Reports and Technical Reports	All Technical Reports Collection

1976-01

TPS, A Text Processing System: Primer and Reference Manual

Barksdale, Gerald Leonard

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/30256

NPS72Ba76011

NAVAL POSTGRADUATE SCHOOL
Monterey, California

TPS, A TEXT PROCESSING SYSTEM

Primer and Reference Manual

by

G. L. Barksdale, Jr.

W. B. Meyer

January 1976

Approved for public release; distribution unlimited.

Prepared for: Naval Postgraduate School
Monterey, California

NAVAL POSTGRADUATE SCHOOL
Monterey, California
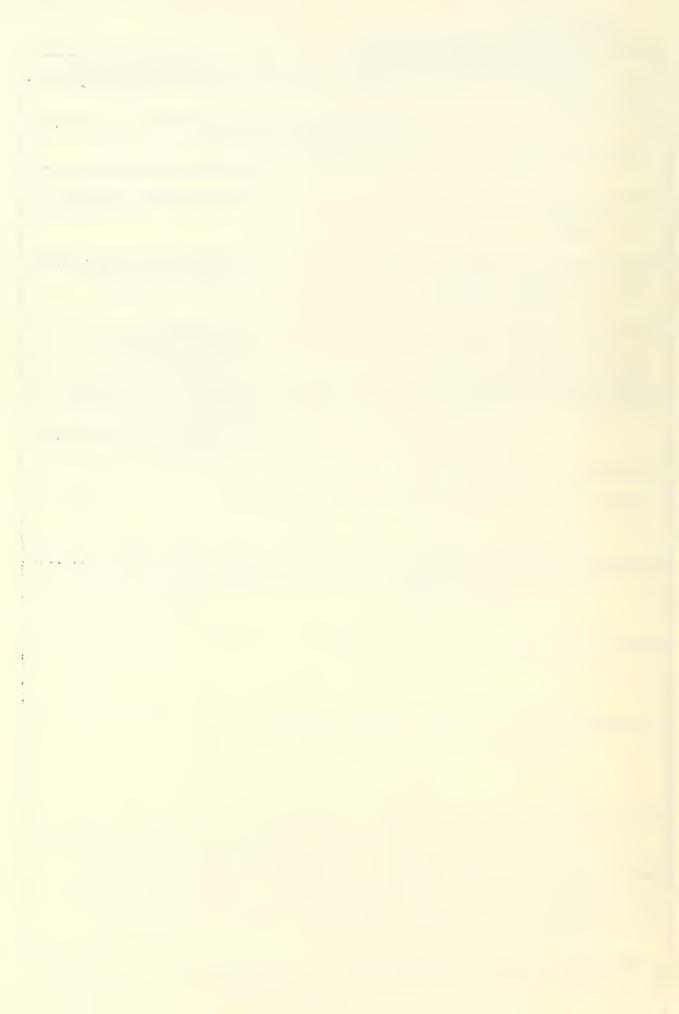
Rear Admiral Isham Linder
Superintendent

Jack R. Borsting
Provost

Reproduction of all or part of this report is authorized.

This report was prepared by:

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS72BA76011	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TPS, A TEXT PROCESSING SYSTEM - PRIMER AND REFERENCE MANUAL		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) G. L. Barksdale, Jr. and W. B. Meyer		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Computer Science Group (Code 72) Naval Postgraduate School Monterey, California 93940		12. REPORT DATE 22 January 1976
		13. NUMBER OF PAGES 77
14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

TPS is a card-based word processor for use with the IBM OS/MVT operating system. TPS commands are imbedded in the free-format text input stream. User defined macros (with arguments) and a report generator macro library, (TPSLIB) provide automated formatting facilities similiar to those available from a competent technical typist. This report is a primer for TPS and TPSLIB; appendices provide detailed reference information and examples.

DD FORM 1 JAN 73 1473   EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TPS, A TEXT PROCESSING SYSTEM

Primer and Reference Manual

Computer Science Group
Naval Postgraduate School
Monterey, CA 93940

.

G. L. Barksdale, Jr.
W. B. Meyer

TABLE OF CONTENTS

I. <u>TPS, A TEXT PROCESSING SYSTEM</u>

TPS is a word processing program which runs on the IBM
360 under the OS/MVT operating system. TPS commands allow
total user ccntrol over text placement on the output page,
as well as providing many services which aid the user in the
preparaticn of indices, tables, and equations.   The
free-format input for TPS is prepared on standard
keypunches; upper/lower case output is produced by TPS.
This document is an example of the formatting capabilities
of TPS.

In addition to its basic commands, TPS provides a
macro-definition facility which allows the user to define
arbitrarily complex TPS command sequences (with run-time
parameters). Using this facility, a library of "standard"
macros has been developed.   The macros defined in TPSLIB
provide the author of a report, technical paper, or thesis
with automated formatting facilities similiar to those which
would be provided by a competent technical typist.

A. INPUT CONVENTIONS

In its simplest form, TPS requires user knowledge of
three "commands":

-PG Print the rest of the current line buffer and skip to
     the top of the next page.

-PP Print   the   current line buffer and begin a new

paragraph, skipping and indenting.

/ "Capitalize" the letter which follows.

Let us consider a simple example of keypunched input
which illustrates these rules:

****

-PG-PP/THIS IS A SIMPLE EXAMPLE OF
COMMAND USAGE IN /T/P/S. /IT WILL RESULT
IN THE TWO PARAGRAPHS WHICH FOLLOW.
-PP/A NEW PARAGRAPH FOR THE EXAMPLE.

This is a simple example of command usage in TPS. It
will result in the two paragraphs which follow.

A new paragraph for the example.

****

Example 1

In Example 1, the card input did not fill all 80
columns of the card. Note that all the blanks between the
last word of card 1 (OF) and the first word of card 2
(COMMAND) have been compressed to a single space in the
output. This "blank squeezing" will always occur unless
commands are given to suppress it.

Since few keypunchers can produce perfect output, a
correction feature is provided in TPS. It gives you a
simple way to correct errors as you punch cards. It won't
solve all your problems, but it can save you a lot of card
duplication. If you discover that you have made an error,
you can delete characters in error by punching the "¢"
character. Thus AM¢ND is equivalent to AND, and TJIS¢¢HAT
is THAT. You may not use the "¢" to correct errors on the
previous card, however.

B. OUTPUT

You have probably been wondering how to get upper and
lower case output. To do so requires operator installation
of a special (TN) print train on one of the high-speed line
printers.   This interrupts normal service, and is therefore
generally available only once per day (in the very early
morning).   Two output classes have been established for TN
output: SYSOUT=M is printed on regular lined paper and
SYSOUT=N is printed on unlined bond.

Since the TN train has numerous "special" characters for
which the keypunch has no keys, their use is made possible
by "capitalizing" available characters, e.g., /+ is ± and /=
is ≠. A list of the upper/lower case character relations is
given below.

          ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
          abcdefghijklmnopqrstuvwxyz0123456789
          ⌐ ⌐•-∂°∎{/±⌐)+—+}≤≥⌐[−]≠!⟨   UPPER CASE
          #,$.-∂%*</+_)¢|&>:;¬'?"=!(lower case

A typical user produces several draft versions of his
document. Draft versions of TPS output may be printed with
the normal printer (PN train, SYSOUT=A) thus allowing you to
proofread your text (debug your TPS input) without waiting
for the overnight turnaround of the TN output.

C. A GENERAL OVERVIEW OF TPS FACILITIES

TPS provides control over many different aspects of text
placement. Expressed in general categories, they are:

• LINE CONTROL: centering and justification

- MARGIN CONTROL: indentation, left and right margins
- DUMP AND SKIP: line dump, paragraphing, paging, skips
- PAGE NUMBERING: mode and location of page numbers
- REVISION FLAGGING: documentation aids
- INDEXING: index and table of contents generation
- FORMULAS: superscript and subscript aids
- MACROS: textual substitution with parameters
- DEBUGGING: TPS aids
- BASIC FUNCTIONS: commands having local effect
- GLOBAL CONTROL: global functions

Command definitions have been grouped according to the above divisions in Appendix A. The paragraph at the beginning of each section of Appendix A provides an explanation of the TPS environment pertinent to the command group; reading these paragraphs is probably the best way to discover the range of facilities which TPS provides. An alphabetical index to the commands is given in Appendix B.

Default values of control settings are shown in Appendix C. Appendix D lists the PARM values which may be specified to control TPS execution.

D. TPSLIB

TPSLIB is a set of macros written using commands from TPS which automatically perform the formatting and indexing of the title page, abstract, table of contents, references, and appendices for papers, reports, and theses. By using TPSLIB, the processing power of some of the more advanced features of TPS may be easily utilized by the author.

1. <u>Punching your input deck</u>

Since TPS reads cards continuously (column 1 is a continuation of the field ending in column 80 of the preceding card), non-blank characters in columns 80 and 1 are joined together. In addition, TPS treats a sequence of blank characters as though it were but a single blank, thus squeezing out extra blanks in the input. To achieve line justification (even left and right margins), TPS regards blanks as "paddable" characters; this sometimes results in the expansion of a single blank in the input text into two (or even more) blanks on output. This implies that you may leave any spacing that you like between two words as long as there is at least one blank space; however, if you need an exact number of blank spaces in the text, you should use the commands -FWnn&, -TB, or use as many special characters [&] as you need blank spaces; this character will be transformed into blanks. If you need [&] in your text, you should use the -TRcc command properly (be sure you are not using a TPSLIB macro, because these macros take advantage of this character). Except for the ordering of the words, there is generally no correlation between the placement of words on the input cards and their position on the output. Input is truly free-format.

2. Macros

Macros are a set of instructions designed to perform a specific task. They can be viewed as subroutines. The number of arguments required by the macro (0 or more) is specified at the time the macro is defined; when calling a TPSLIB macro, the user must provide input for all the required arguments. Macros are called using the character [@] followed by the macro name; TPSLIB macro names are one character long and are listed in appendices F, G and E, along with the function they perform.

7

When calling macros that require arguments, be sure not to leave any blank space between the macro name and the argument. The special character [!] is used to indicate the end of each argument. Make sure that you don't leave any blanks between the end of each argument and the [!], or indentation and skipping may not be properly performed. Some macros will automatically capitalize letters; refer to appendixes that present a deck set up for a thesis, with all available macros called (appendix H), and the output that they cause (appendix I).

3. New Macros

You might want to build some macros for your own use. And if you do, they may be placed anywhere after the TPSLIB macros.

Scme suggestions for the adventuresome:

* get a listing of the current TPSLIB macros by running TPS with the PARM='I' option;

* make sure you read Appendix A;

* use appendix F to determine which characters are still available for macro names;

* if you wish to replace a previously used macro definition with one of your own, purge the original definition with -MPn unless you know that your definition uses no more macro storage than the TPSLIB definition;

* since macro storage is almost full when using TPSLIB, you will prcbably need to use the MSIZE parameter (Appendix D);

* if you need to use cells, use only cells from 100 to 599; the remaining cells are used by TPSLIB.

E. DECK SET UP

     To use TPSLIB, select the appropriate deck setup from
the following list:

     1. Upper Case Only


```
// (normal job card) (Quick turnaround debugging)
// EXEC PGM=TPS
//STEPLIB DD DSN=SYS3.TPS,DISP=SHR,UNIT=2314,VOL=SER=LINDA
//SYSPRINT DD SYSOUT=A
//SYSIN  DD DSN=SYS3.TPSLIB,DISP=SHR,VOL=SER=LINDA,UNIT=2314
//       DD *
   (manuscript)
/* (orange card)
```


     2. Upper and Lower Case


```
// (normal job card) (Overnight turnaround debugging)
// EXEC PGM=TPS,PARM='TN'
//STEPLIB DD DSN=SYS3.TPS,DISP=SHR,UNIT=2314,VOL=SER=LINDA
//SYSPRINT DD SYSOUT=M
//SYSIN  DD DSN=SYS3.TPSLIB,DISP=SHR,VOL=SER=LINDA,UNIT=2314
//       DD *
   (manuscript)
/* (orange card)
```


     3. Upper/Lower Case Bond Paper


```
// (normal job card) (Overnight final run)
// EXEC PGM=TPS,PARM='TN'
//STEPLIB DD DSN=SYS3.TPS,DISP=SHR,UNIT=2314,VOL=SER=LINDA
//SYSPRINT DD SYSOUT=N
//SYSIN DD DSN=SYS3.TPSLIB,DISP=SHR,VOL=SER=LINDA,UNIT=2314
```

```
//        DD *
    (manuscript)
/* (orange card)
```

To use TPS without TPSLIB replace the //SYSIN DD DSN=SYS3.TPSLIB card with a //SYSIN DD * card in each of the above decks. TPS automatically provides half-track 2314 VBS blocking on the output data set (SYSPRINT) if no blocking is explicitly specified.

F. KEYPUNCHING SERVICES

The normal keypunching services of the W. R. Church Computer Center are available for TPS users. Please note, however, that keypunch jobs must consist of 40 or fewer pages of text. Standard first in, first out service will be given to a user's first TPS keypunch job submitted each day; time-available punching will be done on any TPS keypunch requests in excess of the 40 page limit. No priority keypunching service is available for TPS users.

II. TIPS ON TPSLIB USAGE

This section will show the usage of some macros available in TPSLIB and will give some hints for effective use of TPS.

The title of this section was made with macro [1], and was called as

@1TIPS ON TPSLIB USAGE!

As you can see, macro 1 is used for first order headers; it automatically skips to a new page, capitalizes, indents and indexes properly.

A. HELPFUL HINTS

This title was made with macro [2]; observe the indentation, indexing and automatic capitalizing of letters; it was called as:

@2HELPFUL HINTS!

Since TPS and TPSLIB are a text processing language, there are certain procedures which are useful (and necessary) knowledge to users of advanced features of the system. The following ideas have proved useful in making TPS do as much work as possible during the preparation of your manuscript.

* macro [X] was used to print this star and skip an extra line;

* macro [Y] was used to print this one;

    ▪ tc print this square, macro [Z] was used; it takes no arguments.

    * when calling macros that automatically store page number (see Appendix E), it is convienent to use a separate card for the macro call (with its arguments). When ycu are through punching this card, duplicate it for the table of contents (changing to the appropriate contents macro name). Remember that all macros that register the page number and appear in the text must also appear in the table of contents in the same order; as an example, if you call a macro like

@1INTRODUCTION!

you should, at the same time, punch another card like

@AINTRODUCTION!

this second card will form the table of contents entry;

    * "TABLE OF CONTENTS" (@C) and contents entries must appear as the last element of the deck;

    * "SPECIAL ABSTRACT" should be run separately.

    * "LIST OF FIGURES" must appear after the appendixes, and cannot appear before any figure that is placed in the manuscript text;

    * try to begin new phrases, paragraphs, etc, on new cards; it will be easier to make corrections later;

    * cell number 099 always contains the current page number, and can be referenced or manipulated by the user;

    * restrictions:

        ▪ the manuscript should not have more than 999 pages;

        ▪ the "TABLE OF CONTENTS" should not have more than 200 entries;

        ▪ these restrictions may be changed by modifying the TPSLIB macros;

B.   ADVANTAGES

TPSLIE provides a   convenient   mechanism   for   producing both draft and final versions of the manuscript. Since all indexing is done automatically, you will be able to insert, delete, or reword any section, paragraph, or figure without having to rearrange the indexing or the table of contents. In addition, only those portions of the manuscript which were changed need be proofread again.

TPSLIB output is accepted by the Thesis Office in lieu of a final typed copy (assuming you follow the normal format requirements).

C.   TPS CCMMANDS

The following TPS commands are usually required in the course of preparing a long document. For a complete discussion of TPS commands refer to Appendix A. An index to the commands is found in Appendix B.

-DB - dump the buffer and begin a new line;

-SKnnn - dump the buffer and begin a new line after skipping nnn lines;

-PP - dump the buffer, skip one line, indent properly to begin a new paragraph;

-PG - dump the buffer and skip to the top of next page;

-EM - all text after this command will be evenly distributed on each line, and aligned with left and right margins, until it is found cne of the three next commands; all those four commands are global;

-CN - text will be evenly distributed around the center column between right and left margins;

-LJ - text will be left justfied;

-RJ - text will be right justified until it is found one of the three above commands;

-UC - make characters upper case until a -LC command is encountered;

-LC - make characters lower case;

/ - capitalize if in lower case mode and will make the character lower case if in upper case mode;

-TB - tabulate the buffer pointer according to some default values; (c.f., -ATnnn for absolute positioning).

III.   TPS HISTORY AND CURRENT STATUS

As any large and complex program grows, it goes through numerous phases. The program generations represented by the present TPS and TPSLIB systems are discussed in the following sections.

A.   HISTORY

The immediate ancestor of TPS is the FMS system developed at UCLA in the late 1960's. Originally designed for use on an IBM 360/20 with 16K bytes of storage, the FMS system was subsequently converted to run in an OS/MVT environment (it fit easily into a minumum size region). This implementation forms the basis for the present TPS system.

In late 1972, work on the present TPS system was begun by G. L. Barksdale, Jr. with design goals of enhancing the ease with which text could be prepared and maintained using only a keypunch and batch processing facilities. Features of the TPS system include indexing, indirection, generalized macro and instruction arguments, revision flagging, and input correction capabilities. The TPSLIB facility was developed by W. B. Meyer as a means of organizing the myriad bookkeeping operations necessary in the preparation of a lengthy manuscript (his thesis, in this case).

B.   DEVELOPMENT STATUS

TPS is no longer in active development. Although there
are numerous features which could reasonably be added to
this already complex program (about 3400 cards of /360
assembly language), no further changes to the system are
anticipated.

APPENDIX A

TPS COMMAND DESCRIPTIONS

This appendix contains a complete list of the available TPS commands. These commands have been grouped into functional areas, with an brief introductory section for each area. In addition, an alphabetical list of TPS commands is provided in the Appendix B.

```
*********************************************
                LINE CONTROL
*********************************************
```

Line control action is initiated at the time the line buffer is dumped. Thus, only the last line control command given is followed.

-CN   CeNter lines
      Center between the right and left margin all lines
      printed after the execution of this command including
      any lines in an incomplete state. The margins used are
      those which were set at the time the lines group in
      progress was begun. Centering assumes that the line
      initially started at the left margin.

-EM   Even Margins
      Simultaneously right and left justify, by inserting
      extra blanks between words, all text found between the
      right and left margins. The direction of blank
      insertions is switched before each line is generated to
      balance the overall page print density. Text not
      between the left and right margins is unaffected by

justification.

-LJ   Left Justify
       Left justify all line groups (lines are left unchanged
       in   the   line   group   buffers).   Before   special
       justification techniques are applied all lines have
       this format before printing.

-RJ   Right Justify
       Right justify all lines following the execution of this
       command.

```
********************************************
                MARGIN CONTROL
********************************************
```

Three commands are provided for setting margins. The number
which is a part of each command refers to the position in
the print line: 001 is the leftmost print position; 132 is
the rightmost.   Default values for these parameters are
shown in Appendix C.

-ISnnn   Indentation Set
       Set the paragraph indentation to "nnn". If it is equal
       to the left margin, one blank line is inserted before a
       paragraph conforming to standard block paragraphing
       conventions.

-LMnnn   Left Margin set
       Set the left margin to "nnn". This is not in effect
       until the current line group is dumped.

-RMnnn   Right Margin set
       Set the right margin to "nnn". As in -LM this command
       is not in effect until the current line group is
       dumped.

```
**********************************************
```
DUMP AND SKIP
```
**********************************************
```

TPS produces up to 5 line buffers for each printed line. These are collectively known as the line group, and consist of the main line and optionally the overprint, the bold face, the superscript, and the subscript lines. The printing (dumping) of the line group is implicitly forced whenever the accumulated text in the buffer extends beyond the right margin. Explicit control of dumping and skipping is obtained with the following commands.

-DB Dump line group Buffer
     Dump all text found in the line buffers observing all
     line justification conventions, with the exception of
     even margins; normal carriage control is taken. If the
     line buffers are empty, no print action is taken. the
     line output pointer is subsequently set to the left
     margin constant.

-DE Dump buffer allowing Even margins
     Dump all text found in the line buffers in exactly the
     same manner as the -DB except that the Even Margin
     option, if set, is observed. This command might be
     useful if an absolute tab or new paragraph command was
     used for setting up marginal notes while text between
     the margins is required to be right and left justified
     in order to blend with the rest of the text.

-PG New PaGe
     Dump the last line group, start a new page, and
     establish a new header displacement. Top of page
     numbering, if requested, is performed automatically.
     No action is taken if already at top of page.

-PP   New ParagraPh

Dump the line buffers as in a -DB. The output position
is set to the paragraph indentation constant. If the
paragraph indentation is equal to the left margin or if
-CC0 (single space, special option) has been specified,
an extra normal carriage control is performed.

-PSnnn   Conditional Page Skip

Dump the current line buffer and then skip to the next
page if "nnn" times the current carriage control value
(i.e., nnn*2 if -CC2) lines are not available on the
current page.

-SKnnn   SKip lines over page boundaries

Dump the line buffers, including carriage control, and
insert "nnn" lines without regard to page boundaries.

-SPnnn   Skip lines or new Page

Dump line buffers including carriage control and insert
"nnn" lines. If the skip goes over a page boundary
skipping stops at the top of the next page after header
spacing.

```
*********************************************
              PAGE NUMBERING
*********************************************
```

Page numbers are automatically generated and printed at the
bottom of each standard page (see -PB). If bottom numbering
is not desired, other forms may be selected from the
following list.

-ALnnn Absolute Left Margin

Set the absolute left margin to value "nnn". The
absolute margins are used to compute page number
locations.

-ARnnn   Absolute Right margin

>   Define the right absolute margin for page numbering.
>   Its initial value is equal to the default right margin.
>   See also -AL.

-PBnnn   Page number at the Bottom of the page

>   Generate the page number "nnn" centered between the
>   absolute left and right margins three lines below the
>   last printable line on the page as specified by the -BD
>   command. See also -PN.

-PNnnn   top of Page Numbering

>   Generate the current page as "nnn" right justified on
>   the first printable line of a page, before header
>   spacing is taken. Since TPS generates output one line
>   at a time this form of page numbering will not be able
>   to set a proper page number until the next page is
>   started. Page numbers are incremented automatically. If
>   "nnn" is equal to "***" then page numbering will be
>   suppressed.

-PR   Page number in Roman

>   Generate the current page number as a centered lower
>   case Roman number at the bottom of the current page.
>   The value of the page number to be used is set with a
>   PB or PN command. The current page number may be reset
>   with a -PB or -PN command.

-P# Print the current page number

>   The value of the current page number will be printed
>   whenever this command is encountered. Special control
>   over page number placement can be achieved by combining
>   this command with "ON" conditions.

-P+nnn Add "nnn" to the current page number
     The contents of the current page number are incremented
     by "nnn"

-P-nnn Subtract "nnn" from the current page number
     The contents of the current page number are decremented
     by "nnn".

```
*******************************************
              REVISION FLAGGING
*******************************************
```

These   commands   provide   a   mechanism   for   flagging   text |
changes.   The revision character is   placed   on   the   output |
line 4 spaces to the right of the current absolute right |
margin. |

-RE Revision Begin
     This command flags the beginning of revised text.   All
     text following this command will be flagged with the
     current revision flag ("|" is default)   until   the   -RE
     command is encountered.

-RCa Revision Character set to "a"
     Change the current revision flagging character to "a".

-RE Revision End
     Signal the end of revised text.

```
*******************************************
                  INDEXING
*******************************************
```

One thousand cells, numbered 000 thru 999 are available for
index   generation   and   counting.   They are initialized to a
zero value.   A warning message is issued whenever an attempt
is   made   to   store   a   new   value   into a cell containing a

non-zero value. (the message is a bit nonsensical: ZERO OPERAND NOT PERMITTED). Since cell contents may be incremented and decremented, cells provide a primitive counting mechanism for use in text generation. In addition, cells may be indirectly referenced by any command which contains a numeric parameter field by replacing the "nnn" with ".nnn"; the second form of parameter returns the value which is currently stored in cell "nnn" whereas the first parameter form returns the value "nnn".

-A#nnn Print cell value in alphabetic caps
    Alphabetic interpretation of the numeric value cf the cell will be printed. Thus, a call value of 1 results in a "A", and a value of 4 in a "D". Values greater than 26 or less than 1 will produce an error message.

-B#nnn Print cell value in lower case alpha
    Operation is the same as for A#, except that letters are lower case, i.e., 1 is "a" and 4 is "d".

-C+nnn Add 1 to the contents of cell "nnn"
    The value currently stored in cell "nnn" is incremented by 1 and the result stored in cell "nnn".

-G#nnn Get and print the number in cell "nnn"
    The number stored in cell "nnn" is retrieved and printed. This command is useful for generation of indices or Tables of Contents when used in conjunction with the -S# command.

-H#nnn Print cell value in superscript numbers
    The cell value is printed as a superscript number. Useful for footnotes and reference citations.

-I#nnnvvv Initialize cell "nnn" to value "vvv"

> The unsigned value "vvv" is stored in cell "nnn" for later use. The value "vvv" must be terminated with a blank

-NLn Number Length for G# H# R# Q#

> The value of the cell will be printed right-justified in a field of length "n". The default field length is 2. If the value is longer than the current value of "n", the additional places will be printed.

-Q#nnn Print cell value in lower case Roman

> Same as R#, except numerals are lower case.

-R#nnn Get and print the value in cell "nnn" in Roman numerals

> The numeric value stored in cell "nnn" is fetched and then printed in lower case Roman numerals

-S#nnn Store the value of the current page number in cell "nnn"

> The value of the current page number is stored in cell "nnn". If cell "nnn" does not have a zero value, then an error is indicated. Cells may be zeroed with the -Z# command.

-Z#nnn Zero cell "nnn"

> Initialize the indicated cell to a zero value.

****

FORMULAS

****

Primitive facilities for formula representation are provided by the commands of this section. The introduction to the -DE command provides background information necessary for understanding of the formula generation commands. Note that while superscript and subscript lines can be generated, these special lines provide none of the features necessary for making composite characters (i.e., overprint, underline); neither are there facilities for generating super or subscripts on super or subscript lines. When complex positioning of arguments is required, use of absolute tabs (-AT) and single spaced lines (-CC1) is recommended.

-ML   Return to current relative position in the Main Line
      Output all the following text on the main line starting
      at a point immediately following the last character
      generated. This is a possible termination for the -SU,
      -SB, or -OP commands. See also -MX.

-MX   Return to last main line position used
      Place all text following this command on the main line,
      starting after the last used position in the main line.
      See also -ML.

-NS   No Sub or superscripting
      Turn off super-subscript mode. Be sure that the line
      buffers are completely empty before issuing this
      command or incompletely generated subscript and
      superscript lines will not be printed.

-SE   Start suBscript
      Save the location of the line output pointer and insert
      text in the Subscript line. See also -SU.

-SU   Start sUperscript
      Save the line output pointer and insert all text in the
      superscript line. This mode may be suppressed by an -SB
      or -NS.

```
****************************************
                 MACROS
****************************************
```

The TPS macro facility provides a mechanism for shorthand
representaticn of text phrases (or tabular formats, etc.)
which are frequently used. TPS macros may have an arbitrary
number of arguments (zero or more). To use a macro, it must
first be defined; a prototype macro definition is:
                   -MDa...macro body...-MEa
where   "a"   is the macro name and "...macro body..."
represents the sequence of commands and text which are to be
executed whenever the macro is invoked (used). The macro
body may contain any TPS command; except for the -NC, these
commands are not executed at definition time.

To use a previously defined macro, we command "@a" (or
"-MUa").   This causes TPS to read new text from the macro
body rather than the normal input stream.   Input text is
taken from the macro body until one of the following
conditions occurs:

(a)   all text in the macro body has been processed.
Action: Return to the invoking text stream.

(b)   a macro invocation is required by the macro body text.
Action: Invcke the requested macro. Note: the requested
macro must not be the macro you are currently executing.

(c) a request for reversion to the invoker's input stream is encountered (-MR or !).

Action: Revert to the invoking text stream, processing the text until a -MR or ! command is encountered. Return to the macro body and process the remaining text.

Note that the ! may be used in place of the "nnn" command argument within a macro body; when this occurs, the parameter value is obtained from the invoking text stream. Reading of the invoking text stream continues until a matching ! is encountered.

(d) a request for reversion to the master text stream is encountered (-MT).

Action: Revert to the master text stream, processing the text until a -MR or ! command is encountered. Return to the macro body and process the remaining text.

Example:-MDE/ELECTROMAGNETIC /COMPATABILITY-MDE   Definition
@E CONSIDERATIONS ...   Use
The macro named "E" is invoked whenever the phrase "Electromagnetic Compatability" is required in the text.

Example:-MD:!-AT030!-AT040!-AT050!-ME: Definition
@:10!45!63!17!@:92!13! 2!13!   Use
The macro named ":" is defined in order to make a table with four columns of numbers; these columns begin respectively at column 15 (the LM value), 30, 40, and 50. The macro ":" should be purged (-MP:) so that the same name may be reused whenever a new table definition is needed.

Three macros have been predefined in TPS; they are:
-MD1-PG-SK004-=L-CN-MR-¬L-SK002-EM-ME1
-MD2-SKC02-FS010-=L-MR-¬L-SK001-ME2
-MD3-SK001-FS010-UN-MR-NU-SK001-ME3
 They are intended to be used for
(1) @1 ...your chapter title...!
(2) @2 ...your major heading...!

(3) @3 ...your sub heading...!

A set of 40 "report writing" macros is available in TPSLIB.
Use of TPSLIB is discussed elsewhere in this manual.

-MC   Macro directory Clear
      Clear all macro descriptions enabling new ones to be
      specified.

-MDa Macro Definition
      Define the start and name of a new macro.   All
      instructions and text following this instruction, until
      an -MEa command, are stored as a macro named "a".

-MEa Macro definition End
      Terminate the scope of an -MDa command. this command
      must be preceded by an -MDa command or an invalid
      command message will be issued.

-MFa Macro oFf
      Declare macro "a" to be temporarily non-existent. This
      command can be nullified by the -MOa command or by
      re-defining the macro. The macro itself is not changed
      by this command.

-MOa Macro On
      Declare a macro as existing after a -MFa has been
      issued. This function is performed automatically by an
      -MDa.

-MPa Purge Macro "a"
      Remove the macro definition for macro "a" from storage.
      This command is required whenever a redefinition of
      macro "a" is made which lengthens the original macro
      body; shorter redefinitions of macro "a" are allowed
      without use of the "purge" command.

-MR   Macro Revert to the last used input stream
     Permit a macro to release control to the card reader
     input stream and obtain control again. This is used to
     supply parameters to a macro. A shortened form of the
     -MR command is the "!".

-MT Revert to the Text stream for next argument
     Macros which wish to retrieve their arguments from the
     input text stream rather than the next higher macro
     body level must use the -MT command rather than the -MR
     command which only fetches the argument from the
     immediately calling macro body.

-MUa   Use a previously defined Macro
     Insert all the text in macro "a" into the input stream
     at this point. A shortened form of the -MUa command is
     the "@a".

        ****
              SPECIAL MACROS: THE ON CONDITION
        ****

Heading and footing text can be automatically printed at the
top and bottom of each page through the use of special
parameterless macros called "ON" condition macros (after
PL/I).   For an "ON" condition macro to be invoked, it must
be defined and explicitly enabled. The -PG command should
not be used in these macros. The last command of the ON
macro must be a dump (-DB) or a skip (-SKnnn). For example,
the footing of this appendix was generated with the
following command sequence:
-OD2-SK003/APPENDIX /A-AT045-P#-DB-OE2-OO2-PE***
The page heading macro is simply:
-OD1 heading text -CN-DBOE1-OO1
When the ON condition is true, the macro body associated
with the condition is executed. The environment specified

in the PSG (see Appendix C) may be switched and restored in the ON macro (-Sn and -Rn) if special effects are desired; the -FC and -FR commands are automatically issued as part of the execution of an ON condition (these commands checkpoint and restore the text environment when a top or bottom of page condition occurs). An example of undesired global effects during heading or footing occurs when the last line of text on a page invokes -=L (this state is not reflected in the PSG); both foot and head macros will then be interpreted according to the -=L conventions.

Another use of the -OD2 is the generation of bottom of page footnotes. The body of the footnote is given in the -OD2 and the length of the page is adjusted to reflect the space required by the footnote (-BDnnn). Since the first definition of a macro must be the longest (there's no way to purge an -ODn), take care. By defining each footnote element as a separate (regular) macro, you need less space in the body of the -OD2 since you may pick up the footnote elements by refering to them by macro name. The last actions in the -OD2 should be to reset the -BD and -OF2.

-ODn Define "On" condition macro
    For "n"=1, a macro which is invoked whenever the top of
    the page is encountered is defined. For "n"=2, a
    bottom of page macro is defined. These macros can have
    no text stream arguments.

-OEn End "On" condition macro
    End the definition of a heading (n=1) or footing (n=2)
    macro.

-OFn Disable (off) the "n" "ON" condition
    The indicated heading or footing macros are turned off.

-OOn Enable "ON" condition "n"
      The indicated "ON" condition is enabled. Whenever the
      top or bottom of page is encountered, the appropriate
      macro will be invoked.

              ****
                              DEBUGGING
              ****

TPS is a programming language and your text program will
probably contain some errors. Most errors can be found with
only the -=C option; the more mysterious the error, the more
information you will want.

-=C Command summary print
      Print a summary of each command encountered. This is a
      useful debugging aid which still enables utilization of
      the resultant output.

-¬C No Command summary (default)
      Do not print the command summary.

-=I Intersperse source cards
      This is the most detailed level of debugging printout.
      The resulting text output is unusable.

-¬I No interspersed listing (default)
      Do not print source card images.

-=S Print sequence numbers
      The text in columns 73-80 is printed as sequence
      numbers. You should give the -SL072 command or use the
      SM=(1,72) parameter on the EXEC card.

```
***********************************************
                BASIC FUNCTIONS
***********************************************
```

-ATnnn   Absolute Tab
     Set the line location pointer to "nnn". No position
     testing is performed allowing text to be placed
     anywhere in the 132 positions available per line.

-BF   Bold Face font
     Over strike all non-blank characters in the main line
     following this command. This   implementation   over
     strikes three times.

-BSnnn   Back Space
     Backspace the line output pointer "nnn" positions.   If
     the left margin is exceeded the line output pointer is
     set to the current value of the left margin.

-FMnna Fill with "a", leaving a right margin of "nn"
     When this command is encountered, the character "a" is
     printed until the line is filled, with a right margin
     of "nn" remaining. It is useful for Index and Table of
     Contents work.

-FWnna Fill With "nn" "a"'s
     Exactly "nn" "a"'s will be be printed.

-LC   Lower Case
     Convert to lower case all text following this command
     before printing. This is used to terminate an -UC
     command.

-NC Read the Next Card immediately

Whenever this command is encountered, the balance of the card is considered to be commentary, and the next card is read.

-NF Normal Face font

Resume normal single strike printing of the main line. This nullifies the effect of -BF.

-NU No Underlining

Terminate underlining.

-OP OverPrint

Save the location of the line output pointer and insert text in the overprint line. See also -ML and -MX.

-SSnnn Skip Spaces

Add "nnn" to the current value of the line output pointer. If the resulting location is greater than the right margin then the line output pointer is set equal to the right margin.

-TB TaB (relative)

Set the line output pointer to the first value saved in the relative tab table that is greater than the current value of the line output pointer.

-UC UpperCase

Convert all characters following this command to upper case before processing. -LC will reset lower case mode.

-UN Start UNderlining

Underline all non-blank characters placed in the main line following this command.

-"" As is text
Dump the following text with blanks unsuppressed,
returning to normal mode when the next control command
is encountered (any control command). Note that even
margin (-EM) requirements override this command. See
the -TR command for a discussion of techniques for
representing "significant blanks."

--- Output the character "-"
Print the current attention character.

-|* Start comment.
Text following this command is not printed and the line
output pointer remains unchanged. All commands except
the -*| will not be executed.

-*| End comment.
Resume normal un-commented text processing.

-=L Leave input unchanged
No translation of input is preformed; this is useful
when card images or quantities of text in upper case
with special symbols are required.

-¬L Normal translation of input (default)
Normal translation of the input stream is carried out.

/a Change print case
If TPS is in upper case mode the character "a" is made
lower case. If in lower case mode the character is made
upper case.

// Output the character /

```
********************************************
              GLOBAL CONTROL
********************************************
```

The collection of parameters which specifies the global
control values is known as a program status group (PSG).
The control values specified in the PSG may be saved and
subsequently restored to one of several PSG's; thus, context
switching can be done efficiently and concisely by means of
PSG saves and restores. The control values in the PSG are
listed in Appendix C.

-BDnnn Body length Definition
     Set the total number of lines on a page less 3 (the
     usual paper setting in the printer is three lines from
     the top), starting from the top of the page. If the
     length of the page set is greater or equal to the
     current line number then a new page is started after
     the line group currently being generated is completed.
     If nnn is *** pagination is suppressed.

-CAx   Change the Attention character
     Change the attention character (default "-") to the
     single character specified. A case change qualification
     is not accepted in this special case and the use of the
     case changed alternate for this character is lost.

-CCn   Automatic Carriage Control
     Insert n-1 blank lines between each line group printed.
     If "n" is equal to 0 then an additional blank line is
     generated between paragraphs. Issuing the command with
     a non-zero "n" resets the inter-paragraph spacing. The
     default settings are -CC2 -CC0.

-CF Correction Off
     Disable the operation of the correction character (the
     "¢" symbol).   This command does not take effect   until
     the following card is read.

-CO Correction On
     Turn   on   the   correction feature.   The punching of "n"
     consecutive " symbols will cause   the   "n"   preceding
     characters to be deleted.

-CQx Case Qualification character change
     Modify   the   case   change qualifier (default "/").   The
     use of the case changed alternate for this character is
     lost.

-HDnnn Set the HeaDer spacing
     Set   the   total   number   of   lines   (less   three) to be
     skipped at the top of each new page. Since the IBM 1403
     printer is usually set to start printing three lines
     from the top of a page these three lines must be   taken
     into consideration when setting the header.

-CN Output is Not printed on tne printer
     Do   not   print any output directed to the printer after
     the execution of this command. A line group in progress
     will   therefore   not   be   printed   unless   it is dumped
     before the command is issued.

-OY Output is permitted to print
     Restart printer output by cancelling the   -ON   command.
     The   contents of the line group in progress will not be
     printed unless the line group is dumped before printing
     is enabled.

-Rn   Reset PSG using reset area n
       Restore all the format description elements described
       in Appendix C to what they were at the time of the last
       -Sn. If the nth reset area has not been set this
       instruction is ignored. "n" must be between 0 and 4.

-RS   ReSet to default PSG (Program Status Group)
       Reset all pertinent overall format variables to the
       original default values. All of them will not be in
       effect until the start of the next line. See also -Rn.

-SFnnn Set the First input column
       The value "nnn" sets the first column of the input card
       which will be read. May also be set in the
       PARM='SM=(NNN,MMM)' field of the EXEC card.

-SInnn Set Last cclumn of input card
       The last column of the input card which will be read is
       set to "nnn".

-Sn   Save the current PSG in reset area "n"
       Save overall PSG format controls in save area "n". "n"
       must be between 0 and 4. see also -Rn.

-** reinitialize TPS
       Reinitialize TPS and begin processing of the text which
       follows.

-STiiijjj... SetTab list
       Set up to fifteen relative tab positions to the values
       specified. Values should be in ascending order and the
       last value in the list must be followed by at least one
       blank. If the new tab list is shorter than fifteen
       values, only the first replaced tab positions are
       changed.

-TRab Translate "a" to "b"

> Every occurrence of "a" as an output character will be changed to "b" just before printing. The characters "a" and "b" may be case qualified. In order to control the placement of blanks in the output text, we may choose a character not used in the text and utilize it as a filler; subsequent to all justification, this "significant blank" character is translated into a normal blank for printing. For example, -TR¬ will cause every "¬" in the current and subsequent line groups to be printed as a blank.

-ULa Change the UnderLine character

> Replace the underline character with "a". This is useful for strike-over and other special effect text. The underline character "a" may be upper case (preceded by a "/").

-=B Brief Mode on

> When brief mode is "on", the characters "∂" and "!" may be used in place of "-MU" and "-MR" respectively.

-¬B Brief Mode off

> Whenever the characters "∂" and "!" are required in the text, this command must be issued.

APPENDIX B

TPS COMMAND INDEX

APPENDIX C

DEFAULT PROGRAM STATUS GROUP

Underline character	-UL_
body length (72 lines)	-BD076
Tab list (15 elements)	-ST020025030...090
Header length	-HD004
Indent	-IS020
Right margin	-RM075
Left margin	-LM015
Carriage control	-CC2-CC0
Justification	-EM

In additicn to the above information, the following state information is saved and restored by -FC and -FR whenever -ODn macros are invoked:

mode flags for case (upper or lower), line adjustment, bold face, block paragraphing, blank squeezing, underlining, and super/sub-scripting.

Value of quantities not in PSG

Absolute right margin	-AR075
Absolute left margin	-AL015

APPENDIX D

EXEC PARAMETERS

TPS EXEC card parameters

Parameters for TPS execution may be specified on
the EXEC card. The EXEC card is of the form:
// EXEC PGM=TPS,PARM='p1,p2,...,pk'
where p1,...,pk are given by one or more of the
parameters shown below:

TN   upper/lower case printer is desired; must also
            specify //SYSPRINT DD SYSOUT=M OR N.

SM=(M,N)   source margins: begin at "m"; end at "n"

COPY=N   number of output copies required; n < 4.   A
            //SYSUT2 DD SYSOUT=M card is needed if n > 1.

REPRO   copy file specified by SYSIN to SYSPRINT. If copy
            option was specified in a previous parm, a SYSUT2
            file is also required.

C   see -=C command

SEQ   see -=S command

D   upper case output (default PN train)

I   see -=I command

L   see -=L COMMAND

MSIZE=N   macro storage area size (default is 8192 bytes).

APPENDIX E

TPSLIB DESCRIPTION

This appendix describes all macros in TPSLIB, indicating the arguments and when page storing and upper case are automatically performed.

A - 1st order table of contents
   * arguments: 1
      ▪ title
   * automatic indexing
   * automatic paging
   * upper case

B - books in list of references
   * commas, underlining, etc automatic
   * arguments: 5
      ▪ Author
      ▪ Book title
      ▪ Pages
      ▪ Editor
      ▪ Year

C - table of contents thesis
   * jumps page and writes TABLE OF CONTENTS
   * no arguments

D - contents of distribution list
   * arguments: 3
      ▪ Who to get copy of thesis
      ▪ Number of copies
      ▪ Remainder of lines

E - thesis face/one author
* arguments: 5
* See example above

F - 4th order table of contents
* arguments: 1
* Title
* automatic indexing
* automatic paging

G - auxiliary macro utilized by TABLE OF CONTENTS; fills the
rest of the current line with "." and prints the "next"
stored page number. Can be used when special effects are
required in the Table of Contents.

H - thesis face/two authors
* see example above

I - 5th order table of contents
* equal to macro F above

J - special abstract/two authors
* see example above

K - table of contents papers
* equal to macro C

L - list cf references
* jumps page and writes LIST OF REFERENCES
* no arguments
* autcmatic paging

M - appendix table of contents
* this macro will place in table of contents any title
of text produced by macro [6] (appendixes)
* arguments: 1
* Title
* automatic paging

N - list cf figures

* jumps page and writes LIST OF FIGURES
* no arguments
* automatic paging

O - list of figures contents
* arguments: 1
  * Title of figure
* automatic indexing
* automatic paging

P - list of references - papers in magazines
* arguments: 6
  * Author
  * Title of paper
  * Magazine where published
  * Volume number
  * Pages referenced
  * Date published

Q - skips a page for figures
* arguments: 1
  * Title of figure
* automatic indexing of figure
* automatic capitalization
* automatic page sotring (for list of figures only)
* this macro will, in fact, skip two extra pages, one
with the figure title printed and other without it, so you
can place a figure in the page laterally; in both cases the
correct page number will appear;

R - abstract
* no arguments
* this macro will skip a page and write ABSTRACT at the
top of next one;

S - 2nd order table of contents
* equal to macro A above

T - 3rd order table of contents

* equal to macro F above

U - special abstract one author
    * see example above

V - refer to figure that last appeared before this call
    * no arguments
    * this macro, when called, will print in text [Fig. nn], where nn is the number of the last figure that appeared before the call to this macro; this will allow you to insert and pull out figures to the last minute without having to change the text all over

W - header for papers
    * see example above

X - skip an extra line and print a star (like the one you see on the beginning of this paragraph)
    * no arguments

Y - skip just one line and print a star
    * like macro X above, except that no extra line is skipped

Z - skip one line and print a square adequately indented
    * see example above

1 - first order header
    * arguments: 1
        • Title
    * skip page
    * automatic capitalization
    * automatic indexing
    * page storing

2 - second order header
    * arguments: 1
        • Title
    * automatic capitalization
    * automatic indexing

* page storing

3 - third order header

    * arguments: 1

        ■ Title

    * automatic indexing

    * page storing

4 - fourth order header

    * arguments: 1

        ■ Title

    * automatic indexing

    * page storing

5 - fifth order header

    * arguments: 1

    * automatic indexing

    * page storing

6 - appendix header

    * arguments: 1

        ■ Title of appendix

    * automatic capitalization

    * automatic indexing (for appendix number)

    * page storing

    * for second and third order headers inside appendixes, macros [9] and [0] (zero) should be used, that perform the same task as macros [2] and [3] but do not have the page storing facility (remember that items inside appendixes do nct appear in table of contents)

7 - any top of page (first order header) title without indexing

    * arguments: 1

        ■ Title

    * automatic capitalization

    * no indexing

    * page storing

8 - to be used for listings items

    * arguments: 1

         ■ Name of item or whatever

    * automatic indexing

    * no page storing

    * this macro can be used also for list of references, for anything different of books (macro B) or papers (macro P)

9 - seccnd order header without page storing

    * equal to macro 2 without page storing

    * to be used inside of appendixes, annexes, etc

0 - macro zero: third order header without page storing

    * equal to macro 3 above except that no page storing is provided

    * to be used inside appendixes, etc, since it should nct appear in table of contents

+ - refer to next figure in text

    * equal to macro [V] except it refers to next figure that appears in text

* - refer to next to the next figure in text

    * equal to macro [V] except it refers to next to the next figure that appears in text after the call to this macro

$ - to be called at end of abstract

    * arguments: none

    * this macro will initialize TPSLIB properly

% - to be called in place of or after abstract in case of a paper

    * equal to macro [$] above

APPENDIX F

SUMMARY OF TPSLIB MACROS

ALPHABETICAL LISTING

- A 1st order contents
- B list of references
- C table of contents thesis
- D contents of distribution list
- E thesis header one author
- F 4th order contents
- G auxiliar macro for table of contents
- H thesis header two authors
- I 5th order contents
- J special abstract two authors
- K table of contents papers
- L header for list of references
- M appendix header contents
- N list of figures
- O contents of list of figures
- P list of references
- Q figure in whole page
- R abstract
- S 2nd order contents
- T 3rd order contents
- U special abstract one author
- V refer to last figure
- W header for papers
- X skip extra line and print star
- Y print star no extra line skipping
- Z indent and print square
- 1 first order header
- 2 2nd order header
- 3 3rd order head
- 4 4th order head
- 5 5th order head
- 6 appendix header
- 7 any top of page title without indexing
- 8 item list (auto number and indent)
- 9 2nd order header without page storing
- 0 3rd order header without page storing
- + refer to next figure
- * refer to next to the next figure
- $ to be called at end of abstract
- % dummy abstract; initialize TPSLIB for a paper

FUNCTIONAL GROUPING

* Cover Pages
  * W paper
  * E thesis (one author)
  * H thesis (two authors)

* Abstracts
  * R abstract
  * U one author (special)
  * J two authors (special)
  * $ end of (thesis)
  * % end of (paper)

* Headings
  * 1 1st order
  * 2 2nd order
  * 3 3rd order
  * 4 4th order
  * 5 5th order
  * 6 1st order (appendix)
  * 9 2nd order (appendix)
  * 0 3rd order (appendix)

* Contents
  * A 1st order
  * S 2nd order
  * T 3rd order
  * F 4th order
  * I 5th order
  * M appendix
  * G appendix special
  * C thesis header
  * K paper header

* Figures
  * Q page for
  * N list of
  * O contents of list
  * V ref. to last
  * + ref. to next
  * * ref. to next to next

* Formatting
  * X * line
  * Y * line
  * Z ∎ line
  * 8 numbered line
  * 7 top of page title

* References
  * L header
  * B book
  * P paper

* Distribution
  * D header and list

APPENDIX G

EXAMPLES

The usage of header macros is presented on the next pages.
The arrow => indicates the arguments that those macros take.

Observe that the page numbering follows the instructions of
the Thesis Manual.

=> TITLE

=> This Page Was Made With Macro [W]
=> Month and Year

=> Author of Paper

(52)

=> TITLE

                        by

                    => Author
                => Rank of Author
            => Degrees Attained by Author

        Submitted in partial fulfillment of the
            requirements for the degree of

            => PAGE MADE WITH MACRO [E]

                    from the
            NAVAL POSTGRADUATE SCHOOL
                => Month and Year

Author:

Approved by:
            --Thesis Advisor

                                    o
            ----------------------------------Second Reader

            ----------------------------=> Department Chairman

            -------------------------------------Academic Dean

                        3   (53)

=> TITLE

                      by

                    => Author
                 => Rank of Author
            => Degrees Attained by Author

                    => Author
                 => Rank of Author
            => Degrees Attained by Author

        Submitted in partial fulfillment of the
             requirements for the degree of

             => PAGE MADE WITH MACRO [H]

                    from the

             NAVAL POSTGRADUATE SCHOOL
                 => Month and Year

Authors:
            _____

            _____

Approved by:
            _____Thesis Advisor

            _____Second Reader

            _____=> Department Chairman

            _____Academic Dean

                         3   (54)

ABSTRACT

This is the abstract. It should not have more than 18 lines. MACRO [R] should be called at beginning of abstract; this macro will automatically skip a page, give the correct page number (4) and write 'ABSTRACT' at top of page. All that the user has to do is to place the abstract text after macro [R] is called. After the abstract text is completed, MACRO [$] should be called to properly initialize TPS for thesis body. Macros [R] and [$] do not take any arguments. In the case of a paper, nor macro [R] or [$] should be called, but macro [%] should be called to properly initialize TPS for paper body.

As an example, this sequence of macros, in the case of a thesis, would be called as:

@R
abstract text
@$

```
                      => TITLE




                      => Author
                  => Rank of Author
            => Degrees Attained by Author
```

=> Macro   [U] was called to write this special
abstract; page skipping and page numbering suppression are
done automatically. This macro takes 11 arguments:

    * the 4 arguments above;
    * this text that is an argument and should be an exact
copy of the thesis abstract;
    * the 6 arguments on the bottom of page.

    To assure proper indentation, the arguments below
should be called in the following format and spacing:
!Master of Science in
!Advisor Name
!Subject of Degree
!Department Name
!Month and year
!Group or Department!

```
=>Master of Science in        Advisor: =>Name
=>Whatever                             =>Computer Science
=>Month 1975                           =>Group.
```

=> TITLE

=> Author
=> Rank of Author
=> Degrees Attained by Author

=> Author
=> Rank of Author
=> Degrees Attained by Author

    => This special abstract is intended for thesis with two authors. Macro [J] was called to write this page. Refer to last page (special abstract with one author) for details.

=>Master of Science in   Advisor: =>Name
=>Whatever   =>Computer Science
=>Month 1975   =>Group

APPENDIX H

SAMPLE TPSLIB PROGRAM

This appendix presents an image of a deck prepared to run a thesis program for one author.

The output of this deck appears in the next appendix.

Each line represents a card. Observe the sequencing of macros and the spacing inside them.

```
@E
TITLE OF THESIS!
/AUTHOR OF /THESIS!
/LIEUTENANT---/COMMANDER!
/DEGREES /ATTAINED BY /AUTHOR!
MASTER OF SCIENCE IN COMPUTER SCIENCE!
/DECEMBER 1975!
/CHAIRMAN, /COMPUTER /SCIENCE /GROUP!
@R
-PP/THIS IS THE ABSTRACT'S TEXT.
@$
@1INTRODUCTION!
-PP/THIS BEGINS THE THESIS BODY. /SOME   EXAMPLES   OF   MACROS
WILL BE SHOWN.
@2FIGURES!
-PP/WILL   SHOW HOW A FIGURE CAN BE PLACED IN BODY AND HOW TO
REFERENCE IT.
-PP/NEXT FIGURE WILL BE @+.
@QFIRST FIGURE!
-PP/WE SAW MACRO /Q PREPARING BODY FOR A FIGURE, @V.
@3/THIRD /ORDER!
-PP/THIS IS AN EXAMPLE OF A THIRD ORDER HEADER.
@1ANOTHER FIRST ORDER!
@2ANOTHER SECOND ORDER!
-PP/AND FROM NOW ON ALL THE USER HAS TO DO  IS  TO  TYPE  IN
CARDS, FREE
FORMAT, HIS
THESIS OR PAPER TEXT,
AND THE OUTPUT WILL BE FORMATTED  IN  ACCORDANCE  WITH  THE
/THESIS /MANUAL.
@6FIRST APPENDIX!
-PP/THIS WILL BE THE FIRST APPENDIX.
@L
@B
/ENSLOW!
/MULTIPROCESSING!
1 TO 20!
/COMTRE /CO.!
1975!
@8/ANYTHING   YOU   LIKE   CAN  BE  EASILY  PLACED  IN  LIST OF
REFERENCES WITH THIS MACRO 8.!
@7INITIAL DISTRIBUTION LIST!
-RJ/NO. /COPIES-DB-EM
@D/DEPARTMENT /CHAIRMAN, /CODE 72!
1!/COMPUTER /SCIENCE /GROUP
-DB/NAVAL /POSTGRADUATE /SCHOOL
-DB/MONTEREY, /CALIFORNIA 93940!
-PG-PB006@N
@C/FIRST /FIGURE!
@C
@AINTRODUCTION!
@SFIGURES!
@T/THIRD /ORDER!
@AANOTHER FIRST ORDER!
@SANOTHER SECOND ORDER!
@MFIRST APPENDIX!
-DE-UCLIST OF REFERENCES-LC@G
-DE-UCINITIAL DISTRIBUTION LIST-LC@G
-DE-UCLIST OF FIGURES-LC@G
-PG
```

APPENDIX I

SAMPLE TPSLIB OUTPUT

The   pages   which follow were produced by running the
text deck shown in Appendix H   with   the   control   cards   of
section I.E.3.

TITLE OF THESIS

by

Author of Thesis
Lieutenant-Commander
Degrees Attained by Author

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the
NAVAL POSTGRADUATE SCHOOL
July 1975

Author: _____

Approved by:
          _____Thesis Advisor

          _____Second Reader

          _____Chairman, Computer Science Group

          _____Academic Dean

BLANK PAGE

ABSTRACT

This is the abstract's text.

4

I. INTRODUCTION

This begins the thesis body. Some   examples   of   macros
will be shown.

A. FIGURES

Will show how a figure can be placed in body and how to
reference it.

Next figure will be Fig 1.

Figure 1 - FIRST FIGURE

Figure 1. FIRST FIGURE

We saw macro Q preparing body for a figure, Fig 1.

1.   <u>Third</u> <u>Order</u>

     This is an example of a third order header.

II.   ANOTHER FIRST ORDER

A.   ANOTHER SECOND ORDER

And from now on all the user has to do is to type in
cards, free format, his thesis or paper text, and the output
will be formatted in accordance with the Thesis Manual.

APPENDIX A

FIRST APPENDIX

This will be the first appendix.

LIST OF FIGURES

(70)

LIST OF REFERENCES

1. Enslow, <u>Multiprocessing</u>, p.   1 to 20, Comtre Co, 1975.

2. Anything you like can be easily placed in list of references with this macro 8.

(71)

INITIAL DISTRIBUTION LIST

No. Copies

1. Department Chairman, Code 72   1
   Computer Science Group
   Naval Postgraduate School
   Monterey, California 93940

TABLE OF CONTENTS

.

This page is generated after all referenced ·
pages have been printed. NPTS has no
mechanism for forecasting page numbers. (73)

Copies

Defense Documentation Center   2
Cameron Station
Arlington, Virginia 22314

Dean of Research   2
Code 023
Naval Postgraduate School
Monterey, California 93940

Library (Code 0212)   2
Naval Postgraduate School
Monterey, California 93940

Computer Science Group (Code 72)   1
Naval Postgraduate School
Monterey, California 93940

Computer Sciences Department   1
Naval Electronics Laboratory Center
271 Catalina Boulevard
San Diego, California 92152

Prof. G. L. Barksdale, Jr. (Code 72Ba)   10.
Computer Science Group
Naval Postgraduate School
Monterey, California 93940

W. R. Church Computer Center (Code 0211)   100
Naval Postgraduate School
Monterey, California 93940

Prof. G. Bradley   1
Prof. G. Brown
Prof. N. Schneidewind   1
                                                           1