm` and `irm`, and sets `lm = ilm`, `rm = irm`. `".em"`, `".rm"`, and `".er"` act as follows:

`.em` : `lm = lm + lme`, `rm = rm + rme`.

`.rm` : `lm = lm - lmr`, `rm = rm - rmr`.

`.er` : `lm = ilm`, `rm = irm`.

Note that unlike project GENIE RUNOFF, these commands are not implemented recursively, and to align back the text, one must remember how much it was indented. `".er"` restores margins to their original values, not to those in effect before the last `".em"` or `".rm"`.

Negative numbers are permitted. On the other hand, these commands do not affect the position of page headings, which are always printed according to the main line layout as specified by `".ll"`.

All three commands may specify optionally a number of blank lines to be skipped

### 2.3 Paragraph Formatting

Paragraphs are only meaningful in "FILL" mode. In this mode, they are recognized by one or more leading blanks at the beginning of a card. No matter how many blanks are used, paragraph indentation will be performed as specified by `".pi"` (see below).

`.ps`            `.paragraph spacing n`

This specifies how many lines are to be inserted between paragraphs. This is the exact number of lines and is not in addition to the current setting for line spacing. Thus, in double-spaced text, `".ps 1"` is permitted (even though it is not standard practice). (default value is 1).

`.pi`            `.paragraph indentation n`

This specifies the indentation of the beginning of a paragraph. (default value is 5.)

`.pu`            `.paragraph undentation n`

Same as `".pi -n"`.

## 2.4 Special Line Justification and Control

These commands pertain to the next card only (unlike GENIE System). If the next card is not a text line, i.e. if it starts with a ".", these commands will have no effect. Line justification is relative to the current left and right margins as set by ".ll" and ".em", ".rm", or ".er" (see above).

All commands in this section may optionally specify a number of blank lines to be skipped.

`.ce`            `.center` the next line (sp)

The next line is centered between the current left and right margins. If it is too long, it may overflow beyond the margins. Centering may then be lost since the first character cannot be printed left of the first physical column on the output device. On the other hand, a string may be intentionally printed "slightly to the left or to the right" of the center by including a few blanks on one side of the string to be centered. A slash "/" should then be used to force RUNOFF into counting those blanks as part of the string.

In "FILL" mode, a "centered" line does not participate in the filling. It is processed separately: a logical line break is assumed before and after it.

`.in`            `.indent` n (sp)

Indent the next line N spaces. If N not provided, 5 is assumed. If the next card is long and produces several output lines, only the first one is indented, as one might expect.

`.un`            `.undent` n (sp)

Identical to ".in -n" .

## 2.5 Heading and Paging

`.he`            `.header` xxxxxxxx

RUNOFF accepts a heading to go on the first line of each page. This can be specified through the parameter, which follows the standard string parameter conventions as explained in Section 2.0. Default convention is a null (or absent) header. The same effect can be achieved by the command ".he" alone on the card.

`.hm`            `.heading mode (param)`

(param) determines how the header is going to be printed. (param) may be any of the following. Only one parameter is allowed for each ".hm" command; several commands can be used however. Their effect is cumulative, or if conflicting, the latest wins.

#### BLACK

The header will be printed in black (default condition).

#### RED

The header will be printed in red, and ribbon will be switched to black after printing it, so that text always starts in black on top of page. Problems may arise if text is being printed in red, using the "°R" command. Any (unexpected) page break will force return to black ribbon. This inconvenience may be corrected in some later version of RUNOFF.

#### LINE N

This command determines on which line the header is printed. Beware that this number must be smaller than the first line number for text printout (as defined using the ".pl" command, see Section 2.2). Otherwise, results are not guaranteed. (Default setting is 4.)

`.pm`            `.paging mode (param)`

This command determines the placing of the page number. The same comments concerning the use of (param) apply here as for the ".hm" command (one parameter per command, but several commands naturally possible; see above for details).

#### TOP

The page number is placed right justified on the header line. This is the initial default condition.

#### BOTTOM

Page numbers are centered on the bottom of the page, the line on which they are printed is at IPEND+2 (for the definition of IPEND, see the ".pl" command in Section 2.2).

## OFF

No page number is printed, but the header is still printed unless null. This command remains in effect until either ".pm top" or ".pm bottom" is found.

## PREFIX XXXXXX

## SUFFIX XXXXX

These commands define the prefix and suffix strings which are printed on both sides of the page number. The full page string is then printed in the form:

[prefix][page number][suffix]

(Note that no blank separators are added so that the page number may be printed, e.g., as follows: "-12-", by setting both prefix and suffix equal to "-"; as was used for printing this manual )

Both strings follow the standard string parameter conventions as explained in section 2.0; in particular, either string may be null. The default condition is as follows: prefix is set to "Page " and suffix is set to null.

.pa            .page n

If N is present, insert a page break, and start numbering the next page with N. Otherwise, turn the paging mode to off, and reset page number to zero. (The meaning of page breaks is explained in the note after ".ej" below.)

.ej n            .eject n

Insert a page break if either there are fewer than N lines left on the page or N is not present. (Thus, note that ".ej" is equivalent to ".ej 1000", not to ".ej 0" as one might have expected.)

**Note:** When a page break is performed (with ".ej" or any other command), printing on the current page is terminated. But the next page is not started yet until some real output (text or spacing) is requested after the page break. In particular, several contiguous page breaks do not result in empty pages of output. (Rather, this could be achieved by ".ej", ".sp", ".ej", ...)

## 2.6 Lines and Spacing

In this section, all commands (except obviously ".sp") may optionally specify a number of blank lines to be skipped.

**.ss**            .single space (sp)

Single space all lines, within paragraphs if in FILL mode, or in any case if in NOFILL mode. This is the initial state.

**.ds**            .double space (sp)

Double space all lines, within paragraphs if in FILL mode, or in any case if in NOFILL mode.

**.sp**            .space n

Output N line spaces. If N is not provided, 1 is assumed (same as a blank card in single space). In case of page overflow all remaining blank lines to be output are deleted and a page break is made (see note after ".ej" above). This command does not guarantee n blank lines. If that number is really needed, e.g. for manually drawing a figure, then one should use the two-command sequence: ".ej n" followed by ".sp n".

**.br**            .break (sp)

The lines before and after this command will not be run together in fill mode. A blank card performs as a break, but gives a line feed too. In a sense, ".br" allows a separation in paragraphs but with the normal line spacing and no indentation. Thus, this very paragraph is separated from the next by a break command.

Note that ".br n" is equivalent to ".sp n".

2.7 Miscellaneous

`.ul`            `.underline (sp)`

The following line is underlined. The line will however include a terminal underlined blank (even if card column 80 has a non-blank character). It is usually better to surround the string to be underlined by "'-' and "'E" as explained in Section 1.2. This command may optionally specify a number of blank lines to be skipped.

`.li`            `.literal`

The next line is taken as part of the text whether or not it begins with a dot.

`.ix`            `.index [phrase],[phrase]`

RUNOFF saves the whole string in an index table. This table is sorted after an ".en" is found, and printed out in alphabetical order. No more than 200 entries are permitted in the index, since the present implementation keeps the strings and indexes in core. (More freedom will come when the tables will be kept on files.) Output format is now restricted to one column. An index command may specify a simple entry: one phrase with no comma. Otherwise, it may specify a category and an entry within that category: the comma is then required, the preceding phrase defines the main category, and the entry proper follows the comma. All entries within a category are printed under the category's name, in alphabetical order (see index at end of manual for examples.)

`.pr`            `.print`  
`.np`            `.noprnt`

These two commands are intended to relieve the typewriter from a little of the burden imposed on it. In noprnt mode, nothing is printed, but processing is done as usual, as far as indexing, line, and page numbers. When the ".pr" command is found, paper is adjusted to reflect the current line number, and print out resumes until a new ".np" command is found. To avoid confused mixing of printouts, ".np" generates a page break (see note after ".ej" in Section 2.5 above). This set of commands permits keeping an accurate index, and prints only the pages which need to be printed. Most useful when inserted at page breaks.

.26           .punch 026 input  
.29           .punch 029 input

These two commands assure compatibility between usual punches (026) and the IBM character code (punch 029). A few characters of the punch 026 need to be recoded, and they are when the flag 026 is on. This removes from the character set the following characters:

& # % [ @ which are recoded into + = ( ) '

Initial condition is punch 026.

For full details of punching specifications, see tables following the Summary.

.en           .end of input

This command stops card input. It gives a page break. It must be followed by a blank card at the end of the deck to satisfy IBM's system requirements. RUNOFF prints out the index table if the print flag is on, and will print out statistics about the previous input deck. RUNOFF then pauses, and restarts when the start button is pushed. All default conditions are then valid again.

## SUMMARY

In the summary, the following conventions are used for parameters.

Symbol	Meaning
N	a number
C	a character
S	a string
P	a general parameter

Any symbol may be followed by \*, indicating that its use is optional as that parameter of the command. If an optional parameter is left out, the effect is to replace it by the usual default condition. The use of + following a break type indicates that the actual break is dependent on the value or presence of the parameter. A "(sp)" indicates a special optional parameter specifying a number of blank lines to be skipped.

Abbreviation	Command	Type of Automatic Section Break	Section
.ad	.adjust (sp) . . . . .	line . .	2.1
.br	.break (sp) . . . . .	line . .	2.6
.ce	.center (sp) . . . . .	line . .	2.4
.ds	.double space (sp) . . . . .	line . .	2.6
.ej	.eject N* . . . . .	page+	2.5
.em	.expand margin N*, N* (sp) . .	line . .	2.2
.en	.end of input . . . . .	page . .	2.7
.er	.end reduction (sp) . . . . .	line . .	2.2
.fi	.fill (sp) . . . . .	line . .	2.1
.he	.header S . . . . .	none . .	2.5
.hm	.heading mode P . . . . .	none . .	2.5
.in	.indent N* (sp) . . . . .	line . .	2.4
.ix	.index S, S* . . . . .	none . .	2.7
.li	.literal . . . . .	none . .	2.7
.ll	.line layout N*, N*, N*, N* . . .	line . .	2.2
.ma	.margin . . . . .	line . .	2.4
.nf	.nofill (sp) . . . . .	line . .	2.1
.nj	.nojust (sp) . . . . .	line . .	2.1
.np	.noprint . . . . .	line . .	2.7
.pa	.page N* . . . . .	page . .	2.5
.pi	.paragraph indentation N . . . .	none . .	2.3
.pl	.page layout N*, N*, N* . . . .	page . .	2.2
.pm	.paging mode P . . . . .	none . .	2.5
.pr	.print . . . . .	line . .	2.7
.ps	.paragraph spacing N . . . . .	none . .	2.3

.pu	.paragraph undentation N . . .	none . .	2.3
.rm	.reduce margin N*, N* (sp) . .	line . .	2.2
.ss	.single space (sp) . . . . .	line . .	2.6
.sp	.space N* . . . . .	line . .	2.6
.un	.undent N* (sp) . . . . .	line . .	2.4
.ul	.underline (sp) . . . . .	line . .	2.7
.26	.punch 026 input . . . . .	none . .	2.7
.29	.punch 029 input . . . . .	none . .	2.7

### Default Conditions and Editing Conventions

The list of commands which follows is "executed" at each initialization.

```
.26
.pr
.pl 7, 61, 4
.he
.hm black
.hm line 4
.pm top
.pm prefix °L°K/page /
.pm suffix
.pa 1
.ll 12, 72, 1, 4
.ss
.fi
.ad
.ps 1
.pi 5
```

These default conditions were established more or less arbitrarily. The user is strongly urged to decide once and for all on his own conventions. It may be helpful to note them in the space above, facing the original conventions set by the authors. Experience shows that it is helpful to also decide in advance on a number of other editing or layout matters (e.g., disposition of section titles, ".ej n" preceding them, etc.).

The authors often use the following set of conventions. Line layout is ".ll 1, 61, 1, 4" and must be used with pre-adjustment of the typewriter's physical margin (e.g. to 15), so that printing time is about 10% smaller. This may interfere with undentation. Page layout is taken as the default ".pl 7, 61, 4", and the header is on line 4. Page numbers, when printed, appear on the bottom of the page, as with the commands:

.pm bottom  
.pm prefix -  
.pm suffix -

Within chapters, section titles are underlined; they are preceded by ".sp 5", ".ej 10" and followed by ".sp 3", and the appropriate index commands. Whenever necessary, subsections (e.g. "4.7.1 Initialization") are used without underlining; they are preceded by ".sp 3", ".ej 6". The most important thing is to define your own set of conventions (if possible within the American standards) and stick to them.

It seems appropriate here to remind the user of a few American standard printing conventions:

- Generally use single space, except for college term papers (".ss")
- One inch margins should be left on top and bottom of each page (".pl 7, 60" or ".hm line 7", ".pl 10, 60" depending on interpretation).
- Left and right margins should be 1.5 and 1 inches respectively (".ll 12, 72" if the physical margin of the typewriter is set as usual). Note that University thesis regulations require a 2 inch margin on the left. This makes binding easier (".ll 18, 72"). In both cases, the number of characters per line (60 or 55) should be kept in mind when using "NOFILL" mode.
- Paragraphs should be separated by one blank line (".ps 2" in single space, ".ps 3" in double space).
- Standard paragraph indentation is 5 (".pi 5").
- Within the text, two blanks should follow either of ".", ";" or ":". One blank is required after ",",
- Use double quotes preferably to single quotes for actual quoting of text from an external source. Single quote is better used only as an apostrophe, for possessive forms. Long quotations should appear single spaced, indented on both sides relative to the main text, with no quote marks.
- Punctuation marks should be placed inside or outside a pair of quotes or parentheses in their natural place except that ".", ";" and "," should be always typed inside the quote marks, even if not part of the quote. This usage is often overlooked in scientific literature as it can be rather confusing.

Character set

The character set available is a dialect of EBCDIC: "greater than" and "less than" signs are replaced by square brackets; the "not" sign is missing; finally, the escape character ("logical or" in EBCDIC) is in fact printed "°".

```

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
                                0 1 2 3 4 5 6 7 8 9
a b c d e f g h i j k l m n o p q r s t u v w x y z
, ; . : ! ?   + - * =   $ ¢ %   ( ) [ ] " _ ' ° # @

```

In general, unknown characters will be interpreted as blanks (including for breaking and filling lines). Two characters are printed as blanks, but not considered as blanks for defining words, breaking at the end of the line, etc. These are the logical not and a special character: punch codes -,7,8 and 12,7,9 respectively. This last special character is automatically substituted for blanks when they appear within the scope of a group command (°G ... °E).



## RUNOFF Index

adjust 8,17  
ball,  
    switch 3  
    type 3  
bottom,  
    paging mode 12  
break 8,14,17  
center 11,17  
character set 20  
control characters 5  
default conditions 18  
double space 14,17  
editing conventions 18  
eject 13,17  
end of input 16,17  
end reduction 9,17  
expand margin 9,17  
fill 8,17  
header 11,17  
heading mode 12,17  
indent 11,17  
index 15,17  
layout 9,9  
line 17  
line break 7,17  
line layout 9,17  
literal 15,17  
logical line 4  
margin 9,17  
nofill 8,17  
nojust 8,17  
noprint 15,17  
off,  
    paging mode 13  
page 13,17  
page break 13  
page layout 9,17  
paging mode 12,17  
paragraph 10  
paragraph,  
    definition 4  
    indentation 10,17  
    spacing 10,17  
    undentation 10,18  
parameter,  
    numeric 7  
    string 7  
physical line 4  
prefix 13  
print 15,17  
punch,  
    026 16,18  
    029 16,18  
reduce margin 9,18  
single space 14,18

## RUNOFF Index

space 14,18  
suffix 13  
top,  
    paging mode 12  
undent 11,18  
underline 15,18