PRIME Technical Update

OBSOLETE

Subject: Runoff

Number: 16

Revision: 0

Date: October 14, 1975

Applicable Hardware: All CPU's

Applicable Software: Supports Text Editors

Documentation Impact: Supplements PDS User Guide (MAN 1879)

Abstract: Defines and provides operating rules for RUNOFF, Prime's Text Formatter. RUNOFF accepts commands on-line or edited into text to control margins, indention, line spacing, column width, page numbering, running heads, and many other features of an ASCII source file. It produces a formatted output to SPOOL or a designated disk file.

This bulletin is one in a series of documentation supplements that supply current information on Prime hardware, software and documentation products. Prime Technical Updates introduce product improvements and revisions, and update existing Prime Computer user documentation.

PRIME Computer, Inc. 145 Pennsylvania Avenue, Framingham, Mass. 01701/(617) 879-2960

P/N PTU-3065

RUNOFF

Text processors fall into two main categories; text editors and text formatters. By text editors is meant document oriented text, rather than source program text. Pure text "editors" are rare and usually have formatting capabilities as well. Prime RUNOFF, strictly speaking, is a text formatter or, more appropriately, a re-formatter. Original source input, in free form, is entered using the standard Prime text editor (ED).

RUNOFF has a wide range of formatting options, including multiple columns per page, which are specified either in "command" mode or in the body of the text itself. In all, there are 60 RUNOFF commands which can, with three exceptions, be input from either the TTY (command mode) or from the text. A few other commands, while not treated as an error, do not perform any useful function from the TTY (for example, page ejection before any text has been read. There are some other commands which behave differently from the TTY or the input file, but these commands, for the most part, are intended for more sophisticated users and need not concern the novice or the user who just wants to "get it printed."

Since the RUNOFF program forms a cohesive system, some general notions and definitions are in order. Command mode gives the user the opportunity to enter commands to RUNOFF before text processing begins. This allows file specific commands to be included in the file, but run specific commands to be issued each run. An example of the difference is that the definition of the tab character or phantom hyphen (see below), as well as which portions of the text are to be right justified, paragraph indentation, identifying a header or footer (a header at the bottom of the page), etc. are all appropriate to a given file and should be included as part of the file. However, the paper size, the device to print the file on, whether perforation marks are to be printed (for continuous paper roll terminals), etc. all depend on the particular run and not on the file. The only restriction on command mode is that actual text cannot be entered from the TTY.

The command format is consistent between command mode and text processing. All commands begin with a decimal point (.) in column 1, followed the command name or an acceptible abreviation (see table at end) followed by a parameter if required. The only command which accepts multiple parameters is the .TAB command which can accept up to 20 tab stops. For convenience, in command mode, the initial period may be omitted, but in processing mode, the period, in column 1, is mandatory to distinguish commands from text. Commands can be entered in either upper or lower case.

The physical paper description consists basically of specifying the size of the paper, the size of the margins (white space around text), and the number of columns on a page. All width specifications are in terms of characters (spaces) and all length specifications are in terms of lines. Attribute specifications (e.g. number of columns, starting

page number, etc.) are always just numbers. The following conventions will be used throughout the rest of this document; the letter "m" is used to indicate a number of characters, the letter "n" is used to indicate a number of lines, and the letter "i" is used to indicate an attribute. In addition, whenever a character is to be used or specified, the designation "char" will be used. The physical paper description includes a specification of length and width (including any margins), a top and bottom margin, a side margin (both left and right are always the same), a center margin (between columns) if multiple columns per page are used, and the number of columns. RUNOFF is set up to a default specification of 85 characters X 66 lines (standard 8 1/2 X 11 inch paper), single column, with side margins of 7 caracters (spaces) and a top margin of 7 lines and a bottom margin of 5 lines. The maximum page size is 170 characters X 132 lines.

Text formatting consists of several options. The main ones are adjusting and filling of text. Filling means that words are taken from the input file as needed to fill up a line in the output text, regardless of the physical layout of the input file. Thus, many input file lines or only a part of one input file line may all go into one output line. Words are taken from the input file until the next word will not fit on the current line. At that point, a break (line advance) occurs in the output text and the word will be placed on the next line. A word is defined as a sequence of non-blank characters. Thus, the sequence "this-is-one::long+word" is treated as a single word of 22 characters (not including the "'s) whereas the sequence "con cat en ation" is treated as 4 words. If a word is too long to fit on an empty output line, as much of the word as will fit is placed on the line and the remainder is placed on the next line. All words in the output text are separated by a single space, regardless of the number of spaces in the input file, except those words which end with the characters: . ! ? : ; and) or the character pairs: ." !" ?" :" and ;" which are followed by two spaces. In no-filling mode, the input text is transferred, as is, line by line, to the output text (except phantom hyphens, blank substitutes, and tabs, see below). All spaces are copied as well as words. If the input line is too long to fit on a single output line, the line is "chopped" and the remainder is put onto the next line. Fill mode produces what is commonly referred to as a "ragged right margin" whereas no-fill mode is generally useful for tables which are already formatted (except, perhaps, for tabs) in the input file. In fill mode, there is also the option of right justification of the text so that the right margin will be straight, rather than "ragged." In adjust mode, the line is filled and then spaces are inserted to pad the line out to the right margin. The actual adjusting algorithm is complicated, but basically, it tries to spread out the pad spaces uniformly across the line. The double spaces following sentence terminators (. ! etc.) are treated as a single space for adjusting to avoid concentrating spaces between sentences. Also, since usually the number of pads to be inserted is small compared to the number of spaces on the line, adjusting alternates between padding from the right and padding from the left. However, the first line of a paragraph will always be padded from the right. Obviously, adjusting does not make much sense when not also filling so the modes

are coupled to allow only the following three possibilities: fill + no-adjust, fill + adjust, and no-fill + no-adjust.

Paragraphing in text is usually indicted either by indenting the first line or by skipping a line and not indenting. RUNOFF allows either possibility in fill mode. Paragraphs are signaled either explicitly (.PARAGRAPH) or implicitly (first character on input line a space). In either case, a break occurs after the previous line and, if not indenting, one printing line is skipped (unless the previous line was a blank line). The lead space for implicit paragraphing is removed when no indenting is performed. One consequence of implicit paragraphing is that a line must not begin with a space unless it is the first line of a new paragraph. In order to start a line with spaces but not make it a paragraph (tables, e.g.), the user has several options. The first, of course, is to leave fill mode. The second is to start the line (1st character) with a tab (see below), and the third is to change the left margin. Margin changing is often referred to as "sub-paragraphing" and consists of changing the left margin, right margin, or both. Four commands exist to allow moving either margin both in and out.

RUNOFF recognizes seven special characters: # = page number (in headers and footers only); .RUBOUT. ('377) = phantom hyphen; .NULL. ('200) = blank substitute; " = erase previous character (command mode only); ? = kill line character (command mode only); a tab character; and % = symbol delimiter (input file lines only). The characters shown, except #, represent default characters which are changeable by commands. In no case can a character be "doubled up" and if an attempt is made to define a character to a value which is already assigned to another character, an error results. Besides the setable characters, the following characters are illegal: .NL., .SPACE, .COMMA, /, #, *, +, >, ., ?, !, ;, :,), (, and ". The erase and kill charcters operate exactly as in the standrd text editor (ED) in command mode but are not recognized in the input file. Since the input is usually created with the text editor, there is no reason for these characters (functionally) to appear in the text whereas the default characters (",?) used in command mode are frequently used in normal text.

The phantom hyphen character is used to mark places, within words, where an actual hyphen can be inserted should the entire word not fit on the output line. Unless actually used to hyphenate a word, phantom hyphens will never appear in the text. This is true in both fill and no-fill modes even though the actual check for hyphenation only occurs in fill mode (even if not adjusting).

The blank substitute character is a "place holder" character which will be treated as part of the word it is in (exactly as if it were a letter), but is printed as a space. This is true in both fill and no-fill modes, though its major utility is in fill mode to force blanks which will not be thrown away like real spaces. In adjust mode, blank substitute characters are not eligible to receive pad spaces.

The tab character has no default value and is set each time tab stops are either set or changed. The reason or this is that the tab will

usually be a printing character which may need to be changed from time to time to avoid conflicts in the text. The tab character actually appears in the input file (unlike the editor tab which inserts spaces which would be thrown out by RUNOFF) but is recognized as a tab only in no-fill mode with the following exception: if the tab character is the first character on the input file line in fill mode, a break occurs and that input line is processed in no-file mode. At the end of that line, the mode reverts to the previous mode. Tab stops are specified relative to the current left margin and will therefore change, in absolute position on the line, if the left margin is changed. Therefore, if tabs are set to 10 and 20, and the left margin is at character position 15, the first printed character following the tabs will be at character positions 25 and 35. Note that tab stops do not indicate a number of spaces to skip (insert), but the starting position of the first character. In the previous example, if the tab character (if printed) was at character position 25, the next character would be printed at position 35. In other words, like a typewriter, if a tab is encountered at a tab stop, advance to the next tab stop. Unlike a typewriter, however, if tabs are specified past the last tab stop, or more tabs than tab stops appear on an input line, they will be ignored.

The page number character, #, is recognized as such only in a header or footer. A header appears at the top of a page and a footer appears at the bottom. A page can have both a header and a footer. In addition, headers (and footers) can be specified differently for even and odd page numbers or the same for all pages. It is possible, for example, to have different headers for even and odd pages, but the same footer on all pages. The format for both footers and headers is identical and consist of three portions, a left justified portion, a centered portion, and a right justified portion. It is not necessary to have all three. The left and right justified protions are set up according to the margins and paper width in effect at the time the header or footer is specified. Therefore, if paper size or format is to be changed, it should be changed before the header or footer is specified. Note that this is automatic if the header or footer is in the file and the paper format is specified in command mode. Headers are placed within the top margin and footers within the bottom margin by the following rules:

Header line = (top margin+1)/2
- if the top margin is odd, overlay the middle line
- if the top margin is even, overlay the upper of the middle two lines

Footer line = (last line-1)+(bottom margin+2)/2
- if the bottom margin is odd, insert above middle line
- if the bottom margin is even, insert between middle lines (Note: a footer will expand the bottom margin by 1 line)

The page number charcter will be replaced by a four column page number (leading zeroes are not printed) wherever indicated. The page number can be inserted as many times as desired in the header and footer. For

example, on page 2, the following sequence is transformed as follows:

```
        Page # of 10   -----> Page    2 of 10

On page 100,           -----> Page  100 of 10

On page 9999,          -----> Page 9999 of 10
```

Note that it is not equivalent to follow the # by three spaces or three more #'s. On page 2;

```
    Page #### xx   -----> Page    2   2   2   2 xx
```

A header or footer will be rejected if the total number of characters specified, counting all #'s as 4 characters, exceeds the available width (between margins) of the page.

The % sign is used to delimit symbols. Symbols are 1 to 6 character names which can be assigned a "value" of up to 30 characters. Symbol insertion is performed on the level of reading lines from the input file (symbols are not recognized in command mode though they can be defined in command mode). Since the substitution is accomplished at the lowest level in line processing, symbols may be replaced by commands, text, or whatever. For example, if a symbol named "cmd" is defined as ">center this", the input line: ".%cmd%" will be interpreted as: ".>center this" which is a command and will be treated as such. A maximum of 60 symbols can be defined at any one time. Symbol insertion is conditional in the sense that if reference is made to an undefined symbol, the reference is deleted. Thus, if the symbol "XYZ" is undefined, the line: ABC%XYZ%DEF will appear as: ABCDEF. A special set of symbols are defined for use in parameter passing to insert files (see descriptions of .INSERT and .RETURN commands). These are the special names %0%, %1%, ..., %9%. Since the % sign may often appear in text as a %, the symbol delimiter can be defined as a different character. Alternatively, the sequences %% and %.NL. will be treated as a single % rather than being deleted as undefined references.

The normal startup of RUNOFF is to type the command;

```
OK, RUNOFF fname
GO
COMMANDS
$
```

where fname is the name of the input file. If the name is not specified, RUNOFF will ask fot the "INPUT FILE: ". If a name is supplied, the operation is equivalent to specifying it in the initial command. If the name is again omitted, RUNOFF assumes that the input file is already open on DOS file unit 1. If a number is supplied (1,3-5,7-15), it is assumed that the input file is open on that DOS unit. Unit numbers 2 (output), 6 (command file), and 16 (index file) cannot be used as input units. Once all command mode commands have

been specified, typing a null line (C/R only) will begin processing of the input file.

RUNOFF has three options for output devices, the TTY, a file, or the line printer (actually, a SPOOL file). The default mode is to print to a SPOOL file only. The only combination of devices which is not allowed is to print to both a file and a SPOOL file.

Upon exit, RUNOFF will close all files that it explicitly opened. That means that all named files will be closed but all files indicated either by default (unit 1 for input, unit 2 for output - SPOOL) or by explicit unit number will be left open.

A break is a conditional line advance based on whether a line advance has just occured. An eject is a conditional page eject based on whether an eject has just occured. With a few exceptions, noted under the descriptions of the particular commands, an eject causes a page to be ejected, regardless of the number of columns per page and the current column.

Errors are treated in a uniform way in command and processing modes and are of two types: unrecognized command and illegal command. In the case of an illegal command, the entire command will be printed (if in processing mode) followed by the message "ILLEGAL COMMAND". For unrecognized commands, the command word is printed, followed by "UNRECOGNIZED". In all cases, if the error occurs in processing mode, a break and an eject is caused and return is made to command mode. Processing can be continued, if appropriate, by typing a null line (C/R only) in response to the $ prompt.

Following is an alphabetical list of all RUNOFF commands. Unless otherwise noted, a command is valid in both command mode or processing mode. For any command specified in command mode, the initial decimal point is optional. In the descriptions, minimum abreviations are indicated by a hyphen (as in BM-ARGIN) and all options are indicated as "m" spaces, "n" lines, "i" attribute number, and "char" for a character.

. = .null.

Begin processing input file.

.* text --->

Comment line. The "text" is used to document the RUNOFF input file and will be ignored by RUNOFF.

.+ text --->

Insert "text" verbatim into output. Any mode change implied by verbatim text is in effect only for this line. This command causes a break.

.> text --->

Center "text" between the left and right margins and insert verbatim. Any mode change implied by verbatim text is in effect only for this line. This commmand causes a break.

./text/text/text/

Apportion text into left, center, and right sections. The left portion will be left justified to the current left margin, the right portion will be right justified to the current right margin, and the center portion will be centered between the current margins. "Current" margins include any left and right indentation. The text is inserted verbatim and any mode change implied is in effect only for this line. This command causes a break.

.. text --->

Command escape. The double period (..) causes the "text" to be interpreted in the current mode with the first character a single period. Note that use of the .+ command inserts the "text" verbatim and this command inserts the text as if it were not preceded by a period.

.A-DJUST

Enter adjust and fill mode. This command causes a break.

.B-REAK

Cause a break.

.BL-ANK char

Set the blank substitute character to "char". The default setting is .NULL. = ^200.

.BM-ARGIN n

Set the bottom margin to n lines. Footer placement is recalculated as necessary. This command causes a break and a page eject. If n=Ø, the bottom margin is reset to 5 lines.

.C-OLUMNS i

Set the number of columns to i. All margins are recalculated as necessary. This command causes a break and a page eject. If i=Ø, the number of columns is reset to 1. (Any current indentation is set to Ø) If i>1, the default column margin (.CMARGIN) is 5 spaces.

.CM-ARGIN m

Sets the inter-column spacing to m spaces. This command is illegal if single column is in effect. This command causes a break and a page eject. If m=Ø, the column margin is reset to 5 spaces. (Any current indentation is set to Ø).

.D-EFINE symbol value

Define "symbol" to have "value". The symbol may be any length, but may not contain imbedded commas, parantheses, or spaces and only the first six characters will be used. "Value" may contain any characters an can be up to 3Ø characters in length. A maximum of 6Ø symbols may be defined at one time.

.EF-OOTER /text/text/text/

Set up footer for even pages. The slashes (/) are mandatory to delimit the left, center, and right portions of the footer. If a portion is not specified, its position must be indicated by two consecutive slashes. For example, a footer with a center portion only would be specified as: .EFOOTER //center//.

.EH-EADER /text/text/text/

Set up header for even pages. The slashes (/) are mandatory to delimit the left, center, and right portions of the header. If a portion is not specified, its position must be indicated by two consecutive slashes. For example, a header with a left portion only would be specified as: .EHEADER /left///.

.E-JECT

Causes a page eject (regardless of columnation).

.ER-ASE char

Set the command mode erase character to "char". If the default is changed ("), it is not possible to set it back to ". This command is illegal in the input file.

.F-ILL

Enter fill mode. This command causes a break.

.FO-OTER /text/text/text/

Set up footer for all pages. The slashes (/) are mandatory to delimit the left, center, and right portions of the footer. If a portion is not specified, its position must be indicated by two consecutive slashes. For example, a footer with a right portion only would be specified as: .FOOTER ///right/.

.FILE fname
.FILE

Specifies the output file. If no name is specified, the output is put onto unit 2 (SPOOL) which is assumed to be open. Note that this command does not cause a break and the output file can be switched in "mid-stream". If fname already exists, RUNOFF will ask if it is O.K. to delete the old fname and, if not, will ask for a new name.

.FR-OM i

Define the first page number to be printed. This is page number (see .PAGEN), not sequential page. If i=0, printing will begin at page 1.

.H-EADER /text/text/text/

Set up header for all pages. The slashes (/) are mandatory to delimit the left, center, and right portions of the header. If a portion is not specified, its position must be indicated by two consecutive slashes. For example, a header with a center portion only would be specified as: .HEADER //center//.

.HY-PHEN char

Set the phantom hyphen character to "char". The default setting is .RUBOUT. = ^377.

.I-NDENT m

Indent left margin m spaces. If m=0, the left margin is indented 5 spaces. Indentation is relative to the current left margin.

.INDEX string

Make entry into index file. The entire "string", up to but not including the .NL., will have the current page number appended to it and be written to the index file (see .IXFILE). The .INDEX command should immediately follow the reference being indexed to insure that the proper page number is used. If the .INDEX command immediately follows a page eject due to putting the last word on the previous page, the previous page number will be used.

.INS-ERT fname(PØ,Pl,...,P9)
.INS-ERT i(PØ,Pl,...,P9)
.INS-ERT (PØ,Pl,...,P9)

Insert a new input file. This command is used to insert text from alternate files without losing the initial (or previous) input file(s). Inserts may be nested up to 13 levels deep. If fname is specified, that file will be opened and the input stream assigned to it. If a unit number (i) is specified, that unit is assumed to be open and the input stream is assigned to it. If no name or unit number is specified, the next unit is assumed to have already been opened and the input stream is assigned to it. In command mode, this last option is equivalent to typing a null line to begin processing. This command is useful, in conjunction with the .RETURN command, for such operations as inserting names and addresses, contained in a separate file, into a standard form letter. See the .RETURN command for examples. The parameters enclosed in parentheses are optional, but if used, each parameter can be from 1 to 6 characters long and a maximum of 1Ø parameters (separated by commas) may be passed to the insert file. These paramters are referenced by the special symbol names %Ø%, %1%, ..., %9%. Each level of insert file has its own definition of parameters, but parameters can be "passed through" multiple levels by using commands such as .INSERT fname(%2%,P2,%Ø%). If a parameter is null (as in Pl,,P2), any reference to it will be treated as an undefined symbol and deleted. Thus, if multiple parameters are concatenated to form a variable length string, a reference of the form : %1%%2%%3% can be used to reference up to 18 characters. However, if less than 13 are needed as in: .INSERT fname(NAME,Abcdef,ghi) the following reference will be treated as:

%Ø% = %1%%2%%3% ----> NAME = Abcdefghi.

.IX-FILE fname

Define index file. The file "fname" will be openned on unit 16 and all .INDEX references will be written to it. Each use of .IXFILE will truncate and close the file openned by a previous .IXFILE command. Since .INDEX assumes unit 16 to be open, this can be done before starting RUNOFF and then not specifying an .IXFILE command. Upon exit, only an index file actually openned by RUNOFF will be closed. If fname already exists, Runoff will ask if it is O.K. to delete the old fname and, if not, will ask for a new name.

.K-ILL char

Set the command mode kill character to "char". If the default is changed (?), it is not possible to set it back to ?. This command is illegal in the input file.

.L-ENGTH n

Set the physical paper length (including margins) to n lines. Header and footer placement is recalculated as necessary. This command causes a break and a page eject.

.N-FILL

Leave fill and adjust mode. This command causes a break.

.NFILE

Specifies that no output is to go to files.

.NA-DJUST

Leave adjust mode. This command causes a break.

.NE-ED n

This command is used to specify that n printing lines are needed for a body of text to follow. By printing lines is meant actual lines of text rather than page lines. These are different whenever double, triple, or more spacing is used. If n lines remain on the current page, this command has no effect. If n lines do not remain, the command causes a break and an eject to the top of the next page or column if multiple columns are used. If n=Ø, 1 line is assumed.

.NP-ARAGRAPH

Causes paragraph indentation to be set to Ø spaces (no indentation) and to signal that a blank line is to be guarenteed between paragraphs.

.NPAU-SE

Do not pause between pages.

.NPE-RFORATE

Do not print page perforations. In .NPERFORATE mode, file output is preceded by a word = '401 which signals the line printer driver to use FORTRAN page controls and each new page is begun with a page eject code. This is invisible to TTY output.

.NT-TY

Do not print on the TTY.

.OF-OOTER /text/text/text/

Set up footer for odd pages. The slashes (/) are mandatory to delimit the left, center, and right portions of the footer. If a portion is not specified, its position must be indicated by two consecutive slashes. For example, a footer with left and center portions only would be specified as: .OFOOTER /left/center//.

.O-HEADER /text/text/text/

Set up header for odd pages. The slashes (/) are mandatory to delimit the left, center, and right portions of the header. If a portion is not specified, its position must be indicated by two consecutive slashes. For example, a header with left and right portions only would be specified as: .OHEADER /left//right/.

.P-ARAGRAPH
.P-ARAGRAPH m

Cause a new paragraph to begin. If m=Ø, the previous indentation mode (including .NPARAGRAPH) is used. If m is not Ø, paragraph indentation is set to m spaces and a blank line will not be put between paragraphs. Paragraph indentation is always from the current left margin as specified by both the .SMARGIN (side margin) and .INDENT commands. This command causes a break.

.PAG-EN i

Specify new starting page number. The next page to be printed, or the page after the current page if the command is given in the middle of a page, will be numbered i (as a replacement for the # character).

.PAU-SE

Pause between pages. When a new page is ready to be printed, RUNOFF will ring the terminal bell and wait for a character to be input before printing the page. This character can be any character at all. However, it should probably be a non-printing character since the primary use of .PAUSE mode is to allow setting up new paper for the page.

.PE-RFORATE

Print perforation marks between pages on both the TTY and the output file.

.PI-CTURE n

This command is used to indicate that n physical page lines (as opposed to printing lines) are needed for the later insertion of a figure. If n lines remain on the current page, these lines are skipped after the current output text line is completed, if in fill mode (i.e. a break does not occur). If n lines do not remain, the page is completed normally and n lines are skipped on the next page (or column). If n is larger than the number of available lines (not including top and bottom margins) on a page, as many pages (columns) and lines are skipped as are needed for a total of n lines. If one or more .PICTURE commands are encountered before an outstanding picture request has been satisfied, the n's will be added together as if one command specified all the lines.

.Q-UIT

Exit RUNOFF. If .QUIT is encountered in the input file, the current page is ejected and the output and any input files which were explicitly opened by RUNOFF are closed and return is to DOSV/M command mode (OK,). RUNOFF can be restarted by typing "S" or "S fname". If .QUIT is typed while in command mode, it is possible to lose the last page of output. When end-of-file is encountered in the primary (as opposed to an alternate) input file, it is as if a .QUIT were encountered.

.R-INDENT m

Indent right margin m spaces. If m=Ø, the right margin is indented by 5 spaces. Indentation is relative to the current right margin.

.RE-TURN i

Return to previous input (.INSERT) file. If i=Ø, the current file will be closed if it was actually opened by RUNOFF. If i=1, the current file will be left open. In all cases, return is to the previous input level. If a .RETURN is encountered in the primary input file, return is to command mode. If .RETURN 1 is used, a null line or .INSERT with no parameter will continue in the original file as if nothing had happened. This can be used to allow dynamic parameter changes during processing from the TTY. If .RETURN is used (i=Ø) to command mode, then resumption is only possible if the file were not explicitly opened by RUNOFF. Otherwise, only .INSERT fname or .INSERT i can be used to reusme file processing from a new file. The .INSERT/.RETURN combination is implemented using a file unit stack to insure proper nesting of input files, down to 13 levels. .RETURN always returns 1 level and .INSERT always goes down 1 level. A typical use of these commands for form letters is shown below: (Names and addresses file previously opened on unit 4.)

```
Commands            Primary Input       Form letter         Names &
--------            -------------       -----------         Addresses
                                                            ---------
RUNOFF CONTRL
$DEFINE date
$.NULL.
                    .INSERT FORM(Mr.)
                                        %date%
                                        Dear %Ø%
                                        .INSERT 4
                                                            name1
                                                            .RETURN 1
                                        text
                                        .INSERT
                                                            address1
                                                            .RETURN 1
                                        text
                                        .EJECT
                                        .RETURN
                    .INSERT FORM(Mrs.)
                                        %date%
                                        Dear %Ø%
                                        .INSERT 4
                                                            name2
                                                            .RETURN 1
                                        text
                                        .INSERT
                                                            address2
                                                            .RETURN 1
                                        text                    .
                                        .EJECT                  .
                                        .RETURN                 .
                    .QUIT                                   .RETURN
```

The necessity of opening the names and addresses file on unit 4 could be avoided as follows:

```
Commands          CONTRL            DUMMY             Names &
--------          ------            -----             Addresses
                                                      ---------
RUNOFF CONTRL
$.NULL.
                  .INSERT DUMMY
                                    .INSERT NAMES
                                                      .RETURN 1
                                    .RETURN

                                    FORM
                                    ----
                  .INSERT FORM
                                    text
                                    .INSERT
                                                      name1
                                                      .RETURN 1
                                    text
                                    .INSERT
                                                      address1
                                                      .RETURN 1
                                    text
                                    .EJECT
                                    .RETURN
                  .INSERT FORM                          .
                      .                                 .
                      .                                 .
                      .                                 .
                  .QUIT                                 .RETURN
```

The use of DUMMY and the initial .RETURN 1 in the names and addresses file is to open the names file on the proper unit for that level of nesting and leave it open.

.RU-NDENT m

Undent (move out) the right margin by m spaces. If m=0, the right margin is reset to the original right margin specified by the side margin (.SMARGIN).

.S-KIP n

Skip n printing lines. This command causes a break.