SMOKE SIGNAL BROADCASTING

TEXT PROCESSOR

COPYRIGHT NOTICE

WARRANTEE INFORMATION

The Text Processor is provided AS IS without warrantee. Reasonable care has been taken to insure that the Text Processor Software operates as described in the Text processor manual. If you do find a situation in which it does not operate as described, please contact us. We will attempt to correct any errors brought to our attention but we make no guarantee to do so.

CONTENTS

PREFACE

The SSB Text Processing System is one of the most complex programs released by SSB to date. With this in mind, the following recommendations should be noted by the user.

Do not expect to master the system with one reading of the manual. The entire document should be read lightly the first time through, followed by a more rigorous reading. The "Reference Manual" section is very concise and contains detailed descriptions of all of the commands of the processor. This is the section which should be studied extensively.

Since the system is so complex, many results may occur which are contrary to the user's intentions. If strange output is encountered, reread the sections of the manual covering the commands being used. As more experience is gained, the system will become an invaluable tool, but as with any complex system, it takes time to learn its operation.

INTRODUCTION TO TEXT PROCESSING.

This world is producing millions of words of text each day. There are words in newspapers, magazines, books, catalogs, pamphlets, letters, documents, and manuals, and they all need to be organized before publication. It would certainly be a never ending task if all text formatting and organization were to be done manually. It simply would not get done. Thanks to computers and programs called text processors, text formatting (sometimes called word processing) becomes a fairly trivial task. The text processor allows for convenient and precise page formatting and organization. The final copy becomes extremely readable and neat, which are desirable features of any printed matter.

Just what can be done with text processors? The simplest functions perform exact page fitting. In other words, if the text page should have one inch margins with a page number centered at the bottom of each page, and perhaps a special title at the top of each, the processor will automatically provide these, given the appropriate commands. Line justification is another feature provided. Several types are available which include left-hand justification (left edge straight, right edge ragged), right-hand only justification (left ragged, right straight), left and right (both edges are straight), and center justification (both edges ragged but lines centered). An extensive text processor will provide features which will allow special operations such as footnote processing. The SSB Text Processing System supports all of the above features.

To gain some insight into the use of a text processor, several specific examples will be given using the SSB Text Processor's command set. The commands used by text processors vary from system to system but many are used in the same fashion. The SSB Text Processor uses an intermixed command and text method. To issue a particular command to the processor, it is necessary to start the command in column number one of a new line and begin the command with the control character, a period ('.'). This is the method used by most of the large scale system formatters including NROFF*, which is the system the SSB Text Processor has been modeled after.

Before any specific examples are shown, a description of the 'environment' will be given. The environment refers to the basic page and formatting features which will be in effect unless otherwise specified. The initial or default environment is very important. The SSB processor, without any command information, will perform left and right justification with a line length of 65 characters (the standard 8 1/2" page line width). Page length is initially defined to be 66 lines which is the standard for 11" paper and 6 lines/inch spacing. Other initial environment features provide for the passing of blank lines to output, and

---

*NROFF is a text formatting program written at Bell Laboratories. It runs on many large operating systems, including the UNIX Time Sharing System.

for any line starting with a space or spaces to create a new line with the spaces now treated as unpaddable space characters*. With the environment initialized as above, it is possible to take any text file not having special command information embedded in it and still receive a sensibly formatted output. This is an important feature which is often overlooked by many processor designers. The environment may, of course, be changed or modified at any time by the use of special commands to allow for more personalized and detailed formatting jobs.

Let's take a look at some specific commands of the SSB processor. One of the simplest commands is the center lines command, .CE N, where N is the number of lines to be centered within the current line width. To use this command, as with any of the commands, it is only necessary to place the command right before the lines it is to affect. For example:

```
.CE 2
The Design of Text Processors
An Introduction
```

will cause the two lines listed to be neatly centered on the page. It can be seen that this is much easier than trying to manually calculate the correct spacing.

The initial environment was previously described. All of the parameters may be easily changed by the commands which directly affect them. One of the commands is .LN N and is used to set the current line length. To set the line length to 50, all that is necessary is a command line which reads as follows.

```
.LN 50
```

The length is now 50 columns. Another parameter easily set is the page length using the command .PL N, where N is the number of lines per page desired. Some other commands which change environment parameters include .FI and .NF which turn fill mode on and off (no fill) respectively. Fill means that as many words which will fit within the current line length are placed on each output line. This gives a straight left text edge and a slightly ragged right one. No fill simply copies the input lines directly to the output. It should be noted that 'fill' must be on for any justification to occur. The justification feature may be turned off using .NJ for 'no justification' or the type of justification may be set using .JU X. The X is the selection character and may be null which turns justification on in the mode it was previously defined, it may be R for right hand, C for centered, or N for normal (left + right). Left justification is obtained by turning 'fill' on and justification off.

---

*Unpaddable spaces are characters which appear as spaces on the output but are not recognized as such by the processor. This means these spaces will not be 'spread out' by the justification routines.

Another environmental parameter is the capitalization mode. This feature of the SSB Text Processing System allows an upper case only terminal to be used for preparation of text which will later be output on a hardcopy device having lower case capabilities. The commands .CP and .NC turn this feature on and off respectively. If this mode is on, all letters are automatically converted to lower case unless preceeded by a '@'. The '@' should be thought of as a typewriter shift key in its function. Another feature also enabled in this mode is similar to the 'shift and lock' on a typewriter. By typing a '^' all characters following will be upper case until another '^' is encountered.

It is often desirable, for readability, to use multiple spacing between lines. The SSB processor will allow this using the command .MS N where N is the space count desired and defaults to double spacing (N=2) if no value for N is given. The single space mode can be restored by either using .MS 1 or .SS for 'single space'.

Another group of commands deal with left margins and indentation. The left margin is normally set to Ø since the output device usually provides its own left margin (determined by paper positioning). Some applications require a wider margin at which time .LM N may be used to redefine it to be N spaces wide. Indent is similar to the left margin control with one difference. .LM N preserves the line length and simply moves the line to the right N spaces. .IN N, on the other hand, effectively reduces the line length by N columns in order to preserve the right hand margin. Setting the indent back to Ø will restore the full line length. Another form of indenting can be done by the use of the single indent command .SI N. Single indent is identical to indent except it is automatically restored to Ø after the line is output. It should be noted that the commands for left margin, indent, and single indent are additive in that if the following string of commands is issued:

```
.LM 1Ø
.IN 8
.SI 5
```

the resultant output line would be preceded by 23 spaces. Succeeding lines are preceded by 18 spaces assuming another .SI command was not used.

A note of caution is necessary concerning a characteristic of several of the processor's commands. Most commands will perform only their specified function but some also cause a line 'break'. A break is the forcing of output of the line currently being collected in the line buffer. Normally a line is kept in the buffer until the specified line length has been reached, at which time justification may or may not occur, depending on the mode enabled (also assuming that 'fill' is turned on). The break will cause the partial line to be output without being filled,

but the appropriate justification will be performed. This is useful for starting new paragraphs or new blocks of text. Some of the commands which cause a break are .CE, .FI, .NF, .IN, and .SI. Sometimes it is desirable that these commands do not cause a break. This can be done by using the 'no break' control character, ':'. So far, all commands have been shown preceeded by the normal control character, a period. To set an indent of 10 and not cause a break, the following should be used:

```
:IN 10
```

The colon may be used with any command, whether the command normally causes a break or not.

It is often necessary to produce a section of one or more blank lines. The space command, .SP N, can be used to output N blank lines. The space command also causes a break. If N is not specified, the processor will output 1 blank line. It may be required that the blank lines all be on the same page, maybe for later insertion of a photograph or illustration. The SSB Text Processor allows this by using the 'save space' command, .SV N, where N is the number of lines required. If there are not N lines remaining on the current page, no line is output but instead, printing continues and the count (N) is saved for later use. When the next page is reached, the 'output saved space' command may be used, .OS, which will then produce the remembered number of blank lines. A convenient method for using .OS will be given later. Another similar command is the 'need lines' command, .NL N, where N is a line count. This command says that there must be N lines remaining on the current page, and if there are not, eject to the next page. This is convenient for keeping special blocks of text together (keep them from being split by page boundaries), or for not starting a new paragraph at the bottom of a page if only 1 or 2 lines will fit.

The commands which have been described so far will allow very nice page formatting. If these were all that were available in a text processor, much time and effort would be saved. The SSB Text Processing System, however, offers many more commands and much more versatility. One feature often needed in documents or manuals is the page title. There are many different ways of providing titles but the SSB processor uses a title command which has the form:

```
.TL 'field1'field2'field3'
```

where field1 is left-adjusted, field2 is centered, and field3 is right-adjusted. Any one or all of the fields may be present. Another feature supported in the SSB processor is the ability to print the current page number in the text. Any place a percent sign (%) appears, it will be replaced by the page number. A few examples will clarify the use of the title command.

```
.TL 'Main Title'''
.TL ''Centered Title'Date'
.TL ''-%-''
```

The first line will left adjust "Main Title" on the page. The second example causes "Centered Title" to be centered and "Date" to right adjusted. The final example will cause the current page number to be printed between two dashes.

Now it would be nice if there was some way of getting the title (and maybe a few other commands) to automatically execute at the top and/or bottom of each page of output. The SSB processor offers two advanced features to perform this task: macros and traps. A macro is a set of commands grouped together and given a name. When this name is later referenced, the entire group of commands will be executed. Essentially, what results is the ability to write special programs using the command set of the processor to do specific tasks such as headers, paragraphs, special titles, etc. The trap allows the user to specify a certain line on the output page where a specific macro is to be executed. To solve the title problem stated above it is convenient to define two macros, a header macro and a footer macro. The purpose of the header is to perform a sequence of commands to make the top of each new page appear the same. The footer macro works at the bottom of each page. Suppose it was required that the top of each page have three blank lines followed by a centered title and the bottom of each is to contain a centered page number between dashes. The following macros and trap placement would satisfy this requirement.

```
.DM HD
:SP 3
.TL ''Page Title''
:SP 3
..

.DM FT
:SP 3
.TL ''-%-''
:PG
..

.AT 1 HD
.AT -7 FT
```

The .DM command is used to define a macro and the first one listed in the example defines the header macro called HD. The header macro will space down 3 lines (without causing a break since the no break control character (':') was used), print a centered title, and finally print 3 more blank lines without causing a break. The last line of the header macro definition is '..' and is the command for terminating a macro definition. The second macro defined is FT and is used for the footer. Upon execution it will space down 3 lines (without a break), print a

centered page number, and eject to the next page. The .AT commands were used to set the trap locations. .AT 1 HD causes the header macro to be executed at line 1 of every new page while .AT -7 FT causes the footer macro FT to be executed at the 7th line from the bottom of each page. The ability to specify trap locations and define macros makes titles and footers extremely simple and efficient.

One of the important aspects of using a text processor is the ability to make a few minor command changes and greatly change the final copy. As an example, suppose at the last minute it was decided that it would look better if there were four blank lines at the top of each page rather than three. If the document were being prepared by hand it would be necessary to retype the entire work to obtain the extra space. Using a small text processor it would only be necessary to go back and change the line count before each title. The SSB Text Processor and its ability to define macros means only one line in the entire text file needs to be changed. The second line of the header macro which is currently ':SP 3' would be changed to read ':SP 4'! One simple change and the desired result is obtained! It should be kept in mind that when preparing documents with a processor supporting macro capability, all of the often-used command strings should be defined in a macro so simple global changes may be easily performed if so desired.

There are more advanced features supported in the SSB Text Processing System. One of these is the ability to do conditional command execution. There are four forms of this command:

```
.IF O .XX
.IF E .XX
.IF N .XX
.IF !N .XX
```

where O and E stand for Odd and Even page number respectively, and N can be a number, a number register (to be explained shortly), or an expression containing numbers and number registers. The exclamation point is the 'NOT' operator and .XX is any command or macro name. The command works as follows; IF the condition is true (page is odd or even, or the number or expression is greater than zero) the command XX is executed, otherwise it is not. Preceding the expression by '!' will cause the command or macro to be executed only if the condition is not true (less than or equal to zero). The following special header macro definitions will illustrate the use of this command.

```
.DM HD
:SP 3
.IF O .TL '''Title'
.IF E .TL 'Title'''
:SP 2
..
```

```
.DM HD
:SP 3
.IF %-1 .TL ''Title''
:SP 2
..
```

The first header defined causes the title to be right-adjusted on odd numbered pages and left-adjusted on even pages. The second definition will print a centered title on each page except page number one since the value of the expression will be zero when the page number is one (remember that the '%' represents the current page number).

Another feature contained in the SSB processor is the ability to use number registers. Two types exist, one which allows the user to read and access certain system parameters including the date, page number, current indent, left margin, current column position, current line on the page, and line length. The second type are user definable and can be used exactly as variables would be used in a program. Number registers are the single letters A-Z and the percent sign (%) already introduced. Several other number register features are supported by the SSB processor, including auto increment, assigning values to the registers, use in expressions (as seen in the .IF command), and the ability to print any register on the output in either Arabic, capital Roman, or small Roman numerals.

Some processors, including SSB's, allow communication between the processor and the operator during actual text processing. Three of these commands take on the following form:

```
.ST
.TM any string
.GI any string
```

The first command will stop the processing and print 'STOP' on the user's terminal. This may be desirable if special paper positioning is required or other special action is needed. When the processor has been stopped it may be restarted by typing any character on the terminal except an 'S' which will halt processing. The second command listed will send 'any string' to the terminal as a special message. It may be used before the 'STOP' command to issue special instructions to the operator. The last command will 'Get Input' from the terminal and insert it into the output stream. 'Any string' can be used for a prompt. An example where this command is quite useful is in the preparation of form letters. The processor may prompt the operator for names and addresses which are then typed in at the terminal and automatically inserted into the text!

One final command will be described in this introduction, the 'divert text' command. Sometimes it is desirable to save text currently encountered for later use. An example of this is when trying to do footnotes. It would be nice if immediately

after the footnote reference was made, the actual footnote text could be typed, but saved for later insertion at the bottom of the page. The mechanism which allows this sort of operation is called a 'diversion' and is only available on the more complex text processors, such as SSB's. Two forms of the diversion usually exist:

```
.DI XX
.DA XX
```

where .DI instructs the processor to divert the following text into a diversion space named XX and .DA says to divert and append to the diversion space named XX. During diversion, all normal text processing still takes place, but rather than outputing the text to the printer, the text is written to a special place internal to the processor. The diversion process continues until the command for a divert is found without a name specifier. To recall the diverted text, it is only necessary to call it by name, exactly as macro calls are performed.

As an advanced exercise and demonstration of the diversion process (as well as many other processor commands) a complete set of macros for handling footnotes will be described. The reader should note that the following example is very complex and several readings will probably be required in order to fully understand its operation.

```
.NR B 7
.DM HD
:SP 2
.IF %-1 .TL 'FOOTNOTE TEST'''
:SP 2
.AU 1
.NR X Ø
.NR W Ø-#B
.IF #V .TR
.NS
..
.DM FO
.NR V Ø
.IF #X .FT
.CH FO -#B
:PG
..
.DM NM
.TL ''-%-''
..
.DM BF
.DA TX
.EV 1
.IF !#+X-1 .SA
..
```

- continued on next page -

```
.DM EF
.BR
.EV Ø
.DI
.NR W -#V
.CH FO #W
.IF #N-#P-#W .CH FO #N+1
..
.DM SA
--------------------
.BR
..
.DM TR
.BF
.NF
.FE
.FI
.EF
..
.DM FN
.DI FE
..
.DM FT
.EV 1
.NF
.TX
.RM TX
.DI
.FI
.EV Ø
..
.AT 1 HD
.AT -#B FO
.AT -4 NM
.CH FO 7Ø
.AT -#B FN
.CH FO -#B
.EV 1
.AU 1
.LN 55
.EV Ø
```

This example is quite similar to the one given in the "NROFF Users' Manual" written by J. Ossanna, of Bell Laboratories. To use these macros, merely insert their definitions at the beginning of the text file, and immediately after a footnote reference has been made, call macro BF. Following the call, simply type the footnote text and end it with a call to EF.

A description of the macros follows. The first line defines number register B and sets it equal to 7. Number register B is used to specify the size (in lines) of the bottom margin. A header macro definition follows (HD) and provides several functions. After spacing down two lines, the title is output

unless it is page number one (the IF command). Two more lines are produced and the auto increment value is set to one. Number register X is cleared and it is later used to keep track of the number of footnotes on the current page. Next, W is set to the location of the bottom margin trap and will later be adjusted as necessary if footnotes are added. The IF #V command checks to see if there is any remaining footnote text from the previous page and if so they are reprocessed (number register V contains the line count of the last diversion). Finally, the 'no space' mode is turned on to suppress any spaces which might otherwise get printed needlessly at the top of the page.

The footer macro, FO, clears the diversion count, V, and checks the value of X. If X is not zero (meaning there were footnotes on the page), macro FT is invoked. The footer is then restored to its original location by using the Change command as defined by B. The last command does a page eject. Macro NM is used to print a centered page number at the bottom of each page.

The begin footnote macro, BF, starts with a divert append into the diversion space called TX. The environment* is switched, and if it is the first footnote on the page, macro SA is invoked which outputs a set of dashes as a simple footnote separator line. Diversion of the footnote text continues until macro EF is called. At this time a 'break' is executed and the original environment is restored. The diversion is stopped with the DI command. Number register W is updated by the number of diverted lines and the footer trap line is changed to compensate for the added footnote lines. Finally, if the number of diverted lines was great enough to move the footer trap up past the current line position, the trap is reset to the next line. TR is responsible for rediverting any lines of footnote text which will not fit on the page. It is very unusual for this to happen but this may occur if a footnote is very long and is referenced near the bottom of the page.

Macro FT is used for reading back the diverted text. It switches environments, sets the no fill mode, and calls TX, the actual footnote text. TX is then removed from the macro list, the fill mode is restored, and the environment switched. The last group of lines is used to define the trap locations of the various macros. The header is set to line one, and NM is set to execute four lines from the bottom of the page. The trap for the footer is planted at -#B, then moved past the bottom of the page while FN is also placed at -#B. FO is then moved back as originally placed so in effect both FO and FN are placed at the same line, but trap FN can only occur if the footer trap is moved up by the occurrence of a footnote. The last lines switch to environment one and initialize it for a line length of 55 and auto increment of one.

---

*Environment switching is a feature supported by many of the larger text processors (including SSB's) which allows all of the major environment parameters to change simultaneously.

As a final example of how a text processor can be used, a sample section of text will be given. The text is shown first with the commands and then as the text processor would output the final copy.

```
.SP 2
.CE 2
^TEST OF SEVERAL^
^PROCESSOR COMMANDS^
.SP
.SI 5
@THIS EXAMPLE SHOWS HOW COMMANDS AND TEXT CAN BE INTERMIXED
FOR LATER PROCESSING BY A TEXT PROCESSOR.
@THE EXAMPLE STARTED BY CENTERING TWO LINES FOLLOWED
BY A SINGLE INDENT TO SIGNIFY THE START OF A PARAGRAPH.
@THE CAPITALIZATION MODE IS ON AND THE UPPER CASE SHIFT
CHARACTERS ARE BEING USED.
.SP
.LM 10
.LN 45
.JU C
@THE ADJUST MODE WAS JUST CHANGED TO CENTERING
AS WELL AS A LINE LENGTH OF 45.
@THE LEFT MARGIN WAS SET TO 10 TO GIVE A NICELY
CENTERED NARROW LINE.
@SPECIAL EFFECTS LIKE THESE ARE EASILY ACCOMPLISHED.
.SP
.LM 0
.LN 65
.JU N
@THE PARAMETERS WERE JUST SWITCHED BACK SO THE
LINE APPEARANCE WILL BE RESTORED.
THIS IS A SHORT EXAMPLE BUT SHOULD SHOW HOW THE
COMMANDS CAN BE INTEGRATED WITH THE TEXT.
```

This example appears in its expanded form on the next page.

This introduction to text processing is intended to be only that and is not a complete treatment of the subject. Many commands and features have been omitted. The ones included are the most general and the most used commands which offer the user a great deal of control and flexibility. Hopefully some eyes have been opened to the wide variety of applications of the text processor.

EXPANDED EXAMPLE

TEST OF SEVERAL
PROCESSOR COMMANDS

This example shows how commands and text can be intermixed for later processing by a text processor. The example started by centering two lines followed by a single indent to signify the start of a paragraph. The capitalization mode is on and the upper case shift characters are being used.

The adjust mode was just changed to centering as well as a line length of 45. The left margin was set to 10 to give a nicely centered narrow line. Special effects like these are easily accomplished.

The parameters were just switched back so the line appearance will be restored. This is a short example but should show how the commands can be integrated with the text.

*NOTE: This entire user's manual was prepared using the SSB Text Editing System and the SSB Text Processing System.

COMMAND SUMMARY

Command Form	Initial Value	Default Argument	Cause Break*	Explanation
I. PAGE CONTROL				
.PL +N	66 lines	66 lines	no	Page length.
.PG +N	N=1	-	yes	Eject to next page.
.PN +N	N=1	ignored	no	Page number.
.LM +N	N=Ø	previous	no	Left margin.
.NL N	-	N=1	no	Need N lines.
II. TEXT FILLING, ADJUSTING, AND CENTERING				
.BR	-	-	yes	Break buffer.
.FI	fill	-	yes	Fill output lines.
.NF	fill	-	yes	No fill or justification.
.JU C	jst,norm	just.	no	Justify on.
.NJ	just.	-	no	No justification.
.CE +N	off	N=1	yes	Center N input lines.
III. VERTICAL SPACING				
.MS N	prev	N=2	no	Multiple spacing.
.SS	single	-	no	Single space lines.
.SP N	-	N=1	yes	Space N lines.
.SV N	-	N=1	no	Save N lines.
.OS	-	-	no	Output saved lines.
.NS	space	-	no	No-space mode on.
.RS	-	-	no	Restore spacing.
IV. LINE LENGTH AND INDENTING				
.LN +N	65	prev	no	Line length.
.IN +N	N=Ø	prev	yes	Indent.
.SI +N	-	N=1	yes	Single indent.
.PI ST	-	-	yes	Put string in indent.
V. MACROS, DIVERSIONS, AND LINE TRAPS				
.DM XX	-	ignored	no	Define or redefine a macro.
.AM XX	-	ignored	no	Append to a macro.
.RM XX	-	ignored	no	Remove macro or diversion.
.DI XX	-	end	no	Divert out to macro "XX".
.DA XX	-	end	no	Divert and append to "XX".
.AT -N XX	-	-	no	Set trap at line N.
.CH -N -M	-	-	no	Change trap location.
.CH XX -M	-	-	no	" " "
..	-	-	no	End macro specification.

---

*The use of ':' as the control character (instead of '.') suppresses the break function.

Command Form	Initial Value	Default Argument	Cause Break	Explanation
VI. NUMBER REGISTERS				
.NR X +N	-	-	no	Number register.
.AU +N	Ø	prev	no	Set auto increment.
.AR	arabic	-	no	Arabic numbers.
.CR	arabic	-	no	Capital Roman numbers.
.SR	arabic	-	no	Small Roman numbers.
VII. TABS AND TAB CHARACTERS				
.TA N,..	none	none	no	Set tab columns.
.TF C	un.sp.*	un.sp.*	no	Set tab fill character.
.TC C	none	none	no	Set tab character.
VIII. THREE PART TITLES				
.TL 'left'center'right'			no	Define title.
.LT +N	65	prev	no	Length of title.
IX. CONDITIONAL INPUT COMMANDS				
.IF C COMMAND		-	no	If true, do command.
.IF !C COMMAND		-	no	"
.IF N COMMAND		-	no	"
.IF !N COMMAND		-	no	"
X. ENVIRONMENT SWITCHING				
.EV N	N=Ø	N=Ø	no	Change environments.
XI. SPECIAL CONTROL COMMANDS				
.CP	no caps	-	no	Capitals mode on.
.NC	no caps	-	no	No caps mode.
.ST	-	-	yes	Stop processing.
.EX	-	-	yes	Exit processor.
.PS	no pass	-	no	Pass text without proc.
.RP	-	-	yes	Repeat entire file.
.DH	-	-	yes	Double height line**. NORMAL PRINT
.DW	-	-	yes	Double width line**. ENHANCE PRINT
.DB	-	-	yes	Double height and width**.

---

*Un.sp. = unpaddable space character.
**These commands require the output device
to support double dimensioned character printing.

Command Form	Initial Value	Default Argument	Cause Break	Explanation
XII. EXTERNAL COMMUNICATION				
.TM ST	-	-	no	Send string to terminal.
.GI ST	-	-	no	Get line from terminal.
XIII. MISCELLANEOUS				
.*	-	-	no	Comment field.
XIV. UNDERLINE				
.UL	-	-	no	Underline next input line.
XV. DISK ORIENTED COMMANDS				
.IC C	">"	">"	no	Set item character.
.OF NAME	-	-	no	Open data file.
.CF	-	-	no	Close data file.
.RI S	-	-	no	Read item from file.
.NI N	-	N=1	no	Move to next item.
.NB N	-	N=1	no	Move to next block.

SPECIAL CHARACTER DEFINITIONS

Character	Meaning
\	Standard escape character.
@	Force capital letter.
^	Set capital letter mode.
#	Number register specifier.
.	Basic control character.
:	No break control character.

NUMBER REGISTERS

Register	Meaning
A-B	User definable
C	Current column count
D	Day of the month
E	End of data file flag
F	User definable
G	.GI & .RI character count
H	User def.
I	Current indent
J-K	User def.
L	Current line length
M	Month
N	Line count on page
O	Current left margin
P	Current page length
Q-U	User def.
V	Last diversion line count
W-X	User def.
Y	Year (2 digits)
Z	User def.
%	Page number.

REFERENCE MANUAL

INTRODUCTION

All input lines to the processor which are to be interpreted as commands should be started with the control character (a '.' or ':') in column one followed immediately by the two letter command. If the characters are not system command names or user defined macros, the line will be ignored. The 'nobreak' control character (':') may be used with any command to suppress normal line breakage during processing. Only a single command reference is permitted on any one line.

The following detailed command descriptions reference numerical arguments either as N, +N, or -N. N means any argument is taken as absolute and any previous value is simply replaced by the new value. +N is used when the argument may take any form of a number (either positive, negative, or absolute). Valid arguments of this form are +4, -10, and 3 where the old value would be incremented by 4, decremented by 10, and replaced by 3 respectively. Arguments of the form -N may use absolute values or negative values which are subtracted from the current page length (to reference N number of lines from the bottom of the page). When expressions are involved using the +N argument, the entire N is evaluated before the increment or decrement is applied (e.g. -6-3 will decrement the value by 3). Certain commands requiring arguments will keep the last argument assigned if the argument field is left empty when the command is called.

I. PAGE CONTROL

The page control commands are used to set the physical page parameters such as length, width, margins, numbering, etc. Top and bottom margins are not automatically provided and should be defined by the user with macros as described in a later section.

.PL +N — Set page length to N lines. Initial value is 66 lines and is reset to 66 if no argument is given. Does not cause a break. The maximum N is 255.

.PG +N — Eject to next page. If N is given the new page number will be adjusted accordingly. The page number is automatically incremanted if no argument is given and the command does cause a break. Max N is 255.

.PN +N — Set the page number to +N. If .PN occurs before the first break or first text, it will be set for the first page. The value is initially 1 and the command does not cause a break. The maximum page number is 255.

.LM +N — Set the left margin according to +N. The entire output line will be offset to the right by the number of spaces the current LM is defined. Initially there is

no margin (N=Ø) and no break occurs. Left margins should not exceed 1ØØ.

.NL N Need N lines on the page. If the distance to the next trap position or the bottom of the page is less than N, the paper is advanced to the next trap position (blank lines output). Otherwise no action takes place. No break occurs and the default argument is N=1.

II. TEXT FILLING, ADJUSTING, AND CENTERING

The following commands affect the appearance of individual lines of text. Two important parameters are referenced, Fill and Justify. The default fill mode is to fill output lines with as many words as possible without exceeding the set line length value. Any extra words are saved for output on the next line. A word is defined to be any string of characters separated by a space or spaces. If two words are to be separated by a space but are not to be split across line boundaries or separated by the justification routines, the unpaddable space character, "\ " (slash space) may be used. The default justification mode is left and right, giving straight margins on both sides. Filled lines which contain too few character positions to completely fill out the specified line length are padded with spaces until the correct length is achieved. The space filling or padding is done from alternate sides of the page as each line is justified to eliminate 'white rivers' which may otherwise occur in the text. No hyphenation is performed. It is important to note that fill must be on in order for the justification to be performed, but fill may be on by itself. If fill mode is off, characters are passed exactly as they appear on the input file.

.BR Break the line currently being filled in the buffer. The line is output after specified justification is done but no further filling or padding is attempted. Input lines beginning with spaces and empty text lines (blank lines) also cause a break.

.FI Fill mode is turned on and subsequent output lines are filled. This command causes a break.

.NF Turn off fill mode (nofill). Following input lines are neither filled or justified, but are copied to the output exactly as they appear on input, without regard to the current line length. Causes a line break.

.JU C Justification is enabled. If fill mode is off, adjusting will be deferred until it is back on. If the justify type character, "C", is present the justification type is set as follows: N sets for normal (default, left and right), R sets right only justify, and C will center lines (both margins ragged). If the type character is absent, justification is turned back on with the type previously used. No break is caused.

.NJ Turn justification off. If fill is on, the resultant output line will have a straight left and a ragged right edge. No break is caused and the justify type remains unchanged.

.CE +N Center the next N input lines. A break occurs before the command and then automatically after each line is output. If the resultant line is longer than the current line length, the output line will be left hand adjusted. The maximum count is 255.

III. VERTICAL SPACING

All line spacing defaults to standard single spacing. It may be set at any time by using the MS command. If the line spacing is N, N-1 blank lines are inserted after each output line. The occurrence of a trap will terminate any remaining spacing count. Contiguous space should be saved by using the SV and OS commands.

.MS N Set multiple line spacing to N. N-1 blank lines are inserted after each output line. No break is caused and if N is not specified the value of 2 will be used (double spacing). Max value is 255.

.SS Set single space mode. No blank lines are output after text lines and no break occurs.

.SP N Space N lines. The number of output lines is limited to the distance to the nearest trap or bottom of the page. If nospace mode is on, no spaces are output. If no value for N is given, it defaults to 1. SP causes a break.

.SV N Save N lines of space. If the distance to the next trap (or the bottom of the page) is greater than N, N lines are output, otherwise no lines are immediately output but the count (N) is saved for later output (see OS). Subsequent SV commands will overwrite any previously remembered N. Nospace mode has no effect. The command does not cause a break and the default value for N is 1.

.OS Output saved space. This command is used to output any previously saved space from the SV request. The remembered count is cleared after calling OS and nospace mode has no effect. A break does not occur.

.NS No-space mode is turned on. The no-space mode inhibits SP requests and PG requests without a next page number. This mode is automatically turned off after the output of a line of text. No break is caused.

.RS Restore space mode. If the nospace mode is on, it is turned off with this command without causing a break.

IV. LINE LENGTH AND INDENTING

Using the following set of commands, the user has complete control over the line length and various forms of indenting. The line length includes all indent spaces but does not include left margin spacing. As long as the fill mode is turned on, the resultant output line will be less than or equal to the current line length minus the indent. Line lengths of less than 6 columns are not permitted.

.LN +N — Set line length. The initial value is 65 columns and the command does not cause a line break. Line lengths must be between 6 and 255 columns inclusive.

.IN +N — Set the line indent according to N. With a line length of L and an indent of N, N spaces are output before each line and the remaining text is restricted to a size of L-N. Initially the indent is Ø and the command causes a break.

.SI +N — Single indent N spaces. Only the next output line will be indented by the amount specified by N. Note that single indenting may be done backwards into an indent field. (e.g. if indent is 1Ø, SI -4 would temporarily set the overall indent to 1Ø-4 or 6). IN and SI counts are cumulative and the final value may not be negative! This command causes a line break.

.PI ST — Put string in indent field. The string represented by "ST" (leading spaces ignored), is inserted into the field normally filled with spaces by the indent count. If the string is longer than the indent count, the string will be truncated so it will not extend past the indent field.

V. MACROS, DIVERSIONS, AND LINE TRAPS

A macro is a set of commands and/or text which can be assigned a name and called by name at a later time. All macro names are two characters long and must be different from any names already in existence in the system command name table. Macros are defined or redefined by using the LM command, or by using the output diversion command, DI. Macros already in existence may be appended to by using the AM or DA commands. If a macro is named XX, it may be invoked by an input line beginning with ".XX". A trap may also be placed at a specific vertical page placement to cause automatic macro execution at that point by using the AT command. During macro definition, number registers are not expanded into numeric values but are at the time the macro is executed. No other special character translation is done during macro definitions (e.g. tab expansion, etc.). Keep in mind that macros may be any combination of commands, macro calls, and text, but a macro may not define another macro (it may create a diversion).

A diversion is treated as a macro upon execution but is created in a different manner. Processed output may be diverted into a macro space for such purposes as footnote processing or vertical page size determination for conditional changing of page parameters (number register V contains the last diversion line count). All normal processing takes place during a diversion except left margins. It is standard practice to read back the diverted text in 'nofill' mode to suppress further line processing.

If at any time during macro definitions or diversion creation the macro space is overflowed, a system error will be generated and processing will be halted. None of the macro commands cause breaks in the line filling.

.DM XX Define or redefine a macro with the character name XX. The actual macro begins with the next input line. The macro definition is copied until the termination character ".." is found starting in column 1. Macros may not contain DM requests but may create diversions.

.AM XX Append to the macro named XX. This command acts exactly like DM except the following input lines are appended to an existing macro rather than creating a new named space.

.RM XX Remove macro or diversion. The macro named XX is removed from the name list and subsequent calls to this name will have no effect.

.DI XX Divert output into the macro space named XX. The macro named XX is defined or redefined at this point. All normal text processing occurs during diversions except left margin page offsetting is not done. The diversion process is ended when another DI or DA is encountered. Diversions can not be nested! The count of the number of lines last diverted is kept in number register V for possible later reference.

.DA XX Divert append version of DI. The same rules apply for both commands.

.AT -N XX At line N invoke macro XX. Any macro previously planted at line -N is replaced by XX. N is measured from the top of the page (Ø or 1 may be used to represent the top) and -N is measured from the bottom of the page (e.g. if the page length is 66, line -1 represents line 66). If no macro name is given with the command, the trap located at line -N, if any, is removed.

.CH -N -M Change trap. See next.

.CH XX -M Change the trap planted at line -N to occur instead at line -M. Alternately, change the location of the trap for macro XX to line -M. If there is not a trap set at -N, the request is ignored.

.. Terminate a macro definition.

VI. NUMBER REGISTERS

Number registers are a type of variable used during processing. There are two classifications, user definable and system. Number registers have single character names (A through Z and '%'). Number registers may be used any time a number is expected in a command and also may appear imbedded in text. There are two methods of referencing a number register:

```
#X
#+X
```

where '#' is the register designator character and X is the name of the register. When using '%' it should not be preceeded by the '#'. The '+' in the second example specifies that the number register is to be auto incremented prior to its use and it will retain the new incremented value. The auto increment amount is set using the AU command. When a number register reference is encountered it is converted to decimal, lower case Roman, or upper case Roman, as determined by the mode set. Number registers appearing in macro definitions are not converted until the macro is actually executed. Number registers may also be used to construct expressions any time a number is expected in a command (expressions may not be imbedded in text). The expressions are evaluated left to right and may contain only the operators '+' and '-'.

.NR X +N Assign a value to number register X. This command should only be used to assign values to user definable number registers.

.AU +N Set the auto increment amount to +N. Any time a register is referenced as "#+X", the AU value will be added to it prior to its actual use.

.AR Arabic numbers. See below.

.CR Capital (upper case) Roman numbers. See below.

.SR Small (lower case) Roman numbers. Number registers will subsequently be converted into Arabic, capital Roman, or small Roman respectively. This mode is initially Arabic and also applies to the outputting of page numbers using the '%'.

The following is a list of the system and user definable number register names.

Register	Meaning
A-B	User definable
C	Current column count
D	Day of the month
E-F	User def.
G	Get input (.GI) character count
H	User def.
I	Current indent
J-K	User def.
L	Current line length
M	Month
N	Line count on page
O	Current left margin
P	Current page length
Q-U	User def.
V	Last diversion line count
W-X	User def.
Y	Year (2 digits)
Z	User def.
%	Page number

VII. TABS AND TAB CHARACTERS

The currently defined horizontal tab character is replaced by the required number of fill characters corresponding to the distance to the next defined tab stop column (on the line currently being filled). The fill character is normally the unpaddable space character but may be defined by using the TF command. Up to 20 tab stops may be defined and should be set in ascending order. Initially no tab stops are defined and the tab character is null. Any non alphanumeric character may be defined as the tab character. It should be noted that using tabs with the fill mode turned on can result in nonsensical output tab fields since the user may not know what the current output column is.

.TA N,.. Tab stop settings. The default tab stops are all null (none) and a total of 20 may be defined. The stop values may be separated by spaces, commas, or any other nonnumerics, e.g. TA 10,20,25,40.

.TF C Set the tab fill character. This is normally the unpaddable space character but may be defined to any nonnumeric printable character. If 'C' is not specified the fill defaults to the unpaddable space character.

.TC C Define the tab character. Initially the tab character is null (none) but may be defined to any nonnumeric printable character. If 'C' is not specified the tab character again becomes null.

VIII. THREE PART TITLES

Very convenient titling may be performed by using the TL command. Three fields may be used for left, centered, and right justification of titles. All 3 fields may be used or any combination of fields. The justification is done with respect to the title length which is independent of the defined line length. This length is initially 65 columns. The use of TL has no effect on current line accumulation (does not cause a break). .TL is usually used in header and footer macros. For example, .TL ''-%-'' will print the page number in the center of the title length.

.TL 'LEFT'CENTER'RIGHT'
: Place titles adjusted according to field. The strings represented by "LEFT", "CENTER", and "RIGHT" are respectively left adjusted, centered, and right adjusted within the current title length. Any of the fields may be empty and any nonnumeric printing character may be used in place of the field delimiter "'". The "%" character will be replaced by the current page number in Arabic or Roman representation.

.LT +N
: Set title length. The lengths of titles and lines are separate parameters. Indents do not apply to titles but left margin adjustment does.

IX. CONDITIONAL INPUT COMMANDS

Input command and macro calls may be performed on a conditional bases. Chained conditionals are permitted as in: IF #A IF #B .XX.

.IF C COMMAND See next

.IF !C COMMAND "

.IF N COMMAND "

.IF !N COMMAND
: IF is the conditional command. "COMMAND" can be any system command or macro name. "C" is a built in condition code and can be either O or E to represent Odd or Even page numbers respectively. "N" is any number and can be a number, a number register, or any combination of these in the form of an expression using addition and subtraction. If the condition is true (the built in condition is satisfied or the number is greater than zero), the command or macro named is executed, otherwise the command is ignored. If "C" or "N" are preceeded by a '!' (not), the command is executed if the condition is false or the number is less than or equal to zero.

X. ENVIRONMENT SWITCHING

There are a number of parameters which control the text processing and are grouped together and called the environment. These environment parameters may be changed all at once using the switch command. There are two environments, Ø and 1. They both have identical initial values for all parameters. Parameters within these environments are those associated with:

line length	vertical line spacing
indenting	centering count
adjusting	auto increment
filling	partially collected words
title length	partially collected lines

All other parameters are global, or in other words, they are not switched with the environment but remain unchanged. Examples of global values include left margin, page number, current line number, number registers, trap tables, and macro definitions. Since partially collected words and lines are kept with the environment, switching environments will not cause a break and will also preserve any left over words.

.EV N Change to environment N where N can be Ø or 1. If N is left null, environment Ø is assumed.

XI. SPECIAL CONTROL COMMANDS

The following commands control certain aspects of the processor. The double height and width commands are hardware dependent. You should refer to the "adaption" section of this manual for details.

.CP Turn capital letter mode on. When enabled, this mode will allow the use of an upper case only terminal to prepare text for later output to a device which supports both upper and lower case. Each character is automatically converted to lower case unless it is immediately preceeded by a '@' at which time that character remains upper case. Strings of characters may be kept in upper case by enclosing them between up arrows "^". The "@" is like a shift key and the "^" acts like a shift and lock key.

.NC Turn off capitals mode. Initially this mode is off and the special capitalization characters ("@" and "^") are ignored.

.ST Stop causes processing to temporarily halt and the word "STOP" is output to the terminal. At this time, typing an "S" will cause all processing to be stopped and the processor will be exited. Typing any other character will cause processing to continue. The stop command does cause a line break.

.EX Exit the processor. Text processing is stopped just as if all input had been finished. This command is useful in conjunction with the IF command.

.PS Pass all input to the output. This command is primarily intended as a debugging aid since it allows all input (including command lines) to be passed to the output. No command interpretation or processing is done and once in this mode, the remaining text will be passed until the end of the input file is reached.

.RP Repeat processing on file. This command will cause the file to be 'rewound' and all processing to be repeated. This is useful for some form letter type applications.

.DH Print the next line in double height characters. This feature requires special hardware on the output device. Consult "Adaptions" for details. SET FOR PRINTER CHANGE

.DW Print the next line in double width characters. Requires special hardware. SET FOR PRINTER CHANGE

.DB Print next line in both double height and double width characters. Requires special hardware.

XII. EXTERNAL COMMUNICATION

Two commands exist which allow for communication between the processor and the user during actual text processing. The TM command is useful for sending special instructions to the terminal such as paper adjustment or character font change information. The GI command can be used in form letter preparation or insertion of special text strings while processing is taking place.

.TM ST Send a message to the terminal. ST may be any string of characters or words. The leading blanks are ignored. The message is simply output to the terminal and may be used before the Stop command to issue special instructions.

.GI ST Get input from the terminal. If ST is present (any string), it is output to the terminal as a prompt message. Characters typed from the terminal following the execution of GI are automatically inserted into the input stream for text processing. This command can be used to get name and address information for form letter preparation. The 'get input' function is terminated by typing a carriage return, therefore, only one line of text may be entered with each GI command executed. After completion of the command, the number register G contains the character count of the string typed (not including the carriage return).

XIII. MISCELLANEOUS

The following describe some of the smaller features of the text processor.

.* Comment field. This may be used to insert comments into the input text and will be ignored by the processor. No output is created with this command (the comment is not passed to the output).

Special Characters

\ Standard escape character. This character is used to remove special meaning from a character. For example, if a percent sign ("%") is needed in the output it is necessary to preceed it with the "\", otherwise it will be interpreted as the page number (e.g. \%). To print a backslash, "\\" must be used.

@ Force upper case letter if in the capitals mode (CP). This acts similar to the 'shift' key on a typewriter. Example: "@test" will be output with an upper case "T" and lower case "est".

^ Upper case string delimiter. This character acts similar to the 'shift and lock' key on a typewriter. As an example, ^this is a test^ would cause "this is a test" to be output in all upper case characters. The capitals mode must be on (CP).

Number register specifier. When an alphabetic character is immediately preceeded by a "#" it will be interpreted as a number register. Example: "#A" refers to number register "A".

. The period is the basic command control character. If in column one, it specifies a two character command or macro name follows.

: The colon is the no-break control character. It functions exactly like the period, but will suppress breaks caused by various commands.

% Page number symbol. Any place the percent sign appears, it will automatically be replaced by the current page number.

Special notes

A. Any time input is being typed into the processor, typing a 'control X' will delete that line and issue a "?" as a prompt.
B. The processor automatically makes sure there are two spaces after ".", "!", or "?". This does not apply to punctuation immediately followed by another character.

XIV. UNDERLINE

The following command permits underlining of words but may only be used with printer devices which support single character backspace capability. Unpredictable results will occur when trying to use this command on printers not supporting backspace.

.UL Underline the next input line. The following line of text (single or multiple words) will result in the output being underlined. Only alphanumeric characters are underlined.

XV. DISK ORIENTED COMMANDS

The following commands deal with the use of a "data file". The data file is a set of "blocks", with each block being divided into "items". An item can be any set of text or processor commands followed by an "end of item" character. The "end of item" character is initially a '>' but may be redefined using the .IC command (see below). The end of a block is specified by a null or empty item (two successive end of item characters form a null item; e.g. End of block>>) There are processor commands which allow inserting items into text (see .RI), skipping items (see .NI), moving to a new block (.NB), and the ability to open and close data files. For a specific example of these commands, see the Form Letter example in the MACRO LIBRARY section.

.IC C Set the end of item character. This character is initially a '>' but may be defined to any nonalphanumeric printable character. If 'C' is not specified, the character defaults back to a '>'.

.OF NAME Open a data file. This command will prepare the specified file for reading. If 'NAME' is specified on the command line (it should follow standard file spec rules) that file will be opened if found on the disk. If 'NAME' is not specified on the command line, the processor will prompt the terminal with: "DATA FILE NAME? " at which time the desired file name should be entered. If a file is already open, the .OF command will be ignored by the processor. It is only possible to have one file open at any one time. Closing a file using .CF will allow another file to then be opened.

.CF Close data file. If a data file is opened, it will be closed and not allow any more data to be read from it. If no file is open, the command has no affect.

.RI Read item from input file. If a file has been opened, the RI command will cause input from the file until an "end of item" character is read. The end of item character will be returned as a space if in the fill mode, or a carriage return if in nofill mode. If an S appears on the calling command line (.RI S), no

character will be returned for the end of item character. In other words, the character will be 'S'uppressed. If there are no items remaining in the current block, .RI will have no affect. The RI command will also be ignored if no file has been opened. After reading data with the RI command, number register C will contain a count of the number of characters just read in.

.NI N Move to next item. Normally sequential items are read by using the RI command. It is often desirable to skip items while processing text from a data file. The .NI command is used to skip one or more items in a block. If N is present on the calling line, it should be a number (or number register) which specifies the number of items to be skipped. If N is not present, the default is one item to be skipped. NI will not move past the end of a block.

.NB N Move to next block. The use of NI and RI commands cause the sequential reading of items and will never move into the next block. It is necessary to use the .NB command to advance to the next block. If N is specified (a number or number register), N-1 blocks will be skipped. (example: If .NB 2 were specified, the next block would be skipped over and the next data read would be from the block following). If N is not specified, it defaults to 1. If there are no more blocks left in the data file and the .NB command is used, number register E will be set to one to designate an End of file condition.

USING THE TEXT PROCESSOR

I. BRINGING UP THE SYSTEM

The disk processor command file name is "PR". The general syntax for the PR command is:

PR,<file spec>[,<list of file specs>]

The <file spec> designates which text file is to be processed. If the text to be processed is divided among several files, each file spec may be listed separately on the calling line separated by commas. A special feature supported by PR is the ability to process files from any number of discs on systems containing a limited number of drives. Substituting a '*' for the <file spec> any where on the calling line where a <file spec> is expected will cause the processor to halt and output to the terminal:

CHANGE DISKS AND TYPE A KEY

At this time, insert the disk containing the continuation file(s) and type any key to restart processing. It should be noted that the ability to process multiple files with one calling line should only be used when the files are actual continuations of the same text. The processor treats them as if they were all part of the same file, continuing page numbers, indenting, page width, etc., just as if the first file had never ended.

Another feature supported by the processor is the ability to automatically process a macro definition file prior to processing any of the files specified. Upon the execution of PR, the disk in drive Ø is searched for a file named 'MACRO'. If none is found, the processor starts processing the first file specified. If a MACRO file is present, it is read in and processed, just as if it had been the first file specified in the calling line. This is useful for defining all often used macros in this file so it is not necessary to redefine them in each processor text file prepared.

A few examples will clarify the calling of PR:

```
PR,CHPTR1
PR,Ø:CHPTR1,1:CHPTR2,*,Ø:CHPTR3
```

The first example will process the file named CHPTR1. The file MACRO will also be processed if it exists. The second example will first try to process the file MACRO, then process the files CHPTR1 on drive Ø and CHPTR2 on drive 1. The processor will then halt and output the 'CHANGE DISK' message to the terminal because of the '*' used as a file spec. After changing disks in drive Ø and typing a key, the processor will process the file named CHPTR3 on drive Ø.

When the procesor is called, the following message will be output to the terminal:

PAGE LIMITS?

and is used to specify a particular block of pages to be processed. Typing a carriage return will cause all pages to be processed and output. Typing two numbers separated by a space or a comma will cause only the pages between those numbers (inclusive) to be output. For example, typing:

10,16

will result in only pages numbered 10 through 16 to be output. If just one number is entered, the processor will start outputing at that page number and continue to the end of the file. It should be noted that the processor always starts numbering the first page as number one unless instructed otherwise. As the processor is working, it may be stopped at any time by typing a "control C" on the terminal. (This feature is only supported on computers using a serial type interface (MP-S) as the terminal interface port.) The processor will respond with:

..BREAK..

output to the terminal. At this time processing may be continued by typing any character except an "S" which will cause the processor to be exited.

II. GENERAL USE

There are several things to keep in mind while preparing text for the text processor. Remember that all commands must begin in column one. It is usually most convenient to begin each sentence on a new line for easy future editing. Macros should be used as often as possible. The reason for this is to keep global changes as simple as possible, e.g. change only one line in a macro as opposed to changing single commands scattered throughout the file. It is not necessary to understand how the macros provided in this manual work in order to use them. All that is necessary is to know how to use them which is thoroughly explained. As experience is gained with the processor, you will be able to create your own special purpose macros for easy formatting.

MACRO LIBRARY

The following macro descriptions range from simple header and footer macros to a very complex footnote macro. It is not necessary to understand how the macros work, just how to use them. Each macro includes a description of what it does and how it can be used.

I. HEADERS AND FOOTERS

These macros are used to define top and bottom margins and also specify the contents of these margins, such as page numbers, titles, etc. Almost all processing jobs will require some sort of header and footer. Usually the macro definitions are placed at the beginning of the file (they need to appear before they are called for execution). The "AT" command is used to set the trap location (the line at which the macro should automatically execute) of each of the macros. Headers are set to line 1 and footers to a specific distance from the bottom of the page. Once these macros have been defined and their trap locations set, they can be forgotten about since the processor will do all the rest of the work. The first macro is a simple header macro which provides two blank lines, a centered title, and two more blank lines at the top of each page.

```
.DM HD
:SP 2
.TL ''CENTERED TITLE''
:SP 2
.NS
.OS
..

.AT 1 HD
```

All of the header macros will contain a NS and OS command. NS will suppress any unnecessary spacing which may occur due to the unpredicted appearance of a SP command. For example, if the start of a new paragraph just happens to start at the top of a new page, there is no reason for the paragraph macro to space down two lines, since we are at the top of the page. NS will keep this from happening. The OS command instructs the processor to output any 'saved space' from the previous page. The next header is a little fancier. It does everything the previous one does except the titling is done a little differently. Here, if the current page number is even, the title is left hand justified. If the page is odd, the title is right hand adjusted.

```
.DM HD
:SP 2
.IF E .TL 'EVEN TITLE'''
.IF O .TL '''ODD TITLE'
:SP 2
.NS
.OS
..

.AT 1 HD
```

Subtitles may be used by simply placing a second TL command which contains the subtitle. The last header example is for those using a printer which uses separate sheets of paper (as opposed to continuous fed). This macro will issue a message to the terminal which instructs the operator to insert a new sheet of paper, before each page of text is processed. The paper should be set up such that the first line printed will be the top line of the paper. The operator will have to type a character on the terminal after each stop to restart the processor. Remember that typing an "S" will halt the processor.

```
.DM HD
.TM INSERT NEW SHEET
:ST
:SP 2
.TL ''TITLE''
:SP 2
.NS
.OS
..

.AT 1 HD
```

Footer macros are similar to headers except they are set to execute at the bottom of a page. For example, specifying AT -6 FO would cause the macro called FO to automatically execute at the 6th line from the bottom of the page. The first footer gives a five line bottom margin with the page number between 2 dashes centered on the page, 3 lines from the bottom.

```
.DM FO
:SP 2
.TL ''-%-''
:PG
..

.AT -5 FO
```

It is often desirable to have page numbers on every page except page number one. The following footer will do exactly that.

```
.DM FO
:SP 2
.IF %-1 .TL ''-%-''
:PG
..

.AT -5 FO
```

There are several other types of header and footer macros which can be created. Some of these appear in the macros which follow.

II. PARAGRAPHS AND HEADINGS

There are many forms of paragraphing. The SSE Text Processor does not restrict one to using one particular form. One type of paragraph is to produce one blank line and start the first line of the paragraph indented five spaces. The following macro does just that:

```
.DM PP
.SP
.SI 5
..
```

To use the paragraph macro, simply call it by name any time a new paragraph is desired (e.g. type ".PP" in column one). One little feature which may be added to the macro is a need lines command, NL. In the following example, NL 3 is used to tell the processor that we desire at least three lines be left on the page before a new paragraph is started. This will keep one or two lone lines from being placed at the bottom of the page.

```
.DM PP
.SP
.NL 3
.SI 5
..
```

Many other types of paragraph macros may be created along the same lines as those presented.

Another useful macro can be created for major heading creation. One type of major heading might have a centered title spaced two lines down from the last line of text. The macro to accomplish this may look as follows:

```
.DM MH
.SP 2
.CE
..
```

To use this macro, type ".MH" when the heading is desired. The next line should contain the heading title. For example:

```
Line of text.
.MH
Heading Title
```

The last two macro examples are quite simple, but show how even two or three lines of commands may be replaced by a single macro call. This is quite useful if these operations are going to be repeated many times throughout a document.

III. FOOTNOTES

The following set of macros is all that is required to do very efficient and easy footnote handling. A description of how they actually work is contained in the introduction of this manual. To use these macros, it is only necessary to include their descriptions at the beginning of your file. As soon after a footnote is referenced in the text, call the macro BF (begin footnote) to begin the footnote. Immediately following this call, type the contents of the footnote, followed by a call to the macro EF (end footnote). The following serves as an example:

```
Text here referencing a footnote*.
.BF
*Footnote contents typed here and
may be several lines long.
.EF
```

It should be noted that the footnote macros contain their own header and footer macros which may be modified as desired. These macros should be the first lines of a file.

```
.NR B 7
.DM HD
:SP 2
.IF %-1 .TL 'FOOTNOTE TEST'''
:SP 2
.AU 1
.NR X Ø
.NR W Ø-#B
.IF #V .TR
.NS
..
.DM FO
.NR V Ø
.IF #X .FT
.CH FO -#B
:PG
..
.DM NM
.TL ''-%-''
..
```

- continued -

```
.DM BF
.DA TX
.EV 1
.IF !#+X-1 .SA
..
.DM EF
.BR
.EV Ø
.DI
.NR W -#V
.CH FO #W
.IF #N-#P-#W .CH FO #N+1
..
.DM SA
--------------------
.BR
..
.DM TR
.BF
.NF
.FE
.FI
.EF
..
.DM FN
.DI FE
..
.DM FT
.EV 1
.NF
.TX
.RM TX
.DI
.FI
.EV Ø
..
.AT 1 HD
.AT -#B FO
.AT -4 NM
.CH FO 7Ø
.AT -#B FN
.CH FO -#B
.EV 1
.AU 1
.LN 55
.EV Ø
```

Please remember that it is not necessary to fully understand how these macros work as long as you know how to use them.

IV. TWO COLUMN OUTPUT

The SSB processor does not support backward line feeds so it is necessary to use some operator intervention in order to produce two column output. The following set of macros will produce two column output, each column being 31 characters wide. When the text of the first column reaches the bottom of the page, the string "REPOSITION PAPER" will be output to the terminal and a "STOP" command is executed. At this time the operator should reposition the paper to the top of the page and then restart the processor by typing any key but "S".

```
.LN 31
.NR A Ø
.DM HD
.IF #A .PA
:SP 2
.AU 1
.IF !#+A-1 .TL ''title''
.IF #A-1 :SP
:SP 2
.IF #A-1 .LM 34
..
.DM FO
:SP 2
.LM Ø
.IF #A-1 .TL ''-%-''
.IF #A-1 .NR A Ø
:PG
..
.DM PA
.TM REPOSITION PAPER
:ST
.PN %-1
..
.AT 1 HD
.AT -5 FO
.BR
```

It should be noted that these macros also contain their own special set of header and footer macros which may be modified as desired.

V. FORM LETTERS

The last set of macros and examples deal with form letters. These macros are shown with some sample text and make extensive use of disk data files. This example should be thoroughly studied before trying to make use of disk data file commands. The RP (repeat) command is used so that the file is repeated over and over, until the end of file has been reached in the data file (number register E is non zero). The macro creates a name and address header at the top of each page. Following is "Dear (persons name)" and the text of the letter. The sample program is shown below, followed by the sample data file, and then a sample of the output produced by the processor.

```
.OF
.JU N
.NF
.DI NM
.RI
.BR
.DI
.IF #E .EX
.SP 6
.NM
.RI
.RI
.SP 3
.FI
Dear
.NM
.SP
.SI 5
We are writing to you to inform you that your
.RI
Insurance policy is about to expire.
Your policy number is
.RI
and expires on
.RI S
\.
If you desire renewal, please send payment by
the end of this month.
If payment is not received, your policy will be terminated.
.RI
Thank you for your attention to this matter.
.SP 2
.NF
Thank you
.SP 3
 Agent
.NB
.RP
```

The sample data file is as follows:

```
John Doe>
1313 Riverside Ave.>
Akron, Ohio 44225>
Fire>
F3-4322-946>
March 15, 1975>>
Bill Jones>
1111 Crescent Street
Apartment #12>
Kingston, New York 10011>
Automobile>E5-4936-263>March 14, 1975>
This is your second and final notice!>>
Hiram Johnson>
RR #3>
Lotson, Virginia 32004>
Life>
B1-2234-123>
March 12, 1975>>
```

As can be seen in the above sample data file, items may be placed one per line, or multiples per line as desired. The following is the output obtained from the first block of the data file.

SYSTEM ADAPTATIONS

There are three features which can be user adjusted. These are treated separately below.

I. MACRO STORAGE SPACE

The macro storage space is presently set to approximately 4K and resides at the top of the first 12K block of memory. In 99% of all applications, this space will be much more than sufficient. If more memory is available, and you are requiring more macro space, the size of this space can be expanded. The end of the space address is stored at location $0217 (LMACRO) and may be changed as needed.

II. DOUBLE CHARACTERS

Three commands exist in the processor which require special printer hardware. These ara double height (DH), double width (DW), and double both (DB). Some commercially available printers will print single lines of double size characters if a special control character is received prior to the line. The double height control character ($12) is stored in location $021C. The double width control character ($0E) is stored in location $021D. These may be changed as required.

III. SUSPENDING EXECUTION OF THE TEXT PROCESSOR

Typing Control C during the execution of the text processor will suspend the processing and print "..BREAK.." on the terminal. Typing the letter "S" will cause the processor to exit to the operating system. Any other letter will cause processing to continue. The text processor is configured to use the MIKBUG PIA terminal interface. If you are instead using an ACIA for terminal I/O, the base address of the ACIA should be placed in locations $0213.

```
                     2:          WITH    WI=80,NDMP
                     4: *
                     5: *
                     6: * SSE 6800 TEXT PROCESSING SYSTEM
                     7: *
                     8: *
                     9: * COPYRIGHT 1978 BY
                    10: *
                    11: * SMOKE SIGNAL BROADCASTING
                    12: * 6304 YUCCA
                    13: * HOLLYWOOD, CA 90028
                    14: *
  000E              15: EDIT    EQU     14
                    16: * EDIT DATE: 06-13-78  (NO ITS NOT FRIDAY)
                    17:
  0030              18:         ORG     $0030
                    19:
                    20: * TEMPORARY STORAGE
                    21:
                    22: * NUMBER REGISTERS
                    23:
0030 0002           24: NMREGS  RMB     2       A-B
0032 0001           25: COLCNT  RMB     1       C
0033 0001           26: DAY     RMB     1       D
0034 0001           27: EOFF    RMB     1       E
0035 0001           28:         RMB     1       F
0036 0001           29: GCNT    RMB     1       G
0037 0001           30:         RMB     1       H
0038 0001           31: IND     RMB     1       I
0039 0002           32:         RMB     2       J-K
003B 0001           33: LLN     RMB     1       L
003C 0001           34: MNTH    RMB     1       M
003D 0001           35: LINCNT  RMB     1       N
003E 0001           36: LFM     RMB     1       O
003F 0001           37: PGL     RMB     1       P
0040 0005           38:         RMB     5       Q-U
0045 0001           39: LDIV    RMB     1       V
0046 0002           40:         RMB     2       W-X
0048 0001           41: YEAR    RMB     1       Y
0049 0001           42:         RMB     1       Z
                    43:
                    44: * SPECIAL DISK STORAGE
                    45:
004A 0001           46: EOIF    RMB     1
004B 0001           47: EORF    RMB     1
004C 0001           48: ITEM    RMB     1
004D 0001           49: FILOPN  RMB     1
                    50:
                    51: * SINGLE STORAGE
                    52:
004E 0001           53: ULFLG   RMB     1
004F 0001           54: GDNUM   RMB     1
0050 0001           55: ADD     RMB     1
0051 0001           56: SUB     RMB     1
0052 0001           57: BNUM    RMB     1
```

```
0053 0001          58: NPGN    RMB     1
0054 0001          59: INC     RMB     1
                   60:
                   61: * MACRO SAVE BLOCK
                   62:
0055 0002          63: NUMPNT  RMB     2
0057 0001          64: EXCHR   RMB     1
0058 0002          65: LSTNUM  RMB     2
005A 0001          66: CMFLG   RMB     1
005B 0001          67: MBFLG   RMB     1
005C 0002          68: MBFPNT  RMB     2
005E 0001          69: NOCR    RMB     1
005F 0001          70: DONE    RMB     1
0060 0001          71: FLBF    RMB     1
0061 0001          72: ATFLG   RMB     1
                   73:
                   74: * MORE SINGLE STORAGE
                   75:
0062 0001          76: LEFT    RMB     1
0063 0001          77: TFILF   RMB     1
0064 0001          78: NOFL    RMB     1
0065 0001          79: INNUM   RMB     1
0066 0001          80: NEG     RMB     1
0067 0001          81: SIGN    RMB     1
0068 0001          82: NSP     RMB     1
0069 0001          83: PGN     RMB     1
006A 0001          84: PASCHR  RMB     1
006B 0001          85: SPSPF   RMB     1
006C 0001          86: DOCAP   RMB     1
006D 0001          87: NOOUT   RMB     1
006E 0001          88: TOUTL   RMB     1
006F 0001          89: PTFL    RMB     1
0070 0001          90: SIN     RMB     1
0071 0001          91: MINDIS  RMB     1
0072 0001          92: EV      RMB     1
0073 0001          93: NOEXP   RMB     1
0074 0002          94: NXTTAB  RMB     2
0076 0001          95: TABFLG  RMB     1
0077 0001          96: COLCN2  RMB     1
0078 0001          97: IND2    RMB     1
0079 0002          98: NXTTRP  RMB     2
007B 0001          99: SVDSPC  RMB     1
007C 0001         100: FINMAC  RMB     1
007D 0001         101: NEGT    RMB     1
007E 0001         102: IFFLG   RMB     1
007F 0001         103: MACCNT  RMB     1
0080 0001         104: PASFLG  RMB     1
0081 0001         105: NONUMS  RMB     1
0082 0001         106: DWFLG   RMB     1
0083 0001         107: DFMFLG  RMB     1
0084 0001         108: SPIFLG  RMB     1
0085 0001         109: DIVFLG  RMB     1
0086 0001         110: DIVFL2  RMB     1
0087 0001         111: RIFLG   RMB     1
0088 0001         112: CRSUP   RMB     1
```

```
0089 0001        113: NCOUNT RMB    1
008A 0001        114: PSCNT  RMB    1
008B 0002        115: INFCP  RMB    2
008D 0002        116: XTEMP2 RMB    2
008F 0001        117: PRNTR  RMB    1
0090 0001        118: TLPP   RMB    1
0091 0001        119: LOWPG  RMB    1
0092 0001        120: HIPG   RMB    1
                 121:
0093 0001        122: SBFLG  RMB    1
0094 0001        123: LLN2   RMB    1
0095 0002        124: MACNAM RMB    2
0097 0002        125: MACTMP RMB    2
0099 0002        126: LSTAVL RMB    2
009B 0002        127: FSTAVL RMB    2
009D 0002        128: STPOUT RMB    2
009F 0002        129: NXTMAC RMB    2
00A1 0002        130: NXTOUT RMB    2
00A3 0002        131: XMAC   RMB    2
00A5 0001        132: TSIN   RMB    1
00A6 0001        133: TIND   RMB    1
00A7 0001        134: TLLN   RMB    1
00A8 0001        135: SUPL   RMB    1
00A9 0001        136: SWRDF  RMB    1
00AA 0001        137: CAP    RMB    1
00AB 0001        138: SCAP   RMB    1
00AC 0001        139: TPOS   RMB    1
00AD 0001        140: DELIM  RMB    1
00AE 0001        141: TCNT   RMB    1
00AF 0001        142: MCNT   RMB    1
00B0 0002        143: TTLPNT RMB    2
00B2 0001        144: ENDLIN RMB    1
00B3 0001        145: TAB    RMB    1
00B4 0001        146: TFILL  RMB    1
                 147:
                 148: * ENVIRONMENT PARAMETERS
                 149:
00B5 0002        150: AUTO   RMB    2
00B7 0002        151: ROM    RMB    2
00B9 0002        152: WIDTH  RMB    2
00BB 0002        153: FILFLG RMB    2
00BD 0002        154: PFLG   RMB    2
00BF 0002        155: PCHAR  RMB    2
00C1 0002        156: CNJ    RMB    2
00C3 0002        157: RTJ    RMB    2
00C5 0002        158: MSC    RMB    2
00C7 0002        159: CNTFLG RMB    2
00C9 0002        160: JUST   RMB    2
00CB 0002        161: TLN    RMB    2
00CD 0004        162: BUFPNT RMB    4
00D1 0004        163: BUFEND RMB    4
00D5 0004        164: EBFEND RMB    4
                 165:
00D9 0002        166: CMNPNT RMB    2
00DB 0002        167: SPCPT1 RMB    2
```

```
00DD 0002       168: SPCPT2 RMB    2
00DF 0002       169: TEMP   RMB    2
00E1 0002       170: TEMP2  RMB    2
00E3 0002       171: TEMP5  RMB    2
00E5 0002       172: TEMP6  RMB    2
00E7 0002       173: RETRLG RMB    2
00E9 0002       174: INDEX  RMB    2
00EB 0002       175: XTEMP  RMB    2
00ED 0002       176: MACEND RMB    2
00EF 0001       177: CRF    RMB    1
                178:
  0110          179:        ORG    $0110
                180:
0110 0014       181: TABS   RMB    20
0124 0001       182: TABEND RMB    1
0125 000C       183: NUM    RMB    12
```

```
  0200                1:         ORG     $0200
                      2:
                      3: *
                      4: * >>>>> PROGRAM ENTRY POINT <<<<<
                      5: *
0200 7E023B           6: START   JMP     INTRO
                      7:
                      8: * JUMP TABLE
                      9:
0203 7E7286          10: OUTCH   JMP     $7286    TERMINAL CHARACTER OUTPUT
0206 7E7289          11: INCH    JMP     $7289    TERMINAL CHARACTER INPUT
0209 7E7283          12: MON     JMP     $7283    ADDRESS IN MONITOR TO EXIT TO
020C 7E1883          13: PINIT   JMP     PRNIT    PRINT INITIALIZATION
020F 7E18A3          14: POUCH   JMP     PROUCH   PRINTER CHARACTER OUTPUT
                     15:
0212 0E              16:         FCB     EDIT     REVISION CONTROL INFO
0213 0000            17: ACIADR  FDB     $0000    ADDRESS OF CONSOLE ACIA IF PRESEN
0215 1E9F            18: MACROS  FDB     ENDTP    MACROS CAN START AT END OF PROGRA
0217 2EFD            19: LMACRO  FDB     $2EFD    LAST AVAILABLE FOR MACROS
0219 01FF            20: STACK   FDB     $01FF    H/W STACK AREA
                     21:
021B 3E              22: ITEMCH  FCB     $3E      ">" DEFAULT ITEM CHARACTER
021C 12              23: DHCHAR  FCB     $12      DOUBLE HEIGHT CONTROL CHAR
021D 0E              24: DWCHAR  FCB     $0E      DOUBLE WIDTH CONTROL CHAR
                     25:
                     26: * VERSION STRING
                     27:
021E 322E3820        28: VERSTR  FCC     "2.8  "
0223 0D0A04          29:         FCB     13,10,4
                     30:
                     31: * DISK ROUTINES
                     32:
0226 7E7291          33: ZFLSPC  JMP     $7291
0229 7E7294          34: ZGCHAR  JMP     $7294
022C 7E7297          35: ZGNCHR  JMP     $7297
022F 7E729A          36: ZANCHK  JMP     $729A
0232 7E72A9          37: ZTYPDE  JMP     $72A9
0235 7E7783          38: CDFM    JMP     $7783
0238 7E7786          39: DFM     JMP     $7786
                     40:
                     41: * MAIN PROGRAM STARTS HERE
                     42:
023B BE0219          43: INTRO   LDS     STACK    *** SETUP STACK ***
023E BD02CF          44:         JSR     CLRSPC   GO CLEAR SPACE
0241 9791            45:         STAA    LOWPG    SET PAGE LIMITS
0243 978F            46:         STAA    PRNTR
0245 9790            47:         STAA    TLPP
0247 4A              48:         DEC     A
0248 9792            49:         STAA    HIPG
024A CE176C          50:         LDX     #BANNER  SAY HELLO
024D BD1636          51:         JSR     PSTRNG     TELL THEM WHO WE ARE
0250 CE021E          52:         LDX     #VERSTR  AND THEN
0253 BD1638          53:         JSR     PDATA      TELL THEM WHICH REVISION
0256 CE17EC          54:         LDX     #DATSTR  PROMPT FOR DATE
0259 BD1636          55:         JSR     PSTRNG
```

```
025C BD15D5      56:          JSR    GIBUF    GET DATE
025F 7C005A      57:          INC    CMFLG
0262 BD12E9      58:          JSR    CHKNUM   CHECK IF VALID
0265 2416        59:          BCC    INTRO3
0267 9665        60:          LDAA   INNUM
0269 973C        61:          STAA   MNTH     GET MONTH & SAVE
026B BD12E9      62:          JSR    CHKNUM   CHECK FOR DAY
026E 240D        63:          BCC    INTRO3
0270 9665        64:          LDAA   INNUM    GET & SAVE
0272 9733        65:          STAA   DAY
0274 BD12E9      66:          JSR    CHKNUM   CHECK FOR YEAR
0277 2404        67:          BCC    INTRO3
0279 9665        68:          LDAA   INNUM    GET & SAVE
027B 9748        69:          STAA   YEAR
027D CE17CE      70: INTRO3   LDX    #PRQU    PROMPT FOR PRINTER
0280 BD1636      71:          JSR    PSTRNG
0283 BD0206      72:          JSR    INCH     GET RESPONSE
0286 8150        73:          CMPA   #'P
0288 2607        74:          BNE    INTRO4
028A 978F        75:          STAA   PRNTR    SET PRINTER FLAG
028C BD020C      76:          JSR    PINIT    INITIALIZE PRINTER
028F 2015        77:          BRA    INTRO5
0291 CE1808      78: INTRO4   LDX    #LPPSTR  LINES PER SCREEN?
0294 BD1636      79:          JSR    PSTRNG
0297 BD15D5      80:          JSR    GIBUF    GET RESPONSE
029A 7C005A      81:          INC    CMFLG
029D BD12E9      82:          JSR    CHKNUM   CHECK IF NUMBER
02A0 2404        83:          BCC    INTRO5
02A2 9665        84:          LDAA   INNUM    GET AND SAVE
02A4 9790        85:          STAA   TLPP
02A6 CE17EC      86: INTRO5   LDX    #PGSTR   PRMPT FOR PAGES
02A9 BD1636      87:          JSR    PSTRNG
02AC BD15D5      88:          JSR    GIBUF    GET RESPONSE
02AF 7C005A      89:          INC    CMFLG
02B2 BD12E9      90:          JSR    CHKNUM   CHECK NUMBER
02B5 240D        91:          BCC    INTRO6
02B7 9665        92:          LDAA   INNUM    GET AND SAVE
02B9 9791        93:          STAA   LOWPG
02BB BD12E9      94:          JSR    CHKNUM   CHECK HIGH PAGE
02BE 2404        95:          BCC    INTRO6
02C0 9665        96:          LDAA   INNUM    GET IT
02C2 9792        97:          STAA   HIPG
02C4 BD162B      98: INTRO6   JSR    CRLF     OUT CR & LF
02C7 4F          99:          CLR    A
02C8 CE004F     100:          LDX    #GDNUM   CLEAR SPACE
02CB 8D06       101:          BSR    CLRSP2
02CD 2018       102:          BRA    INIT     GO INITIALIZE
                103:
                104: * CLEAR TEMPORARY SPACE
                105:
02CF 4F         106: CLRSPC  CLR    A
02D0 CE0030     107:          LDX    #NMREGS  SET POINTER
02D3 A700       108: CLRSP2  STAA   0,X      CLEAR SPACE
02D5 08         109:          INX             BUMP POINTER
02D6 8C008F     110:          CPX    #PRNTR   FINISHED?
```

```
02D9 26F8       111:         BNE    CLRSP2
02DB CE0093     112:         LDX    #SBFLG   DO SECOND BLOCK
02DE A700       113: CLRSP4  STAA   0,X
02E0 08         114:         INX
02E1 8C00CD     115:         CPX    #BUFPNT
02E4 26F8       116:         BNE    CLRSP4
02E6 39         117:         RTS    RETURN
                118:
                119: * INITIALIZATION AND SETUP
                120:
02E7 CE0110     121: INIT    LDX    #TABS    SET POINTER
02EA 4F         122:         CLR    A
02EB A700       123: INIT25  STAA   0,X      CLEAR TABS
02ED 08         124:         INX
02EE 8C0125     125:         CPX    #NUM
02F1 26F8       126:         BNE    INIT25   FINISHED?
02F3 4C         127:         INC    A
02F4 97C9       128:         STAA   JUST     SET INITIAL PARAMS.
02F6 97CA       129:         STAA   JUST+1
02F8 97B2       130:         STAA   ENDLIN   MARK END LINE
02FA 97EF       131:         STAA   CRF
02FC 97BB       132:         STAA   FILFLG   SET FOR FILL
02FE 97BC       133:         STAA   FILFLG+1
0300 973D       134:         STAA   LINCNT   INIT LINE COUNT
0302 9732       135:         STAA   COLCNT
0304 9777       136:         STAA   COLCN2   SET COLUMN CNT
0306 9769       137:         STAA   PGN      SET PAGE
0308 8641       138:         LDAA   #65
030A 97B9       139:         STAA   WIDTH    SET PAGE WIDTH
030C 97BA       140:         STAA   WIDTH+1
030E 973B       141:         STAA   LLN      AND LINE LENGTH
0310 9794       142:         STAA   LLN2
0312 97CB       143:         STAA   TLN      SET TITLE LENGTH
0314 97CC       144:         STAA   TLN+1
0316 4C         145:         INC    A
0317 973F       146:         STAA   PGL      SET PAGE LENGTH
0319 FE0215     147:         LDX    MACROS
031C DF9F       148:         STX    NXTMAC   INIT MACRO SPACE
031E DF9B       149:         STX    FSTAVL
0320 86FF       150:         LDAA   #$FF
0322 A700       151: INIT3   STAA   0,X
0324 08         152:         INX
0325 BC0217     153:         CPX    LMACRO   FINISHED?
0328 26F8       154:         BNE    INIT3
032A DF99       155:         STX    LSTAVL
032C 09         156:         DEX
032D 6F00       157:         CLR    0,X      SET END OF MACROS
032F 6F01       158:         CLR    1,X
0331 6F02       159:         CLR    2,X
0333 860D       160:         LDAA   #$D      FIX BUFFER
0335 B71AE9     161:         STAA   CMNDBF
0338 86A0       162:         LDAA   #$A0     SET FILL CHAR.
033A 97B4       163:         STAA   TFILL
033C CE1AB7     164:         LDX    #TRAPS
033F 86FF       165:         LDAA   #$FF     INIT TRAPS
```

```
0341 A700       166: INIT4  STAA   0,X
0343 08         167:        INX
0344 8C1AE7     168:        CPX    #TRPEND FINISHED?
0347 26F8       169:        BNE    INIT4
0349 CE18C3     170:        LDX    #LINBUF
034C DFCD       171:        STX    BUFPNT  SET POINTER
034E DFCF       172:        STX    BUFPNT+2
0350 BD0715     173:        JSR    FIXBFE  FIX BUFFER END
0353 DED1       174:        LDX    BUFEND
0355 DFD3       175:        STX    BUFEND+2
0357 CE195E     176:        LDX    #EXTBUF
035A DFD7       177:        STX    EBFEND+2
035C CE1D9D     178:        LDX    #MACTBL CLEAR MACRO TABLE
035F DFED       179:        STX    MACEND
                180:
                181: * MAIN PROCESSOR LOOP
                182:
0361 CE187A     183:        LDX    #MACST  POINT TO NAME
0364 DFEB       184:        STX    XTEMP   SAVE IT
0366 CE1C51     185:        LDX    #TFCB   POINT TO FCB
0369 DF8D       186:        STX    XTEMP2  SAVE IT
036B B6021B     187:        LDAA   ITEMCH  SET ITEM CHAR
036E 974C       188:        STAA   ITEM
0370 C609       189:        LDAB   #9      SET COUNTER
0372 DEEB       190: DPROC1 LDX    XTEMP   GET POINTER
0374 A600       191:        LDAA   0,X     GET CHAR
0376 08         192:        INX            BUMP POINTER
0377 DFEB       193:        STX    XTEMP
0379 DE8D       194:        LDX    XTEMP2  GET DESTINATION
037B A703       195:        STAA   3,X     PUT IN NAME
037D 08         196:        INX
037E DF8D       197:        STX    XTEMP2  SAVE
0380 5A         198:        DECB   DEC     THE COUNT
0381 26DF       199:        BNE    DPROC1
0383 CE1C51     200:        LDX    #TFCB   POINT TO FCB
0386 6F02       201:        CLR    2,X     SET TO DRIVE #0
0388 8604       202:        LDAA   #4      OPEN FOR READ
038A A700       203:        STAA   0,X
038C BD0238     204:        JSR    DFM     CALL DFM
038F 2763       205:        BEQ    DPROC6
0391 A601       206:        LDAA   1,X
0393 8105       207:        CMPA   #5      CHECK FOR NO FILE
0395 2703       208:        BEQ    DPROC2
0397 7E16F3     209:        JMP    DODFM4  GO REPORT ERROR
039A BD0229     210: DPROC2 JSR    ZGCHAR  GET CHARACTER
039D 810D       211:        CMPA   #$D     IS IT CR?
039F 270D       212:        BEQ    DPRO24
03A1 7F005F     213:        CLR    DONE
03A4 CE1C51     214:        LDX    #TFCB   POINT TO FCB
03A7 BD0226     215:        JSR    ZFLSPC  GET FILE NAME
03AA 242C       216:        BCC    DPROC5
03AC 200B       217:        BRA    DPROC3
03AE 7D008A     218: DPRO24 TST    PSCNT   FIRST NAME?
03B1 2706       219:        BEQ    DPROC3
03B3 BD0235     220:        JSR    CDFM    CLOSE DFM
```

```
03E6 7E0A17      221:          JMP     FINIS4   FINISH PAGE
03E9 CE163E      222: DPROC3   LDX     #ILFN    POINT TO STRING
03EC BD1636      223:          JSR     PSTRNG   OUTPUT IT
03EF BD0235      224:          JSR     CLFM     CLOSE DFM
03C2 7E0209      225:          JMP     MON      EXIT
03C5 CE1650      226: DPROC4   LDX     #CHST    POINT TO STRING
03C8 BD1636      227:          JSR     PSTRNG   OUTPUT IT
03CB BD0206      228:          JSR     INCH     WAIT FOR CHAR
03CE BD022C      229:          JSR     ZGNCHR   SKIP CHARACTER
03D1 BD022F      230:          JSR     ZANCHK   CHECK CHARACTER
03D4 24E3        231:          BCC     DPROC3
03D6 20C2        232:          BRA     DPROC2
03D8 812A        233: DPROC5   CMPA    #$2A     IS IT '*'
03DA 27E9        234:          BEQ     DPROC4
03DC 7C008A      235:          INC     PSCNT    BUMP PASS COUNTER
03DF CE1C51      236:          LDX     #IFCB    POINT TO FCB
03E2 8604        237:          LDAA    #4
03E4 A700        238:          STAA    0,X
03E6 BD0238      239:          JSR     DFM
03E9 2709        240:          BEQ     DPROC6
03EB BD0232      241:          JSR     ZTYPDE   REPORT ERROR
03EE BD0235      242:          JSR     CLFM     CLOSE FILES
03F1 7E0209      243:          JMP     MON      RETURN TO DOS
03F4 8605        244: DPROC6   LDAA    #5
03F6 A700        245:          STAA    0,X      SET FOR READ
                 246:
03F8 9669        247: PROC     LDAA    PGN      CHECK PAGE NUMBER
03FA 9191        248:          CMPA    LOWPG    AGAINST LOW PAGE
03FC 2406        249:          BCC     PROC3
03FE C60F        250:          LDAB    #$F
0400 D76D        251:          STAB    NOOUT    SET NO OUTPUT FLAG
0402 200A        252:          BRA     PUNTST
0404 9192        253: PROC3    CMPA    HIPG     AGAINST HIGH PAGE
0406 2303        254:          BLS     PROC4
0408 7E0A17      255:          JMP     FINIS4   IF PAST, FINISH
040B 7F006D      256: PROC4    CLR     NOOUT
                 257:
                 258: * TEST FOR PUNCTUATION
                 259:
040E 96BD        260: PUNTST   LDAA    PFLG     TEST FLAG
0410 8103        261:          CMPA    #3
0412 2607        262:          BNE     PUNTS3
0414 96BF        263:          LDAA    PCHAR    GET SPARE CHAR.
0416 7F00BD      264: PUNTS2   CLR     PFLG     CLEAR PUNCT. FLAG
0419 2037        265:          BRA     JSTFY
041B BD074C      266: PUNTS3   JSR     GETCHR   GET NEXT CHAR.
041E D65F        267:          LDAB    DONE     FINISHED?
0420 2703        268:          BEQ     PUNT35
0422 7E0A08      269:          JMP     FINISH
0425 D6EB        270: PUNT35   LDAB    FILFLG   FILL ON?
0427 2729        271:          BEQ     JSTFY
0429 D6BD        272:          LDAB    PFLG     TEST PUNCT. FLAG
042B C101        273:          CMPB    #1
042D 2219        274:          BHI     PUNTS7
042F 2711        275:          BEQ     PUNTS6
```

```
0431 812E        276:          CMPA   #'.       IS CHAR A '.'?
0433 2708        277:          BEQ    PUNTS4
0435 8121        278:          CMPA   #'!       IS IT '!'?
0437 2704        279:          BEQ    PUNTS4
0439 813F        280:          CMPA   #'?       IS IT '?'?
043B 2603        281:          BNE    PUNTS5
043D 7C00BD      282: PUNTS4   INC    PFLG      SET PUNCT. FLAG
0440 2010        283: PUNTS5   BRA    JSTFY
0442 8120        284: PUNTS6   CMPA   #$20      IS CHAR SPACE?
0444 27F7        285:          BEQ    PUNTS4
0446 20CE        286:          BRA    PUNTS2
0448 8120        287: PUNTS7   CMPA   #$20      CHECK FOR SPACE
044A 27CA        288:          BEQ    PUNTS2
044C 97BF        289:          STAA   PCHAR     SAVE SPARE CHAR.
044E 8620        290:          LDAA   #$20      SET FOR SPACE
0450 20EB        291:          BRA    PUNTS4
                 292:
                 293: * JUSTIFICATION LOOP
                 294:
0452 CE195E      295: JSTFY    LDX    #EXTBUF   FIX EXTRA POINTERS
0455 DFE1        296:          STX    TEMP2
0457 DFD5        297:          STX    EBFEND
0459 DECD        298:          LDX    BUFPNT    GET BUFFER POINTER
045B 810D        299:          CMPA   #$D       IS CHAR. A CR?
045D 2614        300:          BNE    JSTFY3
045F D6BB        301:          LDAB   FILFLG    FILL MODE?
0461 2605        302:          BNE    JSTFY2
0463 DF9D        303: JSTFY1   STX    STPOUT    MARK LAST BUF. POS.
0465 7E05CC      304:          JMP    OUTLIN    OUTPUT LINE
0468 8620        305: JSTFY2   LDAA   #$20      GET A SPACE
046A A700        306: JSTF25   STAA   0,X       SAVE IT
046C 08          307:          INX              BUMP POINTER
046D 9CD1        308:          CPX    BUFEND    END OF BUFFER?
046F 26F9        309:          BNE    JSTF25
0471 2021        310:          BRA    JSTFY6
0473 BD0668      311: JSTFY3   JSR    TSULN     TEST UL CHAR
0476 A700        312:          STAA   0,X       SAVE CHARACTER
0478 7C0032      313:          INC    COLCNT    BUMP COLUMN COUNT
047B 08          314:          INX              BUMP POINTER
047C 9CD1        315:          CPX    BUFEND    END?
047E 2606        316:          BNE    JSTFY4
0480 D6BB        317:          LDAB   FILFLG    FILL MODE?
0482 2702        318:          BEQ    JSTFY4
0484 200E        319:          BRA    JSTFY6
0486 8C195E      320: JSTFY4   CPX    #EXTBUF   BUFFER OVERFLOW?
0489 2604        321:          BNE    JSTFY5
048B 860D        322:          LDAA   #$D       STUFF A C.R.
048D 20D4        323:          BRA    JSTFY1
048F DFCD        324: JSTFY5   STX    BUFPNT    SAVE BUF POINTER
0491 7E03F8      325:          JMP    PROC      REPEAT LOOP
0494 D6BD        326: JSTFY6   LDAB   PFLG      CHECK FLAG
0496 C103        327:          CMPB   #3
0498 2604        328:          BNE    JSTF63
049A 96BF        329:          LDAA   PCHAR     GET CHARACTER
049C 200B        330:          BRA    JSTF65
```

```
049E 8120      331: JSTF63 CMPA  #$20     IS CHAR = SPACE?
04A0 2751      332:        BEQ   ADJSPC
04A2 BD074C    333:        JSR   GETCHR  GET NEXT CHARACTER
04A5 8120      334:        CMPA  #$20    IS IT SPACE?
04A7 274A      335:        BEQ   ADJSPC
04A9 36        336: JSTF65 PSH   A       SAVE CHAR.
04AA 8620      337:        LDAA  #$20
04AC DED1      338:        LDX   BUFEND  GET TO END
04AE 09        339: JSTFY7 DEX
04AF 8C18C2    340:        CPX   #LINBUF-1   LOOK FOR SPACES
04B2 271C      341:        BEQ   JSTFY9
04B4 A100      342:        CMPA  0,X
04B6 26F6      343:        BNE   JSTFY7
04B8 08        344: JSTFY8 INX               BUMP POINTER
04B9 9CD1      345:        CPX   BUFEND
04BB 2713      346:        BEQ   JSTFY9
04BD A600      347:        LDAA  0,X         PICK UP CHARACTER
04BF DFDF      348:        STX   TEMP        SAVE X
04C1 DEE1      349:        LDX   TEMP2
04C3 A700      350:        STAA  0,X         MOVE THE CHAR.
04C5 08        351:        INX
04C6 DFE1      352:        STX   TEMP2
04C8 DEDF      353:        LDX   TEMP        RESTORE X
04CA 8620      354:        LDAA  #$20        SET WITH SPACE
04CC A700      355:        STAA  0,X         SAVE IT
04CE 20E8      356:        BRA   JSTFY8      REPEAT
04D0 32        357: JSTFY9 PUL   A           RESTORE CHARACTER
04D1 7F0076    358:        CLR   TABFLG      CLEAR TABS
04D4 CE0124    359:        LDX   #TABEND     POINT TO TABS
04D7 DF74      360:        STX   NXTTAB      SET NEXT TAB
04D9 DEE1      361:        LDX   TEMP2       RESTORE X
04DB BD0668    362: JSTF95 JSR   TSULN       TEST UL CHAR
04DE A700      363:        STAA  0,X         SAVE LAST CHAR.
04E0 08        364:        INX               BUMP POINTER
04E1 DFD5      365:        STX   EBFEND      SET END
04E3 8120      366:        CMPA  #$20        WAS CHAR A SPACE?
04E5 270C      367:        BEQ   ADJSPC
04E7 8C198B    368:        CPX   #LINBU2     BUFFER OVERFLOW?
04EA 2707      369:        BEQ   ADJSPC
04EC BD074C    370:        JSR   GETCHR      GET NEXT CHAR.
04EF DED5      371:        LDX   EBFEND      GET POINTER
04F1 20E8      372:        BRA   JSTF95
               373:
               374: * ADJUST BUFFER FOR SPACES
               375:
04F3 5F        376: ADJSPC CLR   B           CLEAR COUNT
04F4 CE18C3    377:        LDX   #LINBUF     POINT TO BUF BEGIN
04F7 DFDB      378:        STX   SPCPT1
04F9 A600      379: ADJSP2 LDAA  0,X         LOOK FOR SPACES
04FB 8120      380:        CMPA  #$20
04FD 2609      381:        BNE   ADJS35
04FF 5C        382:        INC   B           INC THE COUNTER
0500 08        383:        INX               BUMP POINTER
0501 9CD1      384:        CPX   BUFEND
0503 26F4      385:        BNE   ADJSP2
```

```
0505 7E05CC      386: ADJSP3 JMP    OUTLIN  OUTPUT LINE
0508 DFDD        387: ADJS35 STX    SECPT2  SET END
050A BD0686      388:        JSR    DELCHR  DELETE INIT. SPACES
050D CE18C3      389:        LDX    #LINBUF POINT TO BEGIN
0510 8620        390:        LDAA   #$20    CHECK MORE SPACES
0512 A100        391: ADJSP4 CMPA   0,X
0514 2707        392:        BEQ    ADJSP5
0516 08          393:        INX            BUMP TIL FIND
0517 9CD1        394:        CPX    BUFEND  END OF BUFFER?
0519 2710        395:        BEQ    ADJSP6
051B 20F5        396:        BRA    ADJSP4  REPEAT
051D 08          397: ADJSP5 INX            BUMP POINTER
051E 9CD1        398:        CPX    BUFEND  FINISHED?
0520 2605        399:        BNE    ADJS55
0522 7C00A9      400:        INC    SWRDF   SET SINGLE WORD
0525 2004        401:        BRA    ADJSP6
0527 A100        402: ADJS55 CMPA   0,X      CHECK NEXT CHAR.
0529 27F2        403:        BEQ    ADJSP5
052B D6C7        404: ADJSP6 LDAB   CNTFLG  CENTERING?
052D 2703        405:        BEQ    ADJSP7
052F 7E06E8      406:        JMP    CNTRIT  GO CENTER LINE
0532 D6C9        407: ADJSP7 LDAB   JUST    JUSTIFICATION?
0534 27CF        408:        BEQ    ADJSP3
0536 D6C3        409:        LDAB   RTJ     RIGHT HAND?
0538 2703        410:        BEQ    ADJSP8
053A 7E0676      411:        JMP    RIGHTJ  GO DO RIGHT
053D D6C1        412: ADJSP8 LDAB   CNJ     CENTER JUST.?
053F 2703        413:        BEQ    ADJSP9
0541 7E0681      414:        JMP    CENTJ   GO CENTER
0544 D6A9        415: ADJSP9 LDAB   SWRDF   CHECK SINGLE
0546 26BD        416:        BNE    ADJSP3
0548 D660        417:        LDAB   FLBF    FLUSHING BUFFER?
054A 26B9        418:        BNE    ADJSP3
054C D662        419:        LDAB   LEFT    WHICH SIDE
054E 273A        420:        BEQ    RINS    GO FROM RIGHT
                 421:
                 422: * INSERT SPACES FROM LEFT
                 423:
0550 CE18C3      424: LINS   LDX    #LINBUF SET POINTER
0553 DFDF        425:        STX    TEMP    SAVE IT
0555 DED1        426: LINS2  LDX    BUFEND  POINT TO END
0557 09          427:        DEX            DEC     THE POINTER
0558 A600        428:        LDAA   0,X     GET CHARACTER
055A 8120        429:        CMPA   #$20    IS IT A SPACE?
055C 26A7        430:        BNE    ADJSP3
055E DEDF        431:        LDX    TEMP    RESTORE POINTER
0560 A600        432: LINS3  LDAA   0,X     GET CHAR
0562 8120        433:        CMPA   #$20    IS IT SPACE?
0564 2707        434:        BEQ    LINS4
0566 08          435:        INX            BUMP POINTER
0567 9CD1        436:        CPX    BUFEND  END OF BUFFER
0569 27E5        437:        BEQ    LINS
056B 20F3        438:        BRA    LINS3   REPEAT
056D C601        439: LINS4  LDAB   #1      SET COUNT = 1
056F BD06AC      440:        JSR    INSSPC  GO INSERT SPACE
```

```
0572 D6C1        441:          LDAB   CNJ      CENTER JUST?
0574 2701        442:          BEQ    LINS5
0576 39          443:          RTS    RETURN
0577 DEDF        444: LINS5    LDX    TEMP     RESTORE POINTER
0579 A600        445: LINS6    LDAA   0,X      GET CHARACTER
057B 8120        446:          CMPA   #$20     IS IT SPACE?
057D 2607        447:          BNE    LINS7
057F 08          448:          INX             BUMP POINTER
0580 9CD1        449:          CPX    BUFEND   END OF BUFFER?
0582 27CC        450:          BEQ    LINS
0584 20F3        451:          BRA    LINS6
0586 DFDF        452: LINS7    STX    TEMP     SAVE X
0588 20CB        453:          BRA    LINS2    REPEAT
                 454:
                 455: * INSERT SPACES FROM RIGHT SIDE
                 456:
058A DED1        457: RINS     LDX    BUFEND   SET POINTER
058C 8620        458:          LDAA   #$20     SET UP SPACE
058E 09          459: RINS2    DEX
058F A100        460:          CMPA   0,X      IS CHAR A SPACE?
0591 27FB        461:          BEQ    RINS2
0593 DFDF        462:          STX    TEMP     SAVE POINTER
0595 DED1        463: RINS3    LDX    BUFEND   GO TO END
0597 09          464:          DEX
0598 A600        465:          LDAA   0,X      GET CHAR.
059A 8120        466:          CMPA   #$20     IS IT SPACE?
059C 262E        467:          BNE    OUTLIN
059E DEDF        468:          LDX    TEMP     RESTORE X
05A0 A600        469: RINS4    LDAA   0,X      GET CHAR
05A2 8120        470:          CMPA   #$20     IS IT SPACE?
05A4 2708        471:          BEQ    RINS5
05A6 09          472:          DEX    DEC      THE POINTER
05A7 8C18C2      473:          CPX    #LINBUF-1   FINISHED?
05AA 27DE        474:          BEQ    RINS
05AC 20F2        475:          BRA    RINS4    REPEAT
05AE C601        476: RINS5    LDAB   #1       SET COUNT = 1
05B0 BD06AC      477:          JSR    INSSPC   INSERT SPACE
05B3 D6C1        478:          LDAB   CNJ      CENTER JUST?
05B5 2701        479:          BEQ    RINS6
05B7 39          480:          RTS    RETURN
05B8 DEDF        481: RINS6    LDX    TEMP     RESTORE POINTER
05BA A600        482: RINS7    LDAA   0,X      GET CHARACTER
05BC 8120        483:          CMPA   #$20     SPACE?
05BE 2608        484:          BNE    RINS8
05C0 09          485:          DEX
05C1 8C18C2      486:          CPX    #LINBUF-1   FINISHED?
05C4 27C4        487:          BEQ    RINS
05C6 20F2        488:          BRA    RINS7    REPEAT
05C8 DFDF        489: RINS8    STX    TEMP     SAVE POINTER
05CA 20C9        490:          BRA    RINS3
                 491:
                 492: * OUTPUT LINE FROM WORK BUFFER
                 493:
05CC 7F00A9      494: OUTLIN   CLR    SWRDF    CLR FLAG
05CF D63E        495:          LDAB   LFM      LEFT MARGIN?
```

```
05D1 7D006F      496:          TST    PTFL     PUT IN INDENT?
05D4 2602        497:          BNE    OUTLI1
05D6 DB38        498:          ADD    B        ELSE ADJUST LEFT
05D8 7F006F      499: OUTLI1   CLR    PTFL
05DB DB70        500:          ADD    B        SIN ADD IN SINGLE IN.
05DD 2B0C        501:          BMI    OUTLI3
05DF 270A        502:          BEQ    OUTLI3
05E1 8620        503: OUTLI2   LDAA   #$20      GET OF SPACE
05E3 37          504:          PSH    B
05E4 BD1645      505:          JSR    OUTCHR   OUTPUT SPACE
05E7 33          506:          PUL    B
05E8 5A          507:          DEC    B        DEC COUNT
05E9 26F6        508:          BNE    OUTLI2
05EB D6BE        509: OUTLI3   LDAB   FILFLG   FILL MODE?
05ED 2711        510:          BEQ    OUTLI5
05EF 8620        511:          LDAA   #$20      SETUP SPACE
05F1 DED1        512:          LDX    BUFEND   GO TO END
05F3 8C18C3      513: OUTLI4   CPX    #LINBUF  EMPTY?
05F6 2719        514:          BEQ    OUTLI6
05F8 09          515:          DEX    DEC      THE POINTER
05F9 A100        516:          CMPA   0,X      IS IT SPACE?
05FB 27F6        517:          BEQ    OUTLI4
05FD 08          518:          INX             BUMP POINTER
05FE DF9D        519:          STX    STPOUT   SET END
0600 CE18C3      520: OUTLI5   LDX    #LINBUF
0603 9C9D        521:          CPX    STPOUT   EMPTY?
0605 270A        522:          BEQ    OUTLI6
0607 A600        523: OUTL55   LDAA   0,X      GET CHARACTER
0609 BD1645      524:          JSR    OUTCHR   OUTPUT IT
060C 08          525:          INX             BUMP POINTER
060D 9C9D        526:          CPX    STPOUT   FINISHED?
060F 26F6        527:          BNE    OUTL55
0611 5F          528: OUTLI6   CLR    B        CLEAR FLAGS
0612 D782        529:          STAB   DWFLG
0614 D764        530:          STAB   NOFL
0616 D7BD        531:          STAB   PFLG
0618 D768        532:          STAB   NSP
061A D770        533:          STAB   SIN
061C 730062      534:          COM    LEFT     SWITCH SP. SIDES
061F CE18C3      535:          LDX    #LINBUF  SET POINTER
0622 DFCD        536:          STX    BUFPNT
0624 CE0110      537:          LDX    #TABS    SET TABS
0627 DF74        538:          STX    NXTTAB
0629 BD15A5      539:          JSR    FIXWD    GO FIX WIDTH
062C CE195E      540:          LDX    #EXTBUF
062F 9CD5        541: OUTL75   CPX    EBFEND   CHECK FOR EXTRA?
0631 2717        542:          BEQ    OUTLI8
0633 A600        543:          LDAA   0,X      GET CHARACTER
0635 08          544:          INX
0636 DFDF        545:          STX    TEMP
0638 DECD        546:          LDX    BUFPNT   TRANSFER IT
063A A700        547:          STAA   0,X
063C 08          548:          INX             BUMP POINTER
063D 9CD1        549:          CPX    BUFEND   CHECK END
063F 2709        550:          BEQ    OUTLI8   OVERFLOW!
```

```
0641 DFCD      551:          STX    BUFPNT  SAVE IT
0643 DEDF      552:          LDX    TEMP
0645 7C0032    553:          INC    COLCNT  BUMP COLUMN COUNT
0648 20E5      554:          BRA    OUTL75  REPEAT
064A CE195E    555: OUTLI8   LDX    #EXTBUF FIX POINTER
064D DFD5      556:          STX    EBFEND
064F BD095C    557:          JSR    PCRLF   OUTPUT CR & LF
0652 96C5      558:          LDAA   MSC     MULTIPLE SPACE?
0654 270A      559:          BEQ    OUTL85
0656 4A        560: OUTL82   DEC    A
0657 2707      561:          BEQ    OUTL85
0659 36        562:          PSH    A       OUTPUT EXTRA SPACE
065A BD095C    563:          JSR    PCRLF
065D 32        564:          PUL    A
065E 20F6      565:          BRA    OUTL82
0660 9660      566: OUTL85   LDAA   FLBF    FLUSHING?
0662 2701      567:          BEQ    OUTLI9
0664 39        568:          RTS
0665 7E03F8    569: OUTLI9   JMP    PROC    GO PROCESS
               570:
               571: * TEST UNDERLINE CHARACTER
               572:
0668 7D004E    573: TSULN    TST    ULFLG   TEST FLAG
066B 2708      574:          BEQ    TSULN2
066D BD12B5    575:          JSR    CLSFY   CLASS CHARACTER
0670 5D        576:          TST    B
0671 2702      577:          BEQ    TSULN2
0673 8A80      578:          ORAA   #$80    SET PARITY
0675 39        579: TSULN2   RTS    RETURN
               580:
               581: * RIGHT HAND JUSTIFY
               582:
0676 BD06DA    583: RIGHTJ   JSR    CNTSPC  COUNT SPACES
0679 CE18C2    584: RIGHT2   LDX    #LINBUF-1
067C 8D2E      585:          BSR    INSSPC  INSERT SPACES
067E 7E05CC    586:          JMP    OUTLIN  OUTPUT LINE
               587:
               588: * CENTER JUSTIFY
               589:
0681 8D57      590: CENTJ    BSR    CNTSPC  COUNT SPACES
0683 57        591:          ASR    B       DIVIDE BY 2
0684 20F3      592:          BRA    RIGHT2
               593:
               594: * DELETE CHARACTERS
               595:
0686 DEDD      596: DELCHR   LDX    SPCPT2  GET POINTER
0688 9CDB      597:          CPX    SPCPT1  EMPTY?
068A 271F      598:          BEQ    DELCH4
068C 9CD1      599:          CPX    BUFEND
068E 270E      600:          BEQ    DELCH3
0690 A600      601:          LDAA   0,X     GET CHARACTER
0692 08        602:          INX            BUMP THE POINTER
0693 DFDD      603:          STX    SPCPT2  SAVE IT
0695 DEDB      604:          LDX    SPCPT1  RESTORE
0697 A700      605:          STAA   0,X     SAVE CHARACTER
```

```
0699 08          606:          INX     BUMP    POINTER
069A DFDB        607:          STX     SPCPT1
069C 20E8        608:          BRA     DELCHR  REPEAT
069E DEDB        609: DELCH3   LDX     SPCPT1  GET POINTER
06A0 8620        610:          LDAA    #$20    SETUP SPACE
06A2 9CD1        611: DELC35   CPX     BUFEND
06A4 2705        612:          BEQ     DELCH4
06A6 A700        613:          STAA    0,X     PUT IN SPACE
06A8 08          614:          INX     BUMP    POINTER
06A9 20F7        615:          BRA     DELC35
06AB 39          616: DELCH4   RTS
                 617:
                 618: * INSERT SPACES
                 619:
06AC 5D          620: INSSPC   TST     B       TEST COUNT
06AD 272A        621:          BEQ     INSSP5  IF NONE, RETURN
06AF 37          622:          PSH     B       SAVE COUNT
06B0 DFDF        623:          STX     TEMP    SAVE X
06B2 DED1        624:          LDX     BUFEND  POINT TO END
06B4 DFDB        625:          STX     SPCPT1  SAVE
06B6 08          626: INSSP2   INX
06B7 5A          627:          DEC     B       DEC THE COUNT
06B8 26FC        628:          BNE     INSSP2
06BA DFDD        629:          STX     SPCPT2  SAVE POINTER
06BC DEDB        630: INSSP3   LDX     SPCPT1
06BE 9CDF        631:          CPX     TEMP    FINISHED?
06C0 270E        632:          BEQ     INSSP4
06C2 A600        633:          LDAA    0,X     GET CHARACTER
06C4 09          634:          DEX     DEC     THE POINTER
06C5 DFDB        635:          STX     SPCPT1  SAVE IT
06C7 DEDD        636:          LDX     SPCPT2
06C9 A700        637:          STAA    0,X     PUT CHARACTER
06CB 09          638:          DEX
06CC DFDD        639:          STX     SPCPT2
06CE 20EC        640:          BRA     INSSP3  REPEAT
06D0 33          641: INSSP4   PUL     B       RESTORE COUNT
06D1 8620        642:          LDAA    #$20    SETUP SPACE
06D3 08          643: INSS44   INX     BUMP    THE POINTER
06D4 A700        644:          STAA    0,X     STUFF SPACE
06D6 5A          645:          DEC     B       DEC THE COUNT
06D7 26FA        646:          BNE     INSS44
06D9 39          647: INSSP5   RTS     RETURN
                 648:
                 649: * COUNT SPACES
                 650:
06DA 5F          651: CNTSPC   CLR     B       CLEAR COUNT
06DB 8620        652:          LDAA    #$20    SETUP SPACE
06DD DED1        653:          LDX     BUFEND  SET POINTER
06DF 09          654: CNTSP2   DEX
06E0 A100        655:          CMPA    0,X     SPACE?
06E2 2603        656:          BNE     CNTSP3
06E4 5C          657:          INC     B       BUMP THE COUNT
06E5 20F8        658:          BRA     CNTSP2
06E7 39          659: CNTSP3   RTS
                 660:
```

```
                    661: * CENTER LINE
                    662:
06E8 8DF0           663: CNTRIT BSR     CNTSPC  GO COUNT SPACES
06EA 9682           664:        LDAA    DWFLG   DOUBLE WIDTH?
06EC 270E           665:        BEQ     CNTRI4
06EE 96B9           666:        LDAA    WIDTH   GET WIDTH
06F0 10             667:        SBA
06F1 48             668:        ASL     A       FIX FOR DOUBLE
06F2 91B9           669:        CMPA    WIDTH
06F4 220C           670:        BHI     CNTRI5
06F6 16             671:        TAB     SAVE
06F7 96B9           672:        LDAA    WIDTH
06F9 10             673:        SBA     SUB     FROM WIDTH
06FA 16             674:        TAB
06FB 57             675:        ASR     B       DIVIDE BY TWO
06FC 57             676: CNTRI4 ASR     B
06FD CE18C2         677:        LDX     #LINBUF-1   SET POINTER
0700 8DAA           678:        BSR     INSSPC  GO INSERT SPACE
0702 7A00C7         679: CNTRI5 DEC     CNTFLG  DEC CENTER COUNT
0705 260B           680:        BNE     CNTRI6
0707 4F             681:        CLR     A
0708 97C7           682:        STAA    CNTFLG  CLEAR FLAG
070A 9663           683:        LDAA    TFILF   GET TEMP FILL
070C 97BB           684:        STAA    FILFLG  SET FILL
070E DED1           685:        LDX     BUFEND  SET POINTER
0710 DF9D           686:        STX     STPOUT  SET END
0712 7E05CC         687: CNTRI6 JMP     OUTLIN  OUTPUT LINE
                    688:
                    689: * FIX BUFFER END POINTER
                    690:
0715 CE18C3         691: FIXBFE LDX     #LINBUF SET POINTER
0718 DFD1           692:        STX     BUFEND
071A 963B           693:        LDAA    LLN     GET LINE LENGTH
071C 90B9           694:        SUB     A       WIDTH CALC. COLUMN NUM.
071E 4C             695:        INC     A
071F 9732           696:        STAA    COLCNT  SAVE COUNT
0721 5F             697:        CLR     B
0722 96B9           698:        LDAA    WIDTH   GET WIDTH
0724 9BD2           699:        ADD     A       BUFEND+1 ADD TO BUFEND
0726 D9D1           700:        ADC     B       BUFEND
0728 97D2           701:        STAA    BUFEND+1    SAVE RESULT
072A D7D1           702:        STAB    BUFEND
072C 39             703:        RTS     RETURN
                    704:
                    705: * RETURN FROM MACRO
                    706:
072D 7F007C         707: RETMAC CLR     FINMAC  CLEAR FLAG
0730 32             708:        PUL     A       FIX STACK
0731 32             709:        PUL     A
0732 32             710:        PUL     A
0733 97C7           711:        STAA    CNTFLG  RESTORE FLAG
0735 CE0055         712:        LDX     #NUMPNT
0738 32             713: RETMA2 PUL     A       RESTORE VALUES
0739 A700           714:        STAA    0,X
073B 08             715:        INX
```

```
073C 8C0062      716:          CPX     #LEFT    FINISHED?
073F 26F7        717:          BNE     RETMA2
0741 7A007F      718:          DEC     MACCNT   DEC MACRO COUNTER
0744 9661        719:          LDAA    ATFLG    DOING AT?
0746 270B        720:          BEQ     GETCH1
0748 39          721:          RTS     RETURN
                 722:
                 723: * CLEAR 'ENDLIN' AND GET CHARACTER
                 724:
0749 7F00B2      725: CLRGET CLR     ENDLIN
                 726:
                 727: * GET NEXT CHARACTER
                 728:
074C BD1606      729: GETCHR JSR     TSTBRK   TEST FOR BREAK
074F 967C        730:          LDAA    FINMAC   FINISH MACRO?
0751 26DA        731:          BNE     RETMAC
0753 9657        732: GETCH1 LDAA    EXCHR    GET EXTRA CHAR.
0755 2703        733:          BEQ     GETCH2
0757 7E1285      734:          JMP     FTCHNM   GET NUMBER
075A 965A        735: GETCH2 LDAA    CMFLG    COMMAND?
075C 270D        736:          BEQ     GETCH3
075E DED9        737: GETC22 LDX     CMNPNT   SET POINTER
0760 A600        738:          LDAA    0,X      GET CHARACTER
0762 810D        739:          CMPA    #$D      C.R.?
0764 2701        740:          BEQ     GETC25
0766 08          741:          INX     BUMP     THE POINTER
0767 DFD9        742: GETC25 STX     CMNPNT   SAVE IT
0769 2029        743:          BRA     FETCHR
076B 9693        744: GETCH3 LDAA    SBFLG    SPECIAL BUFFER?
076D 26EF        745:          BNE     GETC22
076F 965B        746:          LDAA    MBFLG    MACRO BUFFER?
0771 2708        747:          BEQ     GETCH4
0773 BD0FB2      748:          JSR     INMAC    GET CHARACTER
0776 261C        749:          BNE     FETCHR
0778 7E104C      750:          JMP     MCEND    FINISH MACRO
077B 9684        751: GETCH4 LDAA    SPIFLG   SPECIAL INPUT?
077D 2705        752:          BEQ     GETCH5
077F BD0206      753:          JSR     INCH     GET CHARACTER
0782 2010        754:          BRA     FETCHR
0784 9683        755: GETCH5 LDAA    DFMFLG   DEFINE MACRO?
0786 9A5E        756:          ORA     A        NOCR
0788 2607        757:          BNE     GETCH6
078A 9676        758:          LDAA    TABFLG   TABS?
078C 2703        759:          BEQ     GETCH6
078E 7E0BF0      760:          JMP     DOTAB    GO DO TAB
0791 BD166F      761: GETCH6 JSR     INCHR    GET CHARACTER
                 762:
                 763: * FETCH AND CHECK CHARACTER
                 764:
0794 811A        765: FETCHR CMPA    #$1A     END OF FILE?
0796 2605        766:          BNE     FETCH2
0798 975F        767:          STAA    DONE     SET FLAG
079A 7E0A08      768:          JMP     FINISH
079D 810D        769: FETCH2 CMPA    #$D      C.R.?
079F 262C        770:          BNE     FETCH3
```

```
07A1 7F006B      771:          CLR    SPSPF   SPECIAL SPACE?
07A4 D65E        772:          LDAB   NOCR
07A6 2635        773:          BNE    FETC35
07A8 D693        774:          LDAB   SBFLG   CHECK FLAG
07AA 2607        775:          BNE    FETC22
07AC D6B2        776:          LDAB   ENDLIN  END OF LINE?
07AE 2703        777:          BEQ    FETC22
07B0 BD09E1      778:          JSR    FLUSHB  FLUSH BUFFER
07B3 97B2        779: FETC22   STAA   ENDLIN  SET FLAGS
07B5 7F0093      780:          CLR    SBFLG
07B8 7F004E      781:          CLR    ULFLG
07BB D683        782:          LDAB   DFMFLG  TEST
07BD DAC7        783:          ORA    B       CNTFLG
07BF 2608        784:          BNE    FETC25
07C1 D6BB        785:          LDAB   FILFLG  TEST FILL
07C3 2704        786:          BEQ    FETC25
07C5 8620        787:          LDAA   #$20    SETUP SPACE
07C7 2019        788:          BRA    FETC36
07C9 860D        789: FETC25   LDAA   #$D     SETUP C.R.
07CB 2015        790:          BRA    FETC36
07CD D683        791: FETCH3   LDAB   DFMFLG  GET FLAG
07CF DA80        792:          ORA    B       PASFLG
07D1 260A        793:          BNE    FETC35
07D3 D66A        794:          LDAB   PASCHR  PASS CHAR?
07D5 270F        795:          BEQ    FETCH4
07D7 8120        796:          CMPA   #$20    IS IT A SPACE?
07D9 2602        797:          BNE    FETC35
07DB 8A80        798:          ORA    A       #$80 SET PARITY
07DD 5F          799: FETC35   CLR    B       CLEAR FLAGS
07DE D7B2        800:          STAB   ENDLIN
07E0 D76A        801:          STAB   PASCHR
07E2 7F00AA      802: FETC36   CLR    CAP
07E5 39          803:          RTS    RETURN
07E6 811F        804: FETCH4   CMPA   #$1F    CHECK CHAR
07E8 2203        805:          BHI    FETC45
07EA 7E074C      806:          JMP    GETCHR  GO GET CHAR.
07ED D6B2        807: FETC45   LDAB   ENDLIN  END OF LINE?
07EF 271A        808:          BEQ    FETCH5
07F1 812E        809:          CMPA   #'.     PERIOD?
07F3 2706        810:          BEQ    FETC47
07F5 813A        811:          CMPA   #':     COLON?
07F7 2605        812:          BNE    FETC48
07F9 9764        813:          STAA   NOFL    SET NO FLUSH
07FB 7E08A2      814: FETC47   JMP    COMAND  DO COMMAND
07FE 8120        815: FETC48   CMPA   #$20    SPACE?
0800 2609        816:          BNE    FETCH5
0802 976B        817:          STAA   SPSPF   SET FLAG
0804 BD09E1      818:          JSR    FLUSHB  FLUSH BUFFER
0807 86A0        819: FETC49   LDAA   #$A0
0809 20D2        820:          BRA    FETC35
080B D66B        821: FETCH5   LDAB   SPSPF   TEST FLAG
080D 2707        822:          BEQ    FETC55
080F 8120        823:          CMPA   #$20    IS IT SPACE?
0811 27F4        824:          BEQ    FETC49
0813 7F006B      825:          CLR    SPSPF   CLEAR OUT
```

```
0816 D65A        826: FETC55 LDAB   CMFLG   COMMAND?
0818 DA5B        827:        ORA    B       NBFLG
081A DA81        828:        ORA    B       NONUMS
081C DA84        829:        ORA    B       SPIFLG
081E DA93        830:        ORA    B       SBFLG
0820 261C        831:        BNE    FETCH6
0822 91B3        832:        CMPA   TAB     CHECK IF TAB
0824 2618        833:        BNE    FETCH6
0826 DE74        834:        LDX    NXTTAB  GET NEXT TAB
0828 D632        835:        LDAB   COLCNT  GET COUNT
082A 6D00        836: FETC57 TST    0,X     CHECK
082C 27AF        837:        BEQ    FETC35
082E E100        838:        CMPB   0,X     FINISHED?
0830 2503        839:        BCS    FETC58
0832 08          840:        INX    BUMP    THE POINTER
0833 20F5        841:        BRA    FETC57
0835 DF74        842: FETC58 STX    NXTTAB  SAVE POINTER
0837 96B4        843:        LDAA   TFILL
0839 9776        844:        STAA   TABFLG  SET FLAG
083B 7E07DD      845: FETC59 JMP    FETC35
083E D681        846: FETCH6 LDAB   NONUMS  NUMBERS?
0840 261D        847:        BNE    FETCH7
0842 8123        848:        CMPA   #'#     POUND SIGN?
0844 2704        849:        BEQ    FETC65
0846 8125        850:        CMPA   #'%     PERCENT SIGN?
0848 2615        851:        BNE    FETCH7
084A D65E        852: FETC65 LDAB   NOCR    DO C.R.?
084C 37          853:        PSH    B
084D 975E        854:        STAA   NOCR    SAVE VALUES
084F 9773        855:        STAA   NOEXP
0851 BD126D      856:        JSR    CLRNUM  CLEAR NUMBER
0854 BD1357      857:        JSR    PRNU32  PROCESS NUMBER
0857 33          858:        PUL    B
0858 D75E        859:        STAB   NOCR    RESTORE VALUES
085A 24DF        860:        BCC    FETC59
085C 7E074C      861:        JMP    GETCHR  GET CHARACTER
085F 815C        862: FETCH7 CMPA   #'\     BACK SLASH?
0861 2605        863:        BNE    FETC75
0863 976A        864:        STAA   PASCHR  SET PASS CHAR.
0865 7E0749      865:        JMP    CLRGET  GO GET IT
0868 8140        866: FETC75 CMPA   #'@     AT SIGN?
086A 271C        867:        BEQ    CAPIT
086C 815E        868:        CMPA   #$5E    UP ARROW?
086E 2721        869:        BEQ    SETCAP
0870 D6AA        870:        LDAB   CAP     CHECK MODE
0872 DAAB        871:        ORA    B       SCAP
0874 DA5A        872:        ORA    B       CMFLG
0876 26C3        873:        BNE    FETC59
0878 8141        874:        CMPA   #'A     CHECK IF LETTER
087A 25BF        875:        BCS    FETC59
087C 815A        876:        CMPA   #'Z
087E 22BB        877:        BHI    FETC59
0880 D66C        878:        LDAB   DOCAP   DO CAP?
0882 27B7        879:        BEQ    FETC59
0884 8B20        880:        ADD    A       #$20 FORCE TO LOWER
```

```
0886 20B3       881: FETCH8 BRA    FETC59
                882:
                883: * CAP SINGLE LETTER
                884:
0888 D66C       885: CAPIT  LDAB   DOCAP   CHECK MODE
088A 27FA       886:        BEQ    FETCH8
088C 97AA       887:        STAA   CAP     SET FLAG
088E 7E0749     888: CAPIT2 JMP    CLRGET
                889:
                890: * CAP STRING OF LETTERS
                891:
0891 D66C       892: SETCAP LDAB   DOCAP   CHECK MODE
0893 27F1       893:        BEQ    FETCH8
0895 D6AB       894:        LDAB   SCAP    GET FLAG
0897 2705       895:        BEQ    SETCA2
0899 7F00AB     896:        CLR    SCAP    CLEAR IT
089C 20F0       897:        BRA    CAPIT2
089E 97AB       898: SETCA2 STAA   SCAP    SET FOR STRING
08A0 20EC       899:        BRA    CAPIT2
```

```
                     1:
                     2: * COMMAND PROCESSOR
                     3:
08A2 7F00B2          4: COMAND CLR     ENDLIN  CLEAR FLAG
08A5 CE1AE6          5:        LDX     #CMNLBF-1   SET POINTER
08A8 08              6: COMAN2 INX     BUMP    IT
08A9 7C005E          7:        INC     NOCR    SET NO C.R.
08AC DFE5            8:        STX     TEMP6   SAVE POINTER
08AE 7C0081          9:        INC     NONUMS
08B1 BD074C         10:        JSR     GETCHR  GET CHARACTER
08B4 DEE5           11:        LDX     TEMP6   RESTORE POINTER
08B6 7F005E         12:        CLR     NOCR    CLEAR FLAG
08B9 7F0081         13:        CLR     NONUMS
08BC A700           14:        STAA    0,X     PUT CHARACTER
08BE 810D           15:        CMPA    #$D     WAS IT A C.R.?
08C0 26E6           16:        BNE     COMAN2
08C2 7F00B2         17:        CLR     ENDLIN  RESET END LINE
08C5 CE1AE9         18:        LDX     #CMNDBF SET POINTER
08C8 A600           19: COMAN3 LDAA    0,X     GET CHARACTER
08CA 08             20:        INX     BUMP    THE POINTER
08CB E600           21:        LDAB    0,X     GET NEXT CHAR
08CD 08             22:        INX     BUMP
08CE DFD9           23:        STX     CMNPNT  SAVE THE POINTER
08D0 815F           24:        CMPA    #$5F    LOWER CASE?
08D2 2304           25:        BLS     COMAN4
08D4 8020           26:        SUB     A       #$20 SET TO UPPER
08D6 C020           27:        SUB     B       #$20
08D8 CE0A20         28: COMAN4 LDX     #CMNDT  POINT TO TABLE
08DB A100           29: COMAN5 CMPA    0,X     COMPARE FIRST
08DD 260C           30:        BNE     COMAN7
08DF E101           31:        CMPB    1,X     COMPARE SECOND
08E1 2608           32:        BNE     COMAN7
08E3 975A           33:        STAA    CMFLG   FOUND COMMAND
08E5 EE02           34:        LDX     2,X     GET ADDRESS
08E7 AD00           35:        JSR     0,X     GO DO IT
08E9 202F           36:        BRA     FINCM   FINISH COMMAND
08EB 08             37: COMAN7 INX     BUMP    POINTER
08EC 08             38:        INX
08ED 08             39:        INX
08EE 08             40:        INX
08EF 8C0B1C         41:        CPX     #TBLEND TABLE END?
08F2 26E7           42:        BNE     COMAN5
08F4 36             43:        PSH     A
08F5 967F           44:        LDAA    MACCNT  TEST MACRO NUMBER
08F7 8107           45:        CMPA    #7
08F9 32             46:        PUL     A
08FA 2415           47:        BCC     MACOVF  OVERFLOW?
08FC CE1D9D         48:        LDX     #MACTBL POINT TO MACROS
08FF 9CED           49: COMAN8 CPX     MACEND  END?
0901 2717           50:        BEQ     FINCM
0903 A100           51:        CMPA    0,X     COMPARE FIRST
0905 2604           52:        BNE     COMAN9
0907 E101           53:        CMPB    1,X     COMPARE SECOND
0909 2729           54:        BEQ     CALMAC
090B 08             55: COMAN9 INX     FIND    NEXT ENTRY
```

```
090C 08          56:         INX
090D 08          57:         INX
090E 08          58:         INX
090F 20EE        59:         BRA     COMAN8
                 60:
                 61: * MACRO OVERFLOW ERROR
                 62:
0911 CE181B      63: MACOVF  LDX     #OVFSTR POINT TO STRING
0914 BD1636      64:         JSR     PSTRNG  OUTPUT IT
0917 7E0209      65:         JMP     MON
                 66:
                 67: * FINISH COMMAND
                 68:
091A 967E        69: FINCM   LDAA    IFFLG   CHECK FOR IF
091C 270A        70:         BEQ     FINCM1
091E 4F          71:         CLR     A       CLEAR FLAGS
091F 97B2        72:         STAA    ENDLIN
0921 975A        73:         STAA    CMFLG
0923 977E        74:         STAA    IFFLG
0925 7E08C8      75:         JMP     COMAN3  GO DO COMMAND
0928 7F0064      76: FINCM1  CLR     NOFL    CLEAR FLAGS
092B 7F005A      77:         CLR     CMFLG
092E 7C00B2      78:         INC     ENDLIN  SET END LINE
0931 7E074C      79:         JMP     GETCHR  GO GET CHARACTER
                 80:
                 81: * CALL MACRO
                 82:
0934 DFA3        83: CALMAC  STX     XMAC    SAVE POINTER
0936 CE0061      84:         LDX     #ATFLG  POINT TO VALUES
0939 A600        85: CALMA2  LDAA    0,X     GET VALUE
093B 36          86:         PSH     A       PUT ON STACK
093C 6F00        87:         CLR     0,X     CLEAR IT
093E 09          88:         DEX
093F 8C0054      89:         CPX     #INC    FINISHED?
0942 26F5        90:         BNE     CALMA2
0944 96C7        91:         LDAA    CNTFLG  SAVE CNT FLAG
0946 36          92:         PSH     A
0947 7F00C7      93:         CLR     CNTFLG
094A 7C007F      94:         INC     MACCNT  BUMP COUNTER
094D DEA3        95:         LDX     XMAC    RESTORE COUNT
094F 860F        96:         LDAA    #$F
0951 975B        97:         STAA    MBFLG   SET FLAG
0953 97B2        98:         STAA    ENDLIN
0955 EE02        99:         LDX     2,X     GET ADDRESS
0957 DF5C       100:         STX     MBFPNT  SAVE AS POINTER
0959 7E03F8     101:         JMP     PROC    GO PROCESS
                102:
                103: * PRINT C.R. AND L.F.
                104:
095C BD14BC     105: PCRLF  JSR     PUSHX   SAVE X
095F 8D07       106:        BSR     SCRLF   DO CR AND LF
0961 BD1606     107:        JSR     TSTBRK  BREAK?
0964 BD14CE     108:        JSR     PULLX   RESTORE X
0967 39         109:        RTS     RETURN
                110:
```

```
                    111: * SPECIAL CARRIAGE RETURN LINE FEED
                    112:
0968 CE17E5         113: SCRLF   LDX    #CRLFST POINT TO STRING
096B A600           114: SCRLF1  LDAA   0,X     GET CHARACTER
096D 8104           115:         CMPA   #4       IS IT 4?
096F 2706           116:         BEQ    SCRLF3
0971 BD1645         117:         JSR    OUTCHR  OUTPUT CHAR.
0974 08             118:         INX    BUMP    POINTER
0975 20F4           119:         BRA    SCRLF1
0977 966D           120: SCRLF3  LDAA   NOOUT   DO OUTPUT?
0979 2618           121:         BNE    SCRLF4
097B 9690           122:         LDAA   TLPP    LINES PER PAGE?
097D 2714           123:         BEQ    SCRLF4
097F 7C006E         124:         INC    TOUTL   BUMP LINE COUNT
0982 916E           125:         CMPA   TOUTL   MAX?
0984 220D           126:         BHI    SCRLF4
0986 7F006E         127:         CLR    TOUTL   CLEAR COUNT
0989 BD0206         128:         JSR    INCH    WAIT FOR CHARACTER
098C 810D           129:         CMPA   #$D     IS IT C.R.?
098E 2603           130:         BNE    SCRLF4
0990 7E0209         131:         JMP    MON     EXIT PROCESSOR
0993 9685           132: SCRLF4  LDAA   DIVFLG  DIVERTING?
0995 2643           133:         BNE    SCRLF9
0997 7C003D         134:         INC    LINCNT  BUMP LINE COUNTER
099A 963D           135: SCRLF5  LDAA   LINCNT
099C CE1AB7         136:         LDX    #TRAPS  POINT TO TRAPS
099F A100           137: SCRL55  CMPA   0,X     LINE = TRAP?
09A1 2724           138:         BEQ    SCRLF8
09A3 08             139:         INX    GET     TO NEXT
09A4 08             140:         INX
09A5 08             141:         INX
09A6 8C1AE7         142:         CPX    #TRPEND END?
09A9 26F4           143:         BNE    SCRL55
09AB 913F           144:         CMPA   PGL     BOTTOM OF PAGE?
09AD 232B           145:         BLS    SCRLF9
09AF 9653           146:         LDAA   NPGN    GET NEW PAGE NUM.
09B1 2707           147:         BEQ    SCRLF7
09B3 7F0053         148:         CLR    NPGN
09B6 9769           149:         STAA   PGN     SET PAGE NUMBER
09B8 2003           150:         BRA    SCRL75
09BA 7C0069         151: SCRLF7  INC    PGN     BUMP BY ONE
09BD 8601           152: SCRL75  LDAA   #1      SET UP 1
09BF 973D           153:         STAA   LINCNT  SET LINE COUNT
09C1 96A8           154:         LDAA   SUPL    CHECK FLAG
09C3 2615           155:         BNE    SCRLF9
09C5 20D3           156:         BRA    SCRLF5
09C7 7C0061         157: SCRLF8  INC    ATFLG   BUMP AT COUNT
09CA 96B2           158:         LDAA   ENDLIN  SAVE STATUS
09CC 36             159:         PSH    A
09CD A601           160:         LDAA   1,X     GET NAME
09CF E602           161:         LDAB   2,X
09D1 BD08D8         162:         JSR    COMAN4  GO PROCESS
09D4 7A0061         163:         DEC    ATFLG   DEC COUNT
09D7 32             164:         PUL    A
09D8 97B2           165:         STAA   ENDLIN  RESTORE STATUS
```

```
09DA 39          166: SCRLF9 RTS     RETURN
                 167:
                 168: * BREAK FILLED BUFFER
                 169:
09DB 8601        170: BRAK   LDAA    #1       SETUP 1
09DD 913D        171:        CMPA    LINCNT   TEST LINE COUNT
09DF 27E9        172:        BEQ     SCRLF5
                 173:
                 174: * FLUSH WORK BUFFER
                 175:
09E1 9664        176: FLUSHB LDAA    NOFL     NO FLUSH?
09E3 261F        177:        BNE     FLUSH5
09E5 8620        178: FLUSH  LDAA    #$20     SET UP SPACE
09E7 DECD        179:        LDX     BUFPNT   SET POINTER
09E9 8C18C3      180:        CPX     #LINBUF  BEGINNING OF BUFFER?
09EC 2716        181:        BEQ     FLUSH5
09EE DF9D        182:        STX     STPOUT   SET END
09F0 9CD1        183: FLUSH2 CPX     BUFEND   END?
09F2 2705        184:        BEQ     FLUSH3
09F4 A700        185:        STAA    0,X      SAVE CHARACTER
09F6 08          186:        INX     BUMP     POINTER
09F7 20F7        187:        BRA     FLUSH2
09F9 CE18C3      188: FLUSH3 LDX     #LINBUF  POINT TO BUFFER
09FC 9760        189:        STAA    FLBF     SET FLAG
09FE BD04F3      190:        JSR     ADJSPC   ADJUST SPACE
0A01 7F0060      191:        CLR     FLBF
0A04 7F0064      192: FLUSH5 CLR     NOFL     CLEAR FLAG
0A07 39          193:        RTS     RETURN
                 194:
                 195: * FINISH AND CLEAN UP
                 196:
0A08 8DD7        197: FINISH BSR     FLUSHB   FLUSH BUFFER
0A0A CE1C51      198:        LDX     #TFCB    POINT TO FCB
0A0D 8606        199:        LDAA    #6       SET FOR CLOSE
0A0F A700        200:        STAA    0,X
0A11 BD16E4      201:        JSR     DODFM    CALL DFM
0A14 7E039A      202:        JMP     DPROC2
0A17 7C00A8      203: FINIS4 INC     SUPL
0A1A BD0B47      204:        JSR     PAGE     GO PAGE
0A1D 7E0209      205:        JMP     MON      EXIT
```

```
                     1: * COMMAND TABLE
                     2:
0A20 5350            3: CMNDT  FCC     'SP'
0A22 0B24            4:        FDB     SPACE
0A24 5047            5:        FCC     'PG'
0A26 0B47            6:        FDB     PAGE
0A28 4D53            7:        FCC     'MS'
0A2A 0B6B            8:        FDB     MULTS
0A2C 5353            9:        FCC     'SS'
0A2E 0B79           10:        FDB     SNGLS
0A30 4E4A           11:        FCC     'NJ'
0A32 0B7D           12:        FDB     NOJST
0A34 4A55           13:        FCC     'JU'
0A36 0B81           14:        FDB     JST
0A38 4448           15:        FCC     'DH'
0A3A 0CDA           16:        FDB     DUBH
0A3C 4457           17:        FCC     'DW'
0A3E 0CE6           18:        FDB     DUBW
0A40 4442           19:        FCC     'DB'
0A42 0CF0           20:        FDB     DUBB
0A44 4345           21:        FCC     'CE'
0A46 0CFD           22:        FDB     CENTER
0A48 4252           23:        FCC     'BR'
0A4A 09DB           24:        FDB     BRAK
0A4C 2A20           25:        FCC     '* '
0A4E 0B46           26:        FDB     SPACE6
0A50 4649           27:        FCC     'FI'
0A52 0D8B           28:        FDB     FILL
0A54 4E46           29:        FCC     'NF'
0A56 0D84           30:        FDB     NOFILL
0A58 5349           31:        FCC     'SI'
0A5A 0D32           32:        FDB     SIND
0A5C 5049           33:        FCC     'PI'
0A5E 0D4F           34:        FDB     PTIND
0A60 504E           35:        FCC     'PN'
0A62 0CCD           36:        FDB     PGNUM
0A64 4C4D           37:        FCC     'LM'
0A66 0BAB           38:        FDB     LEFTM
0A68 494E           39:        FCC     'IN'
0A6A 0BBB           40:        FDB     INDNT
0A6C 4C4E           41:        FCC     'LN'
0A6E 0BD1           42:        FDB     LENTH
0A70 4E53           43:        FCC     'NS'
0A72 0C03           44:        FDB     NOSPC
0A74 5253           45:        FCC     'RS'
0A76 0C06           46:        FDB     RESPC
0A78 504C           47:        FCC     'PL'
0A7A 0D1D           48:        FDB     PAGEL
0A7C 4350           49:        FCC     'CP'
0A7E 0D46           50:        FDB     STCAP
0A80 4E43           51:        FCC     'NC'
0A82 0D4B           52:        FDB     NOCAP
0A84 4E4C           53:        FCC     'NL'
0A86 0DB5           54:        FDB     NEDL
0A88 5356           55:        FCC     'SV'
```

```
0A8A 0DFC        56:        FDB    SAVS
0A8C 4F53        57:        FCC    'OS'
0A8E 0E18        58:        FDB    OUTSV
0A90 4154        59:        FCC    'AT'
0A92 0E21        60:        FDB    ATL
0A94 444D        61:        FCC    'DM'
0A96 0E6D        62:        FDB    DEFMAC
0A98 414D        63:        FCC    'AM'
0A9A 0EAB        64:        FDB    APMAC
0A9C 524D        65:        FCC    'RM'
0A9E 0EB4        66:        FDB    REMMAC
0AA0 4449        67:        FCC    'DI'
0AA2 0EF4        68:        FDB    DIVERT
0AA4 4441        69:        FCC    'DA'
0AA6 0F13        70:        FDB    DIVAPP
0AA8 5354        71:        FCC    'ST'
0AAA 0D9E        72:        FDB    STOP
0AAC 544C        73:        FCC    'TL'
0AAE 1090        74:        FDB    TITLE
0AB0 4C54        75:        FCC    'LT'
0AB2 1083        76:        FDB    TLEN
0AB4 4348        77:        FCC    'CH'
0AB6 11E1        78:        FDB    CHNG
0AB8 4946        79:        FCC    'IF'
0ABA 1174        80:        FDB    IF
0ABC 4E52        81:        FCC    'NR'
0ABE 1234        82:        FDB    NREG
0AC0 4152        83:        FCC    'AR'
0AC2 1253        84:        FDB    ARB
0AC4 5352        85:        FCC    'SR'
0AC6 1257        86:        FDB    SROM
0AC8 4352        87:        FCC    'CR'
0ACA 125C        88:        FDB    CROM
0ACC 4155        89:        FCC    'AU'
0ACE 1260        90:        FDB    SAUTO
0AD0 5443        91:        FCC    'TC'
0AD2 0C0A        92:        FDB    TABCH
0AD4 5446        93:        FCC    'TF'
0AD6 0C15        94:        FDB    TABFIL
0AD8 5441        95:        FCC    'TA'
0ADA 0C21        96:        FDB    STAB
0ADC 4558        97:        FCC    'EX'
0ADE 0A08        98:        FDB    FINISH
0AE0 544D        99:        FCC    'TM'
0AE2 0C40       100:        FDB    TERM
0AE4 4749       101:        FCC    'GI'
0AE6 0C55       102:        FDB    GETIN
0AE8 4556       103:        FCC    'EV'
0AEA 0C63       104:        FDB    SENV
0AEC 5250       105:        FCC    'RP'
0AEE 0D92       106:        FDB    RPT
0AF0 5053       107:        FCC    'PS'
0AF2 0B1D       108:        FDB    PASS
0AF4 554C       109:        FCC    'UL'
0AF6 14E3       110:        FDB    UNDL
```

```
0AF8 5249       111:          FCC    'RI'
0AFA 14E8       112:          FDB    RDIT
0AFC 4943       113:          FCC    'IC'
0AFE 1505       114:          FDB    ITMCH
0B00 4E49       115:          FCC    'NI'
0B02 1512       116:          FDB    NXTI
0B04 4E42       117:          FCC    'NB'
0B06 1538       118:          FDB    NXTB
0B08 4346       119:          FCC    'CF'
0B0A 1563       120:          FDB    CLSFL
0B0C 4F46       121:          FCC    'OF'
0B0E 1575       122:          FDB    OPNF
0B10 2020       123:          FCC    '  '
0B12 0B46       124:          FDB    SPACE6
0B14 2020       125:          FCC    '  '
0B16 0B46       126:          FDB    SPACE6
0B18 2020       127:          FCC    '  '
0B1A 0B46       128:          FDB    SPACE6
0B1C 00         129: TBLEND   FCB    0
```

```
                    1:
                    2: * PASS FILE ROUTINE .PS
                    3:
0B1D 7F00BB         4: PASS    CLR     FILFLG   FIX FLAGS
0B20 7C0080         5:         INC     PASFLG
0B23 39             6:         RTS
                    7:
                    8: * SPACE ROUTINE .SP N
                    9:
0B24 BD09E1        10: SPACE   JSR     FLUSHB   FLUSH BUFFER
0B27 9668          11:         LDAA    NSP      NO SPACE?
0B29 261B          12:         BNE     SPACE6
0B2B BD12E9        13:         JSR     CHKNUM   CHECK FOR NUMBER
0B2E 9665          14:         LDAA    INNUM    GET NUMBER
0B30 2603          15:         BNE     SPACE2
0B32 7C0065        16:         INC     INNUM    INC BY ONE
0B35 BD0DC4        17: SPACE2  JSR     FNTR     FIND TRAP
0B38 9165          18:         CMPA    INNUM    EQUAL?
0B3A 2502          19:         BCS     SPACE4
0B3C 9665          20:         LDAA    INNUM    GET NUMBER
0B3E 36            21: SPACE4  PSH     A
0B3F BD095C        22:         JSR     PCRLF    OUTPUT CR AND LF
0B42 32            23:         PUL     A
0B43 4A            24:         DEC     A        DEC COUNT
0B44 26F8          25:         BNE     SPACE4
0B46 39            26: SPACE6  RTS     RETURN
                   27:
                   28: * PAGE ROUTINE .PG +N
                   29:
0B47 BD12E9        30: PAGE    JSR     CHKNUM   CHECK FOR NUMBER
0B4A 2407          31:         BCC     PAGE2
0B4C 9669          32:         LDAA    PGN      GET PAGE NUMBER
0B4E BD12D8        33:         JSR     FIXVAL   FIX VALUE
0B51 200B          34:         BRA     PAGE4
0B53 9668          35: PAGE2   LDAA    NSP      NO SPACE?
0B55 2613          36:         BNE     PAGE6
0B57 9653          37:         LDAA    NPGN     GET NEW PAGE NUM.
0B59 2603          38:         BNE     PAGE4
0B5B 9669          39:         LDAA    PGN
0B5D 4C            40:         INC     A        BUMP BY ONE
0B5E 9753          41: PAGE4   STAA    NPGN     SAVE AS NEW
0B60 BD09E1        42:         JSR     FLUSHB   FLUSH BUFFER
0B63 BD095C        43: PAGE5   JSR     PCRLF    OUTPUT CR & LF
0B66 9653          44:         LDAA    NPGN     GET NEW PAGE NUM.
0B68 26F9          45:         BNE     PAGE5
0B6A 39            46: PAGE6   RTS     RETURN
                   47:
                   48: * MULTIPLE SPACE ROUTINE .MS +N
                   49:
0B6B BD12E9        50: MULTS   JSR     CHKNUM   CHECK FOR NUMBER
0B6E 2404          51:         BCC     MULTS2
0B70 9665          52:         LDAA    INNUM    GET NUMBER
0B72 2002          53:         BRA     MULTS3
0B74 8602          54: MULTS2  LDAA    #2       DEFAULT IS 2
0B76 97C5          55: MULTS3  STAA    MSC      SET COUNT
```

```
0B78 39          56:         RTS
                 57:
                 58: * SINGLE SPACE ROUTINE .SS
                 59:
0B79 7F00C5      60: SNGLS   CLR     MSC      CLEAR COUNT
0B7C 39          61:         RTS
                 62:
                 63: * NO ADJUST ROUTINE .NJ
                 64:
0B7D 7F00C9      65: NOJST   CLR     JUST     CLEAR JUST FLAG
0B80 39          66:         RTS
                 67:
                 68: * SET JUSTIFICATION ROUTINE .JU C
                 69:
0B81 97C9        70: JST     STAA    JUST     SET FLAG
0B83 BD12A7      71:         JSR     LDNSKP   GET NEXT CHAR.
0B86 BD12B5      72:         JSR     CLSFY    CLASSIFY IT
0B89 C102        73:         CMPB    #2
0B8B 2609        74:         BNE     JST15
0B8D 814E        75:         CMPA    #'N      NORMAL?
0B8F 2606        76:         BNE     JST2
0B91 4F          77: JST1    CLR     A        ADJUST FLAGS
0B92 97C1        78:         STAA    CNJ
0B94 97C3        79:         STAA    RTJ
0B96 39          80: JST15   RTS     RETURN
0B97 8152        81: JST2    CMPA    #'R      RIGHT HAND?
0B99 2606        82:         BNE     JST3
0B9B 7F00C1      83:         CLR     CNJ      FIX FLAGS
0B9E 97C3        84:         STAA    RTJ
0BA0 39          85:         RTS
0BA1 8143        86: JST3    CMPA    #'C      CENTERED?
0BA3 26EC        87:         BNE     JST1
0BA5 7F00C3      88:         CLR     RTJ      FIX FLAGS
0BA8 97C1        89:         STAA    CNJ
0BAA 39          90:         RTS     RETURN
                 91:
                 92: * SET LEFT MARGIN .LM +N
                 93:
0BAB BD12E9      94: LEFTM   JSR     CHKNUM   CHECK FOR NUMBER
0BAE 240A        95:         BCC     LEFTM2
0BB0 963E        96:         LDAA    LFM      GET MARGIN
0BB2 BD12D8      97:         JSR     FIXVAL   FIX VALUE
0BB5 2A01        98:         BPL     LEFTM1
0BB7 4F          99:         CLR     A
0BB8 973E       100: LEFTM1  STAA    LFM      SET NEW VALUE
0BBA 39         101: LEFTM2  RTS     RETURN
                102:
                103: * SET INDENT .IN +N
                104:
0BBB BD09E1     105: INDNT   JSR     FLUSHB   FLUSH BUFFER
0BBE BD12E9     106:         JSR     CHKNUM   CHECK FOR NUMBER
0BC1 24F7       107:         BCC     LEFTM2
0BC3 9638       108:         LDAA    IND      GET INDENT
0BC5 BD12D8     109:         JSR     FIXVAL   FIX VALUE
0BC8 2A01       110:         BPL     INDNT2
```

```
0BCA 4F          111:         CLR    A
0BCB 9038        112: INDNT2  SUB    A         IND SET INDENT
0BCD 97A6        113:         STAA   TIND      SAVE AS TEMP
0BCF 2014        114:         BRA    LENT25
                 115:
                 116: * SET LENGTH OF LINE .LN +N
                 117:
0BD1 BD12E9      118: LENTH   JSR    CHKNUM    CHECK FOR NUMBER
0BD4 2419        119:         BCC    LENTH5
0BD6 963B        120:         LDAA   LLN       GET LENGTH
0BD8 BD12D8      121:         JSR    FIXVAL    FIX VALUE
0BDB 810E        122:         CMPA   #14       14 OR LESS?
0BDD 2202        123:         BHI    LENTH2
0BDF 860F        124:         LDAA   #15       FORCE TO 15
0BE1 903B        125: LENTH2  SUB    A         LLN SET NEW
0BE3 97A7        126:         STAA   TLLN      SAVE AS TEMP
0BE5 DECD        127: LENT25  LDX    BUFPNT    CHECK POINTER
0BE7 8C18C3      128:         CPX    #LINBUF
0BEA 2603        129:         BNE    LENTH5
0BEC 7E15A5      130:         JMP    FIXWD     GO FIX WIDTH
0BEF 39          131: LENTH5  RTS    RETURN
                 132:
                 133: * DO NECESSARY TABBING
                 134:
0BF0 D632        135: DOTAB   LDAB   COLCNT    GET COUNT
0BF2 DE74        136:         LDX    NXTTAB    POINT TO TAB
0BF4 E100        137:         CMPB   0,X       COMPARE
0BF6 2405        138:         BCC    DOTAB2
0BF8 96B4        139:         LDAA   TFILL     GET FILL CHAR.
0BFA 7E07CD      140:         JMP    FETCH3
0BFD 7F0076      141: DOTAB2  CLR    TABFLG    CLEAR FLAG
0C00 7E074C      142:         JMP    GETCHR    BACK TO GET CHAR.
                 143:
                 144: * SET NO SPACE .NS
                 145:
0C03 9768        146: NOSPC   STAA   NSP       SET FLAG
0C05 39          147:         RTS
                 148:
                 149: * RESTORE SPACE MODE .RS
                 150:
0C06 7F0068      151: RESPC   CLR    NSP       CLEAR FLAG
0C09 39          152:         RTS
                 153:
                 154: * DEFINE TAB CHARACTER .TC C
                 155:
0C0A BD12A7      156: TABCH   JSR    LDNSKP    GET TO NEXT CHAR.
0C0D 810D        157:         CMPA   #$D       IS IT A C.R.?
0C0F 2601        158:         BNE    TABCH2
0C11 4F          159:         CLR    A         CLEAR VALUE
0C12 97B3        160: TABCH2  STAA   TAB       SAVE TAB CHAR.
0C14 39          161:         RTS    RETURN
                 162:
                 163: * DEFINE TAB FILL CHARACTER .TF C
                 164:
0C15 BD12A7      165: TABFIL  JSR    LDNSKP    GET TO NEXT CHAR.
```

```
0C18 810D       166:         CMPA   #$D      IS IT C.R.?
0C1A 2602       167:         BNE    TABFI2
0C1C 86A0       168:         LDAA   #$A0     SET UNPAD SPACE
0C1E 97B4       169: TABFI2  STAA   TFILL    SAVE CHAR.
0C20 39         170:         RTS    RETURN
                171:
                172: * DEFINE TAB COLUMNS .TA 1 2 3 4
                173:
0C21 CE0110     174: STAB    LDX    #TABS    POINT TO TABS
0C24 BD14BC     175: STAB2   JSR    PUSHX    SAVE X
0C27 BD12E9     176:         JSR    CHKNUM   CHECK FOR NUMBER
0C2A 240E       177:         BCC    STAB4
0C2C BD14CE     178:         JSR    PULLX    RESTORE
0C2F 9665       179:         LDAA   INNUM    GET NUMBER
0C31 A700       180:         STAA   0,X      SAVE IT
0C33 08         181:         INX    BUMP     POINTER
0C34 8C0124     182:         CPX    #TABEND  END OF TABLE?
0C37 26EB       183:         BNE    STAB2
0C39 39         184:         RTS    RETURN
0C3A BD14CE     185: STAB4   JSR    PULLX
0C3D 6F00       186:         CLR    0,X      CLEAR LAST
0C3F 39         187:         RTS
                188:
                189: * OUTPUT STRING TO TERMINAL .TM STRING
                190:
0C40 BD12A7     191: TERM    JSR    LDNSKP   GET TO NEXT CHAR.
0C43 A600       192: TERM2   LDAA   0,X      GET CHAR.
0C45 810D       193:         CMPA   #$D      IS IT C.R.?
0C47 2703       194:         BEQ    TERM4
0C49 08         195:         INX    BUMP     THE POINTER
0C4A 20F7       196:         BRA    TERM2
0C4C 8604       197: TERM4   LDAA   #4       SET UP 4
0C4E A700       198:         STAA   0,X      SAVE IT
0C50 DED9       199:         LDX    CMNPNT   SET POINTER
0C52 7E1636     200:         JMP    PSTRNG   GO PRINT STRING
                201:
                202: * GET INPUT FROM TERMINAL .GI PROMPT
                203:
0C55 8DE9       204: GETIN   BSR    TERM     GO PRINT PROMPT
0C57 CE1804     205:         LDX    #QUSTR   POINT TO STR.
0C5A BD1638     206:         JSR    PDATA    OUTPUT IT
0C5D BD15D5     207:         JSR    GIBUF    GET INPUT RESPONSE
0C60 9793       208:         STAA   SBFLG    SET FLAG
0C62 39         209:         RTS    RETURN
                210:
                211: * SET NEW ENVIRONMENT .EV N
                212:
0C63 BD12E9     213: SENV    JSR    CHKNUM   CHECK FOR NUMBER
0C66 2408       214:         BCC    SENV1
0C68 9665       215:         LDAA   INNUM    GET NUMBER
0C6A 2705       216:         BEQ    SENV2
0C6C 8601       217:         LDAA   #1       SET UP 1
0C6E 2001       218:         BRA    SENV2
0C70 4F         219: SENV1   CLR    A        CLEAR VALUE
0C71 9172       220: SENV2   CMPA   EV       PRESENT EV?
```

```
0C73 2601       221:         BNE    SENV3
0C75 39         222:         RTS    YES,     RETURN
0C76 9772       223: SENV3   STAA   EV       SET NEW EV
0C78 9632       224:         LDAA   COLCNT   SAVE COL COUNT
0C7A D677       225:         LDAB   COLCN2
0C7C D732       226:         STAB   COLCNT
0C7E 9777       227:         STAA   COLCN2
0C80 9638       228:         LDAA   IND      FIX THE INDENT
0C82 D678       229:         LDAB   IND2
0C84 9778       230:         STAA   IND2
0C86 D738       231:         STAB   IND
0C88 963B       232:         LDAA   LLN      DO LINE LENGTH
0C8A D694       233:         LDAB   LLN2
0C8C 9794       234:         STAA   LLN2
0C8E D73B       235:         STAB   LLN
0C90 CE00B5     236:         LDX    #AUTO    POINT TO BLOCK
0C93 A600       237: SENV4   LDAA   0,X      GET VALUE
0C95 E601       238:         LDAB   1,X
0C97 A701       239:         STAA   1,X      SWAP VALUE
0C99 E700       240:         STAB   0,X
0C9B 08         241:         INX    GO       TO NEXT
0C9C 08         242:         INX
0C9D 8C00CD     243:         CPX    #BUFPNT  FINISHED?
0CA0 26F1       244:         BNE    SENV4
0CA2 A600       245: SENV6   LDAA   0,X      GET VALUE
0CA4 E602       246:         LDAB   2,X
0CA6 A702       247:         STAA   2,X      SWAP
0CA8 E700       248:         STAB   0,X
0CAA A601       249:         LDAA   1,X
0CAC E603       250:         LDAB   3,X
0CAE A703       251:         STAA   3,X
0CB0 E701       252:         STAB   1,X
0CB2 08         253:         INX    BUMP     THE POINTER
0CB3 08         254:         INX
0CB4 08         255:         INX
0CB5 08         256:         INX
0CB6 8C00D9     257:         CPX    #CMNPNT  FINISHED?
0CB9 26E7       258:         BNE    SENV6
0CBB CE18C3     259:         LDX    #LINBUF  POINT TO BUFFER
0CBE A600       260: SENV8   LDAA   0,X      GET A CHAR.
0CC0 E6C8       261:         LDAB   200,X
0CC2 A7C8       262:         STAA   200,X    SWAP FOR NEW
0CC4 E700       263:         STAB   0,X
0CC6 08         264:         INX    BUMP     TO NEXT
0CC7 8C198B     265:         CPX    #LINBU2  FINISHED?
0CCA 26F2       266:         BNE    SENV8
0CCC 39         267:         RTS    RETURN
                268:
                269: * SET NEW PAGE NUMBER .PN +N
                270:
0CCD BD12E9     271: PGNUM   JSR    CHKNUM   CHECK FOR NUMBER
0CD0 2407       272:         BCC    PGNUM4
0CD2 9669       273:         LDAA   PGN      GET VALUE
0CD4 BD12D8     274:         JSR    FIXVAL   GO FIX VALUE
0CD7 9769       275:         STAA   PGN      SAVE NEW
```

```
0CD9 39          276: PGNUM4 RTS     RETURN
                 277:
                 278: * SET DOUBLE HEIGHT .DH
                 279:
0CDA BD09E5      280: DUBH   JSR     FLUSH   FLUSH BUFFER
0CDD B6021C      281: DUBH1  LDAA    DHCHAR  SET UP H/W CONTROL CODE
0CE0 7C003D      282:        INC     LINCNT  BUMP LINE COUNT
0CE3 7E1645      283: DUBH2  JMP     OUTCHR  OUTPUT CHARACTER
                 284:
                 285: * SET DOUBLE WIDTH .DW
                 286:
0CE6 BD09E5      287: DUBW   JSR     FLUSH   FLUSH BUFFER
0CE9 B6021D      288:        LDAA    DWCHAR  SET UP H/W CONTROL CODE
0CEC 9782        289:        STAA    DWFLG   SET FLAG
0CEE 20F3        290:        BRA     DUBH2
                 291:
                 292: * SET DOUBLE BOTH .DB
                 293:
0CF0 BD09E5      294: DUBB   JSR     FLUSH   FLUSH BUFFER
0CF3 B6021D      295:        LDAA    DWCHAR  SET UP H/W CONTROL CODE
0CF6 9782        296:        STAA    DWFLG   SET FLAG
0CF8 BD1645      297:        JSR     OUTCHR  OUTPUT CHARACTER
0CFB 20E0        298:        BRA     DUBH1
                 299:
                 300: * CENTER N LINES .CE +N
                 301:
0CFD BD09E1      302: CENTER JSR     FLUSHB  FLUSH BUFFER
0D00 BD12E9      303:        JSR     CHKNUM  CHECK FOR NUMBER
0D03 240B        304:        BCC     CENTE2
0D05 96C7        305:        LDAA    CNTFLG  GET OLD COUNT
0D07 BD12D8      306:        JSR     FIXVAL  FIX VALUE
0D0A 97C7        307:        STAA    CNTFLG  SAVE NEW
0D0C 2723        308:        BEQ     PAGEL4
0D0E 2004        309:        BRA     CENTE4
0D10 8601        310: CENTE2 LDAA    #1      DEFAULT TO 1
0D12 97C7        311:        STAA    CNTFLG  SAVE COUNT
0D14 96BB        312: CENTE4 LDAA    FILFLG  GET FLAG
0D16 9763        313:        STAA    TFILF   SAVE AS TEMP
0D18 86FF        314:        LDAA    #$FF
0D1A 97BB        315:        STAA    FILFLG  FORCE FILL MODE
0D1C 39          316:        RTS     RETURN
                 317:
                 318: * SET PAGE LENGTH .PL +N
                 319:
0D1D BD12E9      320: PAGEL  JSR     CHKNUM  CHECK FOR NUMBER
0D20 2504        321:        BCS     PAGEL1
0D22 8642        322:        LDAA    #66     DEFAULT TO 66
0D24 2009        323:        BRA     PAGEL2
0D26 963F        324: PAGEL1 LDAA    PGL     GET LAST VALUE
0D28 BD12D8      325:        JSR     FIXVAL  FIX VALUE
0D2B 4D          326:        TST     A
0D2C 2601        327:        BNE     PAGEL2
0D2E 4C          328:        INC     A       BUMP BY ONE
0D2F 973F        329: PAGEL2 STAA    PGL     SAVE NEW
0D31 39          330: PAGEL4 RTS     RETURN
```

```
                    331:
                    332: * SET SINGLE INDENT .SI +N
                    333:
0D32 BD09E1         334: SIND   JSR    FLUSHB  FLUSH BUFFER
0D35 BD12E9         335:        JSR    CHKNUM  CHECK FOR NUMBER
0D38 24F7           336:        BCC    PAGEL4
0D3A 9670           337:        LDAA   SIN     GET OLD VALUE
0D3C BD12D8         338:        JSR    FIXVAL  GO FIX VALUE
0D3F 9070           339:        SUB    A       SIN
0D41 97A5           340:        STAA   TSIN    SAVE AS TEMP
0D43 7E0BE5         341:        JMP    LENT25
                    342:
                    343: * SET CAPS MODE .CP
                    344:
0D46 860F           345: STCAP  LDAA   #$F     SET FLAG
0D48 976C           346:        STAA   DOCAP
0D4A 39             347:        RTS
                    348:
                    349: * CLEAR CAPS MODE .NC
                    350:
0D4B 7F006C         351: NOCAP  CLR    DOCAP   CLEAR FLAG
0D4E 39             352:        RTS
                    353:
                    354: * PUT IN INDENT FIELD .PI STRING
                    355:
0D4F BD09E1         356: PTIND  JSR    FLUSHB  FLUSH BUFFER
0D52 BD12A7         357:        JSR    LDNSKP  GET TO NEXT CHAR.
0D55 D638           358:        LDAB   IND     GET INDENT
0D57 272A           359:        BEQ    PTIND5
0D59 D75E           360:        STAB   NOCR    SET FLAG
0D5B 5F             361:        CLR    B
0D5C 37             362: PTIND2 PSH    B
0D5D BD074C         363:        JSR    GETCHR  GO GET CHAR.
0D60 33             364:        PUL    B
0D61 810D           365:        CMPA   #$D     CHECK IF C.R.?
0D63 270C           366:        BEQ    PTIND3
0D65 37             367:        PSH    B
0D66 BD1645         368:        JSR    OUTCHR  GO OUTPUT CHAR.
0D69 33             369:        PUL    B
0D6A 5C             370:        INC    B       BUMP COUNT
0D6B D138           371:        CMPB   IND     FINISHED?
0D6D 240E           372:        BCC    PTIND4
0D6F 20EB           373:        BRA    PTIND2
0D71 8620           374: PTIND3 LDAA   #$20    SET UP SPACE
0D73 37             375:        PSH    B
0D74 BD1645         376:        JSR    OUTCHR  OUTPUT IT
0D77 33             377:        PUL    B
0D78 5C             378:        INC    B       BUMP COUNT
0D79 D138           379:        CMPB   IND     FINISHED?
0D7B 25F4           380:        BCS    PTIND3
0D7D 5C             381: PTIND4 INC    B       BUMP COUNT
0D7E D76F           382:        STAB   PTFL    SET FLAG
0D80 7F005E         383:        CLR    NOCR
0D83 39             384: PTIND5 RTS    RETURN
                    385:
```

```
                    386: * SET NOFILL MODE .NF
                    387:
0D84 BD09E1         388: NOFILL JSR    FLUSHB  FLUSH BUFFER
0D87 7F00BB         389:        CLR    FILFLG  CLEAR FLAG
0D8A 39             390:        RTS
                    391:
                    392: * SET FILL MODE .FI
                    393:
0D8B BD09E1         394: FILL   JSR    FLUSHB  FLUSH BUFFER
0D8E 7C00BB         395:        INC    FILFLG  SET FLAG
0D91 39             396:        RTS
                    397:
                    398: * REPEAT COMMAND .RP
                    399:
0D92 BD09E1         400: RPT    JSR    FLUSHB  FLUSH BUFFER
0D95 7C00A8         401:        INC    SUPL    SET FLAG
0D98 BD0B47         402:        JSR    PAGE    GO PAGE
0D9B 7E16FC         403:        JMP    RWND    REWIND FILE
```

```
                     1: * STOP COMMAND .ST
                     2:
0D9E BD09E1          3: STOP    JSR     FLUSHB   FLUSH BUFFER
0DA1 CE17FA          4:         LDX     #STPSTR  POINT TO STRING
0DA4 BD1636          5: STOP1   JSR     PSTRNG   OUTPUT IT
0DA7 BD0206          6:         JSR     INCH     GO GET CHAR.
0DAA 8153            7:         CMPA    #'S      IS IT 'S'?
0DAC 2606            8:         BNE     STOP2
0DAE BD0235          9:         JSR     CDFM     CLOSE FILES
0DB1 7E0A17         10:         JMP     FINIS4   GO FINISH
0DB4 39             11: STOP2  RTS     RETURN
                    12:
                    13: * NEED N LINES .NL N
                    14:
0DB5 BD12E9         15: NEDL   JSR     CHKNUM   CHECK FOR NUMBER
0DB8 2503           16:         BCS     NEDL1
0DBA 7C0065         17:         INC     INNUM    BUMP BY 1
0DBD 8D05           18: NEDL1  BSR     FNTR     GO FIND TRAP
0DBF 9165           19:         CMPA    INNUM    COMPARE
0DC1 254A           20:         BCS     SAVS25
0DC3 39             21:         RTS
                    22:
                    23: * FIND THE NEXT TRAP
                    24:
0DC4 86FF           25: FNTR   LDAA    #$FF     SET MIN DISTANCE
0DC6 9771           26:         STAA    MINDIS
0DC8 CE1AB7         27:         LDX     #TRAPS   POINT TO TRAPS
0DCB D63D           28: FNTR2  LDAB    LINCNT   GET COUNT
0DCD E100           29:         CMPB    0,X      CHECK LOC.
0DCF 240F           30:         BCC     FNTR4
0DD1 A600           31:         LDAA    0,X      GET DISTANCE
0DD3 81FF           32:         CMPA    #$FF
0DD5 2711           33:         BEQ     FNTR5
0DD7 10             34:         SBA     SWAP     REGISTERS
0DD8 9171           35:         CMPA    MINDIS   MIN DISTANCE?
0DDA 2404           36:         BCC     FNTR4
0DDC 9771           37:         STAA    MINDIS   SAVE NEW
0DDE DF79           38:         STX     NXTTRP   SAVE POINTER
0DE0 08             39: FNTR4  INX     BUMP     THE POINTER
0DE1 08             40:         INX
0DE2 08             41:         INX
0DE3 8C1AE7         42:         CPX     #TRPEND  FINISHED?
0DE6 26E3           43:         BNE     FNTR2
0DE8 D671           44: FNTR5  LDAB    MINDIS   GET DISTANCE
0DEA C1FF           45:         CMPB    #$FF
0DEC 2607           46:         BNE     FNTR6
0DEE 963F           47:         LDAA    PGL      SET UP PAGE LEN.
0DF0 903D           48:         SUB     A        LINCNT
0DF2 4C             49:         INC     A        FIX VALUE
0DF3 5F             50:         CLR     B
0DF4 39             51:         RTS     RETURN
0DF5 E600           52: FNTR6  LDAB    0,X
0DF7 9671           53:         LDAA    MINDIS   GET DISTANCE
0DF9 DE79           54:         LDX     NXTTRP   POINT TO TRAP
0DFB 39             55:         RTS     RETURN
```

```
                    56:
                    57: * SAVE SPACE ROUTINE .SV N
                    58:
0DFC 7F007B         59: SAVS    CLR     SVDSPC  CLEAR COUNT
0DFF BD12E9         60:         JSR     CHKNUM  CHECK FOR NUMBER
0E02 2503           61:         BCS     SAVS1
0E04 7C0065         62:         INC     INNUM   GET COUNT
0E07 8DBB           63: SAVS1   BSR     FNTR    FIND TRAP
0E09 9165           64:         CMPA    INNUM
0E0B 2506           65:         BCS     SAVS4
0E0D 7F0068         66: SAVS25  CLR     NSP     CLEAR NO SPACE
0E10 7E0B3E         67:         JMP     SPACE4  GO DO SPACE
0E13 9665           68: SAVS4   LDAA    INNUM   GET COUNT
0E15 977B           69:         STAA    SVDSPC  SAVE COUNT
0E17 39             70: SAVS5   RTS     RETURN
                    71:
                    72: * OUTPUT SAVED SPACE .OS
                    73:
0E18 967B           74: OUTSV   LDAA    SVDSPC  GET REMEMBERED COUNT
0E1A 27FB           75:         BEQ     SAVS5
0E1C 7F007B         76:         CLR     SVDSPC  CLEAR COUNT
0E1F 20EC           77:         BRA     SAVS25  OUTPUT SPACE
                    78:
                    79: * AT LINE N ROUTINE .AT -N
                    80:
0E21 BD12E9         81: ATL     JSR     CHKNUM  CHECK FOR NUMBER
0E24 2428           82:         BCC     ATL35
0E26 BD11D5         83:         JSR     TSTNEG  IS IT NEGATIVE?
0E29 BD1059         84:         JSR     GTNAM   GET NAME
0E2C 963F           85:         LDAA    PGL     GET PAGE LEN.
0E2E 4C             86:         INC     A
0E2F BD12D8         87:         JSR     FIXVAL  FIX THE VALUE
0E32 4D             88:         TST     A
0E33 2601           89:         BNE     ATL1
0E35 4C             90:         INC     A       BUMP BY ONE
0E36 CE1AB7         91: ATL1    LDX     #TRAPS  POINT TO TRAPS
0E39 A100           92: ATL2    CMPA    0,X     COMPARE
0E3B 2712           93:         BEQ     ATL4
0E3D 8D27           94:         BSR     INTRP
0E3F 26F8           95:         BNE     ATL2
0E41 CE1AB7         96:         LDX     #TRAPS  POINT TO TRAPS
0E44 C6FF           97:         LDAB    #$FF    SET REFERENCE
0E46 E100           98: ATL3    CMPB    0,X
0E48 2714           99:         BEQ     ATL5
0E4A 8D1A          100:         BSR     INTRP
0E4C 26F8          101:         BNE     ATL3
0E4E 39            102: ATL35   RTS     RETURN
0E4F D695          103: ATL4    LDAB    MACNAM  GET NAME
0E51 2604          104:         BNE     ATL45
0E53 5A            105:         DEC     B       DEC THE COUNT
0E54 E700          106:         STAB    0,X     SAVE POSITION
0E56 39            107:         RTS     RETURN
0E57 9696          108: ATL45   LDAA    MACNAM+1    GET NAME
0E59 E701          109:         STAB    1,X     SAVE CHAR.
0E5B A702          110:         STAA    2,X
```

```
0E5D 39          111:         RTS     RETURN
0E5E D695        112: ATL5    LDAB    MACNAM  GET NAME
0E60 27EC        113:         BEQ     ATL35
0E62 A700        114:         STAA    0,X     SAVE CHARACTER
0E64 20F1        115:         BRA     ATL45
                 116:
                 117: * INCREMENT TRAP POINTER
                 118:
0E66 08          119: INTRP   INX     FIX     POINTER
0E67 08          120:         INX
0E68 08          121:         INX
0E69 8C1AE7      122:         CPX     #TRPEND FINISHED?
0E6C 39          123:         RTS
                 124:
                 125: * DEFINE MACRO
                 126:
0E6D 965B        127: DEFMAC  LDAA    MBFLG   CHECK DEF FLAG
0E6F 2639        128:         BNE     DEFMA5
0E71 BD0F1E      129:         JSR     OPMAC   GO OPEN MACRO
0E74 2734        130: DEFMA2  BEQ     DEFMA5
0E76 7C0083      131:         INC     DFMFLG  SET DEF FLAG
0E79 7F005A      132:         CLR     CMFLG   CLEAR COMMAND
0E7C BD0749      133: DEFMA3  JSR     CLRGET  GO GET CHARACTER
0E7F 812E        134:         CMPA    #'.     IS IT A PERIOD?
0E81 260E        135:         BNE     DEFM35
0E83 BD0749      136:         JSR     CLRGET  GET NEXT CHAR.
0E86 812E        137:         CMPA    #'.     IS IT A PERIOD?
0E88 2713        138:         BEQ     DEFMA4
0E8A 36          139:         PSH     A       SAVE CHAR
0E8B 862E        140:         LDAA    #'.     SET UP PERIOD
0E8D BD0F7D      141:         JSR     OUTMAC  OUTPUT TO MACRO
0E90 32          142:         PUL     A       RESTORE CHAR
0E91 BD0F7D      143: DEFM35  JSR     OUTMAC  OUTPUT TO MACRO
0E94 810D        144:         CMPA    #$D     IS IT A C.R.?
0E96 27E4        145:         BEQ     DEFMA3
0E98 BD0749      146:         JSR     CLRGET  GET NEXT CHAR.
0E9B 20F4        147:         BRA     DEFM35
0E9D BD0FC7      148: DEFMA4  JSR     CLSMAC  CLOSE MACRO
0EA0 BD0749      149: DEFM45  JSR     CLRGET  GET CHARACTER
0EA3 810D        150:         CMPA    #$D     IS IT A C.R.?
0EA5 26F9        151:         BNE     DEFM45
0EA7 7F0083      152:         CLR     DFMFLG  CLEAR DEF FLAG
0EAA 39          153: DEFMA5  RTS     RETURN
                 154:
                 155: * APPEND TO A MACRO .AP XX
                 156:
0EAB 965B        157: APMAC   LDAA    MBFLG   CHECK FLAG
0EAD 26FB        158:         BNE     DEFMA5
0EAF BD0F51      159:         JSR     OPAPP   OPEN FOR APPEND
0EB2 20C0        160:         BRA     DEFMA2
                 161:
                 162: * REMOVE MACRO .RM XX
                 163:
0EB4 BD1059      164: REMMAC  JSR     GTNAM   GO GET NAME
0EB7 BD1004      165:         JSR     FNDMAC  FIND MACRO
```

```
0EBA 2701       166:         BEQ    REMMA4
0EBC 39         167:         RTS    RETURN
0EBD DF97       168: REMMA4  STX    MACTMP  SAVE POINTER
0EBF EE02       169:         LDX    2,X     GET ADDRESS
0EC1 BD1033     170:         JSR    CHKLST  LAST MACRO?
0EC4 2414       171:         BCC    REMMA6
0EC6 DE97       172:         LDX    MACTMP  GET POINTER
0EC8 A602       173:         LDAA   2,X     GET ADDRESS
0ECA E603       174:         LDAB   3,X
0ECC DE99       175:         LDX    LSTAVL  SET LAST AVAIL
0ECE A700       176:         STAA   0,X
0ED0 E701       177:         STAB   1,X
0ED2 DE9F       178:         LDX    NXTMAC  SET UP NXT MAC
0ED4 DF99       179:         STX    LSTAVL  SAVE AS LAST AVAIL
0ED6 DE97       180:         LDX    MACTMP
0ED8 200A       181:         BRA    REMNAM
0EDA DE97       182: REMMA6  LDX    MACTMP  SET UP POINTER
0EDC A602       183:         LDAA   2,X     GET ADDRESS
0EDE E603       184:         LDAB   3,X
0EE0 979B       185:         STAA   FSTAVL  SET FIRST AVAIL
0EE2 D79C       186:         STAB   FSTAVL+1
                187:
                188: * REMOVE MACRO NAME FROM TABLE
                189:
0EE4 E604       190: REMNAM  LDAB   4,X     MOVE CHAR DOWN
0EE6 E700       191:         STAB   0,X
0EE8 08         192:         INX    BUMP    THE POINTER
0EE9 9CED       193:         CPX    MACEND  FINISHED?
0EEB 26F7       194:         BNE    REMNAM
0EED 09         195:         DEX    DEC     THE POINTER
0EEE 09         196:         DEX
0EEF 09         197:         DEX
0EF0 09         198:         DEX
0EF1 DFED       199:         STX    MACEND  SET NEW END
0EF3 39         200:         RTS    RETURN
                201:
                202: * DIVERT .DI XX
                203:
0EF4 9685       204: DIVERT  LDAA   DIVFLG  CHECK DIV FLAG
0EF6 270D       205:         BEQ    DIVER2
0EF8 7C0086     206: DIVER0  INC    DIVFL2  SET MARKER
0EFB 7E0FC7     207:         JMP    CLSMAC  CLOSE MACRO
0EFE 7F0085     208: DIVER1  CLR    DIVFLG  CLEAR FLAGS
0F01 7F0086     209:         CLR    DIVFL2
0F04 39         210:         RTS    RETURN
0F05 7C0086     211: DIVER2  INC    DIVFL2  SET MARKER
0F08 8D14       212:         BSR    OPMAC   GO OPEN MACRO
0F0A 27F2       213: DIVER4  BEQ    DIVER1
0F0C 7C0085     214:         INC    DIVFLG  SET FLAG
0F0F 7F0045     215:         CLR    LDIV    CLEAR COUNT
0F12 39         216:         RTS    RETURN
                217:
                218: * DIVERT APPEND .DA XX
                219:
0F13 9685       220: DIVAPP  LDAA   DIVFLG  CHECK DIV FLAG
```

```
0F15 26E1       221:         BNE    DIVER0
0F17 7C0086     222:         INC    DIVFL2  SET MARKER
0F1A 8D35       223:         BSR    OPAPP   OPEN FOR APPEND
0F1C 20EC       224:         BRA    DIVER4
                225:
                226: * OPEN A MACRO SPACE
                227:
0F1E BD1059     228: OPMAC   JSR    GTNAM   GET MACRO NAME
0F21 9695       229:         LDAA   MACNAM
0F23 2601       230:         BNE    OPMAC2  PRESENT?
0F25 39         231:         RTS
0F26 BD1004     232: OPMAC2  JSR    FNDMAC  LOOK FOR MACRO
0F29 2604       233:         BNE    OPMAC4
0F2B 8D90       234:         BSR    REMMA4  REMOVE OLD VERSION
0F2D 20F7       235:         BRA    OPMAC2  OPEN MACRO
0F2F 9695       236: OPMAC4  LDAA   MACNAM  GET NAME
0F31 D696       237:         LDAB   MACNAM+1
0F33 8C1E9D     238:         CPX    #MTEND  END OF TABLE?
0F36 2603       239:         BNE    OPMAC5
0F38 7E0FAF     240:         JMP    SYSERR  REPORT ERROR
0F3B A700       241: OPMAC5  STAA   0,X     SAVE NAME
0F3D E701       242:         STAB   1,X
0F3F 969E       243:         LDAA   FSTAVL  GET FIRST AVAIL
0F41 D69C       244:         LDAB   FSTAVL+1
0F43 A702       245:         STAA   2,X     SAVE IN TABLE
0F45 E703       246:         STAB   3,X
0F47 08         247:         INX    BUMP    THE POINTER
0F48 08         248:         INX
0F49 08         249:         INX
0F4A 08         250:         INX
0F4B DFED       251:         STX    MACEND  SET NEW END
0F4D DE9B       252:         LDX    FSTAVL  GET LAST AVAIL
0F4F 201E       253:         BRA    SAVSX
                254:
                255: * OPEN MACRO FOR APPEND
                256:
0F51 BD1059     257: OPAPP   JSR    GTNAM   GET MACRO NAME
0F54 9695       258:         LDAA   MACNAM
0F56 2601       259:         BNE    OPAPP2
0F58 39         260:         RTS    NO      NAME
0F59 BD1004     261: OPAPP2  JSR    FNDMAC  FIND MACRO
0F5C 26D1       262:         BNE    OPMAC4
0F5E EE02       263:         LDX    2,X     GET LOCATION
0F60 BD1033     264:         JSR    CHKLST  IS IT THE LAST ONE?
0F63 240A       265:         BCC    SAVSX
0F65 969B       266:         LDAA   FSTAVL  GET FIRST AVAIL
0F67 D69C       267:         LDAB   FSTAVL+1
0F69 A700       268:         STAA   0,X     SET NEW
0F6B E701       269:         STAB   1,X
0F6D DE9B       270:         LDX    FSTAVL
                271:
                272: * SAVE SPECIAL INDEX
                273:
0F6F 7D0086     274: SAVSX   TST    DIVFL2  TEST MARKER
0F72 2706       275:         BEQ    SAVSX2
```

```
0F74 7F0086      276:         CLR    DIVFL2  CLEAR MARKER
0F77 DFA1        277:         STX    NXTOUT  SAVE POINTER
0F79 39          278:         RTS    RETURN
0F7A DF9F        279: SAVSX2  STX    NXTMAC  SAVE POINTER
0F7C 39          280:         RTS
                 281:
                 282: * OUTPUT TO MACRO SPACE
                 283:
0F7D DFA3        284: OUTMAC  STX    XMAC    SAVE POINTER
0F7F 7D0086      285:         TST    DIVFL2  TEST MARKER
0F82 2704        286:         BEQ    OUTMA0
0F84 DEA1        287:         LDX    NXTOUT  SET POINTER
0F86 2002        288:         BRA    OUTMA1
0F88 DE9F        289: OUTMA0  LDX    NXTMAC
0F8A 6D00        290: OUTMA1  TST    0,X     TEST IF END
0F8C 2718        291:         BEQ    OUTMA4
0F8E 811F        292:         CMPA   #$1F    IS IT $1F?
0F90 220C        293:         BHI    OUTM18
0F92 810D        294:         CMPA   #$D     IS IT C.R.?
0F94 260B        295:         BNE    OUTMA3
0F96 7D0086      296:         TST    DIVFL2  TEST MARKER
0F99 2703        297:         BEQ    OUTM18
0F9B 7C0045      298:         INC    LDIV    BUMP DIV LINE CNT
0F9E A700        299: OUTM18  STAA   0,X     PUT CHARACTER
0FA0 08          300:         INX    BUMP    THE POINTER
0FA1 8DCC        301: OUTMA3  BSR    SAVSX   GO SAVE X
0FA3 DEA3        302:         LDX    XMAC    RESTORE POINTER
0FA5 39          303:         RTS
0FA6 08          304: OUTMA4  INX    BUMP    THE POINTER
0FA7 9C99        305:         CPX    LSTAVL  LAST AVAIL?
0FA9 2704        306:         BEQ    SYSERR  ERROR?
0FAB EE00        307:         LDX    0,X     GET POINTER
0FAD 20DB        308:         BRA    OUTMA1
                 309:
                 310: * REPORT SYSTEM MACRO ERROR
                 311:
0FAF 7E0911      312: SYSERR  JMP    MACOVF  REPORT OVERFLOW
                 313:
                 314: * INPUT TO MACRO SPACE
                 315:
0FB2 DE5C        316: INMAC   LDX    MBFPNT  SET UP POINTER
0FB4 A600        317: INMAC2  LDAA   0,X     GET THE CHARACTER
0FB6 08          318:         INX    BUMP    THE POINTER
0FB7 DF5C        319:         STX    MBFPNT  SAVE IT
0FB9 4D          320:         TST    A       TEST THE CHAR.
0FBA 2606        321:         BNE    INMAC4
0FBC EE00        322:         LDX    0,X     GET LINK
0FBE 26F4        323:         BNE    INMAC2
0FC0 2004        324:         BRA    INMAC5
0FC2 81FF        325: INMAC4  CMPA   #$FF    IS CHAR FF?
0FC4 27EE        326:         BEQ    INMAC2
0FC6 39          327: INMAC5  RTS    RETURN
                 328:
                 329: * CLOSE MACRO SPACE
                 330:
```

```
0FC7 7D0086      331: CLSMAC TST    DIVFL2  TEST MARKER
0FCA 2709        332:        BEQ    CLSMA2
0FCC 4F          333:        CLR    A
0FCD 9785        334:        STAA   DIVFLG  CLEAR FLAG
0FCF 9786        335:        STAA   DIVFL2
0FD1 DEA1        336:        LDX    NXTOUT  SET POINTER
0FD3 2002        337:        BRA    CLSMA3
0FD5 DE9F        338: CLSMA2 LDX    NXTMAC  POINT TO NEXT MAC
0FD7 6D00        339: CLSMA3 TST    0,X     TEST CHARACTER
0FD9 2714        340:        BEQ    CLSMA4
0FDB 6D01        341:        TST    1,X     TEST NEXT
0FDD 2717        342:        BEQ    CLSMA5
0FDF 6D02        343:        TST    2,X     ONE MORE
0FE1 271A        344:        BEQ    CLSMA6
0FE3 6F00        345:        CLR    0,X     CLEAR OUT SPACE
0FE5 6F01        346:        CLR    1,X
0FE7 6F02        347:        CLR    2,X
0FE9 08          348:        INX    FIX     POINTER
0FEA 08          349:        INX
0FEB 08          350:        INX
0FEC DF9B        351:        STX    FSTAVL  SET FIRST AVAIL
0FEE 39          352:        RTS    RETURN
0FEF EE01        353: CLSMA4 LDX    1,X     GET LINK
0FF1 26E4        354:        BNE    CLSMA3
0FF3 7E0FAF      355:        JMP    SYSERR  REPORT MACRO ERROR
0FF6 86FF        356: CLSMA5 LDAA   #$FF    SET UP FF
0FF8 A700        357:        STAA   0,X     SAVE IT
0FFA 08          358:        INX
0FFB 20F2        359:        BRA    CLSMA4
0FFD 86FF        360: CLSMA6 LDAA   #$FF    SET UP FF
0FFF A700        361:        STAA   0,X     SAVE IT
1001 08          362:        INX    FIX     POINTER
1002 20F2        363:        BRA    CLSMA5
                 364:
                 365: * FIND MACRO
                 366:
1004 9695        367: FNDMAC LDAA   MACNAM  CHECK NAME
1006 2717        368:        BEQ    FNDMA4
1008 D696        369:        LDAB   MACNAM+1    GET NAME
100A CE1D9D      370:        LDX    #MACTBL POINT TO TABLE
100D 9CED        371: FNDMA1 CPX    MACEND  FINISHED?
100F 270E        372:        BEQ    FNDMA4
1011 A100        373:        CMPA   0,X     TEST 1ST CHAR.
1013 2604        374:        BNE    FNDMA2
1015 E101        375:        CMPB   1,X     TEST 2ND CHAR.
1017 2708        376:        BEQ    FNDMA6
1019 08          377: FNDMA2 INX    FIX     POINTER
101A 08          378:        INX
101B 08          379:        INX
101C 08          380:        INX
101D 20EE        381:        BRA    FNDMA1  REPEAT
101F DEED        382: FNDMA4 LDX    MACEND  SET POINTER
1021 39          383: FNDMA6 RTS    RETURN
                 384:
                 385: * FIND LAST MACRO ENTRY
```

```
                    386:
1022 A600           387: FNDLST LDAA    0,X       GET CHARACTER
1024 2703           388:        BEQ     FNDLS2    IS IT ZERO?
1026 08             389:        INX     CO        TO NEXT
1027 20F9           390:        BRA     FNDLST
1029 08             391: FNDLS2 INX     BUMP      POINTER
102A DF9F           392:        STX     NXTMAC    SAVE POSITION
102C EE00           393:        LDX     0,X       PICK UP LINK
102E 26F2           394:        BNE     FNDLST
1030 DE9F           395:        LDX     NXTMAC    GET NEXT LOC.
1032 39             396:        RTS     RETURN
```

```
                     1:
                     2: * CHECK LAST MACRO ENTRY
                     3:
1033 8DED            4: CHKLST BSR     FNDLST   FIND LAST ENTRY
1035 08              5:        INX     FIX      POINTER
1036 08              6:        INX
1037 9C9B            7:        CPX     FSTAVL   IS IT FIRST?
1039 2704            8:        BEQ     CHKLS2
103B DE9F            9:        LDX     NXTMAC   GET NEXT
103D 0D             10:        SEC
103E 39             11:        RTS     RETURN
103F 09             12: CHKLS2 DEX     BACK     UP
1040 09             13:        DEX
1041 09             14:        DEX
1042 86FF           15:        LDAA    #$FF     SET UP FF
1044 A700           16:        STAA    0,X      PUT CHARACTER
1046 A701           17:        STAA    1,X
1048 A702           18:        STAA    2,X
104A 0C             19:        CLC
104B 39             20:        RTS     RETURN
                    21:
                    22: * END MACRO EXECUTION
                    23:
104C 967F           24: MCEND  LDAA    MACCNT   GET COUNT
104E 2706           25:        BEQ     MCEND2
1050 7F005B         26:        CLR     MBFLG    CLEAR FLAG
1053 7C007C         27:        INC     FINMAC   SET FINISHED
1056 7E091A         28: MCEND2 JMP     FINCM    GO FINISH
                    29:
                    30: * GET TWO CHARACTER NAME
                    31:
1059 BD12A7         32: GTNAM  JSR     LDNSKP   GET TO NEXT
105C BD12B5         33:        JSR     CLSFY    CLASSIFY IT
105F C102           34:        CMPB    #2
1061 261C           35:        BNE     GTNA6
1063 36             36:        PSH     A        SAVE CHARACTER
1064 08             37:        INX     FIX      THE POINTER
1065 A600           38:        LDAA    0,X      GET CHARACTER
1067 BD12B5         39:        JSR     CLSFY    GO CLASSIFY
106A C102           40:        CMPB    #2
106C 33             41:        PUL     B        RESTORE CHARACTER
106D 2610           42:        BNE     GTNA6
106F 08             43:        INX     ADJUST   POINTER
1070 DFD9           44:        STX     CMNPNT   SAVE IT
1072 C15F           45:        CMPB    #$5F     LOWER CASE?
1074 2304           46:        BLS     GTNA4
1076 8020           47:        SUB     A        #$20 MAKE UPPER
1078 C020           48:        SUB     B        #$20
107A D795           49: GTNA4  STAB    MACNAM   SAVE THE NAME
107C 9796           50:        STAA    MACNAM+1
107E 39             51:        RTS     RETURN
107F 4F             52: GTNA6  CLR     A        CLEAR OUT
1080 5F             53:        CLR     B
1081 20F7           54:        BRA     GTNA4
                    55:
```

```
                    56: * SET TITLE LENGTH .LT +N
                    57:
1083 BD12E9         58: TLEN    JSR     CHKNUM  CHECK FOR NUMBER
1086 2407           59:         BCC     TLEN2
1088 96CB           60:         LDAA    TLN     GET LENGTH
108A BD12D8         61:         JSR     FIXVAL  GO FIX VALUE
108D 97CB           62:         STAA    TLN     SAVE NEW
108F 39             63: TLEN2   RTS     RETURN
                    64:
                    65: * DO THREE PART TITLE .TL '1'2'3'
                    66:
1090 7F00AC         67: TITLE   CLR     TPOS    CLEAR POSITION
1093 BD12A7         68:         JSR     LDNSKP  GET TO NEXT
1096 810D           69:         CMPA    #$D     C.R.?
1098 27F5           70:         BEQ     TLEN2
109A CE1B9D         71:         LDX     #TTLBUF POINT TO BUFFER
109D 7C005E         72:         INC     NOCR    SET FLAG
10A0 DFB0           73: TITLE1  STX     TTLPNT  SAVE POINTER
10A2 BD074C         74:         JSR     GETCHR  GO GET CHAR.
10A5 DEB0           75:         LDX     TTLPNT  RESTORE POINTER
10A7 A700           76:         STAA    0,X     SAVE THE CHAR.
10A9 810D           77:         CMPA    #$D     FINISHED?
10AE 2703           78:         BEQ     TITL12
10AD 08             79:         INX     BUMP    THE POINTER
10AE 20F0           80:         BRA     TITLE1
10B0 7F005E         81: TITL12  CLR     NOCR    CLEAR FLAG
10B3 CE1AE9         82:         LDX     #CMNDBF POINT TO BUFFER
10B6 A6B4           83: TITL15  LDAA    TTLBUF-CMNDBF,X
10B8 A700           84:         STAA    0,X     PUT CHAR.
10BA 08             85:         INX     GET     TO NEXT
10BE 810D           86:         CMPA    #$D     FINISHED?
10BD 26F7           87:         BNE     TITL15
10BF CE1AE9         88:         LDX     #CMNDBF RESTORE POINTER
10C2 A600           89:         LDAA    0,X     GET CHARACTER
10C4 97AD           90:         STAA    DELIM   SAVE DELIMITER
10C6 08             91:         INX     BUMP    THE POINTER
10C7 DFD9           92:         STX     CMNPNT  SAVE IT
10C9 CE1B9D         93:         LDX     #TTLBUF POINT TO BUFFER
10CC DFB0           94:         STX     TTLPNT
10CE 8620           95:         LDAA    #$20    SET UP SPACE
10D0 A700           96: TITLE2  STAA    0,X     SAVE IT
10D2 08             97:         INX     BUMP    POINTER
10D3 8C1BED         98:         CPX     #TTLBUF+80
10D6 26F8           99:         BNE     TITLE2
10D8 BD1141        100:         JSR     CNTTTL  GO COUNT TITLE
10DB D7AE          101:         STAB    TCNT    SAVE COUNT
10DD BD1155        102:         JSR     XFRTTL  TRANSFER TITLE
10E0 BD1141        103:         JSR     CNTTTL  COUNT TITLE
10E3 96CB          104:         LDAA    TLN     GET LENGTH
10E5 10            105:         SBA
10E6 47            106:         ASR     A
10E7 97AF          107:         STAA    MCNT    SAVE MIDDLE COUNT
10E9 C620          108:         LDAB    #$20    GET SPACE
10EB 91AE          109:         CMPA    TCNT
10ED 230F          110:         BLS     TITLE5
```

```
10EF 90AE       111:          SUB     A       TCNT
10F1 DEB0       112:          LDX     TTLPNT  RESTORE POINTER
10F3 E700       113: TITLE4   STAB    0,X     SAVE CHAR.
10F5 08         114:          INX     BUMP    THE POINTER
10F6 7C00AC     115:          INC     TPOS    UPDATE POSITION
10F9 4A         116:          DEC     A
10FA 26F7       117:          BNE     TITLE4
10FC DFB0       118:          STX     TTLPNT  SAVE POINTER
10FE BD1155     119: TITLE5   JSR     XFRTTL  TRANSFER TITLE
1101 BD1141     120:          JSR     CNTTTL  COUNT TITLE
1104 96CB       121:          LDAA    TLN     GET LENGTH
1106 90AC       122:          SUB     A       TPOS FIX POSITION
1108 11         123:          CBA
1109 230D       124:          BLS     TITLE7
110B 10         125:          SBA
110C C620       126:          LDAB    #$20    SET UP SPACE
110E DEB0       127:          LDX     TTLPNT  SET POINTER
1110 E700       128: TITL65   STAB    0,X     PUT CHAR
1112 08         129:          INX     BUMP    POINTER
1113 4A         130:          DEC     A       DEC THE COUNT
1114 26FA       131:          BNE     TITL65
1116 DFB0       132:          STX     TTLPNT  SAVE POINTER
1118 BD1155     133: TITLE7   JSR     XFRTTL  TRANSFER TITLE
111B 96CB       134:          LDAA    TLN     GET LENGTH
111D 97AC       135:          STAA    TPOS    SAVE POSITION
111F 271C       136:          BEQ     TITLE9
1121 D63E       137:          LDAB    LFM     CHECK MARGIN
1123 270A       138:          BEQ     TITL78
1125 8620       139: TITL75   LDAA    #$20    SETUP SPACE
1127 37         140:          PSH     B
1128 BD1645     141:          JSR     OUTCHR  OUTPUT SPACE
112B 33         142:          PUL     B
112C 5A         143:          DEC     B       DEC COUNT
112D 26F6       144:          BNE     TITL75
112F CE1B9D     145: TITL78   LDX     #TTLBUF POINT TO TITLE
1132 A600       146: TITLE8   LDAA    0,X     GET A CHARACTER
1134 BD1645     147:          JSR     OUTCHR  OUTPUT IT
1137 08         148:          INX     GO      TO NEXT
1138 7A00AC     149:          DEC     TPOS    DEC COUNT
113B 26F5       150:          BNE     TITLE8  REPEAT TIL DONE
113D BD095C     151: TITLE9   JSR     PCRLF   OUTPUT CR & LF
1140 39         152:          RTS     RETURN
                153:
                154: * COUNT CHARACTERS IN TITLE
                155:
1141 5F         156: CNTTTL   CLR     B       CLEAR COUNT
1142 DED9       157:          LDX     CMNPNT  SET POINTER
1144 A600       158: CNTTT2   LDAA    0,X     GET CHARACTER
1146 91AD       159:          CMPA    DELIM   IS IT DELIMITER?
1148 2708       160:          BEQ     CNTTT3
114A 810D       161:          CMPA    #$D     IS IT C.R.?
114C 2704       162:          BEQ     CNTTT3
114E 08         163:          INX     BUMP    THE POINTER
114F 5C         164:          INC     B       BUMP COUNT
1150 20F2       165:          BRA     CNTTT2
```

```
1152 DFDB       166: CNTTT3 STX    SPCPT1  SET POINTER
1154 39         167:        RTS    RETURN
                168:
                169: * TRANSFER TITLE TO BUFFER
                170:
1155 DED9       171: XFRTTL LDX    CMNPNT  SET POINTER
1157 9CDB       172:        CPX    SPCPT1  FINISHED?
1159 2715       173:        BEQ    BMPCP2
115B A600       174:        LDAA   0,X     GET CHARACTER
115D 08         175:        INX    BUMP    TO NEXT
115E DFD9       176:        STX    CMNPNT  SAVE
1160 DEB0       177:        LDX    TTLPNT  SET POINTER
1162 A700       178:        STAA   0,X     PUT CHARACTER
1164 08         179:        INX    BUMP    TO NEXT
1165 DFB0       180:        STX    TTLPNT  SAVE
1167 7C00AC     181:        INC    TPOS    BUMP POSITION
116A 20E9       182:        BRA    XFRTTL  REPEAT
116C 2002       183:        BRA    BMPCP2
                184:
                185: * BUMP COMMAND POINTER
                186:
116E DED9       187: BMPCP  LDX    CMNPNT  GET POINTER
1170 08         188: BMPCP2 INX    BUMP    IT
1171 DFD9       189:        STX    CMNPNT  SAVE IT
1173 39         190:        RTS    RETURN
                191:
                192: * IF COMMAND .IF CONDITION .CM
                193:
1174 7F007D     194: IF     CLR    NEGT    CLEAR FLAG
1177 BD12A7     195: IF1    JSR    LDNSKP  FIND NEXT CHAR
117A 8121       196:        CMPA   #'!     IS IT A '!'
117C 2607       197:        BNE    IF3
117E 73007D     198:        COM    NEGT    SET NEG FLAG
1181 8DEB       199:        BSR    BMPCP   BUMP POINTER
1183 20F2       200:        BRA    IF1
1185 815F       201: IF3    CMPA   #$5F    IS IT LOWER CASE?
1187 2302       202:        BLS    IF35
1189 8020       203:        SUB    A       #$20 MAKE UPPER
118B 814F       204: IF35   CMPA   #'O     CHECK IF ODD
118D 2607       205:        BNE    IF4
118F 9669       206:        LDAA   PGN     GET PAGE NUMBER
1191 46         207:        ROR    A       CHECK IF ODD
1192 2428       208:        BCC    IFN
1194 2009       209:        BRA    IFY
1196 8145       210: IF4    CMPA   #'E     EVEN?
1198 2627       211:        BNE    IF6
119A 9669       212:        LDAA   PGN     GET PAGE NUMBER
119C 46         213:        ROR    A       CHECK IF EVEN
119D 251D       214:        BCS    IFN
119F 967D       215: IFY    LDAA   NEGT    CHECK NEG.
11A1 2631       216:        BNE    IF8
11A3 8DC9       217: IF5    BSR    BMPCP   BUMP POINTER
11A5 BD12A7     218:        JSR    LDNSKP  GET NEXT CHAR
11A8 7F0064     219:        CLR    NOFL    CLEEAR FLAG
11AB 812E       220:        CMPA   #'.     IS IT PERIOD?
```

```
11AD 2706       221:         BEQ    IF55
11AF 813A       222:         CMPA   #':      IS IT COLON?
11B1 260D       223:         BNE    IFN2
11B3 9764       224:         STAA   NOFL     SET NO FLUSH
11B5 08         225: IF55    INX    FIX      POINTER
11B6 DFD9       226:         STX    CMNPNT   SAVE IT
11B8 7C007E     227:         INC    IFFLG    SET IF FLAG
11BB 39         228:         RTS    RETURN
11BC 967D       229: IFN     LDAA   NEGT     CHECK NEG.
11BE 26E3       230:         BNE    IF5
11C0 39         231: IFN2    RTS    RETURN
11C1 BD12E9     232: IF6     JSR    CHKNUM   CHECK FOR NUMBER
11C4 240E       233:         BCC    IF8
11C6 DED9       234:         LDX    CMNPNT   GET POINTER
11C8 09         235:         DEX    ADJUST
11C9 09         236:         DEX
11CA DFD9       237:         STX    CMNPNT   SAVE
11CC 9665       238:         LDAA   INNUM    GET NUMBER
11CE 2BEC       239:         BMI    IFN
11D0 27EA       240:         BEQ    IFN
11D2 20CB       241:         BRA    IFY
11D4 39         242: IF8     RTS    RETURN
                243:
                244: * TEST FOR NEGATIVE NUMBER
                245:
11D5 9665       246: TSTNEG  LDAA   INNUM    GET NUMBER
11D7 2A07       247:         BPL    TSTNE2
11D9 9767       248:         STAA   SIGN     SET SIGN
11DB 9766       249:         STAA   NEG      SET NEG
11DD 700065     250:         NEG    INNUM    NEGATE NUM.
11E0 39         251: TSTNE2  RTS    RETURN
                252:
                253: * CHANGE TRAP LOCATION .CH -M -N
                254:
11E1 BD12E9     255: CHNG    JSR    CHKNUM   CHECK FOR NUMBER
11E4 2419       256:         BCC    CHNG3
11E6 8DED       257:         BSR    TSTNEG   NEGATIVE?
11E8 963F       258:         LDAA   PGL      GET PAGE LENGTH
11EA 4C         259:         INC    A
11EB BD12D8     260:         JSR    FIXVAL   FIX VALUE
11EE CE1AB7     261:         LDX    #TRAPS   POINT TO TRAPS
11F1 4D         262:         TST    A
11F2 2601       263:         BNE    CHNG2
11F4 4C         264:         INC    A        BUMP IT
11F5 A100       265: CHNG2   CMPA   0,X      TEST LOCATION
11F7 2723       266:         BEQ    CHNG5
11F9 BD0E66     267:         JSR    INTRP    BUMP POS.
11FC 26F7       268:         BNE    CHNG2
11FE 39         269: CHNG25  RTS    RETURN
11FF BD1059     270: CHNG3   JSR    GTNAM    GO GET NAME
1202 9695       271:         LDAA   MACNAM
1204 27F8       272:         BEQ    CHNG25
1206 D696       273:         LDAB   MACNAM+1
1208 CE1AB7     274:         LDX    #TRAPS   POINT TO TRAPS
120B A101       275: CHNG4   CMPA   1,X      CHECK CHAR.
```

```
120D 2604      276:         BNE    CHNG45
120F E102      277:         CMPB   2,X
1211 2709      278:         BEQ    CHNG5
1213 08        279: CHNG45  INX    BUMP     TO NEXT
1214 08        280:         INX
1215 08        281:         INX
1216 8C1AE7    282:         CPX    #TRPEND  END OF TABLE?
1219 26F0      283:         BNE    CHNG4
121B 39        284:         RTS    RETURN
121C DFE3      285: CHNG5   STX    TEMP5    SAVE POINTER
121E BD12E9    286:         JSR    CHKNUM   CHECK FOR NUMBER
1221 24DB      287:         BCC    CHNG25
1223 8DB0      288:         BSR    TSTNEG   IS IT NEG.?
1225 963F      289:         LDAA   PGL      GET PAGE LENGTH
1227 4C        290:         INC    A
1228 BD12D8    291:         JSR    FIXVAL   FIX VALUE
122B 4D        292:         TST    A
122C 2601      293:         BNE    CHNG6
122E 4C        294:         INC    A        BUMP IT
122F DEE3      295: CHNG6   LDX    TEMP5    RESTORE POINTER
1231 A700      296:         STAA   0,X      PUT CHAR
1233 39        297:         RTS    RETURN
               298:
               299: * SET NUMBER REGISTER .NR X N
               300:
1234 BD12A7    301: NREG    JSR    LDNSKP   GET TO NEXT
1237 BD12B5    302:         JSR    CLSFY    CLASSIFY IT
123A C102      303:         CMPB   #2
123C 2614      304:         BNE    NREG4
123E 36        305:         PSH    A        SAVE
123F BD116E    306:         JSR    BMPCP    BUMP POINTER
1242 BD12E9    307:         JSR    CHKNUM   CHECK FOR NUMBER
1245 32        308:         PUL    A        RESTORE
1246 240A      309:         BCC    NREG4
1248 BD1277    310:         JSR    FNDNUM   GO FIND NUMBER
124B A600      311:         LDAA   0,X      GET CHARACTER
124D BD12D8    312:         JSR    FIXVAL   FIX VALUE
1250 A700      313:         STAA   0,X      SAVE IT
1252 39        314: NREG4   RTS    RETURN
               315:
               316: * SET ARABIC MODE .AR
               317:
1253 7F00B7    318: ARB     CLR    ROM      CLEAR ROMAN
1256 39        319:         RTS    RETURN
               320:
               321: * SET FOR SMALL ROMAN .SR
               322:
1257 8680      323: SROM    LDAA   #$80
1259 97B7      324: ROM2    STAA   ROM      SET FLAG
125B 39        325:         RTS
               326:
               327: * SET FOR CAPITAL ROMAN .CR
               328:
125C 860F      329: CROM    LDAA   #$F
125E 20F9      330:         BRA    ROM2     SET FLAG
```

```
                    331:
                    332: * SET AUTO INCREMENT .AU N
                    333:
1260 BD12E9         334: SAUTO  JSR     CHKNUM  CHECK FOR NUMBER
1263 2407           335:        BCC     SAUTO4
1265 96B5           336:        LDAA    AUTO    GET OLD
1267 BD12D8         337:        JSR     FIXVAL  FIX VALUE
126A 97B5           338:        STAA    AUTO    SAVE NEW
126C 39             339: SAUTO4 RTS     RETURN
                    340:
                    341: * CLEAR NUMBER SPACE
                    342:
126D 5F             343: CLRNUM CLR     B
126E D765           344:        STAB    INNUM   CLEAR OUT NUM
1270 D754           345:        STAB    INC
1272 D74F           346:        STAB    GDNUM   SET FLAGS
1274 D752           347:        STAB    BNUM
1276 39             348:        RTS     RETURN
                    349:
                    350: * FIND NUMBER REGISTER
                    351:
1277 CE0030         352: FNDNUM LDX     #NMREGS SET POINTER
127A 8041           353:        SUB     A       #$41
127C DF55           354:        STX     NUMPNT
127E 9B56           355:        ADD     A       NUMPNT+1 ADD OFFSET
1280 9756           356:        STAA    NUMPNT+1
1282 DE55           357:        LDX     NUMPNT  GET POINTER
1284 39             358:        RTS     RETURN
                    359:
                    360: * FETCH NUMBER FROM BUFFER
                    361:
1285 DE55           362: FTCHNM LDX     NUMPNT  SET POINTER
1287 9C58           363:        CPX     LSTNUM  FINISHED?
1289 2716           364:        BEQ     FTCHN2
128B A600           365:        LDAA    0,X     GET A CHAR.
128D 847F           366:        AND     A       #$7F MASK IT
128F 08             367:        INX     BUMP    THE POINTER
1290 DF55           368:        STX     NUMPNT  SAVE IT
1292 810D           369:        CMPA    #$D     C.R.?
1294 2608           370:        BNE     FTCHN1
1296 7D005E         371:        TST     NOCR    TEST FLAG
1299 2603           372:        BNE     FTCHN1
129B 7E080B         373:        JMP     FETCH5  RETURN
129E 7E07CD         374: FTCHN1 JMP     FETCH3
12A1 7F0057         375: FTCHN2 CLR     EXCHR   CLEAR EXTRA CHAR.
12A4 7E074C         376:        JMP     GETCHR  GO GET CHAR
                    377:
                    378: * LOAD POINTER AND SKIP SPACES
                    379:
12A7 DED9           380: LDNSKP LDX     CMNPNT  SET POINTER
12A9 A600           381: LDNSK2 LDAA    0,X     GET CHARACTER
12AB 8120           382:        CMPA    #$20    IS IT SPACE?
12AD 2603           383:        BNE     LDNSK4
12AF 08             384:        INX     BUMP    TO NEXT
12B0 20F7           385:        BRA     LDNSK2
```

```
12B2 DFD9       386: LDNSK4 STX    CMNPNT  SAVE POSITION
12B4 39         387:        RTS    RETURN
                388:
                389: * CLASSIFY CHARACTER
                390:
12B5 5F         391: CLSFY  CLR    B       CLEAR SPECIFIER
12B6 4D         392:        TST    A       TEST CHAR
12B7 2B1E       393:        BMI    CLSFY4
12B9 815F       394:        CMPA   #$5F    LOWER CASE?
12BB 2306       395:        BLS    CLSFY1
12BD 817F       396:        CMPA   #$7F    TEST FOR PARITY
12BF 2216       397:        BHI    CLSFY4
12C1 8020       398:        SUB    A       #$20 MAKE UPPER CASE
12C3 8130       399: CLSFY1 CMPA   #'0     CHAR A NUMBER?
12C5 2510       400:        BCS    CLSFY4
12C7 8139       401:        CMPA   #'9
12C9 2202       402:        BHI    CLSFY2
12CB 5C         403:        INC    B       IF SO, SET
12CC 39         404:        RTS    RETURN
12CD 8141       405: CLSFY2 CMPA   #'A     IS CHAR A LETTER?
12CF 2506       406:        BCS    CLSFY4
12D1 815A       407:        CMPA   #'Z
12D3 2202       408:        BHI    CLSFY4
12D5 C602       409:        LDAB   #2      IF SO, SET
12D7 39         410: CLSFY4 RTS    RETURN
                411:
                412: * FIX NUMBER VALUE
                413:
12D8 D665       414: FIXVAL LDAB   INNUM   GET NUMBER
12DA 7D0067     415:        TST    SIGN    TEST SIGN
12DD 2708       416:        BEQ    FIXVA4
12DF 7D0066     417:        TST    NEG     TEST FOR NEG.
12E2 2701       418:        BEQ    FIXVA3
12E4 50         419:        NEG    B       NEGATE NUM
12E5 1B         420: FIXVA3 ABA    FIX     VALUE
12E6 39         421:        RTS    RETURN
12E7 17         422: FIXVA4 TBA
12E8 39         423:        RTS
                424:
                425: * CHECK FOR NUMBER
                426:
12E9 4F         427: CHKNUM CLR    A       CLEAR FLAGS
12EA 9767       428:        STAA   SIGN
12EC 9766       429:        STAA   NEG
12EE BD126D     430:        JSR    CLRNUM  CLEAR NUMBER
12F1 5C         431:        INC    B
12F2 D75E       432:        STAB   NOCR    SET FLAGS
12F4 BD12A7     433:        JSR    LDNSKP  GO TO NEXT
12F7 812B       434:        CMPA   #'+     IS IT A '+'?
12F9 2706       435:        BEQ    CHKNU2
12FB 812D       436:        CMPA   #'-     IS IT A '-'?
12FD 260B       437:        BNE    CHKNU4
12FF 9766       438:        STAA   NEG     SET NEG.
1301 08         439: CHKNU2 INX    BUMP    THE POINTER
1302 DFD9       440:        STX    CMNPNT  SAVE IT
```

```
1304 8D1F        441:          BSR     PRNUM    PROCESS NUMBER
1306 240D        442:          BCC     CHKNU6
1308 2007        443:          BRA     CHKNU5
130A 08          444: CHKNU4   INX     FIX      POINTER
130B DFD9        445:          STX     CMNPNT   SAVE IT
130D 8D32        446:          BSR     PRNU28   PROCESS NUM.
130F 2404        447:          BCC     CHKNU6
1311 8D0B        448: CHKNU5   BSR     CLRTHM   CLEAR FLAGS
1313 0D          449:          SEC
1314 39          450:          RTS     RETURN
1315 8D07        451: CHKNU6   BSR     CLRTHM   CLEAR FLAGS
1317 DED9        452:          LDX     CMNPNT   SET POINTER
1319 09          453:          DEX
131A DFD9        454:          STX     CMNPNT
131C 0C          455:          CLC
131D 39          456:          RTS     RETURN
                 457:
                 458: * CLEAR FLAGS
                 459:
131E 7F0057      460: CLRTHM   CLR     EXCHR    CLEAR THEM
1321 7F005E      461:          CLR     NOCR
1324 39          462:          RTS     RETURN
```

```
                    1: * PROCESS NUMBER
                    2:
1325 9767           3: PRNUM  STAA   SIGN     CLEAR SIGN
1327 BD126D         4:        JSR    CLRNUM   CLEAR NUMBER
132A 5C             5:        INC    B
132B D75E           6:        STAB   NOCR     SET FLAGS
132D 7F0054         7: PRNU27 CLR    INC
1330 7C006A         8:        INC    PASCHR
1333 BD074C         9:        JSR    GETCHR   GET NEXT CHAR.
1336 7D0073        10:        TST    NOEXP    DO EXPRESSIONS?
1339 2706          11:        BEQ    PRNU28
133B 7F0073        12:        CLR    NOEXP
133E 7E13E8        13:        JMP    PRNU82   JUMP AHEAD
1341 BD12B5        14: PRNU28 JSR    CLSFY    GO CLASSIFY
1344 C101          15:        CMPB   #1
1346 2505          16:        BCS    PRNU31
1348 2751          17:        BEQ    PRNUM5
134A 7E13D6        18:        JMP    PRNU73
134D 7D0065        19: PRNU31 TST    INNUM    TEST NUMBER
1350 2705          20:        BEQ    PRNU32
1352 36            21:        PSH    A
1353 9665          22:        LDAA   INNUM    GET NUMBER
1355 2058          23:        BRA    PRNUM6
1357 7F0065        24: PRNU32 CLR    INNUM    CLEAR NUMBER
135A 8123          25:        CMPA   #'#      CHECK FOR '#'
135C 2718          26:        BEQ    PRNUM4
135E 812B          27:        CMPA   #'+      IS IT '+'?
1360 2604          28:        BNE    PRNU35
1362 9750          29:        STAA   ADD      SET FOR ADD
1364 20C7          30:        BRA    PRNU27
1366 812D          31: PRNU35 CMPA   #'-      IS IT '-'?
1368 2604          32:        BNE    PRNU37
136A 9751          33:        STAA   SUB      SET FOR SUBTRACT
136C 20BF          34:        BRA    PRNU27
136E 8125          35: PRNU37 CMPA   #'%      IS IT '%'?
1370 266C          36:        BNE    PRNUM8
1372 9669          37:        LDAA   PGN      GET PAGE NUMBER
1374 2039          38:        BRA    PRNUM6
1376 7C006A        39: PRNUM4 INC    PASCHR   SET FLAG
1379 BD074C        40:        JSR    GETCHR   GET CHARACTER
137C BD12B5        41:        JSR    CLSFY    CLASSIFY IT
137F C102          42:        CMPB   #2
1381 2610          43:        BNE    PRNU45
1383 BD1277        44:        JSR    FNDNUM   GO FIND NUMBER
1386 A600          45:        LDAA   0,X      GET VALUE
1388 7D0054        46:        TST    INC      INCREMENT?
138B 2722          47:        BEQ    PRNUM6
138D 9BB5          48:        ADD    A        AUTO ADD IN AUTO
138F A700          49:        STAA   0,X      SAVE NEW
1391 201C          50:        BRA    PRNUM6
1393 812B          51: PRNU45 CMPA   #'+      IS IT '+'?
1395 2647          52:        BNE    PRNUM8
1397 9754          53:        STAA   INC      SET INC.
1399 20DB          54:        BRA    PRNUM4
139B 8030          55: PRNUM5 SUB    A        #$30 BIAS NUMBER
```

```
139D 36          56:          PSH    A
139E D665        57:          LDAB   INNUM    GET NUM
13A0 58          58:          ASL    B        ADJUST
13A1 58          59:          ASL    B
13A2 DB65        60:          ADD    B        INNUM ADD IT IN
13A4 58          61:          ASL    B
13A5 32          62:          PUL    A        RESTORE
13A6 1B          63:          ABA
13A7 9765        64:          STAA   INNUM    SAVE NEW VALUE
13A9 7C004F      65:          INC    GDNUM    SET GOOD
13AC 7E132D      66:          JMP    PRNU27   REPEAT
13AF D651        67: PRNUM6   LDAB   SUB      SUBTRACT?
13B1 2706        68:          BEQ    PRNU65
13B3 16          69:          TAB    DO       SUBTRACT
13B4 9652        70:          LDAA   BNUM
13B6 10          71:          SBA
13B7 2006        72:          BRA    PRNUM7
13B9 D650        73: PRNU65   LDAB   ADD      ADDITION?
13BB 2702        74:          BEQ    PRNUM7
13BD 9B52        75:          ADD    A        BNUM DO ADD
13BF 9752        76: PRNUM7   STAA   BNUM     SAVE NUMBER
13C1 7F0050      77:          CLR    ADD      CLEAR FLAGS
13C4 7F0051      78:          CLR    SUB
13C7 7C004F      79:          INC    GDNUM    SET GOOD
13CA 7D0065      80:          TST    INNUM    TEST NUMBER
13CD 2603        81:          BNE    PRNU72
13CF 7E132D      82:          JMP    PRNU27
13D2 32          83: PRNU72   PUL    A        RESTORE CHAR
13D3 7E1357      84:          JMP    PRNU32
13D6 7D0065      85: PRNU73   TST    INNUM    TEST NUMBER
13D9 2703        86:          BEQ    PRNUM8
13DB 36          87:          PSH    A
13DC 20D1        88:          BRA    PRNUM6
13DE 7F0073      89: PRNUM8   CLR    NOEXP    CLEAR FLAG
13E1 7D004F      90:          TST    GDNUM    TEST GOOD
13E4 2602        91:          BNE    PRNU82
13E6 0C          92:          CLC    SET      CONDITION
13E7 39          93:          RTS    RETURN
13E8 9757        94: PRNU82   STAA   EXCHR    SAVE EXTRA CHAR.
13EA CE0125      95:          LDX    #NUM     POINT TO NUMBER
13ED 9652        96:          LDAA   BNUM     GET NUMBER
13EF 9765        97:          STAA   INNUM
13F1 2704        98:          BEQ    BTOD
13F3 D6B7        99:          LDAB   ROM      ROMAN OR ARABIC?
13F5 2637       100:          BNE    BTOROM
                101:
                102: * BINARY TO ASCII ARABIC
                103:
13F7 5F         104: BTOD    CLR    B
13F8 8164       105: BTOD1   CMPA   #100     NUM > 100?
13FA 2505       106:          BCS    BTOD2
13FC 8064       107:          SUB    A        #100 SUB OFF 100
13FE 5C         108:          INC    B        BUMP NUMBER
13FF 20F7       109:          BRA    BTOD1
1401 5D         110: BTOD2   TST    B        ANY YET?
```

```
1402 2706       111:         BEQ    BTOD3
1404 CB30       112:         ADD    B       #$30 SET HUNDREDS
1406 E700       113:         STAB   0,X     SAVE
1408 08         114:         INX    GO      TO NEXT
1409 5F         115:         CLR    B       CLEAR REGISTER
140A 810A       116: BTOD3   CMPA   #10     NUMBER > 10
140C 2505       117:         BCS    BTOD4
140E 800A       118:         SUB    A       #10 SUB VALUE
1410 5C         119:         INC    B       BUMP NUMBER
1411 20F7       120:         BRA    BTOD3
1413 5D         121: BTOD4   TST    B       ANY?
1414 2705       122:         BEQ    BTOD45
1416 CB30       123:         ADD    B       #$30 ADD BIAS
1418 E700       124:         STAB   0,X     SAVE TENS
141A 08         125:         INX    BUMP    TO NEXT
141B 8B30       126: BTOD45  ADD    A       #$30 ADD IN BIAS
141D A700       127:         STAA   0,X     SAVE ONES
141F 08         128:         INX    BUMP    POINTER
1420 9657       129: BTOD5   LDAA   EXCHR   GET EXTRA
1422 A700       130:         STAA   0,X     SAVE IT
1424 08         131:         INX    BUMP    TO NEXT
1425 DF58       132:         STX    LSTNUM  SAVE POSITION
1427 CE0125     133:         LDX    #NUM    POINT TO NUMBER
142A DF55       134:         STX    NUMPNT
142C 0D         135:         SEC
142D 39         136:         RTS    RETURN
                137:
                138: * BINARY TO ASCII ROMAN
                139:
142E C643       140: BTOROM  LDAB   #'C     SET HUNDREDS
1430 8164       141: BTORO1  CMPA   #100    NUMBER > 100?
1432 2507       142:         BCS    BTORO2
1434 8064       143:         SUB    A       #100 SUBTRACT OFF
1436 E700       144:         STAB   0,X     SET 100
1438 08         145:         INX    BUMP    TO NEXT
1439 20F5       146:         BRA    BTORO1
143B 815A       147: BTORO2  CMPA   #90     CHECK FOR 90
143D 250A       148:         BCS    BTORO3
143F 805A       149:         SUB    A       #90 SUBTRACT OFF
1441 E701       150:         STAB   1,X     PUT CHARACTER
1443 C658       151:         LDAB   #'X     SET TENS
1445 E700       152:         STAB   0,X     SAVE IT
1447 08         153:         INX    BUMP    TO NEXT
1448 08         154:         INX
1449 8132       155: BTORO3  CMPA   #50     CHECK FOR FIFTY
144B 2507       156:         BCS    BTORO4
144D 8032       157:         SUB    A       #50 SUBTRACT OFF
144F C64C       158:         LDAB   #'L     SET 'L'
1451 E700       159:         STAB   0,X     SAVE IT
1453 08         160:         INX    BUMP    THE POINTER
1454 8128       161: BTORO4  CMPA   #40     CHECK FOR 40
1456 250C       162:         BCS    BTORO5
1458 8028       163:         SUB    A       #40 SUBTRACT OFF
145A C658       164:         LDAB   #'X     SET TEN
145C E700       165:         STAB   0,X     SAVE IT
```

```
145E C64C       166:          LDAB   #'L      SET 50
1460 E701       167:          STAB   1,X      SAVE IT
1462 08         168:          INX    BUMP     TO NEXT
1463 08         169:          INX
1464 C658       170: BTORO5   LDAB   #'X      SET UP 'X'
1466 810A       171:          CMPA   #10      CHECK TENS
1468 2507       172:          BCS    BTORO6
146A 800A       173:          SUB    A        #10 SUBTRACT OFF
146C E700       174:          STAB   0,X      SAVE
146E 08         175:          INX    BUMP     POINTER
146F 20F3       176:          BRA    BTORO5
1471 8109       177: BTORO6   CMPA   #9       CHECK IF 9
1473 250A       178:          BCS    BTOR65
1475 8009       179:          SUB    A        #9 SUBTRACT 9
1477 E701       180:          STAB   1,X      SAVE CHARACTER
1479 C649       181:          LDAB   #'I
147B E700       182:          STAB   0,X
147D 08         183:          INX    GET      TO NEXT
147E 08         184:          INX
147F 8105       185: BTOR65   CMPA   #5       CHECK FOR 5
1481 2507       186:          BCS    BTORO7
1483 C656       187:          LDAB   #'V      SET UP 'V'
1485 E700       188:          STAB   0,X      SAVE IT
1487 08         189:          INX    BUMP     POINTER
1488 8005       190:          SUB    A        #5 FIX VALUE
148A 8104       191: BTORO7   CMPA   #4       CHECK FOR 4
148C 250C       192:          BCS    BTORO8
148E 8004       193:          SUB    A        #4 SUBTRACT OFF
1490 C649       194:          LDAB   #'I      SET UP 'I'
1492 E700       195:          STAB   0,X      SAVE CHARACTER
1494 C656       196:          LDAB   #'V
1496 E701       197:          STAB   1,X      SAVE 'V'
1498 08         198:          INX    BUMP     POINTER
1499 08         199:          INX
149A C649       200: BTORO8   LDAB   #'I
149C 4D         201:          TST    A        TEST ONES
149D 2706       202:          BEQ    BTORO9
149F E700       203:          STAB   0,X      SAVE I'S
14A1 08         204:          INX
14A2 4A         205:          DEC    A        DONE?
14A3 20F5       206:          BRA    BTORO8
14A5 DF58       207: BTORO9   STX    LSTNUM   SAVE POINTER
14A7 96B7       208:          LDAA   ROM      CHECK IF SMALL
14A9 2A0E       209:          BPL    BTODON
14AB CE0125     210:          LDX    #NUM     RESET POINTER
14AE A600       211: BTOR92   LDAA   0,X      GET CHARACTER
14B0 8B20       212:          ADD    A        #$20 MAKE SMALL
14B2 A700       213:          STAA   0,X      PUT BACK
14B4 08         214:          INX    BUMP     TO NEXT
14B5 9C58       215:          CPX    LSTNUM   FINISHED?
14B7 26F5       216:          BNE    BTOR92
14B9 7E1420     217: BTODON   JMP    BTOD5
                218:
                219: * PUSH X ONTO STACK
                220:
```

```
14BC 32          221: PUSHX  PUL    A        GET RETURN ADR.
14BD 33          222:        PUL    B
14BE 97E7        223:        STAA   RETREG   SAVE IT
14C0 D7E8        224:        STAB   RETREG+1
14C2 DFE9        225:        STX    INDEX    SAVE X
14C4 96E9        226:        LDAA   INDEX    GET PART X
14C6 D6EA        227:        LDAB   INDEX+1
14C8 36          228:        PSH    A        PUSH ON STACK
14C9 37          229:        PSH    B
14CA DEE7        230:        LDX    RETREG   GET RETURN
14CC 6E00        231:        JMP    0,X      RETURN
                 232:
                 233: * PULL X FROM STACK
                 234:
14CE 32          235: PULLX  PUL    A        GET RETURN ADR.
14CF 33          236:        PUL    B
14D0 97E7        237:        STAA   RETREG   SAVE IT
14D2 D7E8        238:        STAB   RETREG+1
14D4 33          239:        PUL    B        PULL X
14D5 32          240:        PUL    A
14D6 97E9        241:        STAA   INDEX    SAVE X
14D8 D7EA        242:        STAB   INDEX+1
14DA 96E7        243:        LDAA   RETREG   GET RETURN ADR.
14DC D6E8        244:        LDAB   RETREG+1
14DE 37          245:        PSH    B        PUSH BACK ON
14DF 36          246:        PSH    A
14E0 DEE9        247:        LDX    INDEX    LOAD UP X
14E2 39          248:        RTS    RETURN
                 249:
                 250: * UNDERLINE COMMAND .UL
                 251:
14E3 8601        252: UNDL   LDAA   #1       SET UL FLAG
14E5 974E        253:        STAA   ULFLG
14E7 39          254:        RTS    RETURN
                 255:
                 256: *
                 257: * DISK COMMANDS FOLLOW
                 258:
                 259: * READ ITEM .RI [S]
                 260:
14E8 964D        261: RDIT   LDAA   FILOPN   FILE OPEN?
14EA 2718        262:        BEQ    RDIT4
14EC 7F0088      263:        CLR    CRSUP    CLEAR SUP FLAG
14EF 7F0036      264:        CLR    GCNT     CLEAR CHAR COUNT
14F2 9787        265:        STAA   RIFLG    SET FLAG
14F4 BD12A7      266:        JSR    LDNSKP   SKIP JUNK
14F7 BD12B5      267:        JSR    CLSFY    CHECK CHARACTER
14FA C102        268:        CMPB   #2       IS IT A LETTER?
14FC 2606        269:        BNE    RDIT4
14FE 8153        270:        CMPA   #'S      IS IT AN 'S'?
1500 2602        271:        BNE    RDIT4
1502 9788        272:        STAA   CRSUP    SET SUP FLAG
1504 39          273: RDIT4  RTS    RETURN
                 274:
                 275:
```

```
                276: * SET ITEM CHARACTER .IC C
                277:
1505 BD12A7     278: ITMCH  JSR    LDNSKP  GET NEXT
1508 810D       279:        CMPA   #$D     END OF LINE?
150A 2603       280:        BNE    ITMCH2
150C B6021B     281:        LDAA   ITEMCH  GET DEFAULT ITEM CHARACTER
150F 974C       282: ITMCH2 STAA   ITEM    SET CHARACTER
1511 39         283:        RTS    RETURN
                284:
                285: * NEXT ITEM .NI N
                286:
1512 964D       287: NXTI   LDAA   FILOPN  FILE OPEN?
1514 2721       288:        BEQ    NXTI6
1516 BD12E9     289:        JSR    CHKNUM  LOOK FOR NUMBER
1519 9665       290:        LDAA   INNUM
151B 2601       291:        BNE    NXTI2
151D 4C         292:        INC    A       SET UP ONE
151E 9789       293: NXTI2  STAA   NCOUNT  SET ITEM COUNT
1520 964B       294: NXTI3  LDAA   EORF    CHECK IF EOR
1522 2613       295:        BNE    NXTI6
1524 4C         296:        INC    A       SET NON ZERO
1525 9787       297:        STAA   RIFLG   SET FLAG
1527 BD166F     298: NXTI4  JSR    INCHR   GET CHARACTER
152A 9634       299:        LDAA   EOFF    EOF?
152C 2609       300:        BNE    NXTI6
152E 964A       301:        LDAA   EOIF    EOI?
1530 27F5       302:        BEQ    NXTI4   REPEAT TIL FOUND
1532 7A0089     303:        DEC    NCOUNT  DEC THE COUNT
1535 26E9       304:        BNE    NXTI3
1537 39         305: NXTI6  RTS    RETURN
                306:
                307: * NEXT BLOCK .NB N
                308:
1538 964D       309: NXTB   LDAA   FILOPN  FILE OPEN?
153A 2726       310:        BEQ    NXTB6
153C BD12E9     311:        JSR    CHKNUM  LOOK FOR NUMBER
153F 9665       312:        LDAA   INNUM
1541 2601       313:        BNE    NXTB2
1543 4C         314:        INCA   SET     DEFAULT
1544 9789       315: NXTB2  STAA   NCOUNT  SET COUNTER
1546 7D004B     316:        TST    EORF    CHECK FOR EOR
1549 260F       317:        BNE    NXTB5
154B 8601       318: NXTB4  LDAA   #1      SET FLAG
154D 9787       319:        STAA   RIFLG
154F BD166F     320:        JSR    INCHR   GET CHARACTER
1552 9634       321:        LDAA   EOFF    CHECK EOF
1554 260C       322:        BNE    NXTB6
1556 964B       323:        LDAA   EORF    CHECK EOR
1558 27F1       324:        BEQ    NXTB4
155A 7F004B     325: NXTB5  CLR    EORF    CLEAR FLAG
155D 7A0089     326:        DEC    NCOUNT  DEC THE COUNT
1560 26E9       327:        BNE    NXTB4   REPEAT TIL DONE
1562 39         328: NXTB6  RTS    RETURN
                329:
                330: * CLOSE FILE .CF
```

```
                    331:
1563 964D           332: CLSFL   LDAA    FILOPN   CHECK IF OPEN
1565 270D           333:         BEQ     CLSFL4
1567 CE1CF7         334:         LDX     #DFCB
156A 8606           335:         LDAA    #6       SET FOR READ CLOSE
156C A700           336:         STAA    0,X
156E BD16E4         337:         JSR     DODFM    CALL DFM
1571 7F004D         338:         CLR     FILOPN   CLEAR STATUS
1574 39             339: CLSFL4  RTS
                    340:
                    341: * OPEN FILE .OF [NAME]
                    342: *
                    343:
1575 964D           344: OPNF    LDAA    FILOPN   CHECK IF OPEN
1577 2628           345:         BNE     OPNF8
1579 BD12A7         346:         JSR     LDNSKP   GET NEXY
157C 810D           347:         CMPA    #$D      IS IT CR?
157E 2609           348:         BNE     OPNF5    NOT CR SO MUST BE SOMETHING THERE
1580 CE1869         349:         LDX     #NMST    POINT TO STRING
1583 BD1636         350:         JSR     PSTRNG   OUTPUT IT
1586 BD15D5         351:         JSR     GIBUF    GET RESPONSE
1589 CE1CF7         352: OPNF5   LDX     #DFCB    POINT TO FCB
158C BD1712         353:         JSR     YFLSPC   GET FILE NAME
158F 2511           354:         BCS     OPNF6    ERROR?
1591 CE1CF7         355:         LDX     #DFCB    POINT TO FCB
1594 8604           356:         LDAA    #4       OPEN FOR READ
1596 A700           357:         STAA    0,X
1598 BD16E4         358:         JSR     DODFM    CALL DFM
159B 8605           359:         LDAA    #5       SET FOR READ
159D A700           360:         STAA    0,X
159F 974D           361:         STAA    FILOPN   SET FLAG
15A1 39             362: OPNF8   RTS     RETURN
15A2 7E03B9         363: OPNF6   JMP     DPROC3
                    364:
                    365:
                    366: * FIX WIDTH
                    367:
15A5 96A7           368: FIXWD   LDAA    TLLN     GET TEMP LENGTH
15A7 9B3B           369:         ADD     A        LLN ADD TO LENGTH
15A9 973B           370:         STAA    LLN      SAVE NEW
15AB 96A6           371:         LDAA    TIND     GET TEMP IND.
15AD 9B38           372:         ADD     A        IND ADD TO INDENT
15AF 9738           373:         STAA    IND      SAVE NEW
15B1 96A5           374:         LDAA    TSIN     GET TEMP SIND.
15B3 9B70           375:         ADD     A        SIN ADD TO SIND.
15B5 9770           376:         STAA    SIN      SAVE NEW
15B7 4F             377:         CLR     A        CLEAR OLD VALUES
15B8 97A7           378:         STAA    TLLN
15BA 97A6           379:         STAA    TIND
15BC 97A5           380:         STAA    TSIN
15BE 963B           381:         LDAA    LLN      GET LINE LENGTH
15C0 9038           382:         SUB     A        IND SUB INDENT
15C2 9070           383:         SUB     A        SIN SUB S IND.
15C4 810E           384:         CMPA    #14      LESS THAN 15?
15C6 2202           385:         BHI     FIXWD2
```

```
15C8 860F       386:         LDAA   #15      FORCE TO 15
15CA 8196       387: FIXWD2  CMPA   #150     >150?
15CC 2302       388:         BLS    FIXWD3
15CE 8696       389:         LDAA   #150
15D0 97B9       390: FIXWD3  STAA   WIDTH    SAVE NEW WIDTH
15D2 7E0715     391:         JMP    FIXBFE   GO FIX
                392:
                393: * GET INPUT CHARACTERS
                394:
15D5 CE1A53     395: GIBUF   LDX    #SBUF    POINT TO BUFFER
15D8 5F         396:         CLR    B        CLEAR COUNT
15D9 37         397: GIBUF2  PSH    B
15DA BD0206     398:         JSR    INCH     GET CHARACTER
15DD 33         399:         PUL    B
15DE 8118       400:         CMPA   #$18     CONTROL X?
15E0 271C       401:         BEQ    GIBUF6
15E2 8115       402:         CMPA   #$15     OR CONTROL U?
15E4 2718       403:         BEQ    GIBUF6   TREAT SAME AS ^X
15E6 810D       404:         CMPA   #$D      C.R.?
15E8 270A       405:         BEQ    GIBUF4
15EA 811F       406:         CMPA   #$1F     CONTROL CHAR.?
15EC 23EB       407:         BLS    GIBUF2
15EE 5C         408:         INC    B        BUMP THE COUNT
15EF A700       409:         STAA   0,X      PUT CHARACTER
15F1 08         410:         INX    BUMP     THE POINTER
15F2 20E5       411:         BRA    GIBUF2   REPEAT
15F4 A700       412: GIBUF4  STAA   0,X      PUT CHARACTER
15F6 CE1A53     413:         LDX    #SBUF    FIX POINTER
15F9 DFD9       414:         STX    CMNPNT   SAVE IT
15FB D736       415:         STAB   GCNT     SAVE COUNT
15FD 39         416:         RTS    RETURN
15FE CE1804     417: GIBUF6  LDX    #QUSTR   POINT TO STRING
1601 BD1636     418:         JSR    PSTRNG   OUTPUT IT
1604 20CF       419:         BRA    GIBUF
                420:
                421: * TEST FOR BREAK
                422:
1606 FE0213     423: TSTBRK  LDX    ACIADR   GET ADDR OF CONSOL ACIA
1609 270A       424:         BEQ    TSTPIA   0 => PIA, NOT ACIA
160B A600       425:         LDAA   0,X      READ ACIA CONTROL REGISTER
160D 8401       426:         ANDA   #1       TEST RECEIVER BUFFER FULL
160F 2710       427:         BEQ    TSTBR5   NOTHING YET
1611 A601       428:         LDAA   1,X      READ CHARACTER RECEIVED
1613 200A       429:         BRA    TSTBR6   CHECK FOR ^C
                430:
1615 B68004     431: TSTPIA  LDAA   $8004    GET STATUS
1618 44         432:         LSRA           CHECK
1619 2501       433:         BCS    TSTBR4
161B 39         434: TSTBR2  RTS    RETURN
                435:
161C B68005     436: TSTBR4  LDAA   $8005    GET CHARACTER
161F 847F       437: TSTBR6  ANDA   #$7F     MASK CHAR
1621 8103       438: TSTBR5  CMPA   #3       IS IT ^C?
1623 26F6       439:         BNE    TSTBR2
1625 CE1834     440:         LDX    #BRKSTR  POINT TO STRING
```

```
1628 7E0DA4     441:         JMP     STOP1    OUTPUT IT
                442:
                443: * OUTPUT A C.R. AND L.F.
                444:
162B DFEB       445: CRLF    STX     XTEMP    SAVE X REG.
162D CE17E5     446:         LDX     #CRLFST  POINT TO STRING
1630 BD1638     447:         JSR     PDATA    OUTPUT IT
1633 DEEB       448:         LDX     XTEMP    RESTORE X
1635 39         449:         RTS     RETURN
                450:
                451: * PRINT STRING
                452:
1636 8DF3       453: PSTRNG  BSR     CRLF     OUTPUT CR & LF
                454:
                455: * PRINT DATA
                456:
1638 A600       457: PDATA   LDAA    0,X      GET A CHARACTER
163A 8104       458:         CMPA    #4       IS IT TERM?
163C 2706       459:         BEQ     PDATA2
163E BD0203     460:         JSR     OUTCH    OUTPUT IT
1641 08         461:         INX     MOVE     TO NEXT
1642 20F4       462:         BRA     PDATA    REPEAT
1644 39         463: PDATA2  RTS     RETURN
                464:
                465: * OUTPUT CHARACTER
                466:
1645 D685       467: OUTCHR  LDAB    DIVFLG   DIVERTING?
1647 2706       468:         BEQ     OUTCH2
1649 7C0086     469:         INC     DIVFL2   SET FLAG
164C 7E0F7D     470:         JMP     OUTMAC   OUT TO MACRO
164F D66D       471: OUTCH2  LDAB    NOOUT    DO OUTPUT?
1651 2701       472:         BEQ     OUTCH3
1653 39         473:         RTS     RETURN
1654 4D         474: OUTCH3  TST     A        CHECK PARITY
1655 2A0C       475:         BPL     DOOUT
1657 81A0       476:         CMPA    #$A0     IS IT SPACE
1659 2708       477:         BEQ     DOOUT
165B 8D06       478:         BSR     DOOUT    OUTPUT CHAR
165D 8608       479:         LDAA    #$8      SET UP BACKSPACE
165F 8D02       480:         BSR     DOOUT    OUTPUT IT
1661 865F       481:         LDAA    #'_      SETUP UNDER LINE
1663 847F       482: DOOUT   ANDA    #$7F     MASK CHAR.
1665 D68F       483:         LDAB    PRNTR    TO PRINTER?
1667 2603       484:         BNE     DOOUT2
1669 7E0203     485:         JMP     OUTCH    OUT TO TERM
166C 7E020F     486: DOOUT2  JMP     POUCH    OUT TO PRINTER
```

```
                    1:
                    2: * INPUT A CHARACTER
                    3:
166F 37             4: INCHR   PSH     B
1670 DFEB           5:         STX     XTEMP
1672 7D0087         6:         TST     RIFLG
1675 260C           7:         BNE     DATIN
1677 CE1C51         8:         LDX     #TFCB     POINT TO TEXT FCB
167A DF8B           9:         STX     INFCB     SET INPUT FCB
167C BD16D3        10:         JSR     DREAD     DO DISK READ
167F DEEB          11: INCHR2 LDX     XTEMP     RESTORE X
1681 33            12:         PULB    RESTORE B
1682 39            13:         RTS     RETURN
                   14:
                   15: * DATA IN FROM DISK
                   16:
1683 CE1CF7        17: DATIN  LDX     #DFCB
1686 DF8B          18:         STX     INFCB     SET DATA FCB
1688 964B          19:         LDAA    EORF      CHECK FOR EOR
168A 2708          20:         BEQ     DATIN3
168C 7F0087        21: DATIN2 CLR     RIFLG     CLEAR MODE
168F 33            22: DATI25 PUL     B         RESTORE REGS
1690 DEEB          23:         LDX     XTEMP
1692 20DB          24:         BRA     INCHR     DO IN CHAR
1694 BD16D3        25: DATIN3 JSR     DREAD     DO DISK READ
1697 260D          26:         BNE     DATI35    EOF?
1699 914C          27:         CMPA    ITEM      IS IT ITEM CHAR?
169B 2624          28:         BNE     DATIN6
169D 7D004A        29:         TST     EOIF      TST EOI FLAG
16A0 2709          30:         BEQ     DATIN4
16A2 8601          31:         LDAA    #1        SET EOR FLAG
16A4 974B          32:         STAA    EORF
16A6 4F            33: DATI35 CLR     A
16A7 9787          34:         STAA    RIFLG     CLEAR MODE
16A9 20D4          35:         BRA     INCHR2    RETURN
16AB 8601          36: DATIN4 LDAA    #1
16AD 974A          37:         STAA    EOIF      SET EOI FLAG
16AF 9688          38:         LDAA    CRSUP     SUP ON?
16B1 2707          39:         BEQ     DATIN5
16B3 97B2          40:         STAA    ENDLIN    SET END LINE
16B5 7F0088        41:         CLR     CRSUP     CLEAR FLAG
16B8 20D2          42:         BRA     DATIN2
16BA 860D          43: DATIN5 LDAA    #$D       SETUP CR
16BC 7F0087        44:         CLR     RIFLG     CLEAR MODE
16BF 200E          45:         BRA     DATIN8
16C1 D64A          46: DATIN6 LDAB    EOIF      CHECK EOI
16C3 2707          47:         BEQ     DATIN7
16C5 810D          48:         CMPA    #$D       IS IT CR?
16C7 27C6          49:         BEQ     DATI25
16C9 7F004A        50:         CLR     EOIF
16CC 7C0036        51: DATIN7 INC     GCNT      BUMP CHAR COUNTER
16CF 33            52: DATIN8 PUL     B         RESTORE B
16D0 DEEB          53:         LDX     XTEMP     RESTORE X
16D2 39            54:         RTS     RETURN
                   55:
```

```
                    56: * DISK READ CHARACTER
                    57:
16D3 DE8B           58: DREAD   LDX     INFCB    SET FCB
16D5 8D0D           59:         BSR     DODFM    CALL DFM
16D7 2604           60:         BNE     DREAD2
16D9 811A           61:         CMPA    #$1A     END OF FILE?
16DB 2605           62:         BNE     DREAD4
16DD C601           63: DREAD2  LDAB    #1       SET EOF FLAG
16DF D734           64:         STAB    EOFF
16E1 39             65:         RTS     RETURN
16E2 5F             66: DREAD4  CLRB    SET      ZERO
16E3 39             67:         RTS     RETURN
                    68:
                    69: * DO DFM CALL
                    70:
16E4 BD0238         71: DODFM   JSR     DFM      CALL DFM
16E7 2601           72:         BNE     DODFM2   ERROR?
16E9 39             73:         RTS     RETURN
16EA A601           74: DODFM2  LDAA    1,X
16EC 8106           75:         CMPA    #6       IS IT EOF?
16EE 2603           76:         BNE     DODFM4
16F0 861A           77:         LDAA    #$1A     SET EOF CHAR
16F2 39             78:         RTS     RETURN
16F3 BD0232         79: DODFM4  JSR     ZTYPDE   REPORT ERROR
16F6 BD0235         80:         JSR     CDFM     CLOSE DFM
16F9 7E0209         81:         JMP     MON      RETURN TO DOS
                    82:
                    83: * REWIND FILE
                    84:
16FC CE1C51         85: RWND    LDX     #TFCB    POINT TO FCB
16FF 8606           86:         LDAA    #6       CLOSE FILE
1701 A700           87:         STAA    0,X
1703 BD16E4         88:         JSR     DODFM    CALL DFM
1706 8604           89:         LDAA    #4       OPEN FOR READ
1708 A700           90:         STAA    0,X
170A BD16E4         91:         JSR     DODFM    CALL DFM
170D 8605           92:         LDAA    #5       SET FOR READ
170F A700           93:         STAA    0,X
1711 39             94:         RTS     RETURN
                    95:
                    96: * PR'S VERSION FOR DOS68'S "ZFLSPC"
  0003              97: XFN     EQU     $03      OFFSET TO FILE NAME
  0026              98: XDB     EQU     $26      CONTROL INFO LENGTH
  0002              99: XUN     EQU     $02      OFFSET TO UNIT #
                   100:
1712 CE1CF7        101: YFLSPC  LDX     #DFCB    CLEAR OUT THE FCB
1715 C626          102:         LDAB    #XDB
1717 6F00          103:         CLR     0,X
1719 08            104:         INX
171A 5A            105:         DECB
171B 26FA          106:         BNE     *-4
                   107:
171D DED9          108:         LDX     CMNPNT   CHECK FOR UNIT #
171F A601          109:         LDAA    1,X      PEEK AHEAD
1721 813A          110:         CMPA    #':      UNIT DELIMITER?
```

```
1723 2611       111:         BNE    YFL1     NO UNIT #
1725 A600       112:         LDAA   0,X      GET UNIT #
1727 8030       113:         SUBA   #'0
1729 251D       114:         BCS    FSFX     IF < '0 THEN ERROR
172B 8102       115:         CMPA   #2       IF > '2 THEN ERROR
172D 2219       116:         BHI    FSFX
172F B71CF9     117:         STAA   DFCB+XUN    0,1, OR 2; USE AS UNIT #
1732 08         118:         INX
1733 08         119:         INX             GET RID OF "N:"
1734 DFD9       120:         STX    CMNPNT
                121:
1736 CE1CF7     122: YFL1    LDX    #DFCB    GET POINTER FOR STORING FILE NAME
1739 C607       123:         LDAB   #7       MAX CHARACTERS+1
173B 8D1F       124: FS5     BSR    YGNCHR   GET NEXT CHAR FROM FILE NAME/EXT
173D BD022F     125:         JSR    ZANCHK   ALPHA-NUMERIC?
1740 2508       126:           BCS  FS7      NO, END OF FIELD
1742 A703       127:         STAA   XFN,X    NOT END, STORE AS PART OF FIELD
1744 08         128:         INX
1745 5A         129:         DECB            CHECK FOR OVERFLOW
1746 26F3       130:         BNE    FS5
1748 0D         131: FSFX    SEC             ERROR EXIT
1749 39         132:         RTS
                133:
174A 812E       134: FS7     CMPA   #'.      END OF FILE NAME?
174C 260C       135:         BNE    FSOKX    NO, MUST END OF WHOLE THING
174E CE1CFD     136:         LDX    #DFCB+6  PTR FOR STORING FILE EXTENSION
1751 C604       137:         LDAB   #4       MAX EXT SIZE+1
1753 7D1D00     138:         TST    DFCB+XFN+6  HAVE WE BEEN HERE BEFORE?
1756 27E3       139:         BEQ    FS5      NO, MUST BE WORKING ON EXT NOW
1758 20EE       140:         BRA    FSFX     OOPS, CANNOT HAVE "FILENAME.EXT.E
                141:
175A 0C         142: FSOKX   CLC             OK EXIT
175B 39         143:         RTS
                144:
175C DF8D       145: YGNCHR  STX    XTEMP2   SAVE FCB PTR
175E DED9       146:         LDX    CMNPNT   -> INPUT STRING
1760 A600       147:         LDAA   0,X      GET NEXT CHAR TO PROCESS
1762 810D       148:         CMPA   #13      CR?
1764 2703       149:         BEQ    YGNC     YES, DON'T ADVANCE
1766 08         150:         INX             ADVANCE
1767 DFD9       151:         STX    CMNPNT
1769 DE8D       152: YGNC    LDX    XTEMP2   RESTORE X REG
176B 39         153:         RTS
```

```
                    1: * STRINGS
                    2:
176C 20202053       3: BANNER FCC     "   SMOKE SIGNAL BROADCASTING"
1788 0D0A           4:        FCB     13,10
178A 20203638       5:        FCC     "  6800 TEXT PROCESSING SYSTEM"
17A7 0D0A           6:        FCB     13,10
17A9 20202020       7:        FCC     "          VERSION "
17BB 04             8:        FCB     4
                    9:
17BC 44415445      10: DATSTR FCC     'DATE (MM:DD:YY)? '
17CD 04            11:        FCB     4
17CE 54595045      12: PRQU   FCC     'TYPE P FOR PRINTER... '
17E4 04            13:        FCB     4
17E5 0D0A0000      14: CRLFST FCB     $D,$A,0,0,0,0,4
17EC 50414745      15: PGSTR  FCC     'PAGE LIMITS? '
17F9 04            16:        FCB     4
17FA 07            17: STPSTR FCB     7
17FB 53544F50      18:        FCC     'STOP...'
1802 0704          19:        FCB     7,4
1804 3F20          20: QUSTR  FCC     '? '
1806 0704          21:        FCB     7,4
1808 4C494E45      22: LPPSTR FCC     'LINES PER SCREEN? '
181A 04            23:        FCB     4
181B 2A2A2A2A      24: OVFSTR FCC     '**** MACRO OVERFLOW ****'
1833 04            25:        FCB     4
1834 2E2E4252      26: BRKSTR FCC     '..BREAK..'
183D 04            27:        FCB     4
183E 494C4C45      28: ILFN   FCC     'ILLEGAL FILE NAME'
184F 04            29:        FCB     4
1850 4348414E      30: CHST   FCC     'CHANGE DISKS AND HIT KEY'
1868 04            31:        FCB     4
1869 44415441      32: NMST   FCC     'DATA FILE NAME? '
1879 04            33:        FCB     4
187A 4D414352      34: MACST  FCC     'MACRO'
187F 00000000      35:        FCB     0,0,0,0
                   36:
                   37: *PRINTER ROUTINES
                   38:
1883 39            39: PRNIT  RTS     **      REPLACE WITH OWN
1884 001F          40:        RMB     31
                   41:
18A3 39            42: PROUCH RTS     **      REPLACE WITH OWN
18A4 001F          43:        RMB     31
                   44:
                   45: * BUFFER STORAGE AREA
                   46:
18C3 009B          47: LINBUF RMB     155
195E 002D          48: EXTBUF RMB     45
198B 00C8          49: LINBU2 RMB     200
1A53 0064          50: SBUF   RMB     100
1AB7 0030          51: TRAPS  RMB     48
1AE7 0002          52: TRPEND RMB     2
1AE9 00B4          53: CMNDBF RMB     180
1B9D 00B4          54: TTLBUF RMB     180
1C51 00A6          55: TFCB   RMB     166
```

```
1CF7 00A6          56: DFCB    RMB     166
1D9D 0100          57: MACTBL  RMB     256
1E9D 0002          58: MTEND   RMB     2
                   59:
  1E9F             60: ENDTP   EQU     *       END OF FIXED TEXT PR. SPACE
                   61:
                   62:         END
```

*** No Errors.

THE TEXT PROCESSOR DISC

The disc containing the text processor contains the following three versions of the text processor:

PR.$
PRA.$
PRC.$

These three files differ only in that "PR.$" is set up to use DOS68 in the $6000-$7FFF range, "PRA.$" is set up to use DOS68 in the $A000-$BFFF range, and "PRC.$" is set up to use DOS68 in the $C000-$DFFF range.