HEWLETT
PACKARD

TDP
Reference Manual

Manual Part No.
36578-90009 E0888

Printing History

First Edition - August 1988

Copyright Notice

Preface

About this manual

This manual is written for the person familiar with TDP, EDIT/3000, or any text processing system. It is intended to be used as a reference guide; not as a self study introduction to the system. The manual describes the editing and formatting commands available to the user of TDP. If you are using TDP for the first time you may find the following manuals useful:

Using the HP 3000	(Part No. 03000-90121)
MPE V Commands Reference Manual	(Part No. 32033-90006)
OR **MPE XL Commands Reference Manual**	(Part No. 32650-90003)

How to use this manual

This manual is a reference guide to be turned to as and when you require information about a particular feature or command of TDP. The manual is organized as follows:

Chapter 1 This chapter describes some of the important general features of TDP, how to use TDP, how TDP works and provides a guide to TDP commands. If you have experience of using a text processing system, but are unfamiliar with TDP, this is where you should start.

Chapter 2 This chapter describes the terms and conventions used to describe TDP and explains the format of the command descriptions used throughout this manual.

Chapter 3 This chapter describes all of the editing commands used to enter, edit and output text with TDP. The majority of commands use examples to illustrate the use and/or effect of the commands.

Chapter 4 This chapter describes all of the formatting commands used to control the layout and presentation of documents created and edited with TDP. As in Chapter 3, the majority of commands use examples to illustrate the use and/or effect of the commands.

Chapter 5 This chapter describes how to set up and use the TDP spooler.

Chapter 6 This chapter describes the facilities and use of the TDP mass mailing program, MAILER.

Chapter 7 — This chapter describes how the user can add words to the TDP hyphenation exception dictionary, and how user written hyphenation procedures should interface to the system.

Appendix A — This appendix describes the error messages you might encounter while using TDP. It is divided into editor and formatter error messages.

Appendix B — This appendix provides a complete summary of the syntax of all TDP commands described in the rest of the manual.

Appendix C — This appendix describes some technical aspects of TDP, such as workfile structure and how to call TDP programatically.

Appendix D — This appendix provides a comparison of TDP and EDIT/3000 facilities for the user of EDIT/3000.

Appendix E — This appendix describes how to run compilers and programs from TDP.

Appendix F — This appendix describes how formatted output is obtained from TDP and the printers supported by TDP.

Appendix G — This appendix provides a brief introduction to using laser printers with TDP.

Appendix H — This appendix provides some brief notes on those features of TDP that may cause formatting problems for the new user of TDP.

Appendix I — This appendix lists the ASCII character codes used by TDP.

Conventions used in this manual

Bold — Bold text indicates a cross reference either to another section in this manual or to another manual. Bold text is also used to highlight especially important information, typically a warning note.

$\boxed{\text{CTRL}}$ — Boxed words or symbols represent keys on the keyboard.

`CLEAR HOLD?` — Text in this typeface and color represents text that appears on the screen, such as messages, system prompts or text that you type.

Getting help

TDP has a HELP facility which you can call on for information about many aspects of the product. Chapter 1 describes further the use of HELP.

If TDP cannot process a command you have given, you will see an error message to explain the problem. Usually you will be able to type the command or text again and complete your task. For further information on error messages see Appendix A.

If you have a problem with TDP you cannot solve yourself, ask your Office Products Coordinator (OPC) for help. The OPC is responsible for looking after TDP at your office, and for providing any assistance or training you may require. If you do not have an OPC at your office, see the MPE System Manager.

Table of contents

1 Introduction

2 TDP Terms and conventions

3 Editing commands

4 Formatting commands

5 The Spooler

6 Mailer

7 Hyphenation

A Error messages

B Command summary

C Technical information

D Comparison with Edit/3000

E MPE commands and subsystems

F Formatted output from TDP

G Using laser printers

H Formatting hints

I Table of ASCII character codes

Index

Introduction

This chapter introduces some of the general facilities of TDP and provides a guide to the TDP commands described in Chapters 3 and 4.

Starting TDP

To start a TDP session, log on to the system in the normal way and type:

```
:RUN TDP.PUB.SYS
```

(Note this is the normal command to start TDP. It may have been replaced by an installation specific UDC. Your System Manager will provide details.)

TDP displays an opening message and then displays the standard editor prompt (usually " / ") to show that it is ready for you to start typing.

To exit from TDP, type the command `E` (or `EXIT`) after the prompt. If there is material in your workfile, and changes have been made since the last KEEP or TEXT command was issued, TDP will print an message warning you that the workfile has not been saved, and asking if you wish the workfile to be deleted (without saving) and exit. If you answer NO, the workfile remains unchanged and the editor prompt returns for further input or commands. If you answer YES, the EXIT command is executed, and you are returned to the MPE operating system.

Running in batch mode

Although interactive sessions will be the most common way of using TDP, it is possible for TDP to be run in batch mode. For a detailed description of how to run a job in batch mode see the :JOB command, described in the **MPE V Commands Reference Manual** or the **MPE XL Commands Reference Manual**.

Note that TDP provides two special commands for batch jobs (see the BATCHERROROK command described in Chapter 3 and the BATCHERROR command described in Chapter 4).

HELP

Before anything else you should be aware of the TDP HELP facility. HELP provides on your terminal screen a comprehensive guide to TDP commands and facilities. You can access HELP at anytime, and this will often prove more convenient than having to refer to the manual. For an introduction to the HELP system type:

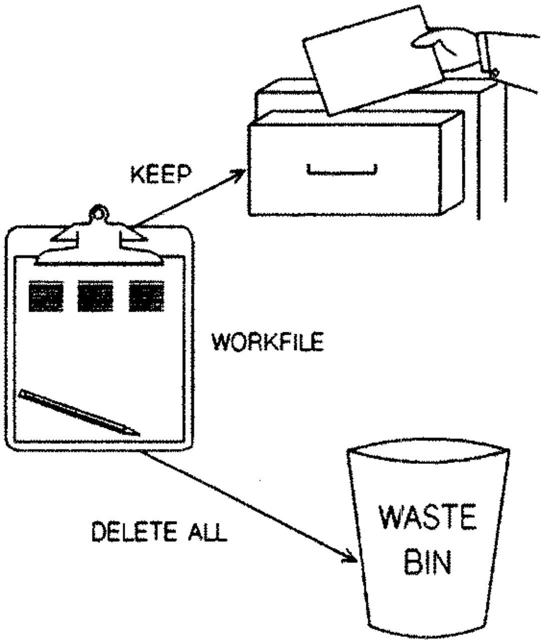
```
HELP
```

To exit from the HELP system, after the HELP system prompt >, type:

```
EXIT
```

Creating new text

When you want to enter material, you type in lines of text at the terminal. These lines are placed in a workfile. You can think of the workfile as your "scratch pad"; all adding, changing, deleting, and so forth is done there. When you have entered all the material, you can copy the workfile to a permanent file on the HP 3000, by means of the KEEP command, or you can delete the material in the workfile, by means of the DELETE ALL command. The material remains in the workfile until you delete it, copy in a different file, or exit from TDP. When you exit normally from TDP, (after a KEEP or DELETE ALL command) the workfile is purged.



Working on a document

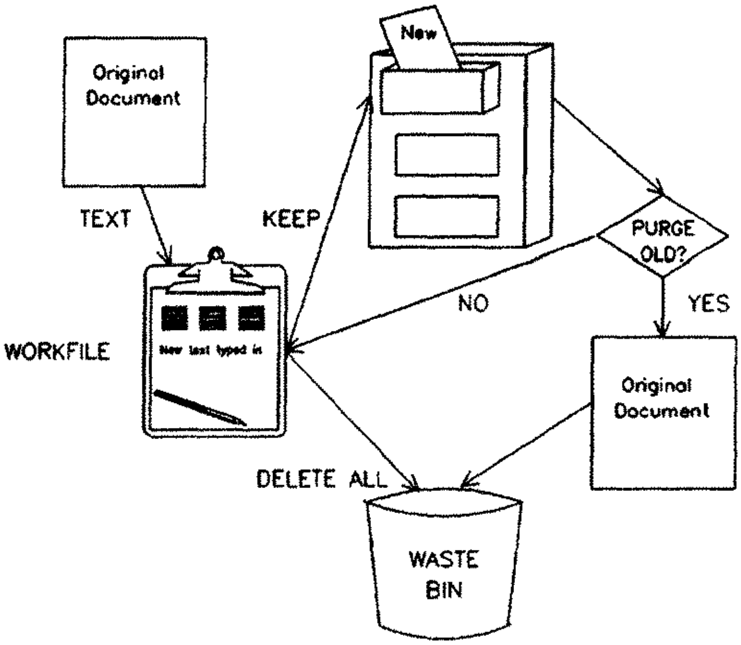
Now let's suppose you want to make changes to the material you have entered and kept as a permanent file. You copy the permanent file into the workfile with the TEXT command, and make the changes to that copy; the permanent file is unchanged at this point. There is no danger of accidentally destroying your only copy since you still have your original text. When the changes are complete, you can KEEP your updated text to a new file with a different file name, thereby retaining the original file under the original name, or you can replace the original contents of your permanent file with the updated version of the text.

When you KEEP the file, TDP automatically assumes that you want to overwrite the old copy of the file. However, to make sure that is what you want TDP asks whether you want to purge the old file of that name:

1 If you answer YES, the old material is purged and the information in the workfile is put in its place.

2 If you answer NO, then it will tell you that the text has not been kept. The workfile remains unchanged and so does your original file. To save the workfile to another file use the KEEP command with the filename you want to give it.

In the case of a system crash, or some other abnormal termination of your session, the workfile remains on the disc as a permanent file.



Identifying the workfile

When a workfile is saved as a permanent file on disc, as a result of an abnormal termination, the file is given an identifier in a standard form so that you can later identify it. The workfile name is an eight character identifier, beginning with K and followed by seven numbers indicating when it was created, as follows:

Kdddhhmm

The first three characters show the Julian day number, and the next four characters show the time in hours and minutes when you started work on the file.

Recovering a workfile is described below, in the section Workfile recovery.

Workfile recovery - Abnormal end of session

To find out the name of the file use the LISTF command. To recover your workfile TEXT in the K-file as you would any other file; you can list the file to see if all your work is there and carry on working on the file if you wish to. You can either keep the file or delete it as you would have done normally. You must specify a name if the file it is to be kept.

If you were adding to the file at the time the session terminated, then you may have lost some of the text. This is because by default TDP only saves the file, when adding, every 99 lines. (To change this default use the SET POSTADD command, telling TDP how often you want the workfile to be saved; for example, every 20 lines.)

If you were in screen mode at the time the system terminated, then you may have lost the range of lines you specified in the SCREEN command.

Elements of TDP

When you run TDP you are working with a number of separate but integrated programs. The most important of these are the editor and the formatter.

Most of your interaction at the terminal will be with the editor. The TDP text formatter obeys the formatting commands you embed in your textfile. The editor and the formatter are controlled by different types of command, which are described later.

The other parts of TDP are:

MAILER	which provides a general purpose mass mailing facility, and is described in Chapter 6.
SPOOLER	which provides a means of printing your formatted output, and is described in Chapter 5.
TDPEX	which maintains an exception dictionary of words not properly handled by the hyphenation routines, and is described in Chapter 7.

The two types of command

TDP recognizes two different types of command:

1 Editor commands (such as TEXT, MODIFY, KEEP, FINAL, etc.) are entered after the (/) prompt, and are immediately executed by the system. See Chapter 3 for a detailed description of each editing command.

2 Formatter commands (such as \CENTER, \LFT, \JUSTIFY, etc.) are entered as a separate formatting command line in the workfile, and are preceded by a backslash (\) or another user defined symbol. The formatting command lines are stored as a permanent part of that file; other formatting commands (such as ^B, ^R, etc.) are entered within the line itself, to indicate intraline formatting. They are executed when the file is output with a FINAL or DRAFT command. See Chapter 4 for a detailed description of each formatting command.

File formats

When you create and keep files in TDP, you are allowed to specify a format for the file (by means of the SET FORMAT command). The file format may be DEFAULT, COBOL or DIARY.

DEFAULT files are kept with an eight character sequence number placed at the end of each line. Most files will have the default format. The increment, used when lines are added or renumbered is set to 1 (changed by means of SET DELTA) and the workfile line length and right margin are set to 72 (changed by means of SET LENGTH and SET RIGHT).

Note that when adding lines of text between two existing lines, TDP will attempt to assign line numbers based upon the value of DELTA. However, if consecutive line numbers are already in use, TDP will assign decimal point line numbers (up to three decimal points). For example, with two existing lines numbered 80 and 81, ADD 80 will cause TDP to assign the new line as 80.1 (if 80.1 was already assigned, the next available number would be 80.01, etc.).

COBOL files are kept with a six character sequence number at the front of each record. Three digits are allowed before the decimal point, and three after. The default value of DELTA is 0.1 and 74 is the default value for RIGHT and LENGTH.

DIARY files have sequence numbers that are composed of a date, time and line number. The sequence numbers are kept as they are for DEFAULT files, but TDP treats the ADD command for DIARY files in a special way. Sequence numbers assigned will represent the date and time. Subsequent PRINT, TEXT, or QT commands recognize these sequence numbers. The MAKE command cannot be used to create DIARY files. The SCREEN command should not be used to edit DIARY files.

Up to 999 lines can be entered for any day and time. Times recorded as sequence numbers are rounded to the nearest quarter-hour. The line number appears after the date and time.

A line may be retrieved by date (Example: LIST 8/13) or date and time (Example: LIST 8/13_12:00a). In addition, a line can be added or referenced by the keywords NOW (current day and time) and TODAY (today's date, time of "zero" for the day). For example:

```
/SET FORMAT = DIARY
/ADD NOW

7/16_2:30p1
```

This is the sequence number of a DIARY file. The 7/16 is the date, 2:30p is the time rounded to the nearest 15 minutes (either a.m. or p.m.), and 1 identifies line number 1. Each keyword references the first occurrence of the date only.

Search facilities with strings

Strings can be used by TDP to locate a position within the workfile. A string is a group of characters enclosed in special characters. A string can consist of a character, a word, or several words. The special characters enclosing the string are called "delimiters". Accepted delimiters include the 19 characters ! " $ & ’ = ^ ~ ‘ @ | \ { : } _ < > and ?.

The system begins searching for a string from the current line pointer, and continues searching until the string is found or the end of file is reached. To have the system begin the search at some other location besides the current line, enter the desired line number in brackets just before the search string.

Example: `/LIST [23] "Yesterday"`

This example begins searching at the LEFT margin of line 23 and lists the first line that contains the word "Yesterday".

Searches for strings are conducted in either a literal or non-literal mode.

When LITERAL is set to false (SET NONLIT), the TDP editor checks for context when searching for a string; thus the search for "is" does not find "this" or "isn't". The TDP editor will also locate a string if the first letter is upper case when the string given began with a lower case character; thus a search for "this" also locates "This", but not "THIS". In addition, the editor will treat blanks within the string in a special way; a search for "Los Angeles" is considered a search for the pair of words "Los" and "Angeles" and the pair will be found if there are no blanks between them, a single blank between them, or many blanks in between.

The search will find a string extending over two or more lines provided that:

1 LITERAL is false

2 The string contains more than one word

3 The line break is between words.

When LITERAL is set to true (SET LIT), the search is done for the string exactly as presented. A search for "is" finds "this" and "isn't"; it will not find "Is". In this case, the editor is looking literally at the string requested; no adjustment is made for upper case starting characters; no attention is paid to characters surrounding the string.

Searching for a particular string occurs in two different contexts. One is to locate positions within the file ; the other is to locate strings for a FIND or CHANGE command. The setting of the LITERAL parameter governs the search mode for both cases. The command option words LIT and NONLIT, however, may be used to override the setting of the literal parameter only for the second case. Thus if LITERAL is false, the command FIND "w","A"/".",LIT finds all the occurrences of the letter "w" regardless of context in the range from the next occurrence of the word "A" to the ".". The search begins with the occurrence of an isolated "A"; not with a character "A" embedded in another word.

In a document full of words, with no complicating symbols, the searching logic when LITERAL is false will find strings as generally expected. The exact logic of when a string is found depends on "appropriate" characters appearing in front of and behind the string in the file. (For the purpose of string searching, every line is considered to contain a blank before the first character of the line and another blank after the last.) The following table shows what is "appropriate":

If the first character is:	the string must be preceded by:
alpha	special or numeric
numeric	special
special	anything

If the last character is:	the string must be followed by:
alpha	special
numeric	special or alpha
special	anything

Thus according to this logic, if the string "AB24X" was in a file and the user searched for "AB24", the string "AB24X" would be found. A search for "AB2" would not find the string.

The TDP editor parameter BLIT allows the user to alter the logic used in finding strings. BLIT is true by default; the command SET BLIT reverses its value. If BLIT is false (and LITERAL is false), then strings ending in either alpha or numeric must be followed by a special character and strings beginning with either alpha or numeric must be preceded by a special character.

The percent (%) sign is used to denote special search strings and octal numbers:

%BLANK or %NIL is a blank line.
%DATE is the current date in the form mm/dd/yy.
%TIME is the current time in the form hh:mm AM (or PM).
%DTTM is the date and time in the form mm/dd/yy, hh:mm AM (or PM).
%FF (%13) is a form feed.
%CR (%15) is a carriage return
%BS (%10) is a backspace
%ES (%33) is an escape
%TB is the tabkey
%LF is a line feed
%X,%Y, %Z are the values stored in the X, Y, and Z buffers in TDP's calculator.
%TERM is the string identifying the user terminal through the SET TERM command.
%WORK is the name of the workfile.
%TEXT is the name of the text file.
%FOUND is the number of occurrences of the search string returned by FIND.
%ERROR is the error code of the last error returned.

The construct Z : : % is available primarily to access the strings %TERM, %WORK and %TEXT in USE file command lines. This use of Z : : will not affect any string stored with the Z : : = or ZP : : = commands.

For example, a USE file may be constructed to perform some manipulation on the workfile, then keep the file, and release it for use by others on the HP 3000. The permanent file can be released from within the USE file through the commands:

```
KEEP
RELEASE Z::%TEXT
```

Special editing keys

Certain keys perform special functions when used in conjunction with the Control key (CTRL). To invoke these functions, hold the (CTRL) key down and press the specified key.

(CTRL)-X

(CTRL)-X is used to erase the current line. MPE responds !!!, carriage return, and line feed; you then enter the new line.

(CTRL)-Y

(CTRL)-Y is used to terminate the processing of a TDP command. For instance, if you wish to stop listing a file after entering the command LIST ALL, you can type control-Y whenever you wish to stop. TDP responds . . . , rings the bell, and prompts you for further input.

(CTRL)-H

(CTRL)-H is used to "erase" characters one at a time, by backspacing over them and entering the correct characters.

(CTRL)-S

(CTRL)-S is used to temporarily suspend a listing on the terminal. You might use this when you wish to inspect a certain portion of the listing, then continue listing the rest of the file at regular speed.

(CTRL)-Q

(CTRL)-Q is used to continue a listing that was suspended with a (CTRL)-S. The listing continues from the point at which it was stopped.

Multipoint Terminals

On multipoint terminals the (Enter) key is used instead of (Return) to transmit data to the computer. Control S,Q and X have no equivalent on a multipoint terminal. (CTRL) H will produce a backspace. The effect of (CTRL) Y can be achieved by pressing and holding the (Break) key until any data being received completes and then pressing the (Enter) key.

Guide to TDP commands

This section provides a guide to TDP commands by describing the functions you might want to perform and listing the TDP command(s) appropriate to that function.

FUNCTION	COMMANDS USED	DESCRIBED IN
Add material	ADD	Chapter 3
	ADDLINE	"
	JOIN	"
	MERGE	"
	MAKE	"
	OVERLAY	"
	INCLUDE	Chapter 4
Blank lines	SPACE	Chapter 4
Branch within USE files	@GO	Chapter 3
	@IF	"
	@L	"
Calculator	=(calculator)	Chapter 3
	=L	"
	=M	"
	=S	"
	=TOTAL	"
	TOTAL	"
Center lines	CENTER	Chapter 4
Change workfile parameters	SET LEFT	Chapter 3
	SET RIGHT	"
	SET LENGTH	"
	SET SIZE	"
	SET	"
Code a file	KEEP,CODED	Chapter 3
	TEXT,CODED	"
Combine lines	FILL	Chapter 3
	GLUE	"
Copy lines	COPY	Chapter 3
Create a new workfile	ADD	Chapter 3
	MAKE	"
	TEXT	"
Delete material	CHANGE	Chapter 3
	COLMOVE	"
	DELETE	"
	WORDMOVE	"

FUNCTION	COMMANDS USED	DESCRIBED IN
Double space	DOUBLE	Chapter 4
	LINESPACE	"
Figures	BOX	Chapter 4
	FIGURE	"
	ILLUSTRATION	"
	TABLE FIGURE	"
	SPACE	"
	NAME RASTER	"
Find material in the workfile	FIND	Chapter 3
	FINDNEXT	"
	FINDNUMBER	"
Flag changes to the file	SET PMARK	Chapter 3
	SET STAMP	"
	LIST,PMARK	"
	LIST,NEW	"
	REVISION	Chapter 4
	RMARGIN	"
Footnotes	FOOTNOTE	Chapter 4
	^#F	"
Headings and footings	HEAD	Chapter 4
	FOOT	"
	HEADLINE	"
Hold file	ADD,HOLD	Chapter 3
	DELETE HOLD	"
	FIND,HOLD	"
	HOLD	"
	INSERT,HOLD	"
	KEEP,HOLD	"
	LIST HOLD	"
	REPLACE,HOLD	"
Hyphenate words	HYPHEN	Chapter 3
	HYPHEN	Chapter 4
	HYPHALLCAPS	"
	HYPHCHAR	"
	HYPHDBL	"
	HYPHFIRSTCAP	"
	HYPHLAST	"
Indent paragraphs	INDENT	Chapter 4
	INLFT	"

FUNCTION	COMMANDS USED	DESCRIBED IN
Index	INDEX	Chapter 4
	NAME INDEX	"
Justify the right margin	JUSTIFY	Chapter 4
	HYPHEN AUTO	"
	RIGHT	"
Laser printing	ACTIVATE	Chapter 4
	DEACTIVATE	"
	ENVIRONMENT	"
	FONT	"
	FONTEQ	"
	FONTID	"
	ILLUSTRATION	"
	LAYOUT	"
	VERIFY ENVIRONMENT	"
	BOLD	"
	GHOST	"
	^B{text}^S	"
	^G{text}^S	"
	^A(text)^N	"
	^F{id}{text}^S	"
Macro strings	M{x}	Chapter 4
	^M{x}	"
	^D	"
Mathematical expressions	EQN	Chapter 3
	EQUATION	Chapter 4
	^+	"
	^-	"
	^>	"
	^<	"
Minimize disc space	CRUNCH	Chapter 3
	UNCRUNCH	"
	FILL	"
Modify the workfile	ADD	Chapter 3
	DELETE	"
	JOIN	"
	OVERLAY	"
	SCREEN	"
	CHANGE	"
	COLINSERT	"
	COLMOVE	"
	COLREPLACE	"

FUNCTION	COMMANDS USED	DESCRIBED IN
Modify the workfile (cont.)	INSERT	Chapter 3
	MODIFY	"
	REPLACE	"
	WORDMOVE	"
Move material within the workfile	MOVE	Chapter 3
	COLMOVE	"
	WORDMOVE	"
Multi-column output	COLUMN	Chapter 4
	CMARGIN	"
	NEXT	"
	LAYOUT	"
New page	NEW	Chapter 4
	NEED	"
	LAYOUT NEW	"
	ACTIVATE	"
	DEACTIVATE	"
Number paragraphs	^#(n)	Chapter 4
	#(n)	"
	^#S	"
Page layout	PAGELENGTH	Chapter 4
	TOP	"
	BOTTOM	"
	LFT	"
	RHT	"
	INLFT	"
	INRHT	"
	VERIFY	"
Page numbers	PAGENO	Chapter 4
	SECTION	"
	ALTERNATE	"
	PAGENOLINE	"
	#P	"
	^#P	"
	^#S	"
Print enhancement	BOLD	Chapter 4
	^B	"
	GHOST	"
	^G	"

FUNCTION	COMMANDS USED	DESCRIBED IN
Print enhancement (cont.)	RED	Chapter 4
	^R	"
Print material	DRAFT	Chapter 3
	ENVELOPE	"
	FINAL	"
	LIST	"
	PRINT	"
Printer control	LPTOP	Chapter 4
	PITCH	"
	PROPORTIONAL	"
	WIDTH	"
Printing control	FINAL,ASK	Chapter 3
	EXIT	Chapter 4
	GO	"
	MARGIN	"
	PAGE	"
	PAUSE	"
	TRY	"
	COPIES	"
Prompt for input	Q	Chapter 3
	Z : :	"
	ZP : :	"
	M(x)=PROMPT(string)	Chapter 4
Renumber the workfile	KEEP,RESEQ	Chapter 3
	RESEQUENCE	"
Screen editing	SCREEN	Chapter 3
Store commands	REDO	Chapter 3
	Z : :	"
	@	"
	@M	"
	@D	"
	@F	"
	@S	"
Sort lines	RESEQUENCE,SORT	Chapter 3
Spelling check	CHECK	Chapter 3
	SPELL	"

FUNCTION	COMMANDS USED	DESCRIBED IN
Store workfile	CLOSE	Chapter 3
	KEEP	"
Table of Contents (automatic)	CONTENTS	Chapter 4
	NAME CONTENTS	"
Table of Figures (automatic)	TABLE FIGURE	Chapter 4
	NAME TABLE FIGURE	"
Tables	ALIGN	Chapter 3
	DELIM	"
	RACK	"
	SET COLSTOPS	"
	TOTAL	"
	=TOTAL	"
	=S	"
	BLANK	Chapter 4
	IMAGE	"
Time and date	TIME	Chapter 3
	^D	Chapter 4
Underline	UL	Chapter 4
	UW	"
	^U	"
	^W	"
	^-	"
Upshift/downshift	UPSHIFT	Chapter 3
	DOWNSHIFT	"
USE files	USE	Chapter 3
	@GO	"
	@IF	"
	@L	"

TDP Terms and conventions

This chapter describes the terms and conventions used to describe TDP commands and facilities in the rest of this manual.

Command descriptions

The command descriptions contained in Chapters 3 and 4 all follow a standard convention. The layout of a command description is illustrated on the following page. All TDP commands follow the same general format: the command name is entered first, followed by the required parameters (if any) and then any optional parameters desired. The syntax conventions used are as follows:

1 Parameters are shown from left to right in their required order of entry.

2 Parameters are required unless enclosed in brackets [ ]. The order shown for optional parameters is not important with regard to the entry order.

3 When only one of two or more parameters must be selected, the choices will be displayed one below the other and enclosed in braces { }.

4 The key words "IN", "TO" and "BY" can be replaced by commas.

Let's take a look at a sample command.

```
JOIN[Q] filename [filerange] TO (linenumber) [BY increment]
                                             [NOTEXT]
```

The JOIN command can optionally be followed by a "Q" (for quiet) to suppress the printing of the results of the join. The filename parameter is required, but the filerange parameter is optional. The linenumber parameter, which follows the "TO" or a comma, is also optional. You have the option of entering "BY" (or comma) followed by an increment, and you can optionally enter ",NOTEXT", which performs the same function as the "Q" option.

All commands may be entered in uppercase or lowercase letters. Example: ADD may be entered as ADD, Add, or add. In addition, many of the commands have abbreviations that can be entered instead of typing the whole command name. These abbreviations are noted in the description of each command, as well as in the Command Summary in Appendix B.

More than one command can be included on a single command line, if the commands are separated by semi-colons.

Command descriptions in Chapters 3 and 4 are presented in alphabetical order to make them easy to reference. If you do not know the name of a certain command, there are several ways to locate it: first, read through the groups of commands described in the **Guide to commands** in Chapter 1; and second, skim the **Command Summary** in Appendix B.

Command name

Command abbreviation

Purpose Describes, briefly, the function of the command

Form The formal syntax of the command.

Description Describes the function and effect of the command, and the effect of any mandatory parameters.

Limitations Describes any limitations on the use of the command where these are not obvious.

Options Describes the effect of any optional parameters.

Examples Provides short examples illustrating the use or effect of the command, where these are not obvious.

Related commands Lists commands having related functions.

Definition of terms

Some of the terms used to describe the commands require some explanation. Understanding the terms will enable you to make maximum use of your TDP system. All of the required parameters are discussed below.

Line number

Each line in the workfile is assigned a unique line number. Normally line numbers are eight digits long, with five digits (maximum) before the decimal point and three digits (maximum) after it. However, COBOL and DIARY format files have a different numbering scheme (see **File formats** in Chapter 1 for further information).

By default, a new workfile is numbered beginning with line number 1, with an increment of 1 for each line entered.

Unless you specify otherwise, when you save your workfile the line numbers are saved as part of the file, and appear associated with the same lines when the file is copied back into the work file for modifications. You may, however, choose to save the file without line numbers; in which case, new line numbers are assigned when the file is next copied into the workfile. If you text an unnumbered file, it remains unnumbered by default for subsequent KEEPs.

In general, it is preferable to keep your files with line numbers to maintain the same line numbers from session to session. This eliminates a great deal of potential confusion when editing from review copies generated at different times during the life of the document.

Current line pointer

TDP maintains a current line pointer which is used as the default position if no position is specified in a command. The current line pointer is usually left at the first column of the next line; LIST and MODIFY, however, leave the pointer at the end of the last line listed or modified.

You may reference the current line pointer with an asterisk (*). Example: LIST * lists the current line. Offsets are also allowed: *+1 means one line after the current line pointer. Note that when a range is specified, the current line pointer is moved to the beginning of the range. Thus, LIST *+3/*+2 lists three lines, starting three lines past the current pointer, and stopping two lines later.

Column positions and locations

Specific position within the line can be referenced with the column number. The first column is 1, the last column 72 (for the default line length). For a maximum length line, the last column is 168.

If a column number is used with a line number in a command to specify the exact position (ie. the location), it appears in parentheses following the line number. For example: 13(37) means the 37th column of line number 13.

If a column number is itself used as a parameter for a command (with COLINSERT, for example), it is not placed in parentheses.

The keywords FIRST and LAST can be used to refer to the first and last non-blank characters in a line or as the first and last records in the workfile. TDP interprets these keywords according to their context: used with column commands, they are interpreted as column specifications; used in commands specifying line numbers, they are interpreted as line number specifications.

Offset numbers can be used to qualify locations: for example, FIRST+10 refers to the tenth column in the line, or the tenth line in the file, depending on the context. Similarly, 13+10 is the tenth line past 13; "The"+10 is the tenth line past the next occurrence of "The"; and *+10 is the tenth line past the current line pointer.

When specifying column positions, you use LEFT to indicate the low column or LEFT setting, and RIGHT to indicate the high column or RIGHT setting. There are two ways of specifying offsets to column positions.

(*+n) and (*-n) locate a position n columns to the left or right of the current column position. Where the offset falls outside the margins an error message is given.

(+n) and (-n) reference a position n non-blank characters to the left or right of the first non-blank character between the margins of the current line. Only characters between the margins are counted and counting continues onto subsequent or previous lines until the offset is satisfied.

Strings

A string is a group of characters enclosed in special characters. A string can consist of a character, a word, or several words. The special characters enclosing the string are called "delimiters". Accepted delimiters include the 19 characters shown on the line below:

! " $ & ' = ^ ~ ` @ | \ { : } _ < > ?

Strings can be used by TDP to locate a position within the workfile. The system begins searching for the string from the current line pointer, and continues searching until the string is found or the end of file is reached.

Range

A range is used to indicate one or more lines, or parts of lines, that are to be affected by a command. A rangelist is a list of ranges, separated by commas.

A range of lines is specified by a starting line or location, then a slash, and then the ending line or location. For example: 3/10 indicates the range of lines from 3 through 10. Similarly, "The"/"." indicates the range starting with the next occurrence of "The" and continuing until the next " . " ; * / * + 10 indicates the range of lines starting at the current line pointer and continuing for ten lines; and 10/"." indicates the range starting with line number 10 and ending with the next line containing " . ".

If the range is indicated by line numbers, then the second number must be greater than the first, or the range will be considered empty, and an error message will be displayed. If one of a list of ranges is empty, TDP displays a message, but completes the operation on the remaining ranges. There are four special range indicators: WINDOW, ALL, FIRST and LAST.

WINDOW specifies the range of lines starting with the next line and continuing for a preset number of lines (default is 18 lines).

ALL specifies all lines in the file.

FIRST specifies the first line in the file and LAST specifies the last line in the file. Thus, LIST FIRST/LAST is equivalent to LIST ALL. In column ranges FIRST and LAST specify the first and last non-blank character.

Filerange

Some TDP commands allow operations on files other than the workfile (e.g. PRINT). A filerange can be specified with these commands in the same way as a range is used with the workfile. With the limitation that *,offset numbers and column references are not allowed, a filerange has the same format as a range. An additional feature of filerange is the ability to specify record numbers with the "#" character. eg PRINT LFILE #2/#4 would print the second, third and fourth record of the file LFILE. The first record in a file is considered to be record #1.

Wordlist

A wordlist is one or more words (to a maximum of 15) enclosed in parentheses and separated by a space. Some commands accept a wordlist as a parameter which permits you to search for up to fifteen separate words at a time. For example, the CHANGE command can search for a list of words, changing each one that is found to an appropriate new word. The only restriction on this command is that the total number of characters must be less than or equal to 72.

Filelist

A filelist can be specified as a list of files separated by commas, as a range of files (filename/filename) or as a set of files identified by one of the wild card characters (@, #, or ?).

Linelist

A linelist is one or more line numbers separated by commas.

Rangelist

A rangelist is a list of ranges separated by commas.

Token

A token is a collection of non-blank characters preceded and followed by a space.

Textfile

The textfile is an external file which is brought into the TDP subsystem (copied into the workfile) with a TEXT command. The contents of the TEXT file become the contents of the workfile.

Workfile

The workfile is a temporary file containing the text upon which TDP operates. If new information is being ADDed, it is placed into the workfile from the terminal. If a file is to be modified, it is first copied into the workfile to prevent the accidental destruction of the original information. Refer to Appendix C for technical details about the workfile.

Join, Merge, Overlay files

An external file which is brought into the TDP subsystem (copied into the workfile) with a JOIN, MERGE, or OVERLAY command is called the join, merge, or overlay file, respectively. (See Chapter 3 for details on these commands.)

Hold file

The hold file serves as a temporary storage file for TDP and is generally used for holding interim information (for example, copying parts of the work file to the hold file and then adding or inserting the hold file into other places within the workfile at some later time).

The hold file is created by TDP when the first HOLD command is entered. The hold file can be kept as a permanent file, listed, or added to. See Chapter 3 for details.

Use file

The use file is an external user file containing TDP commands (and, optionally, text records). The use file is called with a USE command (see Chapter 3).

TDP allows conditional and unconditional branching within use files.

Include file

The include file is an external file referenced by the \INCLUDE command, which is merged into the output file when it is printed (see Chapter 4).

Monitor file

This file can be assigned to store a record of all the commands entered during an editing session. It can also be used to indicate the value of certain parameters during the session, see the MONITOR command (Chapter 3) for details.

Block

In formatting terms, a block is considered to start with the next text line and continues until a formatting command line (normally signified by a \ character) or a blank line is encountered.

Printer

Any output device.

Terminal

Any input device.

Editing commands

Introduction

TDP is a command-driven text processing system. There are two types of commands: the first, described in this chapter, are commands entered on a command line and executed immediately. These are called editing commands. Examples are TEXT, ADD and MODIFY. The other commands, described in Chapter 4, are entered on a separate line in the workfile, retained as a permanent part of that document, and executed only when output is requested with a DRAFT or FINAL command. Examples of formatting commands are CENTER, JUSTIFY, SPACE, and INDENT.

Command format

The conventions used to present the form of each command are described in Chapter 2. Briefly, they are as follows:

1 Parameters are shown from left to right in their required order of entry.

2 Parameters are required unless enclosed in brackets [ ]. The order shown for optional parameters is not important with regard to the entry order.

3 When only one of two or more parameters must be selected, the choices will be displayed one below the other and enclosed in braces { }.

4 The key words "IN", "TO" and "BY" can be replaced by commas.

5 Commands can be entered in upper or lower case, or any combination thereof.

For detailed information on terms used in the command descriptions, please see Chapter 2.

Optional parameters

Virtually all editor commands allow the optional parameter Q, to suppress the listing of processed lines of text. In most cases, the optional parameter NOTEXT is allowed in place of Q. Therefore, the NOTEXT parameter is not generally documented as an option. In the few cases where NOTEXT is not a valid option, this is separately noted under the relevant command.

ABORTSPOOL

Purpose

To unconditionally stop the TDP Spooler.

Form

```
ABORTSPOOL
```

Description

This command stops the TDP spooler immediately with the suspension of any jobs currently being printed. Suspended print jobs will request printing when the spooler is restarted. The command STOPSPOOL would normally be used to stop the spooler since this allows any spoolfiles currently being printed to complete before the spooler is stopped.

Limitations

None.

Options

None.

Related commands

ALTERSPOOL
SHOWSPOOL
STARTSPOOL
STOPSPOOL

ADD

A

Purpose

To enter new material into the workfile.

Form

```
ADD[Q]  [linenumber]  [,HOLD[Q] ]
        [string    ]  [,filename]
```

Description

The ADD command is used to add one or more lines to an existing workfile, or to the beginning of a new workfile. The options "linenumber" and "string" allow you to place the added material at a specific location within the file. The HOLD option allows you to add lines of text which have been stored in the "hold" file.

TDP continues to prompt you for more lines until you signal that the addition is complete by typing // directly after the line number prompt. Alternatively, the // can be typed at the end of the last line, but that line is not included in the file.

Limitations

If you attempt to add a line that is longer than the present value of RIGHT, it will be truncated to the nearest column (not normally within 12 characters of RIGHT) and on a space if possible. A message will be printed. (See the SET command later in this section for instructions on resetting the line length.)

Options

[linenumber] The line number option causes the material to be placed in a specific location. If no line number is specified, the added lines are placed at the end of the existing workfile. If a line number is specified and it does not currently exist in the workfile, the addition begins at that line number. If the specified line number already exists, the addition begins at the next available line number. For example, if lines 2.10 and 2.11 exist, ADD 2.1 will add an extra line at 2.101.

[string] When the string option is used to indicate the placement of added lines, the workfile is searched from the current line pointer to the next occurrence of the search string. The addition will begin on the next available line.

[Q] When the Q option is appended to the ADD command, the line number is not printed as a prompt. However, when the addition is completed, the last line number is shown.

[HOLD] When the HOLD option is selected, the content of the hold file is added at the specified location. The hold file is not cleared by this operation. The Q option used with HOLD suppresses the listing of the material added from the hold file. (See the HOLD command later in this section.)

[filename] The filename option is only valid if you are using, and have previously identified, an HP 2642 terminal. If a filename has been given, the lines of text to be added to the workfile are taken from the file with the given name on the HP 2642 mini disc. NOTE: both filename and HOLD cannot be given simultaneously.

It is possible to temporarily exit from ADD mode to execute any other TDP editing command (with the exception of USE, DELETE ALL or another ADD command), and then return automatically to the point of interruption. Only one command can be specified in this manner; multiple commands are not accepted. Simply type // followed by the desired command.

Note that if you set the prompt to another character (say !), the double slash is still used to terminate add mode. However, to call another command from within add mode, you would type !! and the appropriate command.

Examples

Example 1: `/ADD`

Suppose your workfile contains three lines as follows:

```
 1   This publication is the reference manual for Text and
 2   Document Processor (abbreviated TDP in this
 3   manual), a text processing system used to create,
```

This example, then, prompts you for line number 4 and as many lines as you wish to add:

```
 4   and format text files for draft or final printing
 5   of a variety of documents.
 6   Using TDP you can enter a document at the
 7   terminal, store it in a disc file, print a draft
 8   copy for review, make the necessary corrections, and
 9   print the final document when all changes are made.
10   //
```

The double slash in line 10 terminates the command.

Example 2: `/ADD 3`

Using the same workfile as Example 1, this command would add one or more lines immediately after line 3. If line 3 did not already exist in the workfile, the first added line would be line 3.

```
3.1  This line was added with the command /ADD 3.
3.2  You may add as many lines as appropriate, except
3.3  that if you reach 3.999, you will receive a message
3.4  informing you that no more lines can be added, since
3.5  3.999 is the last available line between 3 and 4.
3.6  //
```

The same addition could be accomplished by the command /ADD 3.1.

Example 3: `/ADD "TDP"`

This example searches from the current line pointer (assuming line 3.6) until it finds the string "TDP", which in this case occurs in line 6. Then you are prompted for the next available line number, which in this case would be 6.1:

```
6.1     Here is the insertion in the line after my search.
6.2     As with other forms of the ADD command, I may add as
6.3     many lines as I wish.
6.4     //
```

The content of the workfile, after all of the additions (in the above examples) are made, is as follows:

```
1       This publication is the reference manual for Text and
2       Document Processor (abbreviated TDP in this
3       manual), a text processing system used to create,
3.1     This line was added with the command /ADD 3.
3.2     You may add as many lines as appropriate, except
3.3     that if you reach 3.999, you will receive a message
3.4     informing you that no more lines can be added, since
3.5     3.999 is the last available line between 3 and 4.
4       and format text files for draft or final printing
5       of a variety of documents.
6       Using TDP you can enter a document at the
6.1     Here is the insertion in the line after my search.
6.2     As with other forms of the ADD command, I may add as
6.3     many lines as I wish.
7       terminal, store it in a disc file, print a draft
8       copy for review, make the necessary corrections, and
9       print the final document when all changes are made.
```

Related commands

ADDLINE
ADDSINGLE
HOLD
INSERT
JOIN
MAKE
MERGE
OVERLAY
RESEQUENCE
SET

ADDLINE
AL

Purpose

To add a single line, included as part of the command, to the workfile.

Form

```
ADDLINE[Q] [linenumber] [,text]
           [string    ]
```

Description

ADDLINE is a special case of the ADD command. The text is added at the specified line number or the next available line number if the line number already exists in the workfile. The added line is displayed with its line number. The ADDLINE command is especially useful in USE files.

Limitations

No other command can appear with the ADDLINE command on the command line, since it would be misconstrued as part of the text. Therefore, you cannot retrieve a line from the hold file with this command since a "HOLD" option would be misconstrued as part of the text.

Note that NOTEXT is not an allowable option with this command.

Options

`[linenumber]` The line number option causes the material to be placed in a specific location. If no line number is specified, the added line is placed at the end of the existing workfile. If a line number is specified and it does not currently exist in the workfile, the addition begins at that line number. If the specified line number already exists, the addition begins at the next available line number.

`[string]` When the string option is used to indicate the placement of an added line, the workfile is searched from the current line pointer to the next occurrence of the search string. The addition will begin on the next available line.

`[Q]` When the Q option is appended to the ADDLINE command, the line is not displayed.

`[text]` Note that the text is not entered in quotes. If no text is supplied, a blank line is added. If no line number or string is specified, the text is added at the end of the file. When the line has been added, you are prompted as usual for another command.

Examples

Example 1: `/Addline 4, this was added with the ADDLINE command`

After this addition, the text file would look like this:

```
4     The original line 4 would be here.
4.1   this was added with the ADDLINE command
5     The original line 5 would be here.
```

Example 2: `/ADDLINE *, this text will be added also.`

In this case, the text in the command line would be added at the current line pointer.

Related commands

ADD
ADDSINGLE
INSERT
RESEQUENCE
SET
USE

ADDSINGLE
AS

Purpose

To add a single line to the workfile.

Form

```
ADDSINGLE[Q]  [linelist]
```

Description

ADDSINGLE is used to add a single line, or a series of single lines to the workfile. After you enter the line, you return to command mode. You may enter this command while in add mode, by typing // followed by the command.

Limitations

You cannot retrieve a line from the hold file with this command. Also, you cannot use // followed by another command within this command.

Options

`[Q]` The Q option suppresses the line number prompt for the line(s) you are adding.

`[linelist]` The linelist option is used to specify one or more line numbers where you want to add a line. You are prompted for one line to be added at each specified line (or the next available line). If no linelist is given, you are prompted for one line to be added at the end of the workfile.

Examples

Example 1: `/ADDSINGLE`

Assuming that the last line in your workfile is line number 10, you will be prompted for line number 11. After you enter the line, you return to command mode.

Example 2: `/ADDSINGLE 2, 4.1, 7.3`

You will be prompted for one line at each of the three positions in the workfile: line number 2 (or the next available line, if line 2 already exists), line 4.1, and line 7.3. After you have entered the three single lines, you return to command mode.

Related commands

ADD
ADDLINE
INSERT
RESEQUENCE
SET

ALIGN

Purpose

To align tabular information on preset column stops.

Form

```
ALIGN(R                 ) [Q]{rangelist}[,ADJUST][,NOTEXT]
     (L                 )
     (C                 )
     (D                 )
     (special character)
```

Description

This command can be used to align columns of tabular material relative to the currently set column stops (not tab stops!). The default stops are 10, 20, 30, 40, 50, 60 and 70. New column settings can be defined with the SET COLSTOP command.

The material for each column is called a "token", and consists of a group of non-blank characters separated by one or more blanks. The last letter of the command determines which character in the token is used to position the token.

The ALIGNR (right), ALIGNL (left) and ALIGNC (center) commands align the indicated character (rightmost, leftmost, or center) on a column stop. The ALIGND (decimal point) and ALIGNspecial (special character, as in ALIGN$) commands align the indicated character one position to the right of a column stop. This allows mixed tokens (ie. some with decimal points or special characters and some without) to be properly aligned.

R = the right-hand character.

L = the left-hand character.

C = the character in the center of the token. If the token has an even number of characters, the character to the left of center will be used for alignment.

D = the first occurrence of a decimal point within the token. If, a token does not contain a decimal point, it is assumed to be to the right of the token.

special character = the first occurrence of the specified character within the token. If a token does not contain the specified character, it is assumed to be to the right of the token.

ALIGN

The ALIGN command follows these rules:

1 If the token is too long to fit at the current column stop, an attempt is made to align it on the next columnstop. If it won't fit there the token and any remaining tokens are placed to the right of the already aligned tokens and are separated by single spaces.

2 When there are more tokens than column stops, the excess tokens are put on the right side of the line with single blanks between them.

3 When aligning tokens would make a line longer than the current line length, an error message is printed and no alignment is carried out.

Limitations

None.

Options

[Q] or [NOTEXT] The Q or NOTEXT option suppresses the listing of the affected lines.

[ADJUST] The ADJUST option takes into account the special escape sequences used by the formatter for underlining and other intraline formatting. "Escape" in this case refers to certain commands embedded in the line -- not to the ESC key or escape sequences specific to the terminal. The tokens will be adjusted so that the printed copy will be aligned properly. However, when automatic paragraph numbers, macros, or date substitution are used, ADJUST cannot make the necessary adjustment since there is no way for TDP to know the length of the string being added.

[rangelist] The rangelist option specifies the range of lines over which the ALIGN command will operate.

Note that to align only a portion of the specified lines, you can change the settings of LEFT and RIGHT, since ALIGN only works on material between the current settings of LEFT and RIGHT. (See the SET command later in this section.)

Examples

Example 1: `/ALIGND 1/2`

Suppose you have input some financial data as follows and want to align it:

```
1  122.67 793.15 16.92 1392 6792.19
2  21.76 39 1926.55 3391.17 29.91
```

Then the command above will align the columns as shown below with the decimal points on the column stops.

```
1   122.67    793.15     16.92   1392        6792.19
2    21.76     39      1926.55   3391.17       29.91
```

Example 2: `/ALIGNL 1/2`

A list of names on the other hand, would probably be aligned by the left-hand character. These names, input without any attention to spacing, generate the list shown below when the example is used.

Original lines input:

```
1  Jenkins Doerr Zalewski Coleman Andersen
2  Moynihan Smith Koehler Stewart McKinney
```

Lines after ALIGN command is used:

```
1  Jenkins   Doerr    Zalewski   Coleman   Andersen
2  Moynihan  Smith    Koehler    Stewart   McKinney
```

Related commands

DELIM
RACK
SET COLSTOP
SQUEEZE
TOTAL
=TOTAL

ALTERSPOOL

Purpose

To alter the status of a spoolfile.

Form

```
ALTERSPOOL file id#,(STOPPED  )
                    (CANCELED )
                    (READY    )
```

Description

The ALTERSPOOL command changes the spoolfile with the given file id# to the given status (see Limitations below). The file id# and existing status of a spoolfile can be determined with the SHOWSPOOL Command.

Limitations

The status changes will only be made if the spooler job is running.

Only the following status changes are allowed.

A READY file can be made STOPPED or CANCELED

A STOPPED file can be made READY or CANCELED

A PRINTING file can be made STOPPED

Users with System Manager capability or HPOFFICE account members with Account Manager capability can alter the status of any spoolfile. Users with Account Manager capability can alter the status of any spoolfile in their account. Users without Account Manager or System Manager capability can only alter the status of spoolfiles they have created.

Options

None.

Example

```
/ALTERSPOOL 23,STOPPED
```

This command alters the status of the spoolfile with file id# 23 to STOPPED.

Related commands

ABORTSPOOL
SHOWSPOOL
STARTSPOOL
STOPSPOOL

CATALOG

Purpose

To display information about files currently stored for an account. NOTE: Account Manager or System Manager capability is required for this command.

Form

```
CATALOG filelist
```

Description

The CATALOG command supplements the MPE LISTF command by accepting file lists and file sets for file information output. CATALOG prints a heading for the list, and one line of information for each file.

A group name can be used if you have Account Manager capability. A group and account name can be used if you have System Manager capability.

The filelist can be specified as an alphabetical range such as APPLE/BANANA; or with the "wild card" characters:

@ - which specifies zero or more alphanumeric characters.

# - which specifies one numeric character.

? - which specifies one alphanumeric character.

These can be mixed with alphanumeric characters to obtain necessary information on a desired file set as in the examples below.

@COMP@	when the @ sign is used in this way, any file whose name contains the letters COMP in any position is listed (COMPUTE, INCOMPLT, etc.).
M##	when the # sign is used, any file beginning with M followed by two digits is listed (M29, M73, etc.).
K#@	When the # sign and @ sign are used like this, any file beginning with K then a number then anything else will be listed (K2379001, K74X12Z, etc.).

Limitations

CATALOG can only be used by users with Account Manager or System Manager capability.

Options

None.

CATALOG

Examples

Example 1: `/CATALOG CHAP1`

This command prints the following lines of output on MPE V:

```
CHAP1.ACC.NAME Created by MGR on mm/dd/yy

Last accessed:  mm/dd/yy   Last modified:  mm/dd/yy
Flags: Store( ) Restore( ) Load( ) Excl.( ) Read( ) Write( )
Record size  80 bytes       End of file at 284
 Block size 640 words      Record limit is 284
  File code   0              Sectors used   95
```

and the following lines on MPE/XL:

```
CHAP1.ACC.NAME Created by MGR on mm/dd/yy

Last accessed:  mm/dd/yy   Last modified:  mm/dd/yy
Record size  80 bytes       End of file at 284
 Block size 640 words      Record limit is 284
  File code   0              Sectors used   95
```

Example 2: `/CATALOG  A/CHAP3`

This example catalogs all files beginning with the letter A; the letter B, and the letter C up to the word CHAP3. Here is a sample display:

```
                             File  Rec.  Blk.                  Create
                     Creator Code  bytes words   EOF  Sectors  Date
ANNES.GRP.ACCT         MGR    0       80   640   140     50    3/22/88
CHAP1.GRP.ACCT         MGR    0       80   640   284     95    4/19/88
CHAP3.GRP.ACCT         USER   0      616  3080   143   6027    6/27/88
```

Related commands

FLUSH

CHANGE

C

Purpose

To replace a string with a new string.

Form

```
CHANGE[Q] (string          ) TO (string    ) [IN rangelist]
          ((wordlist)      )    ((wordlist)) [,ASK        ]
          (column[/column] )    (column    ) [,LIT        ]
                                             [,NONLIT     ]
                                             [,SL         ]
```

Description

The CHANGE command is used to change occurrences of one string, certain columns, or a list of words to a new string. The alternate form of the command is used to make multiple changes of individual words with one command.

Limitations

You can not change across line boundaries, nor make a change that would result in a line longer than the currently set line length. The LIT option may not be used when a word list is specified.

Options

[IN rangelist] The rangelist option allows you to specify the lines in which the column changes are to be made. If no range is specified, the search and subsequent CHANGE are done only on the current line. To search and change the entire file, specify IN ALL.

[ASK] The ASK option queries you each time the search string is found, asking if it should be changed. Answer "yes" to complete the change, and "no" to search for the next occurrence of the string.

[Q] The Q option suppresses listing the lines as they are changed.

[LIT]/[NONLIT] The LIT option means that the change will be made whenever the search string is found -- even if it is in the middle of another word. NONLIT means that the search string is changed only if it is a discrete word. Default setting is NONLIT; the default can be changed to LIT by the SET command.

[SL] The SL option, available only when the HIGHC and LOWC parameters have been set with the SET command, restricts the search to the column positions specified in those parameters. This would allow you to search one column of a multi-column workfile, for example.

CHANGE
C

Examples

Example 1: `/CHANGE "TDP/V" to "TDP" IN ALL`

This example searches all the lines in the text file and changes "TDP/V" to "TDP" whenever that string is found.

Example 2: `/change 14 to "$", 3/6,10`

This command changes column 14 to a "$" in lines 3 through 6, and in line 10. The text would look like this after the change:

```
 1  Here is a list of the charges for various parts
 2  used in your assembly line:
 3  Vendor A      $100.00
 4  Vendor B      $ 77.32
 5  Vendor C      $  5.65
 6  Vendor D      $ 99.99
 7
 8  Later in the document I might
 9  list the total.
10  Total         $282.96
11  That ends the financial section of this
12  document.
```

Example 3: `/CHANGE (is are) TO "will be" IN 35/70, ASK`

Here all occurrences of "is" or "are" in lines 35 through 70 will be displayed, one at a time, and you will be asked if the change should be made.

Example 4: `/CHANGE (this that) TO (that this) IN ALL`

In this example, each of the words is changed to the corresponding element of the new word list whenever it is encountered in the workfile. This form of the command is more efficient than using multiple CHANGE commands.

Related commands

COLINSERT
COLMOVE
COLREPLACE
FIND
MODIFY
REPLACE
SET
WORDMOVE

CHECK

Purpose

To check the spelling of words within a specified range of the workfile (HP SPELL must be installed on your system).

Form

```
CHECK[Q] [rangelist] [,UNN     ]
                     [,OFFLINE ]
                     [,NEW     ]
                     [,LONG    ]
```

Description

The CHECK command can be used to check the spelling of part or all of the workfile. The amount of text that is checked is limited by the LEFT and RIGHT margin settings; any words that lie outside the margin settings are not checked. Any portion of the workfile, or all of it, may be checked for spelling mistakes.

The language and the user dictionary are specified by using the SET LANGUAGE and SET DICTIONARY commands.

When an incorrect word is found, that word is highlighted.

The CHECK command leaves the current line pointer at the end of the last line checked.

Limitations

The spelling of words that cross line boundaries cannot be checked.

Lines beginning with \ in column 1 are treated as formatter command lines and ignored.

Options

[rangelist] The rangelist option is used to specify the lines of the file that contain words that are to be checked for spelling mistakes. If no rangelist is given, only words on the current line will be checked. To check the whole file, specify ALL.

[UNN] If the UNN option is chosen, the lines will be listed without line numbers.

[OFFLINE] To produce a listing on the system printer, select the OFFLINE option. Words that are not found in the dictionaries will be underlined.

[LONG] The LONG option is used to check the spelling of all lines that are longer than a certain length (the length is set with the HIGHC parameter on the SET command).

CHECK

[NEW] The NEW option is used in conjunction with the proof marking feature (see the SET command). Only the lines with proof marks will be checked for misspellings. The lines will be listed with the proof marks shown next to each line.

[Q] The Q option suppresses the listing of those lines that contain no misspelled words.

Examples

Example 1: `/CHECK ALL`

The simplest example. This command checks the spelling of the whole workfile. The workfile is displayed with spelling mistakes highlighted.

Example 2: `/CHECKQ 100/1100`

This command checks the spelling lines 100 through 1100 of the workfile displaying only those lines that contain spelling mistakes. The misspelled words are highlighted.

Related commands

SET
SPELL

CLOSE

Purpose

To close the workfile as a permanent or temporary file, depending on its present status.

Form

```
CLOSE
```

Description

The CLOSE command closes the workfile as a permanent file, if it is a permanent file, or as a temporary file, if it is a temporary file. The command clears the workfile.

The advantage of using CLOSE is that it avoids the time delay of TEXT and KEEP commands. This is significant when working with very large files. If you TEXT a CLOSED file it is not copied to a workfile; it is simply opened and becomes the workfile.

Limitations

The CLOSEd file retains the workfile's K-name. If another name is preferred, use the MPE RENAME command.

Options

None.

Related commands

KEEP
TEXT

COLINSERT
CI

Purpose

To insert a string in front of a specified column or string.

Form

```
COLINSERT[Q] {column} TO string  [IN rangelist]
             {string}            [,ASK        ]
                                 [,LIT        ]
                                 [,NONLIT     ]
                                 [,NOTEXT     ]
```

Description

COLINSERT inserts a string in the specified column, in front of whatever character currently occupies that position. The original characters are shifted to the right to make room for the insertion. Alternatively, a string may be specified to define a column position.

Limitations

If the insertion would extend the line past the current line length, the command is not executed and a message to that effect is displayed.

Options

[IN rangelist] The rangelist option allows you to specify the lines in which the column inserts are to be made. If no range is specified, the search and subsequent COLINSERT are done only on the current line. To search and change the entire file, specify IN ALL.

[Q] or [NOTEXT] The Q and NOTEXT options suppress the listing of the changed lines.

[ASK] If the ASK option is selected, you will be queried before each insertion is made. Answer "yes" if you want the insertion to be completed; otherwise, "no" will cause the insertion to be skipped.

[LIT]/[NONLIT] If a string is used to define the column, the LIT option means that the insertion will be made whenever the search string is found -- even if it is in the middle of another word. NONLIT means that the insertion is made only if the search string is found as a discrete word. Default setting is NONLIT; the default can be changed to LIT by the SET command.

Examples

Example 1: `/COLINSERT "string", "new" IN 20,27,33,52,69,101`

This example inserts the word "new" in front of any string "string" found in the specified rangelist. The characters including and following the string "string" are shifted to the right.

Example 2: `/COLINSERT 1 , "     ", 30/50,ASK`

Here five blank spaces are inserted at the beginning of each line in lines 30 through 50. You are asked if you want the insertion made before each line is changed.

Related commands

CHANGE
COLREPLACE
INSERT
MODIFY
REPLACE

COLMOVE
CM

Purpose

To move characters from one position to another within a line.

Form

```
COLMOVE[Q] columnrange,column [,rangelist]
                              [,NOTEXT   ]
```

Description

The COLMOVE command moves characters around within a line. The characters being moved will be removed from their current position and put into the line starting at the specified column. The original characters in the line are moved to the right or to the left, so that the characters being moved can be inserted where requested.

By using a destination column of 0, you can delete columns from the line. (The same thing can be done with the CHANGE command, using the null string ("") as the newstring.)

Limitations

If the line would extend beyond the current setting for the right margin, the move is not completed and an error message is printed.

Options

`[Q] or [NOTEXT]` The Q or NOTEXT option suppresses the listing of the modified lines.

`[rangelist]` The rangelist option specifies the rage of lines to be processed by the command.

Examples

Example 1: `/COLMOVE 1,9`

If line 1 consists of the numbers from 1 to 9 in columns 1 to 9, the result is:

```
1       234567891
```

Example 2: `/COLMOVE 1/3,8`

Using this command on the result from Example 1, line 1 will now be:

```
1       567891 234
```

Related commands

CHANGE
COLREPLACE
DELETE
MODIFY
WORDMOVE

COLREPLACE

COLREP

Purpose

To replace the character(s) in specified columns of the line.

Form

```
COLREPLACE[Q] (column) TO newstring [IN rangelist]
              (string)              [,ASK        ]
                                    [,LIT        ]
                                    [,NONLIT     ]
                                    [,NOTEXT     ]
```

Description

COLREPLACE is used to replace column positions. It is useful primarily for constructing tables in conjunction with the use of a USE file.

The replacement begins at the specified column and "newstring" overlays whatever characters are currently in that position. If "newstring" is shorter than "string" , blanks are used. COLREP is intended for USE files to allow users to maintain alignment of the right side of the line.

Limitations

None.

Options

[Q] or [NOTEXT] The Q or NOTEXT option suppresses the listing of the modified line.

[ASK] The ASK option queries you each time the search string is found, asking if it should be replaced. Answer "yes" to complete the replacement, and "no" to search for the next occurrence of the string.

[IN rangelist] The rangelist option allows you to specify the lines in which the column replacements are to be made. If no range is specified, the search and subsequent COLREPLACE are done only on the current line. To search and change the entire file, specify IN ALL.

[LIT]/[NONLIT] If a string is used to define the column, the LIT option means that the replacement will be made whenever the search string is found -- even if it is in the middle of another word. NONLIT means that the replacement is made only if the search string is found as a discrete word. Default setting is NONLIT; the default can be changed to LIT by the SET command.

COLREPLACE
COLREP

Example

```
/COLREP "Name of person" TO "Joe's"
```

This example changes this line:

```
1   Name of person phone number is 01 949 8605
```

To this:

```
1   Joe's          phone number is 01 949 8605
```

Related commands

CHANGE
COLINSERT
COLMOVE
MODIFY
WORDMOVE

COPY

Purpose

To copy lines from one part of the workfile to another; the lines remain in the original location.

Form

```
COPY[Q]{rangelist    } TO {linenumber} [BY increment]
                          {string    } [,NOTEXT     ]
```

Description

The COPY command duplicates lines in the workfile and places them in the specified location, numbering them according to your instructions. If the linenumber specified as the destination already exists, the material will be copied starting with the next available line.

If the material to be copied does not fit into the specified location, the copy is not completed and an error message is printed. When necessary, the MOVE command can be used prior to a COPY to enlarge the interval into which the material is to be copied.

It is possible to copy from more than one place in the work file with one command; however, only one destination is allowed. The ranges must be separated by commas, and the destination must be preceded by the word TO.

A range can be specified by a starting string and an ending string. In such a case, only the text between and including the two strings will be copied.

A search string can be specified to determine the destination of the material to be copied. If so, the copied lines will be placed on the next available line after the line containing the search string.

Limitations

The COPY command operates only within the boundaries established by the SET LEFT and SET RIGHT commands.

Options

`[Q]` or `[NOTEXT]` The Q or NOTEXT option suppresses the listing of the lines in their new location.

`[BY increment]` The BY increment option allows you to determine the numbering of the new lines (BY 0.1 would number the new lines at location 24, for example, as follows: 24.1, 24.2, 24.3,etc.). If there isn't room in the specified location using the increment you have specified, a smaller increment will be used.

COPY

Examples

Example 1: `/COPY 27/29 to 113 by 0.1`

Lines 27 through 29 will be entered in the workfile at 113 or the next available line number; the new lines will be numbered in increments of 0.1. So the result will be:

```
113.1  line 27 goes here
113.2  line 28 goes here
113.3  line 29 goes here
```

Example 2: `/COPY 17 to "quarter"`

This example copies line 17 to the next line after the occurrence of the search string "quarter".

Related commands

MOVE

CRUNCH

Purpose

To minimize the amount of disc space used by a file.

Form

```
CRUNCH (filename)
```

Description

The CRUNCH command is used to process a file so it uses less space than in its original form, but can still be returned to its original state. CRUNCH can be used on program source files (mostly upper case), or document files (mostly lower case).

Typical savings are 65%-75% of the original disc space used; however, the command uses a lot of CPU time, so it should not be used with frequently edited files.

Crunched files are read automatically by the PRINT and TEXT commands. However, you may not request DRAFT or FINAL output with a crunched file as the input file. Use UNCRUNCH to restore the file to its original form, then carry on.

Limitations

There are restrictions on the files to be crunched:

1 For files to be crunched as numbered files, sequence numbers must be either eight digits long and at the end of the record or six digits long and at the front. (Those are the two default cases produced by TDP for normal files and COBOL files. Sequence numbers are 8 digits for DEFAULT or DIARY format files, 6 digits for COBOL format files.)

2 The sequence number fields must hold legitimate sequence numbers. (This is no restriction for files kept by TDP.)

3 The file is assumed to contain only printable characters. Any character in the file which is not part of the normal printable character set is translated into a single code. An error message is sent at the time the file is crunched. (Examples of non-printing characters are the (Back space) and the (Tab) key. If a file contained both, the crunching process would translate the two into the same code. The uncrunch process would then produce a file different from the original.)

Options

None.

CRUNCH

Example

```
/CRUNCH SAM
```

The dialogue goes like this:

```
Numbered file?  (answer yes or no)
Size of seq?    (enter 8 for DIARY or DEFAULT files;
                 6 if they are COBOL files)

Output file:   (assign any name as the output file for
               the crunched file.)

Type of file:   (enter DOC if it is a document; and DOCUP if it
                is mostly in upper case.)
```

Related commands

UNCRUNCH

Purpose

To remove material from the workfile.

Form

```
DELETE[Q] {rangelist    } [,HOLD  ]
                          [,NOTEXT]
```

Description

The DELETE command can be used to remove lines or portions of lines from the workfile, or to remove lines from the hold file. The material to be deleted can be identified by line numbers, or line numbers qualified by start/end search strings or simply by a start and end search string. Material to be deleted can also be identified by column position; in the form start line(column position)/end line(column position).

When the command DELETE ALL follows a KEEP command with the CODED option the question "Write over old Workfile?" will be displayed. If you answer "YES" TDP will write over your old workfile to ensure that your data can not be read by anyone else. Since every record of the workfile is being written there can be a considerable delay. If you answer "NO" TDP will dispose of your workfile with the equivalent of an MPE PURGE command. (When MPE does a file purge it doesn't overwrite the file but only deletes the reference to it. Hence it may be possible to recover uncoded data from the workfile in this case.)

Limitations

None.

Options

`[Q] or [NOTEXT]` The Q or NOTEXT option suppresses the listing of the lines that are deleted. The number of lines deleted is displayed.

`[HOLD]` The HOLD option allows you to clear the hold file. Any range or start/end string are ignored. The hold file is not listed as it is cleared.

Examples

Example 1: `/DELETE ALL`

This example clears the workfile by deleting all lines that are currently entered. The lines are not listed when this command is used (even though Q is not used).

Example 2: `/DELETE HOLD`

This command clears the hold file. The entire hold file is deleted; it is not possible to delete only a part of the hold file.

Example 3: `/DELETEQ "Chapter 23"/"End."`

This example searches from the current line pointer until "Chapter 23" is found. The deletion begins with that string, and continues until the word "End." is found.

Example 4: `/DQ */20,31/34,57`

This example deletes from the current line pointer through line 20; lines 31 through 34, and line 57.

Example 5: `/DELETE [23.4] "The"/[31.2]"."`

This example deletes starting with the word "The" found in or after line 23.4, and continuing until a period is found in or after line 31.2.

Related commands

CHANGE
HOLD
MODIFY

DELIM

Purpose

To change the character located halfway between column stops.

Form

```
DELIM[Q] "char" [rangelist] [,NOTEXT]
```

Description

The DELIM command replaces the character halfway between each column stop. Thus, if the column stops are set at their default values of 10, 20, 30, 40, 50, 60, and 70, then the characters in columns 15, 25, 35, 45, 55, and 65 would be changed to the specified character.

This command can be used to "draw" vertical lines between columns of tabular material, or to insert blank characters midway between columns.

Limitations

Note that if any of the material in the columns extends to the halfway point, the column will still be replaced.

Options

`[Q] or [NOTEXT]` The Q or NOTEXT option suppresses the listing of modified lines.

`[rangelist]` The rangelist option restricts the scope of DELIM to the lines specified in rangelist.

Examples

Example 1: `/DELIM "|" 4/6`

This example draws vertical lines in lines 4 through 6 halfway between the currently set column stops.

Example 2: `/DELIM " " 4/6`

This example replaces the characters halfway between the column stops with a blank character.

Related commands

ALIGN
RACK
SET
TOTAL
VERIFY

DISPLAYPARMS

Purpose

To display site-specific and/or group-specific default settings for TDP parameters.

Form

```
DISPLAYPARMS
```

Description

TDP site-specific parameters are held in the file TDPPARMS in the TDPDATA.HPOFFICE group; group specific parameters are held in the file PARMSET within your group and account.

DISPLAYPARMS causes the site specific parameters in TDPPARMS to be displayed. In addition, if group specific parameters have previously been established, then the contents of PARMSET will be displayed for your group. (Refer to the STOREPARMS command for further information.)

Options

None.

Related commands

SET
STOREPARMS
VERIFY

DOWNSHIFT

DOWN

Purpose

To change alphabetic characters to lower case in specified lines in the workfile.

Form

```
DOWNSHIFT[Q] (rangelist) [,ADJUST]
                         [,NOTEXT]
```

Description

The DOWNSHIFT command changes any upper case alphabetic characters in the specified range to lower case. The range can be defined in terms of line numbers or from the occurrence of one search string to the occurrence of a second search string. The command will only process those characters that occur between the left and right margin settings.

Limitations

None.

Options

`[Q] or [NOTEXT]` The Q or NOTEXT option suppresses the listing of the downshifted lines.

`[ADJUST]` The ADJUST option causes the first letter of each word to be left in upper case.

Examples

Example 1: `/DOWNSHIFT 45/70, ADJUST`

In this example, upper case alphabetic characters in lines 45/70 are changed to their lower case equivalent. Because ADJUST is included, the first character in each word is not downshifted.

Example 2: `/DOWNSHIFT "The"/"."`

This example downshifts all alphabetic characters starting with the line containing the first occurrence of "The" after the current line pointer, and continuing until the end of that sentence (found by the ".").

Related commands

CHANGE
MODIFY
UPSHIFT

DRAFT

Purpose

To produce a draft copy of a document.

Form

```
DRAFT[Q]  [FROM filename ]
          [TO filename   ]
          [ASK           ]
          [COPIES=number ]
          [AUTO          ]
          [RO            ]
          [DEFER         ]
          [SP            ]
```

Description

The DRAFT command is used to print one or more formatted copies of a file on the line printer or the user's terminal (or, occasionally, to a disc file where it will be stored for later printing). The result is identical to a final copy except that the line numbers are printed to the right of each line and proof marks are printed if the file is proof marked.

Limitations

Line numbers are not exact since lines are filled during formatting. Line numbers are not printed in the case of multi-column output.

Options

[Q] The Q option suppresses the end of formatting message if there have been no errors during formatting. NOTEXT is not a valid option with this command.

[FROM filename] The FROM filename is the file you wish to print. If none is specified, the workfile is printed. If no FROM filename is given and the workfile is empty, you will be prompted for the filename.

[TO filename] The TO file defines the destination of the formatted output, normally a printer. If the output file is your current terminal, it can be pre-defined through the SET TERM command, in which case a TO filename is not needed. An asterisk (*) preceding the output filename informs TDP that the output is to be formatted for a specific output device such as an HP 2645 terminal or HP 2601 printer. (The absence of an asterisk results in the output being formatted for a standard CRT and placed in a disc file of record size 159 Bytes. The record size can be altered with a file equation, see **Formatted Output**, Appendix C.)

[ASK] The ASK option allows you to set certain last minute formatting commands such as PAUSE and DBL. In addition, four commands are used at this point only for document format control:

1 TRY produces the requested output, then returns and prompts you for more formatting commands.

2 GO produces the requested output, then returns to the TDP editor.

3 PAGE provides for the printing of only selected pages.

4 EXIT causes an immediate return to the TDP editor.

[COPIES] The COPIES option may be used to get multiple copies of the formatted document on the line printer (up to a maximum of 99).

[AUTO] The AUTO option overrides the PAUSE command and suppresses the "SET TOP OF PAGE" message. The user is notified of an "AUTOMATIC TOP OF PAGE" followed by a formfeed and formatting begins immediately.

[RO] The RO option is used when output is to be sent to a separate printer/terminal. The "SET TOP OF PAGE" message is displayed on your terminal rather than the output device and formatting begins when you press RETURN.

[DEFER] The DEFER option is used with the TDP spooler. It makes the initial status of the spoolfile created by the FINAL command "STOPPED". Thus the output will not be printed until the user issues an ALTERSPOOL command to change the spoolfile status to "READY".

[SP] The SP option is used with the TDP spooler. It allows the paper to be changed for one output file and changed back at the end of the file. When the spooler comes to print an SP output file it will print the normal spooler banner, the message "** SPECIAL PRINTING **" and will then wait until a blank is entered from the keyboard (CONTINUE key on the HP 2601) before starting to print the file. When printing of the file is completed printing halts again to allow the paper to be changed back and set to top of page. To resume printing the space bar should be depressed (CONTINUE key on the HP 2601).

Examples

Example 1: `/DRAFT FROM PROP TO *LP COPIES=2`

This example prints two draft copies of the file PROP on the line printer.

Example 2: `/DRAFT FROM CHAP1 TO *HP2688, ASK`

This example prints a copy of CHAP1 on the HP 2688 laser printer. Before printing, you are prompted with a backslash for any formatting commands you wish to enter at this time. Type GO when you are ready to begin the output.

Related commands

EXIT
FINAL
GO
MARGIN
PAGE
PAUSE
TRY

END
E

Purpose

To exit from TDP.

Form

```
END
```

Description

This command terminates the use of TDP and returns you to MPE control. The TDP command EXIT is identical to END.

If the workfile does not contain any material that has been modified since the last KEEP or TEXT command (ie. it is clean), the END command is executed immediately. If, when the END command is issued, the workfile contains material that has been modified and not kept since the last TEXT command (ie. not clean), a warning message is displayed:

```
WARNING: Workfile has not been saved.
Delete workfile and exit?
```

If you respond "yes", any changes to the workfile since the last KEEP or TEXT command are not saved; the workfile is deleted and the TDP session ends. If you respond no, the END command is not executed; the workfile remains unchanged and further editing may continue.

Limitations

None.

Options

None.

Related commands

EXIT

ENVELOPE

ENV

Purpose

To print an address on a single envelope, with a printer terminal.

Form

```
ENVELOPE [filename]
```

Description

The ENVELOPE command is included in TDP for the convenience of users whose business letters have a standard form. It assumes that the first group of lines in the file is the address.

The command searches the beginning of the file until it finds a line that begins with an alphabetic character. TDP prints the message "SET ADDRESS LINE" and waits for you to insert the envelope into the printer terminal and type in a space to indicate that you have done so. It prints the address, spacing over 40 columns before beginning, and then pauses to allow you to remove the envelope. Type a space again to return to TDP command mode.

It is assumed that the address ends when a blank line or a line starting with a backslash (\) is found.

To print a whole series of envelopes, it is most efficient to use the MAILER facility (see Chapter 6).

Limitations

None.

Options

`[filename]` The envelope can be printed from any file if the filename option is used. If no filename is given, the envelope is printed from the work file.

Example

```
/ENVELOPE REPLY
```

An envelope is printed with the address from the file named REPLY.

Related commands

DRAFT
FINAL
MAILER

EQN

Purpose

To check the formatting of a mathematical expression.

Form

```
EQN [linelist] [,HOLD]
```

Description

The EQN command formats and displays a mathematical expression. If no linelist is given, processing begins on the line referenced by the current line pointer and continues until a blank line or a line with a backslash (\) in column 1 is encountered. The workfile is not altered. The embedded \EQ command (see Section 4) is necessary to format the expression for final output. The EQN Editor command is only used to verify that the expression will be printed in the desired format.

EQN will format expressions containing superscripts, subscripts, and fractions. Terms may be nested; a superscript may be a fraction; compound fractions may be used.

Brackets are used to delimit terms. The expression is "coded" as follows:

SUB or <	denotes subscript	These are recognized as special symbols only if immediately followed by a bracket (ie. no intervening blanks).
SUP or >	denotes superscript	
OVER or /	denotes a fraction	

If "OVER" is used, both the numerator and denominator must be enclosed in brackets. If "/" is used and brackets are used in the numerator, then the denominator must be enclosed in brackets.

Note that the mathematical symbols +, -, / and * are not acted upon as mathematical symbols but merely symbols (ie. TDP does not recognize the order of precedence of mathematical operators), thus A/B+C and A/B + C will be formatted differently. In general, terms are delimited by spaces, or brackets as described above for superscripts, subscripts, and division. Refer to Example 2 for clarification.

A bracket will appear in the output only if it is immediately preceded by a quote. Thus, to obtain [a] in the output, you will need to write it as "[a"].

The superscripts and subscripts will be formatted with half-line feeds only on those printers which support half-line feed, such as the HP 2680 or HP 2601. Otherwise, a full line feed will be generated.

Limitations

Processing of an equation begins on a line identified in a linelist or by the current pointer and continues until either a blank line or a line with a backslash (\) in column 1 is encountered.

No more than 255 characters or 63 terms can appear in an expression.

The expression can extend for more than one line. However, the output itself must not extend past one line.

If "SUB", "SUP" or "OVER" are used, they must be in either all upper case or all lower case letters, but not mixed.

Options

[linelist] The linelist option specifies the lines to be processed.

[HOLD] The HOLD option places the formatted expression in the hold file rather than displaying it on the terminal.

Examples

Example 1: `/EQN 4`

If line 4 contains the following expression:

```
   4  Z = X>[2] + Y>[2]
```

then the output will be:

```
     2   2
Z = X + Y
```

The same result would be obtained if line 4 contained:

```
   4  Z = XSUP[2] + YSUP[2]
```

Example 2: `/EQN 5`

If lines 5 and 6 contained the following:

```
   5    esub[v2] = 1/4 - rsub[msub[i]]xsub[i]sup[2] +
   6     1/[4] (2"/rsub[dsub[i]] + rsub[s])
```

The output would be as follows:

```
      1       2   1
e   = - - r  x  + · (2/r   + r )
 v2   4    m  i   4     d     s
            i            i
```

Related commands

\EQUATION (Chapter 4)

EXIT

E

Purpose

To exit from TDP.

Form

```
EXIT
```

Description

This command terminates the use of TDP and returns you to MPE control. The TDP command END is identical to EXIT.

If the workfile does not contain any material that has been modified since the last KEEP or TEXT command (ie. it is clean), the EXIT command is executed immediately. If, when the EXIT command is issued, the workfile contains material that has been modified and not kept since the last TEXT command (ie. not clean), a warning message is displayed:

```
WARNING: Workfile has not been saved.
Delete workfile and exit?
```

If you respond "yes", any changes to the workfile since the last KEEP or TEXT command are not saved; the workfile is deleted and the TDP session ends. If you respond no, the EXIT command is not executed; the workfile remains unchanged and further editing may be continued.

Limitations

None.

Options

None.

Related commands

END

FILL

Purpose

To combine two or more lines in such a way as to maximize the text between the LEFT and RIGHT margins.

Form

```
FILL[Q] rangelist [,RESEQ]
```

Description

The FILL command joins lines within a specified range in such a way that the area between the LEFT and RIGHT margins is "filled" with text. Multiple blanks are reduced to single blanks with the following exceptions.

Where " . ", " ! " or " ? " are followed by two or more blanks or appear at the end of a line, then two blanks will be left in the output.

Limitations

Lines beginning with the a "\" in column 1 will be treated as formatting command lines and will be ignored by the FILL command.

Leading blanks in the line following a formatter command or a blank line are left untouched. (Preserves paragraph indentation.)

Options

[Q] The Q option suppresses the listing of filled lines.

[RESEQ] The RESEQ option controls the line numbering of the filled text. Where RESEQ is not specified new lines are introduced by the FILL command in the same way as if the ADD command had been used. However, to get the maximum number of lines in a given range it would be necessary to select the smallest possible increment and renumber the whole FILL range. This is done by specifying the RESEQ option.

Example

```
/FILL ALL
```

Suppose your workfile looks like this:

```
1                      Especially designed to show the fea-
2     tures of the fill command, this file has hyphenation and
3     some excessive punctuation!  Why?  To    show     what
4          the     FILL      command     does.
5     \FORMATTER COMMAND
6               The "\" of the formatter command signals a new
7     paragraph as would a blank line.
8
9               This is the last paragraph.
```

With LEFT=10 and RIGHT=40 the result would be:

```
1                                Especially
1.1             designed to show the features
2               of the fill command, this file
2.1             has hyphenation and some
3               excessive punctuation!  Why?
3.1             To show what the FILL command
5               does.
6     \FORMATTER COMMAND
6.1                       The "\" of the
6.2             formatter command signals a new
7               paragraph as would a blank
8               line.
9
10                         This is the last
11              paragraph.
```

Related commands

GLUE
SQUEEZE

FINAL

Purpose

To produce a final copy of a document.

Form

```
FINAL[Q]  [FROM filename ]
          [TO filename   ]
          [ASK           ]
          [COPIES=number ]
          [AUTO          ]
          [RO            ]
          [DEFER         ]
          [SP            ]
```

Description

The FINAL command is used to produce formatted output. The output can be produced on a printer such as a laser printer, daisy wheel printer or line printer, it can be directed to a disc file or displayed on the user's terminal. Linenumbers are not reproduced on output produced by the FINAL command.

Limitations

None.

Options

[Q] The Q option suppresses the end of formatting message if there have been no errors during formatting. NOTEXT is not a valid option with this command.

[FROM filename] The FROM filename is the file you wish to print. If none is specified, the workfile is printed. If no FROM filename is given and the workfile is empty, you will be prompted for the filename.

[TO filename] The TO file defines the destination of the formatted output, normally a printer. If the output file is your current terminal, it can be pre-defined through the SET TERM command, in which case a TO filename is not needed. An asterisk (*) preceding the output file named informs TDP that the output is to be formatted for a specific output device such as an HP 2645 terminal or HP 2601A printer. (The absence of an asterisk results in the output being formatted for a standard CRT and placed in a disc file of record size 159 Bytes. The record size can be altered with a file equation, see **Formatted output**, Appendix F.)

[ASK] The ASK option allows you to set certain last minute formatting commands such as PAUSE and DBL. In addition, four commands are used at this point only for document format control:

1 TRY produces the requested output, then returns and prompts you for more formatting commands.

2 GO produces the required output, then returns to the TDP editor.

3 PAGE provides for the printing of only selected pages.

4 EXIT causes an immediate return to the TDP editor.

[COPIES] The COPIES option may be used to get multiple copies of the formatted document on the printer (up to a maximum of 99).

[AUTO] The AUTO option overrides the PAUSE command and suppresses the "SET TOP OF PAGE" message. The user is notified of an "AUTOMATIC TOP OF PAGE" followed by a formfeed and formatting begins immediately.

[RO] The RO option is used when output is to be sent to a separate printer terminal. The "SET TOP OF PAGE" message is displayed on your terminal rather than the output terminal and formatting begins when you press RETURN.

[DEFER] The DEFER option is used with the TDP spooler. It makes the initial status of the spoolfile created by the FINAL command "STOPPED". Thus the output will not be printed until the user issues an ALTERSPOOL command to change the spoolfile status to "READY".

[SP] The SP option is used with the TDP spooler. It allows the paper to be changed for one output file and changed back at the end of the file. When the spooler comes to print an SP output file it will print the normal spooler banner, the message "** SPECIAL PRINTING **" and will then wait until a blank is entered from the keyboard (CONTINUE key on the HP 2601) before starting to print the file. When printing of the file is completed printing halts again to allow the paper to be changed back and set to top of page. To resume printing the space bar should be depressed (CONTINUE key on the HP 2601).

Examples

Example 1: `/FINAL FROM PROP TO *LP COPIES=2`

This example prints two final copies of the file PROP on the line printer.

Example 2: `/FINAL FROM CHAP1 TO *HP2688, ASK`

This example prints a copy of CHAP1 on the HP 2688 laser printer. Before printing, you are prompted with a backslash for any formatting commands you wish to enter at this time. Type GO when you are ready to begin the output.

Related commands

DRAFT
EXIT (Chapter 4)
GO (Chapter 4)
MARGIN (Chapter 4)
PAGE (Chapter 4)
PAUSE (Chapter 4)
TRY (Chapter 4)

FIND

F

Purpose

To locate a specified string in the workfile.

Form

```
FIND[Q] {string      } [IN rangelist] [,HOLD[Q]]
        {(wordlist)}                  [,LIT    ]
                                      [,NONLIT ]
                                      [,NOTEXT ]
                                      [,SL     ]
                                      [,UNN    ]
```

Description

The FIND command is used to locate a specified string in one or more lines of the workfile. The specified range is searched for the string. If a range is not given, the entire file is searched.

The lines containing the string are listed along with the line number, and the number of times the string was found is noted at the bottom of the list.

Note that if a wordlist is used, it must be enclosed in parentheses, and the words separated by spaces. A wordlist cannot be used with the LIT option or with the LIT parameter set.

Limitations

None.

Options

[Q]/[NOTEXT]/[UNN] The Q or NOTEXT option suppresses the listing of the full line; with this option, only the line numbers of the lines in which the search string occurs are printed. If both the NOTEXT and UNN options are selected, only the count of the number of times the string is found will be printed. If the UNN option alone is used, lines are listed without line numbers.

[LIT]/[NONLIT] The LIT option causes the string to be found wherever it occurs ... even if it is in the middle of another word. If NONLIT is specified, the string will be "found" only if it is a discrete word, i.e., preceded and followed by a blank. The default setting is normally NONLIT; this can be changed with the SET command.

[HOLD] If the HOLD option is chosen, the lines that are "found" are added to the end of the hold file. (If the string occurs more than once in a line, the line is added only once.) Line numbers do not appear in the hold file.

[HOLDQ] The HOLDQ option produces a list of the line numbers of "found" lines in the hold file.

[SL] The SL option will restrict the search to the current settings of the LOWC and HIGHC parameters. See the SET command for a description of these parameters.

FIND
F

Examples

Example 1: `/FIND "procedure"`

All lines containing the word "procedure" or "Procedure" will be listed to the terminal.

Example 2: `/FINDQ (this that), 25/101`

Lines 25 through 101 are searched for the words "this" and "that". When either word is found, only the line number is printed (the Q option suppresses listing of the line itself).

Example 3: `/FIND %blank IN ALL`

This command finds all the blank lines in the file, and lists each of the line numbers.

Related commands

CHANGE
FINDNEXT
FINDNUMBER
SET

FINDNEXT

FN

Purpose

To find the next occurrence of a string, or to set the current line pointer to a certain line.

Form

```
FINDNEXT[Q]   {string        } [,LIT   ]
FINDNUMBER[Q] {line[(column)]} [,NONLIT]
                               [,NOTEXT]
                               [,SL    ]
```

Description

The FINDNEXT command searches from the current location until it finds the specified string or reaches the end of the file. The alternate form, FINDNUMBER, sets the current line pointer to the specified line number. FINDNUMBER may also be used to find column positions within a line. The same abbreviation is used for both commands. TDP determines which command it is by the parameter used. When the linenumber or string is found, the line is printed and you return to command mode. If the line number does not exist, the following line will be found and printed.

The command is usually used to set the current location pointer.

Limitations

FINDNEXT will only find the first occurrence of the specified string. For multiple finds, use the FIND command.

Options

[Q]/[NOTEXT] The Q or NOTEXT option suppresses the listing of the line containing the string, or the specified line number.

[LIT]/[NONLIT] The LIT option causes the string to be found wherever it occurs - even if it is inside another word. Thus, "is" would be found where it occurs in the word "history," and so forth. The default setting is NONLIT; that can be changed with the SET command. If LIT has been made the default, than using the NONLIT option means that the string "is" would only be found when it is a separate word.

[SL] The SL option restricts the columns to be searched to the values represented by the HIGHC and LOWC parameters. See the SET command for an explanation of these parameters.

FINDNEXT
FN

Examples

Example 1: `/FINDNEXT "The"`

This command finds the next occurrence of the string "The" and prints the line in which it occurs.

Example 2: `/FINDNUMBERQ 17`

This command finds line 17 and positions the line pointer at that line. Line 17 is not listed. If there is no line 17 in the file, the line pointer is set to the nearest preceding line.

Example 3: `/FN 23(16),NOTEXT`

This command is correctly interpreted as FINDNUMBER because it is followed by a line number. It locates the sixteenth column of line 23, and positions the line pointer there. The line is not printed.

Related commands

FIND
SET

FLUSH

Purpose

To purge a group of files. NOTE: System Manager capability is required to purge across accounts.

Form

```
FLUSH filelist
```

Description

The FLUSH command is an extension of the MPE PURGE command. It will purge all files in the filelist. The filelist can be specified by a list of files, a range of files (filename/filename), or a set of files identified by one of the "wild card" characters (@,#,?).

A group name can be used if you have write access to that group. A group and account name can be used if you have System Manager capability and have write access to the file.

For each file found in the specified fileset or filelist you are asked: PURGE filename? A response of "YES" causes the file to be purge; a response of "NO" leaves the file on disc. If the file can not be purged a message is displayed.

The SET PERMYES command may be used to cause all files in the list to be purged without user interaction.

Limitations

The command will only purge files to which you have write access. You must have System Manager capability to purge files which are not in your log-on account.

Options

None.

Examples

Example 1: `/FLUSH ADAM/CLEAR`

This command purges the range of files beginning with ADAM and continuing, in alphabetical order, until CLEAR is reached.

Example 2: `/FLUSH @COMP@.USER1`

This command purges any file with the letters COMP anywhere in the name (COMPLETE, RECOMPUT, etc.) in the group USER1 of the log-on account.

Example 3: `/FLUSH M#`

This command purges any file whose name has 2 characters, beginning with M and followed by an any integer (M2, M6, etc.).

Related commands

CATALOG

GETPARMS

Purpose

To reset altered set parameters to site-specific or group-specific values.

Form

```
GETPARMS
```

Description

TDP site specific parameters are held in the file TDPPARMS in the TDPDATA.HPOFFICE group; group specific parameters are held in the file PARMSET within your group and account.

If site specific or group specific parameters have been established, the SET parameters are assigned values from these files, when TDP is started. If, during the course of a session, you alter the values of the SET parameters, you can re-initialize them to the values in the TDPPARMS and PARMSET files by issuing a GETPARMS command. (See the STOREPARMS command for further information.)

Limitations

In order to prevent text from being inadvertently lost, this command can only be issued if the workfile is empty.

Options

None.

Related commands

DISPLAYPARMS
SET
STOREPARMS
VERIFY

GLUE
GL

Purpose

To combine two lines into one line.

Form

```
GLUE[Q] rangelist [,NOTEXT]
```

Description

The GLUE command joins each pair of lines in a specified range. Note that trailing blanks on the first line and leading blanks on the second are suppressed. A single blank is left between the contents of the lines.

Limitations

If the resulting line would be too long, the lines are not joined and an error message is displayed, and no further gluing is performed.

Options

`[Q]/[NOTEXT]` The Q and NOTEXT option suppress listing of the lines that have been glued.

Examples

Example 1: `/GLUE 2`

Suppose lines 2 and 3 appear as follows:

```
2  This line had material deleted
3  and so did this one.
```

The command listed for this example adds line 3 to the end of line 2, and deletes line 3. The result is:

```
2  This line had material deleted and so did this one.
```

Example 2: `/GLUE 10/15`

In the case that lines 10 through 15 appear as follows:

```
10  This is an example
11  in which a range
12  of lines can be
13  glued together.  Note
14  that the lines
15  are operated upon in pairs.
```

The result would be as follows for the command /GLUE 10/15:

```
10  This is an example in which a range
12  of lines can be glued together.  Note
14  that the lines are operated upon in pairs.
```

Related commands

FILL

HELP

Purpose

To help the user in selecting and using TDP commands.

Form

```
HELP [command [SYNTAX/ALL]]
     [MENU]
     [FURTHER]
     [topic [keyword/ALL]]
```

Description

The HELP command invokes the help system which provides a brief description of the commands available in TDP. It also offers information on options, parameters and basic concepts.

Limitations

The information given by the help system is not intended to be complete for any given command. Refer to this manual for a more detailed description of any particular command.

Options

`[command[SYNTAX/ALL]]` In the form HELP command, command is the name of a TDP command. TDP displays a description of the command named. The form HELP command SYNTAX causes just the command syntax to be displayed. The form HELP command ALL displays both the description of the command and its syntax.

`[MENU]` The MENU option displays an introductory menu for the help system. The page displays a list of general topics available in the help system. To access one of the topics use the form HELP topic to display the relevant page.

`[FURTHER]` The FURTHER option provides an extension of the general topics available in the help system. These topics are accessed in the same way as those displayed by the MENU option, via the form HELP topic.

`[topic[keyword/all]]` As described above this form of the command is used to access one of the general topic pages listed via the MENU or FURTHER options. Some of the displayed topic pages list a number keywords that provide access to further information by typing HELP topic keyword. For example, the general topic PRINTERS lists the keywords SPOOLING and LASER. To access the information on spooling type
HELP PRINTERS SPOOLING. The alternative form HELP topic ALL will display all of the pages associated with keywords (if any) listed under that topic. For example, HELP PRINTERS ALL will display the general topic page on printers, then the page associated with spooling and finally the page associated with laser printers.

Related commands

PROMPT

HOLD
H

Purpose

To place material in the hold file.

Form

```
HOLD[Q] [rangelist [,APPEND] [,NOTEXT] [,NUM]]
        [filename  [rangelist]                ]
```

Description

The hold file is a storage area separate from the workfile. Lines of text may be taken from the workfile and put into the hold file. Later, these lines of text may be added to the workfile. This can be used for moving text from one file to another.

Material is added to the hold file using the HOLD command (or the HOLD option of the FIND command). The range of lines specified is placed in the hold file. The hold file is unnumbered; new line numbers are assigned when the lines are added back into the workfile.

If the hold file currently contains material, TDP will ask if you want to clear the hold file before adding the new material. If the answer is no, it will ask if you want to append the new material to the old. If the answer is no again, it will do nothing.

Limitations

If text has been placed in the HOLD file, that text will be deleted when a second HOLD command is given without an APPEND option. Before deleting the text, however, the user is first asked if the HOLD file is to be cleared. A "YES" response must be given in order to clear the HOLD file and add the new text.

The HOLD command operates only within the column boundaries established by the SET LEFT and SET RIGHT commands.

The HOLD file can contain a maximum of 2000 lines of text.

Options

[rangelist] The rangelist option determines which lines are held. If no rangelist is given, only the current line is placed in the hold file.

[APPEND] The APPEND option is used to add new material at the end of the existing hold file.

[Q] or [NOTEXT] The Q or NOTEXT option suppresses the listing of the lines being placed in the hold file.

[NUM] The NUM option causes the line numbers to be stored in the hold file with the text. The linenumbers are stored as text and cannot be used to renumber the workfile.

HOLD

H

[filename] The filename option can only be used if you are using, and have previously identified with a SET command, an HP 2642 terminal. In this case, the HOLD command will cause the rangelist (if given) to be written to a file with the name "filename" on the HP 2642 mini disc. The filename is limited to 10 alphanumeric characters and must begin with an alphabetic character.

Examples

Example 1: `/HOLD 20/30`

Lines 20 through 30 will be placed in the hold file. The lines being held will be listed.

Example 2: `/HOLDQ 34/67,APPEND`

This command places lines 34 through 67 into the hold file, at the end of the material currently in the file. The lines are not listed.

Related commands

ADD
DELETE
FIND
INSERT
KEEP
LIST
MOVE
REPLACE

HYPHEN

HYPH

Purpose

To test the hyphenation routine.

Form

```
HYPHEN
```

Description

This command allows you to test the system's hyphenation of certain words or, alternatively, to use the system's hyphenation dictionary to find the correct hyphenation for a word.

After entering the command, the system asks you for the word, then prints it in its hyphenated form.

You exit from the command with a carriage return or CTRL-Y.

Limitations

None.

Options

None.

Examples

```
/Hyphen
Word:  calculation
cal-cula-tion
Word:  (carriage return)
/
```

This example retrieves the system's hyphenation for the word "calculation". Note that there is actually another acceptable hyphen point in the word (cal-cu-la-tion), but TDP is designed to note fewer points, rather than take the chance of indicating too many.

Related commands

\HYPHALLCAPS (Chapter 4)
\HYPHCHAR (Chapter 4)
\HYPHDBL (Chapter 4)
\HYPHEN (Chapter 4)
\HYPHFIRSTCAP (Chapter 4)
\HYPHLAST (CHapter 4)

INSERT

I

Purpose

To insert material into the workfile.

Form

```
INSERT[Q] {line[(column)]} [,HOLD[Q]]
          {string         }
```

Description

The INSERT command is used to insert material in the middle of a line of text. It searches for the string or line and column numbers, then begins the insertion before that position. The portion of the line starting with the string or position is stored away for later use. You are then prompted for the insertion just as if you were using the ADD command. When you signal that the insertion is complete by typing a double slash (//), the partial line that was stored away is added as the last step. If the insertion is to start at the beginning of the line, it is handled as if it were an ADD starting after the preceding line.

Limitations

None.

Options

[Q] The Q option suppresses the listing of the line in which the insertion is to be made.

[HOLD] The HOLD option inserts the lines currently in the hold file at the specified line. HOLDQ suppresses the listing of the lines being inserted from the hold file.

Examples

```
/INSERT [2]"The"
```

Suppose your workfile looks like this:

```
1  This file explains the training course
2  available for TDP users.  The course,
3  designed to thoroughly acquaint the novice
4  with text processing and the TDP
5  system, lasts for 2 days and is held at
6  the regional training centers.
```

In this example, the line pointer is repositioned at the beginning of line 2, and material is added starting before the word "The":

```
2.1  We recommend that at least one person
2.2  from your organization attend the formal
2.3  training course.
2.4  //
2.4  The course,
```

(Note: TDP has saved "The course," to add here.)

The result would be:

```
 1   This file explains the training course
 2   available for TDP users.
2.1  We recommend that at least one person
2.2  from your organization attend the formal
2.3  training course.
2.4  The course,
 3   designed to thoroughly acquaint the novice.
```

The same thing could be done with the command: /INSERT 2(31)

Related commands

ADD
ADDSINGLE
CHANGE
COLINSERT
COLMOVE
COLREPLACE
COPY
HOLD
MERGE
JOIN
MOVE

JOIN
J

Purpose

To add all or part of a disc file to a specified location in the workfile.

Form

```
JOIN[Q] filename [filerange] [TO linenumber] [BY increment ]
```

Description

The JOIN command is used to add material to one point in the work file from any other disc file. The material to be added can be identified by line number or record number. The lines being added are renumbered.

If the JOIN file has a different record length from the work file, or if the file appears to be unnumbered, TDP will ask if the JOIN file is unnumbered. If the reply is "YES", the entire record will be considered data and will be added to the workfile. If the reply is "NO", TDP will assume that the file is numbered.

If the records of the JOIN file are larger than those of the workfile, TDP may change the settings of the LENGTH and RIGHT parameters to accommodate the larger records. A warning is printed when this occurs.

Limitations

This command will not replace or interleave lines (see MERGE or OVERLAY commands). Only files of like format should be joined together; DEFAULT to DEFAULT, DIARY to DIARY and COBOL to COBOL.

If you are going to join a large file to your work file, you will need to SET SIZE to accommodate the added lines.

Options

`[Q]` The Q option suppresses the listing of the material being joined.

`[filerange]` The range option allows you to join just part of the file. You can specify which lines are to be joined either by listing starting and ending line numbers (if the file is numbered), or starting and ending record numbers. If no range is given, the entire file is joined. Optional parentheses can be used to enclose the line range. No comma can appear between the filename and the range.

`[TO linenumber]` The TO line number option determines the destination line number of the material being joined. If none is specified, the material is added at the end of the workfile.

`[BY increment]` The BY increment determines the interval between line numbers (i.e., 1, 10, .1, .01, etc.), overriding any value set by SET DELTA.

Examples

Example 1: `/JOIN EXAMPLE`

The entire file named EXAMPLE will be added to the end of the workfile. The lines will be renumbered starting one after the ending line, and incremented by one for each line (assuming SET DELTA = 1).

Example 2: `/JOIN EXAMPLE 1/10 TO 15 BY 0.1`

This command adds lines 1 through 10 of the file named EXAMPLE to the current workfile, beginning with line 15 of the current workfile (or the next available line number if 15 currently exists). The new lines will be incremented by 0.1.

Related commands

ADD
MERGE
OVERLAY

KEEP
K

Purpose

To save the workfile as a permanent disc file.

Form

```
KEEP [filename] [range] [,ASK  ]
                        [,CODED]
                        [,HOLD ]
                        [,NUM  ]
                        [,RESEQ]
                        [,UNN  ]
```

Description

The KEEP command copies the workfile to a permanent disc file. The entire workfile, or any portion of it, can be kept; the work file itself is not affected by the command.

If the length parameter has been reset, only the portion of the file fitting that parameter will be kept. Thus if the line length has been set to 60 characters, and the workfile has a line length of 72 characters, only 60 characters on each line will be saved.

The actual record length of the KEEP file is equal to LENGTH plus a six byte sequence number for COBOL format files and LENGTH plus an eight byte sequence number for DEFAULT and DIARY format files.

If the workfile was created without sequence numbers, and the KEEP command doesn't specify the NUM option, then the file is saved without numbers. Similarly, if the file was originally kept with numbers, subsequent KEEPs that don't specify UNN will include sequence numbers in the permanent file.

It is possible to automatically insert the date or time whenever a certain file is kept, or to cause a line to be modified when the file is kept. Simply enter the following in the first line in the file:

```
\*DATELINE   column range, linenumber
or
\*TIMELINE   column range, linenumber
or
\*MODIFY      linenumber
```

where line number and column range identify the location where you want the date or time inserted, or the modify performed. Note that DATELINE, TIMELINE, and MODIFY must be all capitals as shown.

Limitations

You cannot KEEP a file into a different account.

You cannot name a file NOW, TODAY, FIRST, LAST or ALL.

Options

[filename] The filename option allows you to assign a name to the file. The file name can also be followed by a lockword as in FILE/LOCKWORD. If no filename is given, the file is saved with the most recent name used in a KEEP, TEXT, MAKE, or NAME command. If you wish to KEEP only a portion of the file, you must specify a name. In that case, the default name for the entire workfile does not change. If the specified or default filename already exists on the disc, the system asks if you want to purge the old file. If you answer "yes", the old file is purged and the new one kept. If you answer "no", the file is not purged and the workfile is not kept. A message to that effect is displayed.

[range] To save only a portion of the file, specify the range of lines that you want kept.

[UNN]/[NUM] The UNN option saves the file without sequence numbers; the NUM option saves a previously unnumbered file with sequence numbers. Note that COBOL and DIARY files will always be kept with line numbers.

[RESEQ] If the option RESEQ is chosen, the file will be kept as a numbered file, and new sequence numbers will be assigned starting at 1 and incrementing by 1. (COBOL files are incremented by 0.1.) The workfile itself is not resequenced. This is the most efficient way to resequence a file. Note that the RESEQ option should not be used when KEEPing a DIARY file.

[HOLD] The HOLD option will keep the hold file as a permanent file. In this case, a filename must be provided. The hold file will be kept as numbered or unnumbered according to the current setting of the numbered flag. That setting can be overridden with the UNN or NUM options.

[CODED] The file can be encrypted for security purposes by using the CODED option. The system will request a "secret" word from you, and then ask you to repeat it for verification. It must be identical whenever typed; even upper and lower case characters must be duplicated exactly. The secret word can be any combination of letters and numbers up to 20 characters long. After the file has been coded, any effort to retrieve the file without using the secret word results in a listing of unintelligible output. To retrieve the original file, use the CODED option with the TEXT command. When a file has been kept with the CODED option TDP assumes that the workfile contained sensitive information. Thus when a command is given which would result in the deletion of the workfile e.g. DELETE ALL or TEXT AFILE, TDP asks the question "Write over old workfile?". This is because when MPE purges a file it only deletes the reference to the data. The data remains in readable form on the disc. Answering "YES" to the question above will result in the workfile being overwritten before MPE is allowed to purge it. **WARNING: if you forget the secret word, there is no way to retrieve it.**

[ASK] The ASK option causes TDP to ask you for a blocking factor and a record limit (maximum number of records to be allowed in in the file) for the file being created. This is useful when TDP is used to prepare files which will be processed by other programs.

Examples

Example 1: `/KEEP ROUGH,RESEQ`

This saves the workfile in a permanent file named ROUGH, with the lines renumbered starting with one and incrementing by one. Note that the KEEP command used with the RESEQ option is more efficient than a RESEQ command followed by a KEEP.

Example 2: `/KEEP PROG1, ASK`

Suppose that the workfile is a data file. This example saves that file with the name PROG1, after asking for the blocking factor and file limits. The ASK option is used since the file will be processed by another program.

Example 3: `/KEEP SALARY, CODED`

Suppose the workfile contains a report on salaries in your company. This example encodes the report, and stores it in its coded form, with a "secret" word associated with it. Without that secret word, no one can access the file, protecting its contents from unauthorized scrutiny.

Here is the dialogue:

```
Input Secret word:  supercode
Repeat secret word:  supercode
Purge old? y
```

The secret word, shown here, would not print at your terminal.

Related commands

CLOSE

LIST
L

Purpose

To list lines from the workfile on the terminal or system printer.

Form

```
LIST[Q] [rangelist] [,EXPAND ]
                    [,HOLD   ]
                    [,LONG   ]
                    [,NEW    ]
                    [,NOTEXT ]
                    [,OFFLINE]
                    [,PMARK  ]
                    [,UNN    ]
                    [,WINDOW ]
```

Description

The LIST command is used to list information from the workfile onto the terminal or system printer. The amount of the line that is listed is limited by the LEFT and RIGHT margin settings. You may list any portion of the file, or all of it.

The LIST command leaves the current line pointer at the end of the last line listed.

Limitations

None.

Options

[Q] or [NOTEXT] The Q or NOTEXT option suppresses the listing, and prints only the number of lines in the specified range. When used with EXPAND, output is shown only for lines which contain non-printing characters. When used with the UNN option no text is displayed but line length statistics are displayed. The right hand column shows the number of lines of the length indicated in the left hand column.

[rangelist] The rangelist option is used to specify the lines you want listed. If no rangelist is given, only the current line is listed.

[UNN] If the UNN option is chosen, the lines will be listed without line numbers.

[OFFLINE] To list the lines to the system printer, select the OFFLINE option. See Appendix F for more information.

[LONG] The LONG option lists all lines within the range that are a certain length or more (the length is set with the HIGHC parameter on the SET command).

[EXPAND] If the EXPAND option is included, each listed line appears as three lines of output: the first shows the line as it appears in the file, except that non-printing characters (including all 8-bit characters) are replaced by blanks;

the second and third lines show the hexadecimal codes for all the characters in the line.

[PMARK] When the PMARK option is used, each line in the range will be listed, with the proof mark listed in the right margin. Refer to the SET command for a description of the use of PMARKs.

[NEW] The NEW option is used in conjunction with the proof marking feature (see SET command). Only the lines with proof marks will be listed, with the proof marks shown next to each line.

[HOLD] The HOLD option is used to list the current contents of the hold file. The entire file will be listed.

[WINDOW] The WINDOW (WIN) option lists the next x lines of the file, where x is a value from 1 to 55 that is assigned with the SET command. The default value is 18.

Examples

Example 1: `/LIST ALL`

The simplest example, this command lists the entire workfile to the terminal.

Example 2: `/LIST 25/50, OFFLINE, UNN`

This example prints lines 25 through 50 on the system printer, without the accompanying line numbers.

Example 3: `/LIST 17, EXPAND`

This example lists line 17, and then two lines containing the hexadecimal code for each character:

```
This example lists line 17, and then two lines containing the
56672676676626677726666233226662766627762666672666766666662766
48930581D0C50C93430C9E5017C01E40485E047F0C9E5303FE419E9E70485
```

Example 4:

```
/LIST 10, WIN
/LIST %blank, WIN
/@12
```

This set of commands allows you to skim through the workfile, examining the first few lines of each paragraph (the number of lines is dependent on the current setting of WINDOW). The first command lists line 10 and the following window. The next command finds the next blank line and lists the window of lines following it. The third command is a special "repeat" command that repeats the previous command 12 times. The repeat command could be set for any number of repetitions.

Related commands

PRINT

MAKE

Purpose

To start a new workfile and initiate ADD mode.

Form

```
MAKE[Q] filename
```

Description

The MAKE command is a convenient way to start a new workfile. It checks to see if the workfile is clean (ie. not modified since the last TEXT or KEEP) and, if not, asks you to either keep or delete the current workfile. Once this has been done, it assigns the specified filename as the default KEEP filename so you can keep the file when desired without specifying a filename. Finally, it places you immediately into ADD mode. Note that the MAKE command does not add to the current workfile.

Limitations

None.

Options

[Q] The Q option suppresses printing of the line number as a prompt. NOTEXT is not a valid option with this command.

Example

```
/MAKE EXAMPLE
```

The system responds:

```
   Making EXAMPLE
```

and then prompts you for line 1, as long as the current workfile was clean.

Related commands

ADD
KEEP
NAME

MAILER

Purpose

To deal with mass mailing applications

Form

```
MAILER [filename]
```

Description

This is a general purpose mass mailing utility. See Chapter 6 for a full description of applications and use. MAILER command prompts at the terminal for specification of the mailing parameters.

Limitations

None.

Options

When the filename option is used the mailing parameter specifications are taken from a command file built during a previous execution of MAILER.

Related commands

ENVELOPE

MERGE

Purpose

To merge a disc file with an existing workfile.

Form

```
MERGE[Q] filename [filerange]
```

Description

The MERGE command merges a disc file with the workfile, interleaving line numbers as appropriate. If a line number exists in both files, the line in the workfile is retained. The disc file itself is not altered by this command.

Limitations

The workfile and the disc file must be the same format, eg. DEFAULT and DEFAULT.

Options

`[Q]` The Q option suppresses listing of the merged lines.

`[filerange]` The filerange option allows you to merge only a selected portion of the disc file with the workfile. If no filerange is given, the entire file is merged.

Examples

Example 1: `/MERGE FISCAL`

This command merges the file named FISCAL with the workfile.

Example 2: `/MERGE FISCAL 1/50`

This command merges lines 1 through 50 with the workfile.

Related commands

JOIN
OVERLAY

MODIFY
M

Purpose

To modify characters within a line of text.

Form

```
MODIFY[Q] [rangelist] [,EXPAND]
```

Description

The MODIFY command is used to replace, delete, or insert characters within a given line. The lines in the rangelist are displayed one at a time for modification. You enter a sub-command under the appropriate position in the line:

"r" followed by a replacement string is typed to replace existing characters in the line.

"d" is typed under each character to be deleted. To delete several characters, type "d" under the first and last; everything in between will be deleted. To delete to the end of the line, type "d" under the first character to be deleted and "e" immediately after it. The "d"'s may be immediately followed by an "i" and a string to be inserted.

"i" followed by an insertion string is typed to insert something into the line (or combined with "d" as explained above).

The modified line is listed again for your inspection and for further modification. When the line is correct as shown, press (RETURN) to enter the modifications into the workfile.

Two additional sub-commands are available:

"s" is typed to indicate where a line should be split into two lines:

"c" is typed to cancel all modifications that have been made to the current line. (CTRL)-Y also cancels the modifications just being made. (Cancel and (CTRL)-Y do not delete a line entered with "split".)

The alternate form of the command allows you to modify the line containing the first occurrence of the specified string. By using the bracketed line number option, you can start the search for the string at a certain point in the workfile (see **Strings** in Chapter 1).

If you have issued a command to modify a list of lines, you can terminate that command with a (CTRL)-Y.

MODIFY is also used in calculator mode to edit a stored expression, and to modify a line of commands. (See @M command.)

Limitations

You will not be allowed to make a replacement or insertion that causes the line to exceed the current line length.

Options

[Q] The Q option suppresses listing of the modified lines.

[rangelist] The rangelist option allows you to specify the lines that are to be modified. If the rangelist is not specified, the current line will be presented for modification.

[EXPAND] The EXPAND option prints each line to be modified as three lines: the first is the line as it appears in the workfile (but with non-printing characters including eight-bit characters replaced by blanks); the second and third contain the hexadecimal codes for all the characters in the line. This option facilitates modification of these non-printing characters.

Examples

Example 1: `/MODIFY 1`

This example presents line number 1 for modification:

```
1        This is the first line in my file.
 Changes:                             iwork
1        This is the first line in my workfile.
 Changes:        rline number 1
1        This is line number 1e in my workfile.
 Changes:                     d
1        This is line number 1 in my workfile.
 Changes:  (carriage return)
```

The first change inserted the word "work" after "my". Note that the blank between "i" and "work" resulted in proper spacing between the words. The second change entered a replacement string "line number 1" starting at the word "the". Note that the string we wanted to replace was not the same length as the replacement string, necessitating the third change. The third change deleted the "leftover" character from the replacement.

Example 2: `/MODIFY [50] "work"`

This example, using the bracketed line number option, begins searching for the string "work" at line 50, and displays the first line that contains the string for modification:

```
57       The LIST command lists the workfile.  More
 Changes:                                      d  d
57       The LIST command lists the workfile.
 Changes:  (carriage return)
```

The same result could be achieved as follows:

```
57       The LIST command lists the workfile.  More
 Changes:                                      de
57       The LIST command lists the workfile.
 Changes:  (carriage return)
```

Example 3: `/MODIFY 1`

```
1      This is the first line in my file.
 Changes:                     deil need to modify.
1      This is the first line I need to modify.
 Changes:  (carriage return)
```

Related commands

CHANGE
COLINSERT
COLMOVE
COLREPLACE
INSERT
REPLACE
SCREEN
WORDMOVE

MONITOR

Purpose

To provide automatic monitoring of a TDP session.

Form

```
MONITOR
```

Description

The MONITOR command provides automatic monitoring of editor commands and of such items as number of active lines and processor time used during a TDP session. After typing the command, you are asked for the monitor file name. Any valid file name may be used, but since the name is not checked to see if there is another file by that name, you must be careful. TDP will not be able to save the monitor file as a permanent file when you exit if a permanent file with that name already exists. Next you are asked for the monitor/verify string. Supply those parameters from the VERIFY COMMAND that you want monitored. They will be checked after each command is entered, and the result stored in the monitor file, along with all commands entered.

Limitations

If the file is not closed when exiting from TDP it is not saved as a permanent file. (See Example below.)

Options

None.

Example

```
/MONITOR
Monitor File Name:       MONITOR
Monitor/Verify String:   ACTIVE,TIME
```

This example sets MONITOR as the file to store the information gathered by the MONITOR file, and asks for information to be collected on the parameters ACTIVE and TIME.

To print the file, enter the command /MONITOR again and you will be asked if you want to "Print Monitor File?" If you answer "yes," the contents of the file will be listed on the terminal. Then you will be asked if you want to "Close Monitor file?" If you answer "yes", the file is closed as a permanent file and monitoring is stopped for this session. The same drill occurs when you EXIT from TDP if MONITOR has been invoked.

Related commands

SET
VERIFY

MOVE

Purpose

To move lines from one part of the workfile to another.

Form

```
MOVE[Q] rangelist TO (linenumber) [BY increment]
                     (string    ) [,NOTEXT]
```

Description

The MOVE command removes text from one location in the work file and places it in a new location. It works just like the COPY command, except that the text is deleted from the original location. The increment option allows you to control the manner in which the lines are numbered in their new location.

The rangelist parameter allows you to move a single line, a group of lines, or multiple groups of lines. If the destination line number already exists, the next available line number is used as the starting position for the new material.

If a string is specified to determine the destination of the material to be moved, the moved lines will be placed on the next available line after the line containing the search string.

Limitations

The TO word can be replaced by the comma only if a single range is being used. If a list of line ranges is given, the word TO must appear.

If lines will not fit in the new location, they are moved to the end of the file and you are told that that has been done.

The MOVE command operates only within the boundaries established by the SET LEFT and SET RIGHT commands.

Options

`[BY increment]` The BY increment option allows you to determine the increment used for the new lines. If the lines being moved will not fit in the space available at the destination line number with the specified increment, the system selects a smaller increment.

`[Q]` The Q option suppresses listing of the moved lines.

Examples

Example 1:

```
/MOVE 5,50
```

This example moves line 5 to line 50, and deletes the line number 5.

Example 2:

```
/MOVE 17/34, 42/49 TO 116 BY 0.01
```

This example moves two groups of lines (17 through 34, and 42 through 49) to line number 116 (or the next available line). The lines are deleted from their original locations, and renumbered with an increment of 0.01.

Example 3:

```
/FINDNUMBER 1
/MOVE "Because"/"increased" TO 19 by 0.01
```

A range can be specified by a starting string and an ending string. In such a case, only the text between and including the two strings will be moved. Suppose a section of your workfile looks like this:

```
11   Using TDP you can enter a document at the terminal,
12   store it in a disc file, type in only the necessary
13   corrections, and print the final copy of the
14   document.  Because the
15   document has never been retyped, both accuracy and
16   efficiency are increased.
17
18   TDP can be used to prepare many kinds of
19   documents.  The amount of typing time is minimized.
20
21   Some of the features that contribute to TDP's popularity
```

The commands shown for this example will reset the line pointer to line number 1 (so the file is searched from the beginning); and move the specified material so your workfile looks like this:

```
11    Using TDP, you can enter a document at the terminal,
12    store it in a disc file, type in only the necessary
13    corrections, and print the final copy of the
14    document.
17
18    TDP can be used to prepare many kinds of
19    documents.  The amount of typing time is minimized.
19.01 Because the
19.02 document has never been retyped, both accuracy and
19.03 efficiency are increased.
20
29    Some of the features contributing to TDP's popularity
```

Related commands

COPY

NAME

Purpose

To assign a default name to a new workfile.

Form

```
NAME filename
```

Description

The NAME command checks to make sure the workfile is clean, ie, has not been modified since the last TEXT or KEEP command. If not, you get an error message. If the file is clean, this command creates a new workfile and assigns a default KEEP filename so you can keep the file in the future without specifying a filename in the KEEP command.

NAME is the same as MAKE, except that add mode is not initiated. NAME is useful when the first material to be entered in a new workfile is entered with a JOIN command.

Limitations

None.

Options

None.

Example

```
/NAME MINE
```

This example assigns the name MINE as the default filename for the workfile. If you later KEEP the workfile without specifying a filename, the file will be kept with the name MINE.

Related commands

KEEP
MAKE

OVERLAY

Purpose

To overlay the workfile with another disc file.

Form

```
OVERLAY[Q] filename [filerange]
```

Description

The OVERLAY command is similar to the MERGE command, in that material is added to the workfile from another disc file. However, with OVERLAY if a line number exists in both files, the line in the workfile is replaced by the line in the overlay file.

Limitations

The workfile and overlay file must be the same format, eg. COBOL and COBOL. The overlay file must be a numbered file.

Options

`[Q]` The Q option suppresses the listing of the added material.

`[filerange]` The filerange option allows you to add a selected portion of the overlay file. The filerange can be specified by line numbers or by record numbers. If no range is specified, the entire file will be added.

Examples

Example 1: `/OVERLAY DRAFT`

This command adds the entire file DRAFT to the existing workfile.

Example 2: `/OVERLAY DRAFT 15/37`

This command adds lines 15 through 37 of DRAFT to the workfile.

Example 3: `/OVERLAY DRAFT #3/#10`

This command adds record numbers 3 through 10 of DRAFT to the workfile.

Related commands

JOIN
MERGE

PRINT

P

Purpose

To print the contents of a file other than the workfile.

Form

```
PRINT[Q] filename [range    ] [,EXPAND ]
                  [filerange] [,NUM    ]
                              [,OFFLINE]
                              [,RNUM   ]
                              [,UNN    ]
```

Description

The PRINT command can be used to print part or all of a permanent disc file. Options allow you to print the file on the system line printer, with or without line numbers or record numbers, and with hexadecimal codes for each character.

When the file is printed, a heading is printed indicating the name of the file, the current time and date, and the number of records and record size. No formatting is done.

Limitations

This command is not used with the workfile (see the LIST command).

Options

[range] The range option allows you to print a selected portion of the file, specified by line numbers, or by record number (noted by a # sign in front of the record number). If no range is specified, the entire file is printed. Note that only one range of lines can be printed with the PRINT command. If there are valid sequence numbers in the last 8 bytes of each record they will be displayed in the linenumber field of the PRINT display.

[OFFLINE] The OFFLINE option directs output to the system line printer. If this option is not selected, the output goes to the terminal by default. You can redirect output by using the file equation, `FILE LP; DEV=devicename`.More details are given in Appendix F.

[Q] The Q option suppresses printing of the heading (filename, time and date). and any error message if the file is not found.

[EXPAND] The EXPAND or EXP option prints three lines of output for each line in the file. The first is the line as it appears in the file; the second and third show the hexadecimal codes for each character, including non-printing characters (and all eight-bit characters).

[UNN]/[RNUM] The UNN option forces printing of the file without line numbers. If a line range is specified with this option, it is ignored, and the entire file is printed. RNUM prints the record number for each printed line. If you use UNN to print a numbered file, the line number is printed as the last eight characters of the line.

Examples

Example 1: `/PRINT WATER`

This example prints the entire file named WATER on the terminal.

Example 3: `/PRINT WATER #1/#15,RNUM`

This example prints the first 15 records of the file WATER, with the record numbers printed beside the line number. The record number has a parenthesis after it. The output goes to the terminal by default:

```
1)  1      This file contains the annual report of the
2)  2      Water Resources Board of San Jose, California.
etc.
```

Related commands

DRAFT
ENVELOPE
FINAL
LIST

PROCEDURE
PROC

Purpose

To call a procedure from TDP.

Form

```
PROCEDURE   [pname [,(G)] [rangelist]]
                  (P)
                  (S)
```

Description

The PROCEDURE command calls and executes a user written procedure which is stored in an MPE/3000 Segmented Library (SL). See the **MPE Segmenter Subsystem Reference Manual** for a description of SLs.

For TDP to be able to call a procedure, certain conventions must be followed when the procedure is written. The procedure must return a boolean value. The order and type of the parameters used in the procedure are as shown in the following SPL/3000, FORTRAN/3000 and PASCAL/3000 examples;

A sample header for an SPL/3000 logical procedure:

```
LOGICAL PROCEDURE SAMPLE (STRING, LENGTH, NUMBER, SPACE);
BYTE ARRAY STRING, NUMBER;
INTEGER LENGTH;
ARRAY SPACE;
```

A sample header for a FORTRAN/3000 logical function:

```
LOGICAL FUNCTION SAMPLE (STRING, LENGTH, NUMBER, SPACE)
INTEGER LENGTH, SPACE (20)
CHARACTER STRING*80, NUMBER*8
```

A sample header for a PASCAL/3000 procedure:

```
$subprogram$
program sample;
Const

Type string168 = packed array[..168] of char;
     word      = -32768..32767;
     string8   = packed array[1..8] of char;
     word20    = array[1..20] of word;

Function sample(Var sling:string168;
                Var length:word;
                Var number:string8;
                Var space:word20):word;
```

The procedure must be written to conform to the following requirements:

1 Return a boolean value.

2 Include these four parameters, in the order shown:

STRING, LENGTH, NUMBER, SPACE

where

STRING is a byte array in SPL/3000, a character array in FORTRAN/3000 or a packed array in PASCAL/3000. It will contain one line of the workfile each time TDP calls the procedure.

LENGTH is an integer value holding the length of the current line being passed by STRING.

NUMBER is a byte array in SPL/3000, a character array in FORTRAN/3000 or a packed array in PASCAL/3000. It will contain the line number of the current line being passed by STRING.

SPACE is a scratch array of 20 words maintained by TDP for the procedure's use. The SPACE array is not initialized for each call of the specified procedure as it iterates through the rangelist of the PROCEDURE command. Consequently, this array may be used to "remember" information in successive calls to the procedure within the rangelist specified. The array SPACE is only initialized when a new procedure is referenced.

These four parameters must be declared by the procedure whether they are used or not.

3 The procedure must correct the parameter LENGTH if it shortens or lengthens the line in STRING and pass this value back to TDP. If LENGTH is negative, the line is deleted.

4 The procedure must return logical TRUE for each successful execution: TDP then calls the procedure again for the next line in rangelist.

5 The procedure must return logical FALSE for an unsuccessful operation. TDP then displays an error message and discontinues further calls to the procedure. Lines in rangelist not yet processed by the procedure remain as they were before the PROCEDURE command was used.

Limitations

A PROCEDURE command can call only logical procedures written in a language which accepts parameters as specified and which produces a boolean result. This procedure must reside in a segmented library.

PROCEDURE

PROC

Options

[pname] The pname option provides the name (or entry point) assigned to the procedure in the SL.

[G]/[P]/[S] The G, P, or S option specifies which SL library is to be searched for the procedure (G=group, P-public, S=system). Default for this option is S.

[rangelist] The rangelist option specifies the range of lines to be affected by the procedure. Positions noted within a line (columns or strings) are ignored; the procedure must affect the entire line. If no rangelist is supplied, only the current line is affected.

Example

```
/PROC DRILL, G, 10/20
```

This example calls a procedure named DRILL that is stored in the group segmented library. The procedure is carried out for each of the lines 10 through 20.

PROMPT

Purpose

To aid the user in determining the next step in the text processing environment.

Form

```
PROMPT
```

Description

The PROMPT command lists the present status of the workfile (assuming a workfile has already been created): how the workfile was established, and whether it has been changed since it was created. Next, it provides a list of suggestions for allowable operations, and prompts you for a command. The content of the suggestions will vary according to the context in which the PROMPT command was issued. However, the example below illustrates one case of the PROMPT command. You remain in PROMPT mode, and will continue to receive suggestions, until you type STOP.

Limitations

None.

Options

None.

Example

```
/PROMPT
```

This command displays the status of the workfile and various suggestions for logical commands at this step. The information that is displayed is as follows:

```
TDP PROMPT MODE
WORKFILE texted from EXPER
WORKFILE is "CLEAN" (not changed since last KEEP or TEXT)

Available commands include:
     LIST lines of the workfile
     MODIFY lines in the workfile
     ADD lines to the workfile
        or DELETE, FIND, CHANGE, MOVE, or COPY
     KEEP the workfile
     PRINT old document
     FINAL (formatted) output from old document
     STOP the prompt mode
     EXIT from TDP
         or get suggestions from HELP
Do what/
```

Related commands

HELP

Q

Purpose

To print a message on the terminal.

Form

```
Q[Q] string
```

Description

The Q command is used to prompt the user for input. It can be stored as one of several commands in a USE file, so the string is printed whenever that file is accessed. (See Z: : for another means of prompting for input.)

Limitations

None.

Options

[Q] If a Q is appended to the command, the string is printed without a carriage return and line feed following it. This allows the input to appear on the same line. NOTEXT is not a valid option with this command.

Example

To use Q with a USE file, you would create a file of commands using TDP, with a Q command included in it. KEEP the file as an unnumbered file. Here is a sample file to prompt the data entry clerk for a list of mailing addresses:

```
QQ "Name:"
Addsingle
QQ "Address:"
Addsingle
QQ "City,State:"
Addsingle
QQ "Zip Code:"
Addsingle
```

We will keep it in a file named MAILADD.

Now, to type in the mailing addresses for 50 people, we first issue the following command:

```
/USE MAILADD
```

You are prompted for the first mailing address. Then enter the command /@49 and you will be prompted for the rest of the addresses. (See the @ command description later in this section.) The resulting workfile will list the 50 addresses, but not the prompts printed by the Q command.

Related commands

USE

QUICKTEXT

QT

Purpose

To access a file for listing and limited editing.

Form

```
QUICKTEXT filename [,UNN]
```

Description

QUICKTEXT is used to list, modify, and find lines in the specified file. The command is useful for gaining fast access to another file when KEEPing the workfile and TEXTing the other file would take too long (for example when editing a very large file). Note that the file is not copied into your workfile, as with the TEXT command. You are working on the only copy of the file. To stress this, a different prompt (**) appears, and you can issue one of four commands:

1 LIST
2 MODIFY
3 FIND
4 E,END,EXIT

LIST and MODIFY are used in the same manner as always, except that ranges must be specified with line or record numbers, and the split (S) sub-command may not be used. Record numbers are indicated by a # sign before the number. Options usually accepted with FIND, other than rangelist, however, are not recognized in QT mode; a rangelist must always be specified.

Limitations

QUICKTEXT can not be used on TDP workfiles.

No other commands will be recognized while in QUICKTEXT mode; QUICKTEXT can not be used with proofmarked files. Commands for execution in QUICKTEXT mode can only be input once QUICKTEXT mode has been entered (commands appearing after the QT command on the same line will only be executed after exiting QUICKTEXT mode).

WARNING: This command modifies the permanent stored version of the file, so USE EXTREME CAUTION. It should not be used while doing normal editing of the same file.

Options

[UNN] The UNN option is used if the file is not numbered. In this case, the range numbers are assumed to be record numbers.

Examples

Example 1:

```
/QUICKTEXT WATER
** MODIFY 16
16      the allocation of resources made possible
Changes:
```

This example shows the QUICKTEXT command used to modify a line in the file named WATER. Changes are made as if you were in regular modify mode, with the exception of the (S) sub-command.

Example 2:

```
/QUICKTEXT WATER
** LIST ALL
 1 This file contains the annual report of the
 2 Water Resources Board of San Jose, California.
37 This is the end of the file.
```

This example shows the LIST command used in QT mode.

Example 3:

```
/QUICKTEXT CHAP1
** FIND "TDP" in 45/47
45 TDP is a command driven text processor.
```

This example shows the FIND command used in QUICKTEXT mode.

RACK

Purpose

To display the current settings of column stops and tabstops.

Form

```
RACK[1]
    [3]
```

Description

The RACK command is used to display the current settings of column stops and tabstops. If a tab is set for that column, a T appears; if a column stop is set for the column, a C appears; if both are set for that column, a B appears. IF the RACK command is used without parameters, it displays two lines of output. The first line marks every 10 column positions. The second line displays a number marking each column position, unless a tab or column stop is set for that position.

Limitations

None.

Options

[1] RACK1 prints only the second line of the output.

[3] RACK3 produces three lines of output; the codes for tab and column stop appear separately on the third line.

Examples

Example 1: `/RACK1`

This command prints the following line of output:

```
123456789T1234C67890123486789T1234C6789T
```

Example 2: `/RACK`

This command prints the following two lines of output:

```
         1         2         3         4
123456789T1234C67890123486789T1234C6789T
```

Example 3: `/RACK3`

This version of the command prints three lines of output for the same column and tab settings:

```
         1         2         3         4
1234567890123456789012345678901234567890
         T    C         B    T    C    T
```

Related commands

ALIGN
DELIM
SET
TAB

REDO

Purpose

To modify and execute the previous command line.

Form

```
REDO
```

Description

The REDO command displays the previous command line and allows you to use the sub-commands associated with the MODIFY command to make modifications to that command (with the exception of the S sub-command). The modified command is then executed.

Limitations

REDO should not be used to modify command lines greater than 80 characters in length.

REDO cannot appear on the same line as other commands.

Options

None.

Example

```
/REDO
```

Suppose your last command was /FINDQ "message". The REDO command will output that command and ask for changes as follows:

```
           FindQ "message"
Changes:         r"proprietary"

           FindQ "proprietary"
Changes:   (carriage return signals that modification is complete)
```

When the modification of the command is complete, the new command is executed.

Related commands

@M

REPLACE

Purpose

To replace lines in the workfile.

Form

```
REPLACE[Q] {rangelist } [,HOLD[Q]]
           {string    }
           [(wordlist)]
```

Description

The REPLACE command is used to replace lines in the workfile, limited by the settings of LEFT and RIGHT. The new material can be added from the terminal or from the hold file.

Each line specified in the rangelist is listed on the terminal and the line number is repeated on the next line to prompt you for the new line.

The command is terminated at the terminal by a CTRL-Y or a double slash. When the replacement is from the hold file, the command automatically terminates when the end of the hold file is reached.

Limitations

None.

Options

[Q] The Q option suppresses the listing of the line being replaced. Only the line number prompt is printed.

[rangelist] The rangelist option specifies the range of lines to be replaced. The command cycles through each line, printing each line and prompting you for a replacement.

[string] If the string option is used, TDP searches from the current line pointer until it finds the first occurrence of the string, then prints the line in which it occurs and prompts for a replacement line.

[(wordlist)] The wordlist option works in a similar manner to the string option, but TDP searches for the first occurrence of any word in the list, and then prints the line in which it occurred and prompts for a replacement line.

[hold] The HOLD option replaces lines in the rangelist with the contents of the hold file. If more than one range is given the lines from the hold file are entered one by one; when the first range is completed, the replacement for the second range starts with the next line in the hold file. HOLDQ suppresses the default listing of the new material.

REPLACE
R

Examples

Example 1: `/REPLACE 1/5`

This command cycles through lines 1 to 5, printing the current line then prompting you for a replacement line.

Example 2: `/R "document"`

This example searches from the current line pointer to the next occurrence of "document", then prints the line in which it occurs and prompts you for a replacement line.

Related commands

CHANGE
COLREPLACE
MODIFY
OVERLAY

RESEQUENCE
RESEQ

Purpose

To renumber the workfile. As an option, the file may also be sorted.

Form

```
RESEQUENCE [line] [BY increment] [,SORT]
```

Description

The RESEQUENCE command renumbers the entire work file. You may specify the new starting line number; default is 1. The increment between the new line numbers is governed by the DELTA parameter of the SET command, or it can be optionally set to any appropriate value. Optionally, the file can be sorted on one or more keys, to a maximum of five.

Limitations

The RESEQUENCE command is time-consuming, since the entire workfile is rewritten. If you wish to renumber your workfile and then KEEP it, it is more efficient to use the RESEQUENCE option of the KEEP command than to use this command followed by a KEEP.

Options

`[BY increment]` The BY increment option determines the increment between the new line numbers, overriding the SET DELTA value.

`[SORT]` The SORT option prompts you for the starting location and length of the keys on which the file is to be sorted. A key is a contiguous set of column positions in the record.

RESEQUENCE
RESEQ

Examples

Example 1: `/RESEQUENCE 100 BY 10`

This command renumbers the file starting with 100 and continuing 110,120,130,...

Example 2: `/RESEQUENCE 10 by 1, SORT`

Say you have a list of names and the zip code for each person's mailing address, in your workfile as follows:

```
1      95127      Andersen, William
2      95127      Doerr, Jerry
3      95131      Stewart, James
4      98055      Butler, Philip
5      97132      Newberg, Robert
6      95035      Kisor, Stanley
7      98055      McLean, Susan
8      98055      Osborn, Michael
```

Now suppose that you want to list the names by zip code, and then alphabetical order within each zip code. The command shown for this example above will produce the following dialogue and resequencing:

```
Number of keys:  2
  Start column:  1
Number of columns:  5
  Start column: 11
Number of columns:  2
/LIST ALL
    10      95035      Kisor, Stanley
    11      95127      Andersen, William
    12      95127      Doerr, Jerry
    13      95131      Stewart, James
    14      97132      Newberg, Robert
    15      98055      Butler, Philip
    16      98055      McLean, Susan
    17      98055      Osborn, Michael
```

Related commands

KEEP

SCREEN

Purpose

To allow full screen editing on Hewlett-Packard terminals. See the Product Data Sheet for information about the terminals which support SCREEN mode.

Form

```
SCREEN [rangelist] [,NUM    ]
                   [,LONG   ]
                   [,ADJUST ]
```

Description

The SCREEN command allows you to take advantage of the full screen editing capabilities of on most Hewlett-Packard terminals. (See the TDP Data Sheet for the current list of devices).

When the command is initiated, TDP attempts to set BLOCK MODE, strap D, and the auto line feed key programatically; if that attempt fails, you are asked to manually adjust the terminal.

If the TERMINAL parameter has not been set (via SET TERM), TDP asks you for the terminal name. Enter the model number of the terminal. (Your answer may include the "HP" prefix.)

Now TDP displays the first range in the rangelist on the terminal. If a range containing more data than can be held in the terminal display memory is specified, the range will be split into parts and processing will proceed as if a rangelist had been given. Use the cursor control keys and the delete and insert character and line keys to edit the lines in the range.

When the lines appear correctly on the screen, press the Enter key. Modified lines are put in the workfile and the next range is displayed. Note that if you are screen editing without the NUM option, TDP may allocate new line numbers to this part of of the text.

You will exit from screen mode automatically when the rangelist has been exhausted; to exit prematurely, manually set block mode off and enter a CTRL-Y. On multipoint terminals the exit procedure is different ; hold the Break key until any data being received completes and then press the Enter key.

On normal exit from screen mode, if the block mode and strap D have been manually adjusted, you will be reminded to reset the terminal.

As described above, where a range will not fit in the terminal memory it is split up into parts. The size of these parts is governed by the SET parameter SCREENMAX. If SCREENMAX is set to zero , the default value, TDP makes a conservative estimate of the amount of terminal memory available in your terminal and displays an appropriate amount of data. If however you wish to maximize the amount of data screened you can alter the value of SCREENMAX with the SET command. For HP 262X terminals and the HP 2382 SCREENMAX should be set to the maximum number of lines to be screened. On HP 264X the situation is more complex since the maximum number of lines which can be held in display memory depends markedly on the length of those lines. To determine the maximum value for SCREENMAX for a HP 264X terminal use the TDP command /TERMCAP. This determines the number of 3 character length lines the terminal will hold. Setting SCREENMAX to this value would result in the terminal display memory being completely filled on each screening leaving no room for any additions to the data. Thus the user may wish to SET SCREENMAX to a convenient percentage of this value to allow for additions to the data.

Limitations

If SCREEN is used without the NUM option TDP has no way of matching up lines on the screen with lines in the workfile; thus you cannot set the margins to SCREEN only a portion of a workfile line ,edit it and get that portion updated in your workfile. For this reason SCREEN without the NUM option sets LEFT=1 and asks you if it can SET RIGHT and LENGTH to the terminal page width. (Terminal page width is 80 for all SCREEN supported terminals except the HP 2626 where it is configurable.)

If you answer NO to this question and then if you enter data on the terminal screen in column numbers higher than the value of RIGHT or LENGTH this data will be truncated when the workfile is updated from the screen.

If the SCREEN command is used with the NUM option the command acts only on data between the LEFT and RIGHT margins consistent with other TDP commands.

Options

[rangelist] The rangelist option specifies ranges of lines to be displayed for local editing.

[LONG] The LONG option allows the user to SCREEN lines incorporating display enhancement escape sequences which make the actual number of characters transmitted to and read from the terminal greater than the actual display width of the terminal. The LONG option is only valid without the NUM option. The LENGTH of the workfile lines must be sufficient to hold up to eighty displayable characters plus the variable number of non-displaying characters necessary to switch the display enhancements on and off.
Thus to display and record screens using the display option, first set the LENGTH parameter to 160 and then use the SCREEN command with the LONG option. TDP will set RIGHT=LENGTH and will then try to display and read up to 160 characters on each line of the screen. WARNING: If you have more displaying characters on a line than the terminal will display (i.e. 80 except for the HP 2626) truncation of the line will occur.

[NUM] The NUM option will include line numbers in the display. When NUM is specified, if the capability exists in the terminal, the left margin of the terminal will be set to the start of text (column 11 for default format files).

The cursor can be moved into the line number field to add or modify existing line numbers. When new lines are inserted, the line number field is left blank. Upon return to the workfile, a line number will be assigned to maintain its relative position. You can provide a line number if desired. You can also modify an existing line number with the potential effect of moving the line within the workfile.

Three error conditions can occur when a numbered display is returned to the workfile:

a If an invalid line number (e.g. ABC) is encountered, the entry of text is stopped, the terminal bell is sounded, and a "B" is displayed in the column between the offending line number and the text of the line. The line number can then be corrected and the Enter key pressed to continue.

b If a duplicate line number is encountered, the result is as described in (a) above except that a "D" is displayed after the line number. To continue, correct the line number and press the Enter key.

c If more lines have been inserted than will fit between two existing line numbers, the extra lines will be written to a file with a name of the form Xdddtttt in which "ddd" is the day of the year and "tttt" is the time of day. After enough space has been made in the workfile, the extra lines can be JOINed.

[ADJUST] The ADJUST option is solely for use when using TDP over a DS1000/3000 link. When this option is used TDP does not issue line feeds when reading from the screen, thus compensating for the extra line feeds issued by the DS software on a 1000/3000 link.

Examples

```
/SCREEN 1/5, 44/66
```

This example displays two ranges for local editing. First, lines 1 through 5 are displayed. When they have been modified and entered, lines 44 through 66 are displayed for modification.

SET
S

Purpose

To set various parameters used by other TDP editing commands.

Form

```
SET

     [BATCHERROROK               ]     [NOSTREAM                   ]
     [BIGBLOCK [[=] ON ]         ]     [NOTAB                      ]
     [         [[=] OFF]         ]     [PERMYES[[=]ON ]            ]
     [BLIT [[=] ON ]             ]     [       [[=]OFF]            ]
     [     [[=] OFF]             ]     [PMARK                      ]
     [COLSTOP [[=] n1,n2,...,n14]]     [PNEWS [[=]ON ]             ]
     [        [[=] SPREAD n     ]]     [      [[=]OFF]             ]
     [COMP                       ]     [POSTADD [=] {n}            ]
     [CONTINUE                   ]     [         {COMPILE   }      ]
     [CONTROLY                   ]     [         {TDP[/3000]}   {C}]
     [DELTA [=] {n}              ]     [PRIORITY {RUN       }[=]{D}]
     [DICTIONARY [=] {filename}  ]     [         {FORMAT    }   {E}]
     [DISPLAY                    ]     [         {SPOOLER   }      ]
     [ERROR [=] {n}              ]     [PROMPT [=] char            ]
     [ESCAPE [=] char            ]     [QUIET                      ]
     [FILETYPE [=] {n}           ]     [RIGHT [=] {n}              ]
     [FORMAT[=]{COBOL  }         ]     [RUN                        ]
     [         {DEFAULT}         ]     [SCREENMAX [=] {n}          ]
     [         {DIARY  }         ]     [SESSQUIET [[=] ON/OFF]     ]
     [HIGHC [=] {n}              ]     [SHORTERROR                 ]
     [LANGUAGE [=] {string}      ]     [SIZE [=] {n}               ]
     [LEFT [=] {n}               ]     [SPOOLER [[=] ON/OFF]       ]
     [LENGTH [=] {n}             ]     [STAMP char                 ]
     [LINE [=] {n}               ]     [STOP                       ]
     [LIT                        ]     [STREAM                     ]
     [LOCPARM [=] {ON }          ]     [SYSERROR [=] {n}           ]
     [            {OFF}          ]     [TAB                        ]
     [LONGERROR                  ]     [TABCHAR [= char]           ]
     [LOWC [=] {n}               ]     [        [= tabkey]         ]
     [MISS[OK]                   ]     [TABSTOP [[=] n1,n2,...,n15]]
     [MPE                        ]     [        [[=] SPREAD n     ]]
     [MUF [[=]ON ]               ]     [TERMINAL {string }         ]
     [    [[=]OFF]               ]     [TERMTYPE {n}               ]
     [MUST                       ]     [VBIGBLOCK [[=] ON ]        ]
     [NO                         ]     [          [[=] OFF]        ]
     [NOBLIT                     ]     [WINDOW [=] {n}             ]
     [NOBREAK                    ]     [YES                        ]
     [NOCOMP                     ]     [ZFILL [[=] ON ]            ]
     [NOCONTROLY                 ]     [      [[=] OFF]            ]
     [NOMPE                      ]     [ {n}                       ]
     [NONLIT                     ]
     [NORUN                      ]
```

SET

Description

The SET command is used to set values for parameters that affect the operation of TDP. Some of the parameters are set to TRUE or FALSE just by using "SET parameter"; others are assigned numeric values; default values or conditions exist for all of the parameters.

Parameters can be automatically set for a file whenever it is texted in by including a special line in the file. The first five characters must be *SET followed by a parameter list. The SET command must be uppercase and no semicolons can appear in the parameter list.

You can store SET parameter values, for future sessions, by issuing a STOREPARMS command.

Options

1 BATCHERROROK - Normally, TDP editor operating in batch mode will terminate if any error occurs. The SET BATCHERROROK command will allow TDP editor to continue even though an error does occur. Once set, the parameter cannot be reset.

2 BIGBLOCK [[=] ON/OFF] - A standard TDP work file contains 9 records per block. The command SET BIGBLOCK ON establishes the work file with 18 records per block. As a result, the amount of memory used is increased, but the number of disc accesses decreases resulting in better performance. The default value is OFF. In the command, the absence of ON of OFF implies ON. NOTE: If VBIGBLOCK is ON, BIGBLOCK is ignored.

3 BLIT [[=] ON/OFF] - to alter the search logic for string searches. The default is ON. If BLIT is OFF, a string beginning with an alphabetic or numeric character will only be found if it is delimited by special characters. If neither ON nor OFF is given in the command, ON is assumed.

4 COLSTOPS [[=] n1,n2,...,n15]
[[=] SPREAD n] - to set the column stops used by the ALIGN command. This can be set in several ways:

a SET COLSTOPS - results in a line of numbers representing column positions being printed. You can space across the line and place a "T" beneath each column position you wish to set to a maximum of 15. NOTE: When this form of the command is used, all previous column stops are deleted.

b SET COLSTOPS n1,n2,...n15 - allows you to set column stops at specific locations. For example, SET COLSTOPS 5,20 will set column stops at column positions 5 and 20.

This form of the command also allows you to delete or modify existing column stops. The commas between the numbers are optional. If provided, they can preserve an existing column stop. For example, if column stops currently exist at locations 10, 20, 30 and 40, the command SET COLSTOPS 22,35 would result in column stops at 22,35 and 40. However, if the command SET COLSTOPS ,,22 was given, the result would be column stops at 10,20,22 and 40.

c SET COLSTOPS SPREAD n - sets "n" column stops evenly spaced across the line between the LEFT and RIGHT margins. For example, with LEFT=1 and RIGHT=72, the SET COLSTOPS SPREAD 6 command produces column stops at 10,20,30,40,50 and 60. Again, 15 is the maximum number of column stops which can be set.

5 COMP - Resets the SET NOCOMP command. You must have Account Manager or System Manager capability to use this command.

6 CONTINUE - Normally, an error resulting from a command executed from a USE file will terminate processing of that USE file. If the command SET CONTINUE is issued, such errors will be reported but processing will continue. To reset this parameter, issue a SET STOP command.

7 CONTROLY - The ability to press Control-Y to interrupt TDP processing can be disabled by a SET NOCONTROLY command. The SET CONTROLY command is used to re-establish the capability.

8 DELTA [=] n - is used to change the default increment used when lines are added or renumbered within the work file. The default values for DELTA are 0.1 for COBOL files, and 1 for all other files. DELTA may be set between .001 and 100, inclusive.

9 DICTIONARY [=] filename - This allows the user to specify which User Dictionary is to be used by the CHECK/SPELL commands. A User Dictionary file must be an ASCII file, with a line length of 32 characters, unnumbered, and with one word per line.

If not set, the default is AMSPELL for American and BRSPELL for British. (For further information on user dictionaries see the HP SPELL Administration manual).

Example: /SET DICTIONARY MYDICT.MYGRP.MYACC

10 DISPLAY - The normal mode of operation is to display all changes being maoe to the work file. If this display has been turned off by a SET QUIET command, it can be re-established through the SET DISPLAY command.

11 ERROR [=] n - to print the text for a particular error number.

12 ESCAPE [=] char - to assign a special character (non-alphanumeric) that will be used by the ADJUST option on the ALIGN command to properly align columns which contain non-printing characters. The default character is the caret (^).

13 FILETYPE [=] n - allows you to assign a file code to your file for the convenience of identifying it in a LISTF or CATALOG listing. Only filetypes 250 through 999 may be set.

14 FORMAT [=] (DEFAULT/COBOL/DIARY) - is used to set the format of the work file to DEFAULT, COBOL, or DIARY. The main difference between these formats is the manner in which sequence numbers are handled. The file format should be set before the first line is entered into the work file. When FORMAT is set, the filetype is also set.

DEFAULT - Most files will have the default file format. In this format, the sequence number of eight character in length is placed at the end of each line when the file is kept. The default value of DELTA is 1 and 72 is the default value for RIGHT and LENGTH.

COBOL - COBOL files are kept with a six character sequence number at the front of each record. Three digits are allowed before the decimal point, and three after. The default value of DELTA is 0.1 and 74 is the default value for RIGHT and LENGTH.

DIARY - The sequence numbers for diary files, which are composed of a date, time and sequence number, are kept as they are for DEFAULT files, but TDP treats the ADD command for DIARY files in a special way. Line numbers assigned will represent the date and time. Subsequent PRINT, TEXT, or QT commands recognize these line numbers.

Up to 999 lines can be entered for any day and time. Times recorded as line numbers are rounded to the nearest quarter-hour. The line number appears after the date and time.

Lines may be retrieved by date (Example: LIST 8/13) or date and time (Example: LIST 8/13_12:00a). In addition, lines can be added or referenced by the keywords NOW (current day and time) and TODAY (today's date, time of "zero" for the day). For example:

```
/SET FORMAT = DIARY
/ADD NOW
```

7/16_2:30p1 This is the sequence number of a DIARY file.
The 7/16 is the date, 2:30p is the time rounded
to the nearest 15 minutes, either a.m. or p.m.,
and 1 identifies line number 1.

15 HIGHC [=] n - to set the upper column boundary for locating search strings in each line. This parameter is used with the LONG option on the LIST, CHECK and SPELL commands and when the SL ("search limit") option is specified with any command. The default (and maximum) value is 255. HIGHC may not be set lower than LOWC.

16 LANGUAGE [=] string - This information is used to select the System and Corporate Dictionaries to be used when checking the spelling. If no language is set, American is assumed.

Example: /SET LANGUAGE BRITISH

For further information on System and Corporate Dictionaries see the HP SPELL Administration manual.

17 LEFT [=] n - to set the left-hand margin of the work file. The LEFT value must be less than RIGHT. The default (and minimum) setting is 1.

18 LENGTH [=] n - to set the line length when creating a work file. LENGTH cannot be less than the value of RIGHT and it cannot be set greater than the record size for the work file. The value of LENGTH must be even. If it is set to an odd number, TDP automatically increases the value by 1. LENGTH can be increased after a file has been texted, but by no more than 22 characters. (Note that a PMARKed file cannot be increased in length in this way.) When a file is created, however, the only limitation is the maximum length size of 168 characters. Length can be decreased at any time, with no restrictions. The default value depends upon the format. Appendix C contains details on work file record size.

19 LINE [=] n - to alter the size of the line on the terminal. If a line is sent to the terminal that is longer than this value, two or more lines will be printed on the terminal. The default line size is 160 characters. Use of the SET TERMINAL command can alter the value of LINE. SETting TERMINAL to HPnnnn or CRT sets LINE to 80.(HP 2601 excepted). SETting TERMINAL to HP 2626 sets LINE to the current page width of the HP 2626.

20 LIT - to cause all searches not overridden with a NONLIT option to be conducted in a literal fashion, i.e., even if the search string occurs within a longer string.

21 LOCPARM [=] {ON/OFF} - is used by the System Manager to control capability of setting individual parameters for each group. ON means allowed, OFF disallowed. (See STOREPARMS for further information.)

22 LONGERROR - Normally, when an error occurs, the user is given both an error number and a message. The error message may be suppressed by issuing the command SET SHORTERROR. In order to return to the default mode, use the command SET LONGERROR.

23 LOWC [=] n - to set the lower column boundary for locating search strings. This parameter is used only if the option word SL is specified. The default (and minimum) setting is 0. LOWC may not be set higher than HIGHC.

24 MISS[OK] - to eliminate the "string not found" error. This parameter is useful for USE files and in batch applications where, if a string is not found, you want to continue with the next command rather than having the job terminate because the error was detected. Use SET MUST to reset this parameter.

25 MPE - Resets the SET NOMPE command. You must have Account Manager or System Manager capability to use this command.

26 MUF [=] (ON/OFF) - Monitor Use File. When executing a USE file, TDP does not reflect the commands being processed. With MUF set, TDP will reflect them back to the user's terminal. This facility is particularly useful when testing USE files.

27 MUST - to reset the default condition changed with the MISSOK parameter. If MUST is set, when a search string is not found, an error is generated and command processing halts.

28 NO - to automatically provide a NO response to the next TDP question. After one question, this parameter is reset and you must provide answers via the terminal.

29 NOBLIT - NOBLIT performs the same function as SET BLIT OFF. See BLIT for details.

30 NOBREAK - to disable the Break key. BREAK remains disabled until you exit TDP. Note that if NOBREAK and NOCONTROLY are set, there is no way to interrupt anything TDP is doing.

31 NOCOMP - to prevent compilations of FORTRAN, COBOL, COBOL II, PASCAL, RPG, and SPL programs from within TDP. To reverse the command use SET COMP.

32 NOCONTROLY - The ability to press Control-Y to interrupt TDP processing can be disabled by a SET NOCONTROLY command. The SET CONTROLY command is used to re-establish the capability. It is also re-established automatically upon a return from a FINAL, RUN or compile command.

33 NOMPE - to prevent use of MPE commands within TDP. To reset the command use SET MPE. Note that if this parameter is set, NOBREAK should also be set; otherwise the Break key can be used to access MPE commands.

34 NONLIT - re-establishes the default non-literal search condition if a SET LIT command has been issued. A non-literal search will find the string only if it occurs as a separate word - thus, the string "is" would not be found in the word "This". Furthermore, if the string is entered in lowercase letters, the string will still be found if the first letter of the word in the text is capitalized - thus, a search for the string "is" would also find the word "Is".

35 NORUN - to prevent the running of programs while using TDP. To reset the command use SET RUN.

36 NOSTREAM - to prevent running job streams from within TDP. All other MPE commands are allowed. To reset the command use SET STREAM.

37 NOTAB - to indicate that TDP is not to perform any tabbing function. Since this is the default condition, it would only be used if a SET TAB command has been issued.

38 PERMYES [[=] ON/OFF] - to set (or reset) all answers to yes/no questions usually asked by TDP to "yes". This is useful if you wish to use a USE file with a set of commands which require a series of "yes" responses, and did not want to be at the terminal the entire time the commands were executing. NOTE: The use of SET PERMYES with no options will change the sense of the value (ie. if it is on, it will be set off).

39 PMARK - to begin proofmarking the file, and increase the length of the text line eight characters to make space for the markings. Once the marks have been made, you are not allowed to increase the line length. The proof marks show the most recent editing operation on any lines. Lines that have been added or edited will have a two letter code that shows the operation, and the Julian day the operation was done. The codes are as follows:

AD = ADD	AL = ALIGN
CH = CHANGE or SCREEN	CM = COLUMNMOVE
DS = DOWNSHIFT	GL = GLUE
JO = JOIN	MD = MODIFY
MG = MERGE	PR = PROCEDURE
RP = REPLACE	RQ = RESEQUENCE
SQ = SQUEEZE	US = UPSHIFT
WM = WORDMOVE	

If PMARK is set, when you enter /SET PMARK, the system begins a dialogue with you. You are first asked if you want to clear old pmarks. The answer should be "yes" if you want to clear some or all pmarks but leave the file with pmark set; otherwise, answer "no". If you answer "no" to the first question, you are then asked if you want to re-set the pmark flag (thereby turning off the pmark facility and clearing all existing pmarks).

Proofmarks are included in DRAFT output on the right side of each line; they are not normally shown on FINAL copy (see Section 4, \Revision Mark).

40 `PNEWS [=] (ON/OFF)` - This facility allows a "welcome" message to be displayed on a user's terminal at the start of a TDP session. TDP looks for a file TDPNEWS.TDPDATA.HPOFFICE. If it exists and PNEWS is set ON, the contents of the file will be displayed. The file must be unnumbered.

41 `POSTADD [=] n` - The lines being added to a file are not permanently stored on the disc when entered. In fact, only after the input of 99 lines (by default) will the information be stored. As a consequence, if the computer system fails, the lines which were added since the last disc update will not be saved. The execution of the command /SET POSTADD 20 will permanently save every 20 lines, thus, the loss of information will be smaller in the case of a system failure. The more frequently disc updates are done, however, the more slowly TDP will perform. The maximum value of POSTADD is 999.

```
            {COMPILE    }
            {TDP[/3000]}       {C}
42 PRIORITY{RUN         } [=] {D}
            {FORMAT     } [=] {E}
            {SPOOLER   }
```

To set the execution priorities of a compilation, a running of TDP, or of running any program from TDP. This setting is maintained for the duration of the TDP run, or until reset by another SET PRIORITY command.

43 `PROMPT [=] char` - to set a special character (non-alphanumeric) to be used as the editor prompt character.

44 `QUIET` - to suppress most of the output from TDP commands. It is equivalent to the use of Q option on individual commands.

45 `RIGHT [=] n` - to set the right margin of the work file. RIGHT must be greater than LEFT, and less than or equal to the line LENGTH parameter. The default setting depends on the format.

46 `RUN` - Resets the SET NORUN command. You must have Account Manager or System Manager capability to use this command.

47 `SCREENMAX [=] n` - to set the maximum number of lines to be sent at any one time to a terminal by the SCREEN command. When SCREENMAX is set to 0, the default value, TDP will set a conservative limit on the amount of data to be sent to a terminal at any one time. To maximize the amount of data screened SET SCREENMAX to the maximum number of lines your terminal can hold. The TDP command /TERMCAP can be used to determine this. Setting SCREENMAX to this value would result in the terminal display memory being completely filled on each screening leaving no room for any additions to the data. It is thus advisable to SET SCREENMAX to a smaller value to allow for additions.

48 SESSQUIET [[=] ON/OFF] - If SESSQUIET is set ON, messages from other users or the operator will be suppressed. NOTE: If no options are included, the sense of the flag is switched (ie. if SESSQUIET is ON and you issue the command SET SESSQUIET, then SESSQUIET will be OFF).

49 SHORTERROR - Normally, when an error occurs, the user is given both an error number and a message. The error message may be suppressed by issuing the command SET SHORTERROR. The command may be reset with SET LONGERROR.

50 SIZE [=] n - to alter the number of records allowed in the work file. The default SIZE=2000. See Appendix C for details. The size of an existing workfile cannot be altered.

51 SPOOLER [[=] ON/OFF] - When SPOOLER is OFF references to the TDP configuration file logical names in a FINAL or DRAFT command result in TDP trying to gain access to the specified terminal to print the formatter output. If the terminal is in use the FINAL command fails. When SPOOLER is set ON the formatter output is passed to the TDP spooler which will print it when the specified device becomes free.

52 STAMP char - to define a character which will be inserted in the proofmarks added to the text lines. You might use a different character for each person using a common file, so that it will be obvious who made the latest change to a line.

53 STOP - to return to the default condition for error handling. When this parameter (opposite of CONTINUE) is set, processing stops when an error occurs during USE file operations.

54 STREAM - Resets the SET NOSTREAM command. You must have Account Manager or System Manager capability to use this command.

55 SYSERROR [=] n - to print the text for a certain file error number.

56 TAB - to enable the tabbing function. When this parameter is set, a specified character is interpreted as the tab character in the ADD, REPLACE, INSERT, and MODIFY commands. As TDP enters the line, it replaces the tab character with the appropriate number of blanks.

57 TABCHAR [= char/TABKEY] - to establish the character (or terminal tab key) to be used as the tab character. The default setting is the caret (^). Note that the caret prints as an up arrow on some terminals and printers. Any character can be set as the tab character. If the TABKEY is used as the tab character, the (Back space) key should not be used to correct typographical errors, as it will be difficult to determine what column should be corrected.

```
58 TABSTOPS [[=] n1,n2,...,n15]
            [[=] SPREAD n       ] - to set tab stops. They can be set in
```
several ways:

a SET TABSTOPS - results in a line of numbers representing column positions being printed. You can space across the line and place a "T" beneath each column position you wish to set to a maximum of 15. NOTE: When this form of the command is used, all previous tab positions are deleted.

b SET TABSTOPS n1,n2,...n15 - allows you to set tab stops at specific locations. For example, SET TABSTOPS 5,20 will set tab stops at column positions 5 and 20.

This form of the command also allows you to delete or modify existing tab stops. The commas between the numbers are optional. If provided, they can preserve an existing tab stop. For example, if tab stops currently exist at locations 10, 20, 30 and 40, the command SET TABSTOPS 22,35 would result in tab stops at 22,35 and 40. However, if the command SET TABSTOPS ,,22 was given, the result would be tab stops at 10,20,22 and 40.

c SET TABSTOPS SPREAD n - sets "n" tab stops evenly spaced across the line between the LEFT and RIGHT margins. For example, with LEFT=1 and RIGHT=72, the SET TABSTOPS SPREAD 6 command produces tab stops at 10,20,30,40,50 and 60. Again, 15 is the maximum number of tab stops which can be set.

```
59 TERMINAL {HP150   }
            {HP2382  }
            {HP2392A}
            {HP26nn  }
            {HP262X  }
            {HP263n  }
            {HP239n}
            {HP700/9n}
            {CRT     }
```

This command is used to specify the terminal being used for editing. Output requested using the FINAL or DRAFT commands without specifying the output terminal is formatted for the terminal specified through this command. SCREEN command also uses this specification of terminal name. SETting TERMINAL to HPnnnn or CRT has the effect of SETting the LINE parameter to 80. The only exceptions being the HP 2601 when no change to LINE is made and the HP 2626 where LINE is set to the configured page width of the terminal.

60 TERMTYPE n - to set the terminal type for MPE. See the HELLO command in the MPE command Reference Manual.

61 VBIGBLOCK [[=] ON/OFF] - A standard TDP work file contains 9 records per block. The command SET VBIGBLOCK ON establishes the work file with 36 records per block. As a result, the amount of memory used is increased, but the number of disc accesses decreases resulting in better performance. The default value is OFF. In the command, the absence of ON of OFF implies ON. NOTE: If VBIGBLOCK is ON, BIGBLOCK is ignored.

62 WINDOW [=] n - to assign a number to be used as line range for a window. The default setting is 18 and the maximum is 55. If WINDOW is specified as a line range, the next 18 lines are listed.

63 YES - to automatically provide a YES response to the next TDP question. After one question, this parameter is reset and you must provide answers via the terminal. It is of most use in a USE file in which a YES/NO response is needed but you do not wish to set the PERMYES flag on.

64 ZFILL [[=] ON/OFF] - The ZFILL parameter, used in conjunction with the Z: : construct (see the Z: : command).

65 n - Lists a Roman Numeral equivalent of the given number. The number must be in the range 1 to 9999.

Related commands

DISPLAYPARMS
GETPARMS
STOREPARMS
VERIFY

SHOWSPOOL

Purpose

To display information about spoolfiles

NOTE: Users with Account Manager or System Manager capability will also be provided with printer status information.

Form

```
SHOWSPOOL
```

Description

The SHOWSPOOL command displays the status of spoolfiles. It prints a heading and then a line of information for each spoolfile . The spoolfile entries are displayed in priority order i.e. spoolfiles at the top of the list are being considered for printing before those lower down.

The information headings are ID#, FILENAME, CREATOR, STATUS and PORTS.

The ID# (file identification number) is assigned by the spooler in numerical order as print requests are received. The ID# is used to reference a spoolfile in the ALTERSPOOL command.

The FILENAME is the name you have assigned to the textfile. EDITOR WORKFILE is displayed where no text file has been specified.

The STATUS of a spoolfile can be READY, STOPPED or PRINTING. A READY spoolfile will be printed as soon as all previous spoolfiles for that device have been dealt with. A STOPPED spoolfile will not be printed until a user changes its status to READY. A PRINTING spoolfile is currently being printed.

The PORT number is the three digit number identifying the port to which the printer is attached. Where more than one device number is shown this is because the TDP configuration file allows a choice of printer. For users with Account or System Manager capability a display of printer status is appended to the spoolfile information.

Limitations

None.

Options

None.

SHOWSPOOL

Example

```
ID# FILENAME            CREATOR               STATUS    PORTS

3   REPORT.PUB.TDP      JIM.SALES,BUS         READY     015
12  MEMO                JOHN.ACCOUNT,GROUP    PRINTING  015
14  LETTER              JACK.TDP,HP36578      STOPPED   041


DEVICE#      STATUS

 012         UNAVAILABLE
 013         IDLE
 015         BUSY
 041         IDLE
```

Device status information is only displayed to users with SM or AM capability.

Related commands

ABORTSPOOL
ALTERSPOOL
STARTSPOOL
STOPSPOOL
SET

SPELL

Purpose

To check and correct the spelling of words within a specified range of the workfile. (HP SPELL must be installed on your system.)

Form

```
SPELL[Q] [rangelist] [,UNN     ]
                     [,OFFLINE ]
                     [,NEW     ]
                     [,LONG    ]
```

Description

The SPELL command can be used to check the spelling of part or all of the workfile. The amount of text that is checked is limited by the LEFT and RIGHT margin settings; any words that lie outside the margin settings are not checked. Any portion of the workfile, or all of it, may be checked for spelling mistakes. The language and the user dictionary are specified by using the SET LANGUAGE and SET DICTIONARY commands.

When a misspelt word is found, the user is prompted for input. The following inputs are accepted:-

(Return)

The word is not corrected and checking continues from the next word.

Type A(dd) and press (Return)

The word is added to the user dictionary. Further occurrences of the word will not be flagged as misspellings.

Type C(orrect) and press (Return)

Possible corrections for the word are displayed. To display more corrections, type C and press (Return). To insert a correction into the workfile, type the number of the correction and press (Return). If the correction will make the line of text too long, the word will not be inserted into the workfile.

Type E(nd) and press (Return)

Terminates spell checking.

Type I(gnore) and press (Return)

Further occurrences of the word will not be flagged as misspellings until one of the following:

a) The SET DICTIONARY command is invoked.
b) The SET LANGUAGE command is invoked.
c) The TEXT command is invoked.
d) The user EXITs TDP.

Type T(ype) and press (Return)

The user is prompted for his/her own correction which is inserted into the workfile. If the correction will make the line too long, the word will not be inserted.

SPELL

The SPELL command leaves the current line pointer at the end of the last line checked.

Limitations

The spelling of words that cross line boundaries cannot be checked. If the correction of the spelling of a word results in a line longer than the currently set line length, no more words are checked.

Lines beginning with "\" in column 1 are treated as formatter commands and ignored.

Options

[rangelist] The rangelist option is used to specify the lines of the file that contain words that are to be checked for spelling mistakes. If no rangelist is given, only words on the current line will be checked. To check the whole file, specify ALL.

[UNN] If the UNN option is chosen, the lines will be listed without line numbers.

[OFFLINE] To produce a listing on the system printer, select the OFFLINE option. Word that are not found in the dictionaries will be displayed with suggestions.

[LONG] The LONG option is used to check the spelling of all lines that are longer than a certain length (the length is set with the HIGHC parameter on the SET command).

[NEW] The NEW option is used in conjunction with the proof marking feature (see the SET command). Only the lines with proof marks will be checked for misspellings. The lines will be listed with the proof marks shown next to each line.

[Q] The Q option suppresses the listing of the lines that contain no misspelt words.

Examples

Example 1: `/SPELL ALL`

The simplest example. This command checks the spelling of the whole workfile. Each line is displayed and the user is prompted for input when a misspelling is found.

Example 2: `/SPELLQ 10/100`

This command checks the spelling of lines 10 to 100 of the workfile displaying only those lines that contain spelling mistakes. The user is prompted for input at each misspelling.

Related commands

CHECK
SET

SQUEEZE
SQ

Purpose

To remove extra blanks from a line.

Form

```
SQUEEZE[Q] [rangelist] [,NOTEXT]
```

Description

The SQUEEZE command is opposite to ALIGN, i.e., it removes extra blanks between tokens on a line, and also leading blanks. Usually SQUEEZE is used to compact material that has been ALIGNed on the column stops. The command shifts the tokens to the left, so that only one blank exists between them.

A token is a collection of non-blank characters with blanks before and after it (the beginning and end of a line act as a blank in this case). Examples: HELLO, 333.7, $1,587.62, co-operative, A, and # . More than one word can be grouped together as a single token by using a necessary blank between the words (see \blank command in Section 4). Example: if @ has been defined as a necessary blank, then DATE@OF@PURCHASE would be considered a token by TDP.

Limitations

None.

Options

`[Q] or [NOTEXT]` The Q or NOTEXT option suppresses the listing of the lines being squeezed.

`[rangelist]` The rangelist option determines which lines are to be squeezed. If no rangelist is specified, only the current line is affected.

Examples

Example 1: `/SQUEEZE`

Suppose the current line lists five tokens, aligned by their left-hand character on the default column stops as follows:

```
1234      1357      5534      9908      6650
```

The command listed above would result in the following line:

```
1234 1357 5534 9908 6650
```

Example 2: `/SQUEEZE 5/8, NOTEXT`

In this example, lines 5 through 8 would be shifted left as in Example 1. However, since NOTEXT was specified, neither the original line nor the resulting line are displayed on the terminal.

Related commands

ALIGN
FILL

STARTSPOOL

Purpose

To start the TDP Spooler.

Form

```
STARTSPOOL
```

Description

This command after checking to see if the spooler is already in operation streams the job which runs the TDP spooling program.

Limitations

Under heavy system load there may be a delay before the spooler job starts and it is also possible that the job limit for your system has been reached in which case the job goes into a WAIT state until an already executing job terminates. For these reasons it is advisable to check after a short delay that the spooler is running by issuing the command /VERIFY SPOOLER.

Options

None.

Related commands

ABORTSPOOL
ALTERSPOOL
SHOWSPOOL
STOPSPOOL
SET
VERIFY

STOPSPOOL

Purpose

To stop the TDP spooler.

Form

```
STOPSPOOL
```

Description

The STOPSPOOL command stops the TDP spooler. Spoolfiles currently being printed are allowed to complete before the spooler stops. The command ABORTSPOOL is used if you wish to stop the spooler unconditionally.

Limitations

None.

Options

None.

Related commands

ABORTSPOOL
ALTERSPOOL
SHOWSPOOL
STARTSPOOL

STOREPARMS

Purpose

To set site-specific or group-specific default settings for TDP parameters.

Form

```
STOREPARMS
```

Description

To set site-specific parameters, you must have Account Manager capability for the HPOFFICE account or System Manager capability for the system and be logged onto the TDPDATA.HPOFFICE group. You may then run TDP and issue SET commands to alter default values for desired parameters. You issue a STOREPARMS command to store those parameters in the file TDPPARMS accessed by TDP at the beginning of each session. This sets parameters that will be used by everyone at the site.

One TDP parameter that can be set in this way is LOCPARM. If it has been set at the site level, individual users can alter default values for their groups. The procedure is as follows:

1 BUILD a file named PARMSET (only 1 record is needed but it must have a record size of 256 bytes.)

2 Run TDP

3 Issue SET commands to alter defaults for desired parameters.

4 Issue a STOREPARMS command

Step 4 stores the parameter settings in the PARMSET file.

Later, when a user runs TDP, TDPPARMS.TDPDATA.HPOFFICE is checked for parameter settings. If LOCPARM is set, TDP checks for the file PARMSET in the log-on group for the group-specific parameters.

Note that parameters set for the site override defaults established in TDP, and that parameters set for the group override those set for the site. The parameters which can be stored are listed below:

```
TABS
PERMYES
QUIET
LITERAL
BERROROK
SHORTERR
SPOOLER
BLIT

HIGHC
LOWC
RIGHT

SESSQUIET
YES
NORUN
NOMPE
STREAMOK
CONTINUE
MISSOK
NOCOMP

VBIGBLOCK
BIGBLOCK
LOCPARMS
PNEWS
ZFILL
MUF
```

```
    LEFT
   PMARK
  LENGTH
FILETYPE
  WINDOW
 POSTADD
  PROMPT
  ESCAPE
TERMINAL
 TABCHAR

TABSTOP settings:
```

Limitations

The workfile must be empty when a STOREPARMS command is issued.

Note that any user logged onto a particular group can change the parameters stored in the PARMSET file of that group, thereby altering the group-specific settings. If you wish to restrict the altering of the group level values to only one user of the group, use the MPE ALTSEC command to change the security on the PARMSET file so that all users can read it, but only the Group Librarian (GL) or Account Librarian (AL) can write to the file. This can be done through the command:

```
:ALTSEC PARMSET;(R:ANY;W:AL,GL).
```

Options

None.

Example

```
:HELLO MGR.HPOFFICE
:RUN TDP.PUB.SYS
/SET LOCPARM ON
/SET VBIGBLOCK ON
/STOREPARMS
```

The command sequence shown here sets two parameters for all users at the site: LOCPARM and VBIGBLOCK. Then an individual user might go on to set certain parameters for all users of that group.

```
:HELLO USER.TECHPUBS
:BUILD PARMSET;DISC=1
:RUN TDP.PUB.SYS
/SET LEFT=1
/SET RIGHT=60
/STOREPARMS
```

Now the default margins for every user of that group are 1 and 60. The user can, of course, override the defaults by issuing SET commands during his/her session.

Related commands

DISPLAYPARMS
GETPARMS
SET
VERIFY

TAB

Purpose

To set up tabbing parameters

Form

```
TAB
```

Description

This command provides a convenient way of setting up the tabbing parameters TABS,TABCHAR and TABSTOPS. (They can also be set with the SET command.)

Up to 15 tabstops can be set in TDP. You can define a tab character such that when TDP encounters this character in the input line it will replace it with the number of blanks necessary to shift the remainder of the input line to the next tabstop.

The terminal tabkey can be defined as the tab character in which case both the terminal and TDP will respond to the tab character. The terminal responds by moving the cursor to the next tabstop and when the line is entered with the (Return) key TDP responds by replacing the tab character with an appropriate number of blanks. Two characteristics of this mode of tabbing should be noted:

1 TDP will set up the tabstops in your terminal only if you have defined the TABCHAR as the TABKEY before setting the TABSTOPS.

2 Use of the (Back space) key in this mode may give unexpected results. e.g. Suppose with a single tabstop at column 10 you enter, starting at column number one, "ABC tab (Back space) DEF". The result on the screen would be:

```
column no   1   5   9T
            ABC     DEF
```

Since the (Back space) character would erase the tab character the result in the workfile would be:

```
column no   1   5   9T
            ABCDEF
```

Limitations

None.

Options

None.

Example

```
/TAB
USE TABS?y
Tab Char:@
    '@' OK?y
Indicate positions with a 'T'
          1234567890123456789012345678901234567890 ...
                   T
```

This would result in tabbing being enabled with '@' being the tab character and with a single tabstop set at column 10.

Related commands

SET
VERIFY

TERMCAP

Purpose

To determine the maximum setting of the SET SCREENMAX parameter for a terminal.

Form

```
TERMCAP
```

Description

This command does a hard reset on the terminal and then sends a series of three character length lines to the terminal. It then homes the cursor and reads the top line to determine the capacity of the terminal. The capacity is then displayed.

Limitations

For use on HP 264X terminals only.

Options

None.

Related commands

SET
SCREEN
TERMSTAT

TERMSTAT

Purpose

To display the strap settings for HP 264x series terminals.

Form

```
TERMSTAT
```

Description

The TERMSTAT command clears the screen and tells how straps A through H are currently set.

Limitations

For use on HP 264x series terminals only.

Options

None.

Related commands

SET
TERMCAP

TEXT
T

Purpose

To copy material from a disc file into the workfile.

Form

```
TEXT filename [filerange] [,CODED]
                          [,UNN  ]
```

Description

The TEXT command copies all or part of a permanent file into the workfile. The current workfile must be clean (ie. not modified since the last TEXT or KEEP). The file can be input with or without the already existing sequence numbers. The material being copied is not listed on the terminal.

For Numbered Files: If no option is specified the TEXT command will assume the input file is numbered and will use a portion of each record of the input file as a sequence number. TDP inspects the first three records of the file; if a character other than a digit is found in the sequence number field of the record, or if the number is out of sequence, TDP assumes the file is unnumbered.

For Unnumbered Files: If the input file is unnumbered, or the sequence numbers are to be used as part of the text, use the UNNUMBERED option. The subsequent KEEP will be unnumbered unless the NUMBERED option is chosen.

Files of type COBOL or DIARY, and PMARKED files, are identified by file type and automatically handled in the required manner. Crunched files may be texted directly; they are automatically uncrunched.

If you are texting over an existing workfile and that workfile was KEPT with the CODED option. TDP asks if you want to write over the old workfile to ensure that the information in it cannot be read. There will be a delay while this process takes place since every record in the workfile is being overwritten.

Setting parameters automatically: Parameters can be automatically set for a file whenever it is TEXTed in by including one special line in the file. The first five characters must be *SET followed by a parameter list. The SET command must be uppercase and no semicolons can appear in the parameter list.

Limitations

None.

Options

[filerange] The filerange option allows you to TEXT a selected portion of the disc file into the workfile. The filerange can be specified by line numbers or record numbers. If no filerange is specified, the entire file is TEXTed.

[UNN] The UNN option texts the file into the workfile without sequence numbers. If the file was kept with sequence numbers, they will be shown as part of the text line.

[CODED] The CODED option allows you to decode a file that has been KEEPed with the CODED option (see the explanation under Options for the KEEP command). You must supply the "secret word" used when the file was KEEPed in order to decode the file. Note that you cannot text a portion of a file when using the CODED option.

Examples

Example 1: `/TEXT SAM, UNN`

The file SAM is assumed to be unnumbered, and is copied as the workfile. If there are sequence numbers, they will appear as part of each line.

Example 2: `/TEXT SAM #20/#30`

Record numbers 20 through 30 will be copied into the workfile (TDP knows records are desired because of the # sign preceding the record number).

Example 3: `/TEXT SAM 462/500`

Line numbers 462 through 500 of the file SAM will be copied into the workfile.

Related commands

KEEP
QUICKTEXT

TIME

Purpose

To display the current date and time.

Form

```
TIME
```

Description

The TIME command displays the current date and time in the same format used when you first run TDP: the day of the week, the date, the time, and the Julian day of the year.

Limitations

None.

Options

None.

Example

```
/TIME

TDP                FRI, JUN 10, 1988, 11:57 AM (DAY #161)
```

TOTAL

Purpose

To compute row and column totals for a range in the file.

Form

```
TOTAL[Q] [rangelist] [,APPEND]
                     [,NOTEXT]
```

Description

The TOTAL command computes the total of the tokens in each row, and in each column in the range of lines. Tokens which are not valid numbers are treated as zero. TOTAL works only on the portion of the line between LEFT and RIGHT; set these parameters to confine your calculations to the numerical table section of your document.

Note that a "$" will be included in the total if it was included for any of the numbers being totaled. Also note that commas are appropriately placed for greater readability.

TOTAL considers a number negative if it is preceded by a minus sign, or enclosed in parentheses, brackets, or <>.

Limitations

None.

Options

[rangelist] The rangelist option allows you to compute totals for a range of lines, or multiple ranges of lines.

[Q] or [NOTEXT] The Q or NOTEXT option suppresses listing of all output as the totals are computed.

[APPEND] The APPEND option causes row totals to be entered at the right-hand section of each row of text, and a line to be inserted at the bottom of the range of lines with column totals and the grand total. You will need to use the ALIGN command to align this inserted line.

TOTAL

Examples

```
/TOTAL 4/7, APPEND,NOTEXT
```

This command totals the rows and columns in lines 4 through 7 and inserts row and column totals. Suppose that lines 4 through 7 are as follows:

```
4        172.50      6672.89       135.35
5         99.67      4327.98       531.59
6        696.13      1976.23       334.66
7        461.10      2331.68       251.96
```

After the command above is executed, the lines look like this:

```
4      172.50       6672.89        135.35 6,980.74
5       99.67       4327.98        531.59 4,959.24
6      696.13       1976.23        334.66 3,007.02
7      461.10       2331.68        251.96 3,044.74
8      1,429.40 15,308.78 1,253.56 17,991.74
```

Line 8 contains column totals, and the grand total as the token farthest to the right. The right-hand column of lines 4/7 contains the row total for each row. Now we use /ALIGND ALL to align the tokens as shown below:

```
4      172.50   6672.89    135.35  6,980.74
5       99.67   4327.98    531.59  4,959.24
6      696.13   1976.23    334.66  3,007.02
7      461.10   2331.68    251.96  3,044.74
8    1,429.40 15,308.78  1,253.56 17,991.74
```

Related commands

ALIGN
DELIM
RACK
SET
=TOTAL

UNCRUNCH

UNCRUN

Purpose

To restore "crunched" files to their original form.

Form

```
UNCRUNCH filename
```

Description

The UNCRUNCH command is used to restore files that have been "crunched" to their original form.

Limitations

The file is assumed to contain only printable characters. Any character in the file which is not part of the normal printable character set is translated into a single code. An error message is sent at the time the file is crunched. (Examples of non-printing characters are the Back space and the Tab key. If a file contained both, the crunching process would translate the two into the same code. The uncrunch process would then produce a file different from the original.)

Note that files must be uncrunched before FINAL or DRAFT commands are issued. (TEXT and PRINT will automatically handle crunched files.)

The effect of uncrunching a file that is not crunched is undefined.

Options

None.

Examples

```
/UNCRUNCH SAM
```

The dialogue is shown below:

```
SAM crunched on JUN 10, 88
Uncrunch to original filename?  (answer "yes" if you want to
                                 return the file to its original
                                 filename; if you answer "no",
                                 you will be asked for the
                                 filename you wish to use.)

OVERWRITE OLD FILE?             (answer "yes" unless you made a
(asked whenever a                mistake answering the previous
file already exists              question; if you answer "no", you
with the name you have           will be asked for another filename.)
specified.)
```

Related commands

CRUNCH

UPSHIFT

UPS

Purpose

To shift all alphabetic characters to upper case.

Form

```
UPSHIFT[Q] [rangelist] [,ADJUST] [,NOTEXT]
```

Description

The UPSHIFT command changes all lower case alphabetic characters in the specified range to upper case. The range can be defined in terms of line numbers or starting with the occurrence of one search string and ending at the occurrence of another; you can upshift one word by the command /UPSHIFT "word", or multiple words by the command /UPSHIFT "word1", "word2", "word3"... If no range is specified, all words in the current line are upshifted. The command observes left and right margin settings.

Non-alphabetic characters are not affected by this command.

Limitations

None.

Options

`[rangelist]` The rangelist option defines the range of lines over which lower case characters are to be shifted into upper case.

`[Q] or [NOTEXT]` The Q or NOTEXT option suppresses the listing of the lines being upshifted.

`[ADJUST]` The ADJUST option causes the first letter of each word to be left in lower case.

Examples

Example 1: `/UPSHIFT 7/11`

This example upshifts all alphabetic characters in lines 7 through 11. The lines will be listed as the shifting is carried out.

Example 2: `/UPSHIFT "The"/"procedure.",NOTEXT`

This command searches from the current line pointer to the first occurrence of the word "The". The upshifting begins there, and continues until the word "procedure." is found. The lines will not be listed.

Related commands

DOWNSHIFT
CHANGE
MODIFY

USE
U

Purpose

To use a pre-stored file of TDP commands.

Form

```
USE[Q] filename [,NOTEXT]
```

Description

The USE command is used to access a file containing a list of TDP editing commands. The file consists of a collection of commonly used commands. That series of commands is later executed when the filename is referenced in a USE command. Thus the USE command simplifies many repetitious tasks; it also allows an experienced user at your site to incorporate more advanced commands for use by less experienced users. The processing of USE files may be modified by SET CONTINUE, SET MISSOK, SET MUST and SET STOP.

USE files can contain USE commands that access other files; the result is a nested command sequence. If you use commands in a USE file which require input to continue (interactive commands), the USE file will wait for input at that point.

Limitations

You cannot include formatter commands in a USE file.

If there is any error in any of the commands in the USE file, the USE command is terminated and any unexecuted commands in the USE file are ignored. To continue after an error use SET CONTINUE. You can test the value of %ERROR (see Chapter1, **Search facilities with strings**) to prevent looping.

You cannot exit from TDP from within a USE file. To achieve this effect, run TDP passing an INFO string containing the EXIT command, for example:

```
RUN TDP.PUB.SYS;INFO="USE usefile;EXIT"
```

Options

`[Q] or [NOTEXT]` If Q or NOTEXT is used, output from commands within the USE file is reduced; specifically, summary output from JOINQ, DELETEQ, and CHANGEQ is suppressed.

With the exception of Q and NOTEXT, no specific options are available with this command. However, the @GO and @IF statements can be used within the USE file to allow branching within the file.

The @GO statement jumps to a specified record number or label. The @IF prints a string on the terminal and branching occurs if the answer is "YES". The @IF statement contains a string, and a record number or label. When the statement is reached, the string is printed on the terminal, and the branching takes place if you answer "YES" to the string. A variation, the @IF NOT statement operates identically except that branching occurs on a "NO" response. One other variation, @IF FILE, interprets the string as a filename and attempts to open the file for READ access. If the file can be read, the branching occurs. (See the @GO and @IF commands later in this Section.)

USE

U

Examples

Example 1: `/USE MAILADD`

Suppose that the file MAILADD contains the following commands:

```
QQ "Name"
Addsingle
QQ "Address:"
Addsingle
QQ "City,State:"
Addsingle
QQ "Zip Code:"
Addsingle
```

The command /USE MAILADD will initiate the execution, in sequence, of all these commands.

Example 2: `/USE EXAMPLE`

A more complicated USE file, including use of ZP : :, Z : :, and @IF and @GO; and showing an embedded USE command, is shown below:

```
 1    V DEPTH
 2    Q"LINE ONE"
 3    @IF "GO TO LINE ONE?" THEN GO TO 1
 4    @IF "GO TO L23?"  THEN GO TO L23
 5    ZP::="File Name:"
 6    @if file "Z::" L99
 7    q"file is absent"
 8    @go L23
 9    @L99 q"file present"
10    @L23 Q"THIS IS LABEL 23"
11    @IF "GO TO TOP?" GO TO 1
12    @IF "RECURSE?" GO L12
13    @IF NOT "Stop?" GO 1
14    END
15    @L12 Q"LABEL L12"
16    U EXAMPLE
17    QQ"RETURNED TO";V DEPTH
18    @GO L23
```

Related commands

@GO
@IF
@L
@S
SET
Q

VERIFY
V

Purpose

To verify certain parameter settings and obtain other information about the workfile.

Form

```
VERIFY

     [ACTIVE      ]     [HOLD        ]     [PROMPT      ]
     [ALL         ]     [LANGUAGE    ]     [QUIET       ]
     [AM          ]     [LEFT        ]     [RESTRICTIONS]
     [BATCHERROROK]     [LENGTH      ]     [RIGHT       ]
     [BIGBLOCK    ]     [LINE        ]     [SCREENMAX   ]
     [BLIT        ]     [LIT         ]     [SESSQUIET   ]
     [CLEAN       ]     [LOCPARM     ]     [SHORTERROR  ]
     [COLSTOP     ]     [LOWC        ]     [SIZE        ]
     [CONFIG      ]     [LOTS        ]     [SM          ]
     [CONTINUE    ]     [MISS        ]     [SPOOLER     ]
     [DELTA       ]     [MUF         ]     [STAMP       ]
     [DEPTH       ]     [MUST        ]     [STOP        ]
     [DICTIONARY  ]     [NO          ]     [TAB         ]
     [DISPLAY     ]     [NONLIT      ]     [TERMINAL    ]
     [ERROR       ]     [NOTAB       ]     [TERMTYPE    ]
     [ESCAPE      ]     [PERMYES     ]     [TIME        ]
     [FATHER      ]     [PMARK       ]     [TOTAL       ]
     [FILE        ]     [PNEWS       ]     [VBIGBLOCK   ]
     [FILETYPE    ]     [POINTER     ]     [WINDOW      ]
     [FORMAT      ]     [POSTADD     ]     [YES         ]
     [HIGHC       ]     [PRIORITY    ]     [ZFILL       ]
```

Description

The VERIFY command is used to determine current settings of certain parameters, and to obtain other information about the workfile. For a more detailed discussion on those parameter which can be set, refer to the SET command. The form VERIFY ALL displays the more commonly used parameters. The form VERIFY LOTS displays all parameters available.

Options

1 ACTIVE - gives the number of lines currently in the workfile.

2 AM - to determine if you have Account Manager capability.

3 BATCHERROROK - displays the current setting of the BATCHERROROK flag.

4 BIGBLOCK - shows whether or not the internal blocking factor has been doubled.

5 BLIT - displays the current setting of BLIT.

6 CLEAN - tells whether the workfile has been modified since the last KEEP or TEXT command.

7 COLSTOP - lists the current settings of the column stops.

8 CONTINUE - displays whether the system is set to continue when an error is found in processing a use file.

9 CONFIG - performs consistency checking on the TDP configuration file TDPCONFG.TDPDATA.HPOFFICE.

10 DELTA - tells the current default increment for line numbering.

11 DEPTH - lists the current USE file level. If, within a USE file, another USE file is opened, the DEPTH parameter is incremented by 1. This parameter is only useful when used from within a USE file.

12 DICTIONARY - shows the current user dictionary.

13 DISPLAY - tells whether lines being added, changed, etc. are listed by default.

14 ERROR - lists all error messages with error numbers.

15 ESCAPE - identifies the "escape" character used with the ADJUST option of the ALIGN command.

16 FATHER - tells what process called TDP. The Job/Sess # is displayed if TDP was invoked from the command interpreter, or User Process # if TDP was called from a program.

17 FILE - lists the temporary K-filename given to the workfile and the name of the file currently being processed.

18 FILETYPE - identifies the file code of the current work file.

19 FORMAT - lists the current format of the workfile: DEFAULT, COBOL, or DIARY.

20 HIGHC - lists the upper column boundary.

21 HOLD - lists the number of lines in the hold file.

22 LANGUAGE - shows which language dictionaries are are currently in use (BRITISH or AMERICAN).

23 LEFT - lists the current setting of the left margin.

24 LENGTH - lists the currently-set line length.

25 LINE - lists the current setting of the line size on the terminal.

26 LIT - lists the current setting for type of search: literal, or non-literal.

27 LOCPARM - shows whether the System Manager has granted the ability to set group parameters.

28 LOWC - lists the lower column boundary.

29 MISS - indicates whether or not an error message will be generated if a string is not found during a search. If MISS is TRUE, an error message will not be generated. It is the opposite of MUST.

30 MUF - displays the setting of MUF (Monitor Use File).

31 MUST - indicates whether or not an error message will be generated if a string is not found during a search. If MUST is TRUE, an error message will be generated. It is the opposite of MISS.

32 NO - shows the current status of the NO flag.

33 NONLIT - lists the current setting for type of search.

34 PERMYES - displays the current state of the PERMYES flag.

35 PMARK - tells whether proofmarking is set ON.

36 PNEWS - displays the setting of PNEWS. If it is set, then TDP will display the welcome message (if the file TDPNEWS.TDPDATA.HPOFFICE exists).

37 POINTER - shows the location of the current line pointer in the file.

38 POSTADD - shows how often the disc is updated when adding lines to the workfile.

39 PRIORITY - displays the current priority settings as established through the SET command.

40 PROMPT - identifies the prompt character.

41 QUIET - tells whether standard output by TDP is being suppressed.

42 RIGHT - lists the current setting of the right margin.

43 RESTRICTIONS - the SET parameters NOCOMP, NOMPE, NORUN and NOSTREAM provide a method of preventing compilations, MPE commands, running programs and streaming jobs from within TDP. The VERIFY RESTRICTIONS command displays the current setting of these parameters together with the setting of SPOOLER.

44 SCREENMAX - shows the current setting of SCREENMAX

45 SESSQUIET - tells whether messages from other users or the operator will be accepted. If SESSQUIET is true, they will not be.

46 SHORTERROR - displays the current setting of SHORTERROR. If it is set, then the user will only get an error number if an error occurs; the error message is suppressed.

47 SIZE - shows the size of the workfile.

48 SPOOLER - shows the current setting of the SPOOLER flag and whether the spooler is running.

49 SM - shows whether or not you have System Manager capability.

50 STAMP - shows the currently-set proofmark stamp character.

51 STOP - tells whether the current setting is to stop processing when an error is found in processing a use file.

52 TAB - displays the currently-set tab stops, the tab condition, and the assigned tab character.

53 TERMINAL - shows the current terminal setting used by TDP.

54 TERMTYPE - shows the MPE terminal type.

55 TIME - tells the number of CPU processing seconds used by TDP so far this session.

56 TOTAL - gives the number of lines currently active in the workfile.

57 VBIGBLOCK - shows whether or not the internal blocking factor has been quadrupled.

58 WINDOW - shows the current number of lines used for a "window".

59 YES - shows the current state of the YES flag.

60 ZFILL - shows the current state of the ZFILL flag.

Related commands

DISPLAYPARMS
GETPARMS
SET
STOREPARMS

WORDMOVE

WM

Purpose

To move tokens around on a line.

Form

```
WORDMOVE[Q] tokenrange,position [,rangelist]
```

Description

The WORDMOVE command moves one or more tokens from one position on a line to another position on the same line. (A token is a collection of non-blank characters preceded and followed by a blank.) If no rangelist is specified, only the current line is affected.

The tokenrange defines the tokens that are to be used. The position specifies their destination. Commas may be inserted between the three parameters.

If position zero is used as the destination for a tokenrange, the tokens in that range will be deleted.

Limitations

None.

Options

`[Q]` The Q option suppresses the listing of the line with the tokens moved.

`[rangelist]` The rangelist specifies the lines in which the move is to be made.

Examples

Example 1: `/WORDMOVE 5,2,8`

This example moves the fifth token to the second token position in line 8. If line 8 was:

```
8  This is a test case to experiment with WORDMOVE
```

then it would now read:

```
8  This case is a test to experiment with WORDMOVE
```

Example 2: `/WM 5,0,33`

This example eliminates the fifth word or token in line 33. If line 33 was:

```
33 This development represents the very worst
```

it will now read:

```
33 This development represents the worst
```

Related commands

CHANGE
COLMOVE
DELETE
MODIFY

Z::

Purpose

To assign a value to Z: : to be used in place of a string in a command line.

Form

```
Z[Q]::=
ZP::= "prompt"
Z::%name
ZS::="string"
```

Description

When the command Z: := is issued, TDP prompts you with "ENTER Z: :=" at which time you enter a string. The system assigns the string you enter to the code Z: :. Thereafter, you can enter Z: : in a command line and TDP substitutes the string for it. This command is useful in setting a target range for repetitive operations (see Example 1). The string assignment is terminated by another Z: : command, or by exiting from TDP. The current value of Z: : is displayed by the command Q"Z: :".

The variation ZP: :="prompt" of this command can be used within a USE file to prompt the user for the input string at the time the USE file commands are executed. The prompt that you supply replaces the default prompt "Enter Z: :".

The ZS variation allows you to set the value of the string from the command line with no prompt.

The construct Z: :%name where %name is a special search string (see Chapter 1) allows the value of the special search strings to be used in a command line. (This is especially useful in USE files). Wherever Z: :%name appears in the command line it is replaced by the value of the special search string %name. This use of Z: : does not affect the value stored by the constructs above.

The effect of Z: :%name may be modified by the SET parameter ZFILL. When ZFILL is OFF (the default) Z: :%name will be replaced in a command line by the value of the referenced special string. When ZFILL is ON the Z: : will be replaced in a command line by its current value and whatever follows the Z: : is treated as textual input to the command line.

Limitations

The string can be a maximum of 72 characters long.

Options

[Q] The "Q" option eliminates the prompt "ENTER Z: :=".

Examples

Example 1:

```
/Z::=
      ENTER Z::=5/8,13/30
```

In this example, you type in "Z: :=". The system responds with "ENTER Z: :=" at which time you enter "5/8,13/30". Thereafter, when you want to LIST, MODIFY, CHANGE, or so forth for those ranges, you need only enter the following commands:

```
/LIST Z::
/MODIFY Z::
/CHANGE Z::
```

Example 2: `/ZP::="Name"`

When this command is executed, TDP responds with the prompt "Name" rather than "ENTER Z: :=". This allows you to create your own prompt, thus better recognize (in a USE file for example) what is being requested.

Example 3: `/FN Z::%X`

If this command is executed with ZFILL OFF (the default) the command will be interpreted as /FN n (where n is the current value stored in the X buffer of TDP's calculator) and TDP will find the line in the workfile whose linenumber is n. With ZFILL ON and assuming nothing had by assigned to Z: : using the Z: := construct the command would be interpreted as \FN %X and TDP would find the next occurrence of the string "n" in the workfile.

Related commands

SET
Q
@
@M
@D
@F
@S

@

Purpose

To repeat the previous command line a specified number of times.

Form

```
@[n]
 [Z]
```

Description

The @n command repeats the last entered command line n times. If n is not specified, the previous command is repeated once. @ must appear in column 1.

Limitations

This command should not be used if the command line to be repeated is longer than 80 characters, since the 81st character onwards will not be repeated and syntactic errors may result.

The @ command cannot be used on a line with other commands.

Options

[n] The n option specifies the number of times the command is to be repeated (up to a maximum of 999).

[Z] If the Z option is specified, the command is repeated Z times, where Z is the result of the last calculation made in calculator mode.

Related commands

REDO
Z: :
@M
@D
@F
@S

@D

Purpose

To display a command line stored in one of the command buffers.

Form

```
@D [A]
   [B]
   [C]
```

Description

The @D command is used to display the command line stored in one of the three command buffers (A, B or C).

Limitations

Only 80 characters are stored; if the command line is longer than 80 characters, the line is truncated.

The @D command cannot be used on a line with other commands.

Options

[A]/[B]/[C] The A/B/C options specify which buffer is to be displayed. If none is specified, buffer A is assumed.

Related commands

Z: :
@
@M
@F
@S

@F

Purpose

To fetch and execute a command line stored in one of the command buffers.

Form

```
@F [A [M] ]
   [B [M] ]
   [C [M] ]
```

Description

The @F command is used to fetch the command stored in one of the buffers and execute it immediately.

Limitations

The @F command cannot be used on a line with other commands. An @GO command can be stored but not executed.

Options

[A] / [B] / [C] The A/B/C options allow you to specify which buffer holds the command you want fetched. If none is specified, buffer A is assumed.

[M] You can modify the command before executing it by using the M option.

Related commands

Z: :
@
@D
@M
@S

@GO

Purpose

To provide unconditional branching within a USE file.

Form

```
@GO[TO] {record number}
        {label}
```

Description

The @GO command is used in USE files to unconditionally branch to another line in the file. The destination of the branch can be a record number or a label assigned with @L command. @GOTO is an alternate form of the of the command.

A branch to a record number is more efficient than a branch to a label.

Limitations

No other command can appear on the same line as an @GO.

Options

None.

Related commands

USE
@IF
@L

@IF

Purpose

To provide conditional branching within USE files.

Form

```
@IF [NOT]{FILE "filename"} THEN GO TO {label          }
         {string         }            {record number}
         {x/y/z  </=/>  n}
```

Description

The form @IF FILE filename branches if a file with the name "filename" can be opened for read access.

The form @IF "string" is used to branch if a "yes" answer is given when "string" is printed on the terminal.

A relational condition can also be specified comparing one of the TDP calculator registers (x, y or z) to a numeric value. For example: IF x>=0 THEN GOTO {label}

The destination can be specified by a label (see @L command) or a record number in the USE file.

Limitations

No other command can appear on the same line as the @IF command.

The maximum value of n is 32767.

Options

[NOT] The NOT option reverses the operation of any of the conditionals.

Examples

Example 1: `@IF "Do you want to try again?"  THEN GO TO L10`

This command prints the string on the terminal and, if the answer input is "yes," branches to the line labeled L10 in the USE file.

Example 2: `@IF NOT "Do you want to stop?"  THEN GO TO 1`

This command asks if you want to stop, and branches to record number 1 in the USE file if the answer is "no".

Example 3: `@IF FILE "DATA" THEN GO TO 15`

This command tries to open a file named DATA and, if it succeeds, branches to record number 15 in the USE file.

Related commands

USE
@GO
@L

Purpose

To label a command line in a USE file.

Form

```
@Ln
```

Description

The @L command is used to assign a label to a command line in a USE file. The label is used to identify that line as the destination for either conditional or unconditional branching operations.

Limitations

The number attached to a label cannot exceed five digits.

The "L" must be upper case.

Every time a GO (label) command is issued, the USE file is searched for that label. Once the USE file is checked out, the labels should be replaced by record numbers for greater efficiency.

Options

None.

Examples

```
10  @L16 FINDNEXTQ "TDP"
20  ADDLINE *+2
30  GOTO L16
```

This example assigns the label L16 to the command line FINDNEXTQ, adds a blank line two lines after the found line, and loops back to find the next line containing the search string.

Related commands

USE
@GO
@IF
@S

@M

Purpose

To modify and execute the previous command line.

Form

```
@M
```

Description

The @M command displays the previous command line and allows you to use the sub-commands associated with the MODIFY command to make modifications to that command (with the exception of the S sub-command). The modified command is then executed.

Limitations

@M should not be used to modify command lines greater than 80 characters in length.

No other commands can be given on the same line.

Options

None.

Example

```
/@M
```

Suppose your last command was /FINDQ "message". The @M command will output that command and ask for changes as follows:

```
          FINDQ "message"
Changes:        r"proprietary"

          FINDQ "proprietary"
Changes:  (carriage return signals that modification is complete)
```

When the modification of the command is complete, the new command is executed.

Related commands

REDO
Z: :
@
@D
@F
@S

Purpose

To store a command line in a buffer.

Form

```
@S[A [= command string]]
  [B [= command string]]
  [C [= command string]]
```

Description

TDP has three buffers which may be used to store command lines. These buffers are called A, B, and C. The @S command is used to store a command list into one of these buffers.

Limitations

Each buffer can store a maximum of 80 characters from the command line.

The @S command cannot be used on a line with other commands.

Options

[A]/[B]/[C] If only a command buffer is specified (A, B or C), the last command string executed is stored in the specified buffer. If no buffer is given, the "A" buffer is assumed.

[command string] If a command string is to be stored, a buffer must be specified. Note that the storing of a string with an @S command does not immediately execute it.

Example

```
 /@SA=TEXT MYFILE;LIST ALL
```

The command line "TEXT MYFILE;LIST ALL" is stored in buffer A without being executed. An @FA command will cause the file MYFILE to be TEXTed in and listed.

Related commands

USE
@D
@F

=

Purpose

To invoke calculator mode.

Form

```
=[x[= {arithmetic expression}]]
=[y[= {arithmetic expression}]]
=[z]
```

Description

The = command is used to perform calculations during a TDP job or session. Any arithmetic expression containing additions, subtractions, multiplications, divisions, and square roots can be evaluated. When the result is greater than 3 digits, commas are inserted for easier reading. The result of a calculation can be printed immediately or stored in an x or y register for later use. The result of the last calculation is stored in z if x or y is not specified. Once a value is placed in x,y, or z, it remains the same until it is replaced by another value, or until the TDP session is over.

Arithmetic expressions are made up using the symbols ( ) + - * / ie. brackets, addition, subtraction, multiplication and division. The operator DIV is used to specify integer division, and the operator SQRT is used to specify the square root function.

A dollar sign in front of a constant will result in a dollar sign being printed in front of the answer.

You may enter arithmetic expressions containing decimal, octal, or hexadecimal constants. Octal values are denoted by a % sign, and hexadecimal by the letter H preceding the constant. The results are given in octal or hexadecimal, respectively, and also in decimal.

The effects of the command can be summarized as follows:

1 =x
=y
=z

Displays the contents of the x, y and z registers, respectively.

2 =x={expression}
=y={expression}

Stores the results of the expression in the x and y registers, respectively.

3 ={expression}

Stores the results of the expression in the z register.

Limitations

None.

Options

=[x]/[y]/[z] To list the contents of one of the registers use the commands =x, =y, =z as appropriate.

Examples

Example 1: `/=y=SQRT(164)`

This command accesses the square root function, computes the square root of 164, and stores the answer in the y register.

Example 2: `/=HA2C+ %21 + 100`

This command converts the three constants to a common number base, adds them together, then prints the answer in all three number bases:

```
    2,721 (5241 OCTAL)(AA1 HEX))
```

Example 3: `/=SQRT(x+y)/(4*1.37)`

This command computes the square root of the sum of the x and y registers, then divides it by the product of 4 times 1.37. Note that x and y may be used within expressions as shown here.

Related commands

=L
=M
=S
=TOTAL

=C

Purpose

To perform calculations with stored expressions.

Form

```
=C
```

Description

The =C Command performs the calculations stored with =S. The =C command may be used in conjunction with the "@" command to handle repeated calculations using x, y and z.

Limitations

The expression must not reference tokens.

Options

None.

Examples

```
=Qx=0
=S=x=x+1;x**;x*x*x
=C
@2
```

This expression gives the result:

```
1 1 1
2 4 8
3 9 27
```

Related commands

=
=M
=S
=L

=L

Purpose

To list a stored expression

Form

```
=L
```

Description

The =L Command lists the expression stored with =S. (See =S for more details.)

Limitations

None.

Options

None.

Examples

```
/=L
```

This example lists the stored expression:

(2*<2>) + <3>

Related commands

=
=M
=S
=TOTAL

=M

Purpose

To modify a stored expression.

Form

```
=M
```

Description

The =M command is used to modify a stored expression. (See =S for more details.) The conventions are identical to those used with the MODIFY command (except that the S sub-command is not allowed).

Limitations

None.

Options

None.

Example

```
/=M
```

This example lists the stored expression and prompts you for modifications:

```
        (2*<2>) + <3>   (2*<2>) + <3>
Changes:
```

Related commands

MODIFY
=
=L
=S
=TOTAL

=S

Purpose

To store an expression for later use.

Form

```
=S[=expression]
```

Description

The =S command is used to store an arithmetic expression for use with =TOTAL. If you do not supply an expression, TDP will prompt you for it. Tokens in pre-set columns can be referenced and used in the expression; the columns are denoted by being enclosed in angle brackets (<>).

Limitations

None.

Options

None.

Examples

Example 1: `/=S=2*(153 + Z)`

This example stores the expression which adds 153 to the result of the last calculation, and multiplies it by 2.

Example 2:

```
/=S
     ENTER EXPRESSION:  (2*<6> + <3>)
```

This example stores the expression which multiplies token 6 by 2, then adds the value in token 3.

Related commands

=
=L
=M
=TOTAL

=TOTAL

Purpose

To apply a stored expression to the contents of a line.

Form

```
=TOTAL[Q] rangelist [,APPEND]
                    [,NOTEXT]
```

Description

The =TOTAL command is used to apply a prestored expression to a range of lines. This would generally be used only with tables of numeric or financial data.

Limitations

None.

Options

`[Q]` or `[NOTEXT]` The Q or NOTEXT options suppress the output listing on the screen, from the =TOTAL command.

`[APPEND]` The APPEND option adds the result of the calculation to the end of each line. A line containing the column totals is also added to the workfile automatically.

Examples

Suppose lines 3, 4, and 5 of your workfile contain the following information on one month's budget and actual expenditures:

```
 2       Item      Budget       Actual      Projected FY
 3     Salaries  30,696.13    35,160.15
 4     Benefits    3069.61      3751.08
 5     Expenses  55,113.00    52,995.82
```

To total both rows and columns, and align the results with the rest of the table, use the following commands:

```
/ =S= 12*<2>
/ SET LEFT = 10
/ =TOTAL 3/5, APPEND
```

Now insert the word "Total" at the beginning of line 6, and use the ALIGN command to align that line. The result is shown below:

```
 2     Item      Budget       Actual        Projected FY
 3     Salaries  30,696.13    35,160.15     421,921.80
 4     Benefits   3,069.61     3,751.08      45,012.96
 5     Expenses  55,113.00    52,995.82     635,949.84
 6     Total     88,878.74    91,907.05   1,102,884.60
```

Related commands

=
=L
=M
=S

Formatting commands

Introduction

TDP is a command-driven text processing system. There are two types of commands: the first, described in Chapter 3, are commands entered on a command line and executed immediately. These are called editing commands. Examples are TEXT, ADD, MODIFY, and so on. The second group of commands, called formatting commands, are described in this chapter. Formatting commands are entered on a separate line in the work file, with a backslash in column 1, and retained as a permanent part of the document. They are executed only when output is requested with a DRAFT or FINAL command. (A small group of formatting commands may be entered anywhere on a line, these are the intraline commands. They are described at the end of this chapter.)

Command format

The conventions used to present the form of each command are described in Chapter 2. Briefly, they are as follows:

1 Parameters are shown from left to right in their required order of entry.

2 Parameters are required unless enclosed in brackets [ ]. The order shown for optional parameters is not important with regard to the entry order.

3 When only one of two or more parameters must be selected, the choices will be displayed one below the other and enclosed in braces { }.

4 Commands can be entered in upper or lower case, or any combination thereof.

For detailed information on terms used in the command descriptions, please see Chapter 2.

Within the formatter commands the following rules apply to the processing of options or parameters.

1 Numbers, strings and the keywords ON and OFF may be preceded by an equals sign.

2 Strings are delimited by matching special characters. However, if an equals sign is to be used as a special character, then the leading optional equals sign must be included.

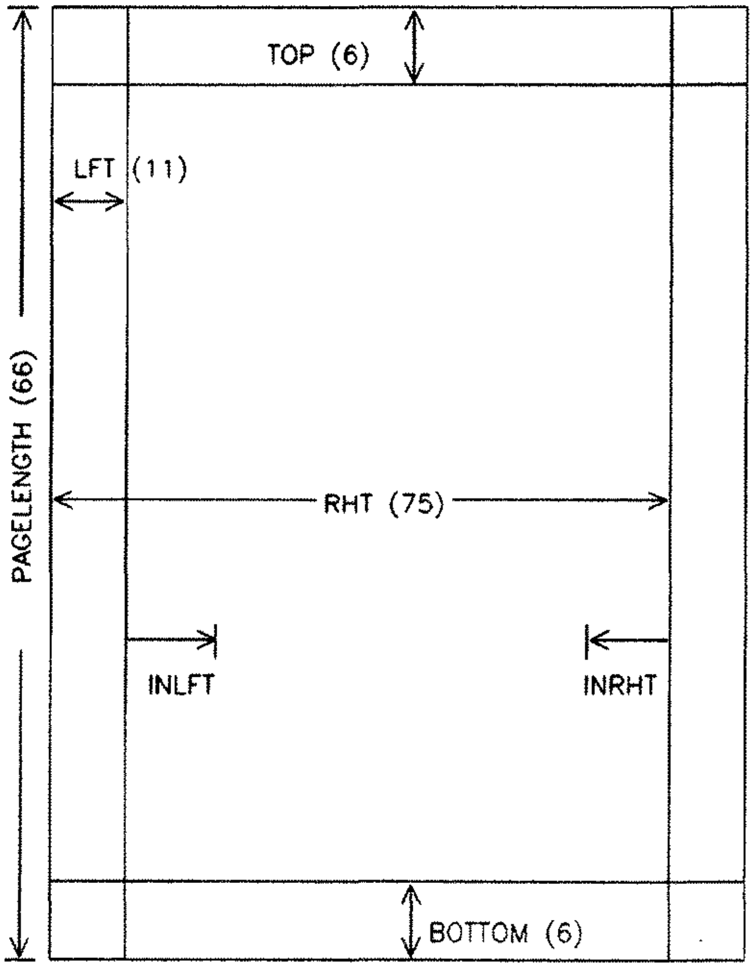
3 A "char" is a string with only one character in it.

4 All options and parameters are shown in their shortest form and may be followed by alphabetic characters, which will be ignored.

5 No commands may follow, on the same line, any command with a BEGIN or END parameter.

Default page parameters

Unless directed otherwise by means of formatting commands, TDP formatter will set default page layout parameters. The diagram below illustrates the default page layout for a 10 pitch device.



ACTIVATE

ACTIV

Purpose

To allow printing on one or more logical pages on a HP 2680 or HP 2688 laser printer.

Form

```
\ACTIVATE [(] n [,n [,n [...]]] ()]
```

Description

The ACTIVATE command is used to add a number of HP 2680 or HP 2688 logical pages to the list of active logical pages which will be used for printing. The number "n" is the number associated with the logical page in the environment file.

The order of use of these logical pages per physical page is from lowest to highest, according to the number of the page. Thus the lowest numbered active logical page is selected on a new page. At the end of a logical page the next highest numbered active logical page is selected. If none exists, a new physical page is started.

Limitations

This command is only valid when used with a HP 2680 or HP 2688 laser printer.

The ACTIVATE and DEACTIVATE commands are ignored until output begins. So, if the ACTIVATE command appears before the first line of text in a file, it will be ignored.

Options

None

Examples

Example 1: `\ACTIVATE 5`

This will add logical page 5 to the list of active logical pages. If logical pages 3 and 12 are currently active, then the next physical page will use logical pages 3, 5 and 12, in that order, to print the text.

Example 2: `\ACTIVATE 5,11,17,22,3,7`

This will include pages 3,5,7,11,17 and 22, in the list of active logical pages. The pages will be used in that order on the next physical page, along with the other active logical pages. If the user is currently using logical page 15, and no higher logical page is active, pages 17 and 22 will be printed on the same physical page as that being written. Printing will then continue on a new physical page.

Related commands

DEACTIVATE
LAYOUT

ALTERNATE
AL

Purpose

To alternate page numbers, headings, footings and sections between the left and right-hand sides of the page.

Form

```
\ALTERNATE [ON ]
           [OFF]
           [REV]
```

Description

The ALTERNATE command places page numbers, headings, footings and sections properly for two-sided printing. The page numbers etc. are placed on the right-hand side of the page for odd-numbered pages and on the left-hand side for even-numbered pages. The last character of the page number is used to determine whether the page is odd or even numbered. For alphabetic characters A,C,E,... are considered as odd, while B,D,F,... are considered as even.

If the setting of ALTERNATE conflicts with any previous occurrence of ALTERNATE or with any of the related commands, HEAD, HEADLINE, FOOT, PAGENO, PAGENOLINE or SECTION, the last occurrence of the command takes precedence.

Limitations

None

Options

[ON] The ON option turns on the command. It is normally used to turn on a command that has previously been turned off; if the command is issued without options set, it assumed to be ON.

[OFF] The OFF option turns off the effect of the command.

[REV] The REV option allows you to reverse the sense of the alternation. Thus, odd numbered pages will have their page numbers etc. placed on the left-hand side, and right-hand side for even-numbered pages.

Examples

This manual was formatted using the ALTERNATE command to control the positioning of page headings and footings.

Related commands

FOOT
HEAD
HEADLINE
PAGENO
PAGENOLINE
SECTION

BACKSPACE
BACK

Purpose
To define a backspace character.

Form
```
\BACKSPACE [char]
```

Description
The BACKSPACE command is used to define a character that will cause the printer to backspace (if possible) when a final or draft copy of the document is prepared. The character that is defined will always be interpreted as a backspace, until a different character is so defined or a BACKSPACE command is encountered with no character specified.

Limitations
The character you define must be a seven-bit character (a character from the ASCII character set).

Options
`[char]` The optional parameter char defines the character to be used as the backspace character.

Examples
```
\BACK = ">"
```

This example assigns > as the backspace character. When a draft or final copy is being prepared, any > found in the file will cause the printer to backspace once.

Related commands
BLANK
ESCAPE
PROMPT

BATCHERROR

Purpose

To set a limit on the number of errors that may occur during formatting in batch mode.

Form

```
\BATCHERROR n
```

Description

The BATCHERROR command defines an upper limit on the number of errors that may occur during formatting. The default is 1 error. Normally, the command should appear at the start of the file, but may be placed anywhere in the file and may be reset any number of times.

Limitations

The maximum value for BATCHERROR is 999.

The BATCHERROR command applies only to errors found in processing command lines; it does not apply to intraline command errors.

Options

None.

BLANK

BL

Purpose

To define a character as a necessary blank.

Form

```
\BLANK [char]
```

Description

The BLANK command is used to define a character as a necessary blank. When that character is entered as part of a line of text, TDP replaces it with a blank when a FINAL or DRAFT command is issued. This command might be used to prevent a phrase from being broken across lines when in format mode, or to prevent extra blanks from being inserted when in justify mode. When a draft or final copy of the document is prepared, real blanks are inserted for all occurrences of the character assigned with this command. The character is returned to normal usage when another character (or a real blank, ie. \BLANK with no parameter) is assigned with this command.

Limitations

The blank character is replaced on the line when the line is first read. Since ASK mode is not started until the first line of output is to be output, the BLANK command should not be entered in ASK mode if the first line of output is to contain necessary blanks.

The character you define must be a seven-bit character (a character from the ASCII character set).

Options

`[char]` The char option defines the character to be used as a necessary blank.

Example

```
\BLANK = "$"
```

This example assigns the dollar sign as a necessary blank. Thus, every time the dollar sign is encountered, a blank appears in the draft or final copy. This might be used to keep a title and name from being separated.

Related commands

BACKSPACE
ESCAPE
PROMPT

BOLD

Purpose

To print one or more lines in bold face print.

Form

```
\BOLD [n]
```

Description

The BOLD command prints the specified number of output lines in bold face print. The entire line is printed in bold face. (To print only certain words in bold face, see the ^B command.) If no parameter value is specified, only the next line is printed bold.

Limitations

This command will only be effective on those output devices which allow bold printing.

Options

[n] The parameter n specifies the number of output lines to be printed in bold typeface.

Example

```
\BOLD 3
```

This example prints the next three lines in bold face type.

Related commands

FONTEQ
GHOST
RED
^B
^G
^R

BOTTOM

BO

Purpose

To set the bottom margin size.

Form

```
\BOTTOM n
```

Description

The BOTTOM command sets the number of lines to be left between the last line of formatted text and the bottom of the page. The default setting is six lines.

The page number and footing are included in the space left by BOTTOM, so make certain that you provide enough room for them, or you will have an error message in the output.

Limitations

The value n must be less than or equal to the current pagelength setting minus the size of the top margin, minus two; and it must be greater than or equal to 0. (See PAGELENGTH and TOP for more details.)

Options

None

Examples

```
\BOTTOM 10
```

This example sets the bottom margin to 10 lines.

Related commands

LFT
PAGELENGTH
RHT
TOP

BOX

Purpose

To leave space for a figure in the text, and draw a box around it.

Form

```
\BOX n [string] [,ASAP] [,FULL]
```

Description

The BOX command leaves the specified number of lines for a figure, prints the "string" as a figure title, and draws a box around the space that has been left.

Normally, space is left for the figure at the bottom of the current page (if enough lines remain) or the top of the next page. The box extends between the current actual left and right margins (including INRHT and INLFT adjustments) at the time the box command occurs. The box is the specified number of lines deep. One blank line is left above the box, and two below (one for the title; the title is placed on the second of the two lines). If there is not room on the current page, the rest of the page is filled with text that follows the BOX command in the file.

If a second BOX or FIGURE command is encountered before the first has been printed, then to preserve the order within the document, the first BOX or FIGURE will be printed. Since only one BOX or FIGURE is placed at the bottom or top of any page, and insufficient space may exist for the first BOX or FIGURE on the current page, blank lines may be left on the current page.

This command does not cause a paragraph break; so may be introduced into a paragraph without affecting it adversely. This means that the BOX command can be introduced in the middle of the paragraph making reference to it, immediately following the reference.

Limitations

The label string is limited to 60 characters.

If a box is requested which is larger than a full page, the largest possible box which will fit on a page is printed.

Boxes are not allowed in headers or footers.

Options

`[ASAP]` The ASAP option causes the box to be placed immediately (if there is room), rather than waiting until normal placement at the bottom of the page; if there isn't room, it is printed at the top of the next page (in which case text is balanced in the normal way).

`[string]` The string option specifies the figure title to be printed. It will be included in the Table of Figures, if that file is being prepared.

`[FULL]` The FULL option causes the box to be placed between the page margins, rather than the current left and right margins. This is used in multi-column mode only when the box is to fill the full width of the page rather than the width of the current column.

Examples

Example 1: `\BOX 10 "Figure 7. This is a sample box."`

This example draws a box around ten blank lines and prints the figure title underneath the box.

```
-------------------------------------------------------------------------------
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
|                                                                             |
-------------------------------------------------------------------------------
```

Figure 7. This is a sample box.

Example 2: `\PARAGRAPH;BOX 12 "Figure 8.  This is a second sample.", ASAP`

This example will cause a paragraph break and output the box if there are seventeen or more lines left on the page. If there are less than seventeen lines the BOX will be printed at the top of the next page.

Example 3: `\NEED 30;BOX 25 "Last sample",ASAP`

This example will force the box to be printed immediately. The NEED command will cause a paragraph break and force a page break if there are not enough lines on the page to print the box. The ASAP option will print the box if there is enough space on the page.

Related commands

DOUBLE
FIGURE
LINESPACE
NEED
PARAGRAPH
SPACE
WIDOW

CENTER
CENT

Purpose

To center one or more lines of text.

Form

```
\CENTER [n]
        [B]
```

Description

The CENTER command is used to center lines of text between the left and right margins. A specified number(n) of lines can be centered, or the next block of text. If no parameter is specified, then all following lines are centered until another mode command (FORMAT, IMAGE, RIGHT) is encountered.

Limitations

CENTER is not allowed in headers or footers.

Options

[B] The B option allows you to center a block of text rather than a specified number of lines. The end of the block is denoted by a formatting command line (normally signified by the \ character) or a blank line.

[n] The n option specifies the number of lines to be centered.

Example

```
\CENTER 3
```

This example centers the next three lines of text.

Related commands

FORMAT
IMAGE
INFORMAT
RIGHT

CMARGIN

CM

Purpose

To define the margin between columns when multiple column output is requested.

Form

```
\CMARGIN [(] {string} [, [string] [, [...]]] [)]
             {n     }    [n     ]
```

Description

The CMARGIN command is used to specify the size of the margin between columns in multiple-column output. If this margin is not set, TDP calculates its own margin based on the number of columns and the left and right margins (excluding INLFT and INRHT).

The [n] parameter specifies the width of the center margins.

The [string] parameter sets the size of the center margins by supplying a string to be printed in the margin(s).

Limitations

CMARGIN must be set before entering multi-column mode (see the COLUMN command). A maximum of 15 column margins may be specified. The total number of characters that can be used for these margins is 128. No formatting is performed on the specified string.

Options

None.

Examples

Example 1: `\CM 4`

This command sets a center margin of four columns between any columns.

Example 2: `\CM (3,7,3)`

This command specifies margins between columns for four or more column output. The first margin is 3 characters, the second 7, and the third to last 3.

Example 3: `\CM ("-*-", "xxxxxxx", "-*")`

This command specifies margins between columns for four (or more) column output. However, it also places the specified characters in those margins. Thus, the first margin is three characters, filled with - * -; the second margin is a row of 7 x's; and the third is two characters, - *.

Related commands

COLUMN

COLUMN
COL

Purpose

To format output into more than one column.

Form

```
\COLUMN {n                           } [,CONTROLLED]
        {(n        [,n         [...]])} [,NEED       ]
        {(n string[,n string[...]])} [,SPACE n    ]
        {END                         }
```

Description

THE COLUMN command is used to specify the start of multi-column output and the number of columns. Formatting returns to single column when a \COLUMN 1 or \COLUMN END appears. When the end of the page, a \NEW command, or another COLUMN command is reached, the material that has accumulated is formatted into the specified number of columns, as close to the same length as possible. The options allow you to control the column breaks and prevent awkward page breaks.

The (n,n,n...) parameter specifies more than one column, and the width of each column. The CONTROLLED option is assumed if this type of specification is used. Thus, you would use \NEXT commands to indicate the start of new columns.

The (n "string", n "string",...) parameter specifies more than one column, the width of each column, the size of the margin between each column, and a string to be printed in those margins.

END causes formatting to revert to single column mode.

Limitations

The maximum number of columns allowed is 16. Any string following the last column is ignored. The total number of characters that can be used for the center margins is 128. No formatting is performed on the specified string.

Note that if INLFT or INRHT are set when a COLUMN command is executed, then the setting will be ignored in calculating the widths of the columns. However, the setting will be applied to the column itself.

The COLUMN command is not allowed in ASK mode.

Options

[CONTROLLED] The CONTROLLED option allows you to control the composition of each column. The output is formatted into one of the columns until you enter the command \NEXT, then the next column is started. If the parameter (n,n,n....) is specified, then output is assumed to be CONTROLLED. If CONTROLLED is not specified, or the parameter (n,n,n...)is not specified, \NEXT is an illegal command.

[NEED] The NEED option prevents a set of columns from being broken over a page boundary. This option will also set the CONTROLLED flag.

[SPACE] The SPACE option allows that number of blank lines to be attached to the end of the multi-column output, provided there are that number of lines on the page, and that the output does not cross a page boundary.

Examples

Example 1: `\COLUMN 2`

This command specifies that the output starting after the command is to be formatted into two columns.

Example 2: `\COLUMN 3, CONTROLLED`

This command specifies that the output following is to be formatted into three columns, with you determining where the column breaks will occur in the material.

Example 3: `\COLUMN 2, NEED`

This command specifies two-column output, and prevents a page break from occurring in the middle of the material.

Example 4: `\COL (20,3,17)`

This command specifies three columns of 20, 3, and 17 characters, respectively. The CONTROLLED option is assumed if this type of specification is used. Thus, you would use \NEXT commands to indicate the start of columns 2 and 3.

Example 5: `\COL (20 "---", 3 "****", 17)`

This example also specifies three columns, with the same column widths as above. However, it also defines the size and composition of the margins between the columns. The first margin is three characters, - - -, and the second is four, * * * *.

Related commands

CMARGIN
NEXT

CONTENTS
CON

Purpose

To automatically compile a table of contents.

Form

```
\CONTENTS [n     ] [,IN n              ]
          [string] [,NEW               ]
          [BEGIN ] [,NO[DOTS]          ]
          [END   ] [,SPACE n [AFTER]]
          [ON    ] [,NEED n            ]
          [OFF   ]
```

Description

The CONTENTS command causes the next [n] line(s) of output to be included in the Table of Contents. Any leading blank characters will be stripped from lines added to the Table of Contents. When the document is formatted, the Table of Contents is constructed and stored in a temporary file, and output after the last page of the document. The Table of Contents is printed without a heading or page number. Options allow you to control the format for the Table of Contents. The [string] parameter specifies a string that will appear in the Table of Contents, referenced to this page, rather than using an existing line of text on the page.

Limitations

The string option is limited to 72 characters and no formatting is performed.

Options

[BEGIN/END] The first BEGIN/END option is used to supply a Table of Contents title other than "CONTENTS", provided it precedes the first \CONTENTS command. Formatting commands to alter the margins, spacing, underlining and so forth can be entered beginning on the line following the \CONTENTS BEGIN command, and must be followed by \CONTENTS END. This is particularly useful when printing a document in another language.

Subsequent BEGIN/ENDs are used to insert information into the Table of Contents that does not appear in the document. Any lines between BEGIN and END are printed in the Table of Contents, but they are not printed in the document. No reference to this page is included.

[ON/OFF] The ON/OFF option can be used to turn on and off the Table of Contents when there are \CONTENTS commands in the file but you do not want or need a Table of Contents (such as when you are printing multiple copies of a portion of the document).

[NEW] The NEW option allows you to force a new page in the Table of Contents before the line specified in the command appears as a Contents entry.

[SPACE n [AFTER]] The SPACE n [AFTER] allows you to insert blank lines before (SPACE n) or after (SPACE n AFTER) the line specified in the CONTENTS command.

[IN n] The IN n option causes the specified line to be indented n spaces from the left margin.

[NO[DOTS]] The NO[DOTS] option suppresses the printing of the dots between the entry and the page numbers (NODOTS), or both the dots and the page numbers (NO).

[NEED] The NEED option functions the same as that in normal formatting. However, it is limited to numeric values only.

Examples

Example 1: `\CONTENTS`

This example adds the next line of text to the Table of Contents.

Example 2: `\CONTENTS "Summary", NEW, SPACE 2 AFTER`

This example adds the string "Summary" to the Table of Contents, keyed to the current page number. This entry starts a new page, regardless of the length of the previous page. Two blank lines will be added after the entry.

Example 3: `\CONTENTS 3, IN 5, NODOTS`

This example includes the next three lines in the Table of Contents, indented five spaces. No dots will be printed connecting the line with the page number.

Example 4:

```
\CONTENTS BEGIN
\CENTER 1;UL 1
Table des Matieres
\SPACE 2
\HEAD = "Table des Matieres (Suite)"
\CONTENTS END
 ...body of document begins here
```

This example sets the heading for the Table of Contents to "Table des Matieres" (the French equivalent). Succeeding pages will be headed "Tables des Matieres (Suite)". After the \CONTENTS END command, you use \CONTENTS to specify entries as usual in the Table of Contents.

Related commands

INDEX
NAME
TABLE FIGURE

COPIES
COP

Purpose To force multiple copies of the formatted output.

Form

```
\COPIES n
```

Description The COPIES command can be issued when more than one copy of the formatted output is required. The numeric parameter identifies the number of copies to be made.

Limitations The limit is 127 copies. This command must placed at the beginning of the file, before any text output. Note that with the TDP spooler only one copy will be made, regardless of the value of COPIES (if the spooler is being used, the COPIES parameter to the FINAL command should be used instead).

Options None.

Example

```
\COPIES 3
```

Will result in three copies of the formatted text resulting from a FINAL or DRAFT command.

Related commands /DRAFT
/FINAL

DEACTIVATE

DEACT

Purpose

To stop printing on one or more HP 2680 or HP 2688 laser printer logical pages.

Form

```
\DEACTIVATE [(] n [,n [,n [...]]] [)]
```

Description

The DEACTIVATE command is used to remove a number of HP 2680 or HP 2688 logical pages from the list of active logical pages. The number "n" is the number associated with the logical page in the environment file.

These pages will not be used after this physical page has been printed.

Limitations

This command is only valid when used with a HP 2680 or HP 2688 laser printer.

At least one logical page must be active at any time. Thus, to deactivate the only active page, a replacement active logical page must be specified first.

The DEACTIVATE and ACTIVATE commands are ignored until output begins. So if the DEACTIVATE command appears before the first line of text in a file, it will be ignored.

Options

None.

Examples

Example 1: `\DEACTIVATE 5`

This will remove logical page 5 from the list of active logical pages used to format the text. If the user is currently using logical page 5 as the current page definition, it will continue to be used to the end of the page. However, on the next physical page, logical page 5 definition will not be used.

Example 2: `\DEACTIVATE 4,9,12,25,7,11`

If the current active logical pages are 1,4,7,9,11,12,15 and 25, and we have already output three pages of text to this physical page, before issuing the command, the system will use logical page 9 to format the next page of text. Even though the user does not appear to have written any data to the fourth page, the laser printing software has already selected that page and will use it to format the text. After completing the page of output, printing will continue on the same physical page, using logical page 15. After that a page eject will occur and printing resume using logical page 1.

Related commands

ACTIVATE
LAYOUT

DELAY

Purpose

To force a write statement for each blank line printed to the terminal.

Form

```
\DELAY
```

Description

TDP normally uses one write statement with the appropriate number of line feed characters to output multiple blank lines. However, some output devices cannot accept multiple line feeds in one record; normally, these would be printers with no internal buffers. The DELAY command forces TDP to generate a separate write statement for each blank line. Output to the line printer (*LP) automatically sets DELAY, since Hewlett-Packard line printers will not handle multiple line feeds.

Limitations

None.

Options

None.

DOUBLE
DBL

Purpose To double-space all or part of a document.

Form

```
\DOUBLE [ON ]
        [OFF]
```

Description Once the command \DOUBLE or \DOUBLE ON is entered, the text will be double spaced until a \DOUBLE OFF command is entered.

Limitations None.

Options

[ON] The ON option switches on double spacing.

[OFF] The OFF option switches off double spacing.

Related commands

LINESPACE
NEED
SPACE

END

Purpose

To mark the end of the text to be formatted.

Form

```
\END
```

Description

The END command is used to indicate the end of the text to be formatted. Unlike the EOD command, it causes no more formatting to take place.

Limitations

None.

Options

None.

Related commands

EOD
EXIT

ENVIRONMENT

ENVIR

Purpose

To define the environment file for the HP 2680 or HP 2688 laser printer.

Form

```
\ENVIRONMENT filename
```

Description

The ENVIRONMENT command is used to identify the environment file required by the HP 2680 or HP 2688 laser printer to define logical pages and available fonts. This command must appear before any text is printed (including blank lines), and should normally appear as the first command in the file.

The groups ENV2680A.SYS (for the HP 2680) and HPENV.SYS (for the HP 2688 and the HP 2680) contain standard environment files supplied with the system. For details of the standard environment files and fonts see the **IFS/3000 Interactive Formatting System Reference Guide**.

In the case of the HP 2688, TDP searches for the environment file in the user's group, then in HPENV.SYS and then in PUB.SYS. For the HP 2680, in addition to the above, TDP also searches in ENV2680A.SYS.

Limitations

This command is only valid when used with the HP 2680 or HP 2688 laser printer. This command must precede any text or command which makes use of the information in the file (eg. ACTIVATE, DEACTIVATE or FONT)

Options

None.

Related commands

ACTIVATE
DEACTIVATE
FONT
FONTEQ
FONTID
ILLUSTRATION
LAYOUT [NEW]
^A
^F

EOD

Purpose

To mark the end of useful data in a file.

Form

```
\EOD
```

Description

The EOD command is used to indicate the end of useful data in an input file to the formatter. This can be used to selectively stop processing of a file by associating the command with an IF statement. It could also be used to stop formatting of a file if only a small section of a large file is required. (END or EXIT may be used as alternatives.) EOD embedded in an INCLUDE file causes no more formatting of the INCLUDE file but formatting does continue at the next level up.

Limitation

None.

Options

None.

Examples

```
\IF NOT *HP2601 EOD
```

In this example, if the text is not being formatted on an HP 2601 printer, the EOD will be executed to terminate processing.

Related commands

END
EXIT
IF

EQUATION

EQ

Purpose

To format text as a mathematical expression.

Form

```
\EQUATION
```

Description

The EQUATION command is used to format the next line(s) as a mathematical expression. Successive lines are considered part of one expression until a blank line or another TDP formatting command is encountered.

Each expression must be preceded by a separate EQUATION command. For more information on entering mathematical expressions, see the EQN command in Chapter 3.

Limitations

No more than 255 characters or 63 terms can appear in an expression.

The expression must appear on one or more lines by itself in order for this command to work properly.

Expression formatting is terminated by either a blank line or a formatting command line (normally signified by the \ character).

Any commands following \EQUATION on the same line will not be executed. EQUATION should not be used to process expressions containing characters from an eight bit character set.

Options

None.

Examples

```
\EQ
c>[2] = a>[2] + b>[2]
(blank line)
```

The EQ command here causes TDP to format the next line as a mathematical expression:

```
  2     2     2
 c  = a   + b
```

The blank line denotes the end of the expression.

Related commands

^+
^_
^>
^<
/EQN

ERROR

Purpose

To suppress the printing of formatter error messages on the terminal.

Form

```
\ERROR (OFF)
       (ON )
```

Description

When an error occurs during formatting, error text is written to a file ERRORLOG in the local group and account. At the end of a formatting run, the user will be prompted to save this file. The file is automatically saved if no permanent ERRORLOG file exists. In batch mode any permanent file will be purged.

While printing the errors in the ERRORLOG file, TDP also prints them to $STDLIST. If you are formatting the document to your terminal the error messages will be displayed at the same time as your document. Using the ERROR OFF command stops the error messages being sent to the terminal. This is reset using the ERROR ON command.

Limitations

If the ERRORLOG file could not be created, or the ERRORLOG file does not accept text, then errors will be reported to the terminal as normal.

Options

None.

ESCAPE

ES

Purpose

To define the character used to signal intraline formatting.

Form

```
\ESCAPE [char]
```

Description

The ESCAPE command is used to assign a character to be used for intraline formatting. The default character is the care ^ To return to the default escape character, enter \ESCAPE with no character parameter. To print the current escape character use a sequence of two escape characters, for example: ^^

Limitations

Must not be the prompt character.

The character you define must be a seven-bit character (a character from the ASCII character set).

Options

None.

Example

```
\ESCAPE "*"
```

This example assigns the asterisk as the escape character. Thereafter, commands for intraline formatting include an * instead of a ^. Thus, ^W(for underlining words only) becomes *W.

Related commands

BACKSPACE
BLANK
PROMPT

EXIT

E

Purpose

To exit from a formatting operation and return to editing operations.

Form

```
\EXIT
```

Description

The EXIT command can be entered following a formatting prompt when using the ASK option with a FINAL or DRAFT command, or when a DRAFT or FINAL operation has been suspended with a CTRL-Y. It can also be embedded in a file to terminate formatting at the point the \EXIT command is encountered. If the EXIT command is embedded in a text file, formatting is terminated without the END OF FORMATTING details.

Limitations

None.

Options

None.

Related commands

/DRAFT
/FINAL
GO
MARGIN
PAGE
PAUSE
TRY

FIGURE

FI

Purpose

To leave space for a figure to be inserted.

Form

```
\FIGURE n [string] [,ASAP] [,FULL]
```

Description

The FIGURE command leaves the specified number of lines for a figure and prints the "string" as a figure title.

Normally, space is left for the figure at the bottom of the current page (if enough lines remain) or the top of the next page. If there is not room on the current page, the rest of the page is filled with text that follows the FIGURE command in the file. The space left is the specified number of lines deep. Two blank lines are left above the figure and three below (one for the title).

If a second FIGURE or BOX command is encountered before the first has been printed, then to preserve the order within the document, the first FIGURE or BOX will be printed. Since only one FIGURE or BOX is placed at the bottom or top of any page, and insufficient space may exist for the first FIGURE or BOX on the current page, blank lines may be left on the current page.

This command does not cause a paragraph break; so it may be introduced in a paragraph without affecting it adversely. This means that the FIGURE command can be introduced in the middle of a paragraph making reference to it, immediately following the reference.

Limitations

The title is limited to 60 characters.

If a figure space is requested which is larger than a full page, the largest space that will fit on a page is reserved.

Figures are not allowed in headers or footers.

Options

`[string]` The string option specifies the figure title to be printed. The title string is limited to 60 characters. It will be included in the Table of Figures, if that file is being prepared.

`[ASAP]` The ASAP option places the figure immediately, if there is room on the current page, rather than waiting for normal placement at the bottom of the page.

`[FULL]` The FULL option causes the figure to be placed between the page margins, rather than the current left and right margins. This is used in multi-column mode only when the figure is to fill the full width of the page rather than the width of the current column.

FIGURE

FI

Examples

Example 1: `\FIGURE 10 "Figure 6. This is a sample."`

This command leaves 10 blank lines for a figure at the bottom of the current page, or the top of the next page. The string "Figure 6. This is a sample." is printed as a figure title below the space that is left.

Figure 6. This is a sample

Example 2: `\FIGURE 10, ASAP`

This command leaves 10 blank lines for a figure immediately after the command is encountered. If there isn't room on the current page, the figure is placed at the top of the next page.

Related commands

BOX

FONT

Purpose

To print a number of lines in a given HP 2680 or HP 2688 font.

Form

```
\FONT {("name"/n [,"name"/n])}    n
      {        id           }
```

Description

The FONT command allows you to print a number of output lines in a given font, or font pair. The font can either be given explicitly or by means of a font identifier. If given explicitly, then the fonts must be contained within brackets, also if two sets are supplied then they will be assumed to be linked (i.e. the first set contains the 96 USASCII printable characters, the second set contains the Roman Extension).

Two character sets may be given in this command. Normally, characters in the first set will be used. Characters in second set will only be used when they occur between ^A and ^N commands or when the character has the eighth bit set (ie. is part of an eight bit character set).

Since more than 96 characters are required to fully define the characters of any European language, a pair of character sets can be linked to each other. The first 96 characters of any pair will be considered to be in the standard USASCII sequence, while the next 96 characters come from the Extended Roman set.

The parameters [n] and [name] are used to associate the numbered or named HP 2680 or HP 2688 font with the appropriate command. If fonts are supplied they must be contained in brackets ( ). If two fonts are supplied then they will be considered as a linked pair, the first forming the normal 96 printable USASCII set and the second set the EXTENDED ROMAN set.

Options

None.

Limitations

This command is only valid when used with the HP 2680 or HP 2688 laser printer.

Where proportionally spaced fonts are used outside format mode, vertical alignments (ie columns, tables) will be lost. Also if a combination of fonts with widely differing widths are used then some alignments may be lost (ie INFORMAT areas).

FONT

Examples

Example 1: `\FONT ("PICA12") 20`

This will print the next 20 output lines in font "PICA12".

Example 2: `\FONT 0 6`

This will print the next six output lines in the font set up to be identifier '0' by the FONTID command.

Related commands

ACTIVATE
DEACTIVATE
ENVIRONMENT
FONTEQ
FONTID
ILLUSTRATION
LAYOUT [NEW]
^A
^F
^N

FONTEQ

Purpose

To associate HP 2680 or HP 2688 fonts with certain TDP commands.

Form

```
\FONTEQ  {BOLD  } [(] "name" [, "name"] [)]
         {GHOST }      n     [,    n   ]
         {NORMAL}
         {ALTER }
```

Description

The FONTEQ command sets up an association between HP 2680 or HP 2688 fonts and their usage within TDP. The command allows the user to define the fonts which are to be used in association with the BOLD and GHOST commands and their intraline equivalents and the character sets accessed with ^A. Any valid font may be associated with BOLD and GHOST.

Two character sets may be associated with each of the BOLD and GHOST commands. Normally, characters in the first set will be used. Characters in second set will only be used when they occur between ^A and ^N commands or when the character has the eighth bit set (ie. is part of an eight bit character set).

Since more than 96 characters are required to fully define the characters of any European language, a pair of character sets can be linked to each other. The first 96 characters of any pair will be considered to be in the standard USASCII sequence, while the next 96 characters come from the Extended Roman set.

It is worth noting that BOLD and GHOST may be associated with the same font, and that not all fonts defined within the environment need to be associated with a use in TDP, via this command or FONTID

The parameters [n] and [name] are used to associate the numbered or named HP 2680 or HP 2688 font with the appropriate command. If two fonts are supplied and they are contained in brackets (), then they will be considered as a linked set, the first forming the normal 96 printable USASCII set and the second set the EXTENDED ROMAN set, or an alternate set.

The ALTER parameter allows the default alternate character set to be defined. This character set will be used if there is no print enhancement in force and an alternate character is asked for or if there is a print enhancement in force but only one font is defined.

The NORMAL parameter allows the normal and alternate fonts to be changed. This is particularly useful when used after a layout command; eg. when moving from landscape to portrait mode printing within the document.

Options

None.

Limitations

This command is only valid when used with the HP 2680 or HP 2688 laser printer.

Where proportionally spaced fonts are used outside format mode, vertical alignments (ie columns, tables) will be lost. Also if a combination of fonts with widely differing widths are used then some alignments may be lost (ie INFORMAT areas).

Examples

Example 1: `\FONTEQ BOLD "BOLD"`

This states that whenever a BOLD or ^B command is found in the input, then the following text will be printed in font "BOLD", from the environment file.

Example 2: `\FONTEQ GHOST ("ROMITAL","ROMITALX")`

This states that whenever a GHOST or ^G command is found in the input, the following text will be printed in fonts "ROMITAL" and "ROMITALX". The font "ROMITALX" will be used for text following a ^A command or if the eighth bit is set on a character.

Example 3:

Suppose you wish to use normal, bold, italics and bold italics within your document. The bold characters can easily be selected by the BOLD and ^B commands, but access to the italics set is not obvious. To achieve the correct printing of the document the following commands could be used; assuming the normal set is font 2; the bold set is font 7; the italics set is font 12; and the bold italics set is font 17.

```
\FONTEQ ALTERNATE 12;FONTEQ BOLD 7,17
```

The italics may be accessed by using the ^A command in either normal or bold mode.

Related commands

ACTIVATE
BOLD
DEACTIVATE
ENVIRONMENT
FONT
FONTID
GHOST
ILLUSTRATION
LAYOUT [NEW]
^A
^B
^F
^G

FONTID

Purpose

To associate an HP 2680 or HP 2688 font with an identifier for use with the FONT or ^F commands.

Form

```
\FONTID id [(] "name"/n [, "name"/n] [)]
```

Description

The FONTID command sets up an association between HP 2680 or HP 2688 fonts and an identifier for later use within TDP. The primary use for the identifier is in the ^F command, which allows you to change fonts in the middle of a line. The other use is with the FONT command which allows you to print a number of lines in a given font.

Two character sets may be associated with any identifier. Normally, characters in the first set will be used. Characters in the second set will only be used between ^A and ^N commands or when the character has the eighth bit set (ie. is part of an eight bit character set).

Since more than 96 characters are required to fully define the characters of any European language, a pair of character sets can be linked to each other. The first 96 characters of any pair will be considered to be in the standard USASCII sequence, while the next 96 characters come from the Extended Roman set.

It is worth noting that different identifiers may be associated with the same font, and that not all fonts defined within the environment file need to be specified in TDP, via this command or FONTEQ.

The parameters [n] and [name] are used to associate the numbered or named HP 2680 or HP 2688 font with the appropriate identifier. If two fonts are supplied and they are contained in brackets ( ), then they will be considered as a linked pair. The first font forming the normal 96 printable USASCII set and the second set the EXTENDED ROMAN set, or the alternate character set.

Options

None.

Limitations

This command is only valid when used with the HP 2680 or HP 2688 laser printer.

Font id's may only be one character long, either alphabetic or numeric.

Where proportionally spaced fonts are used outside format mode, vertical alignments (ie columns, tables) will be lost. Also if a combination of fonts with widely differing widths are used then some alignments may be lost (ie. INFORMAT areas).

FONTID

Examples

Example 1: `\FONTID P "PICA"`

This sets up an association between the internal identifier P and the HP 2680 font "PICA". The font can then be accessed via \FONT P 3, which will print the next three line in font "PICA", or via ^FP. . . .^S, which will print the text between the ^F and ^S commands in the font PICA.

Example 2: `\FONTID 4 20,21`

This sets up an association between the internal identifier 4 and the HP 2688 fonts 20 and 21. When identifier 4 is used in the ^F3 or FONT commands, font 21 will be used for text following a ^A command or for characters with the eighth bit set. Font 20 will be used all for other characters.

Example 3: `\FONTID Q ("ROMAN","ROMANX")`

This sets up an association between the internal identifier Q and the HP 2680 fonts "ROMAN" and "ROMANX". When identifier Q is used in ^F or FONT commands, font "ROMANX" will be used for text following a ^A command or for characters with the eighth bit set.

Related commands

ACTIVATE
DEACTIVATE
ENVIRONMENT
FONT
FONTEQ
ILLUSTRATION
LAYOUT [NEW]
^A
^F
^N

FOOT

Purpose

To define a footing to be printed at the bottom of each page.

Form

```
\FOOT (string) [,A [,REV]]
      (BEGIN ) [,REV     ]
      (END   ) [,I       ]
      (OFF   ) [,L       ]
      (ON    ) [,C       ]
               [,R       ]
```

Description

The FOOT command defines a footing to be printed at the bottom of each page. It is printed to the left of the page number by default. A single-line footing is defined by inserting the footing in the command (the "string" parameter). More than one line is defined by entering the command \FOOT BEGIN, then the footing on as many lines as desired, and finally \FOOT END. The options allow you to determine the placement of the footing.

For a single line footing the footing, section and page number are all placed on the same line, midway down the bottom margin. The line this composite string is on can be controlled by the PAGENOLINE command. If multiple line footings are supplied then again they will be vertically centered within the bottom margin. However, the section and page number will no longer be included. The first line used to print multiple line footings is determined by PAGENOLINE. The page number may be included in a multiple line footing by means of the ^#P command.

If any positional option is supplied, then it will override any value entered via PAGENO, PAGENOLINE, SECTION or ALTERNATE. If no option is supplied the placing of the footings will be determined by the last setting supplied to any of these commands. Also if a positional option is supplied on any of these commands later in the input, then that value will override the value supplied by this command.

Limitations

If a multiple-line footing is used, the BOTTOM margin must be big enough to contain it. (See BOTTOM command for how to reset this margin.)

No formatting of the footing is possible other than the positioning of the text on the line. Simple text enhancements are available by use of the intraline escape sequences. Any formatting command encountered in the footing will be saved, then executed when the footing is output. However, the command will not affect the formatting of the footing, but rather the text that follows the footing.

The following commands are not allowed in footings: BOX/FIGURE, CENTER/IMAGE/RIGHT/FORMAT/SKIP, ILLUSTRATION, INCLUDE, NEW, NEED, PARAGRAPH and SPACE.

FOOT

Options

[L] or [I] The footing is printed on the right-hand side of the page by default. Use the L or I options to print it on the left-hand side.

[C] Use the C option to print it in the center.

[R] The R option resets positioning to the default condition (ie. the right-hand side of the page) when one of the options has been used.

[A[,REV]] It is also possible to have the footing alternate between the right- and left-hand sides of the page. For even numbered pages (if the page number is alphabetic then B,D,F,H,.. are even) the footing is placed on the left-hand side. This can be achieved by use of the A option. The alternation can be reversed by the REV option alone or following the A option. The ALTERNATE command can also set the position of the footing if no option is supplied.

[OFF]/[ON] It is possible to stop the printing of the footing by using the FOOT OFF command. The footing will be printed on this page but not on subsequent pages. All processing involved in printing the footing will still be performed (eg. incrementing page numbers), but no output will appear. To restart printing the footing use the FOOT ON command.

Examples

Example 1: `\FOOT "Revised 12/31/87"`

This example prints the footing "Revised 12/31/87" on the bottom right of each page.

Example 2:

```
\FOOT  BEGIN, ALTERNATE
Revised  12/31/87
Copyright Hewlett-Packard 1987
\FOOT  END
```

This example prints the two lines on the bottom of each page; on the right side for odd-numbered pages, and on the left side for even-numbered pages.

Related commands

FOOTNOTE
HEAD
HEADLINE
SECTION
PAGENO
ALTERNATE
BOTTOM
PAGENOLINE
^#F
^#S
^#P
#P

FOOTNOTE
FOOTN

Purpose

To include footnotes at the bottom of the page.

Form

```
\FOOTNOTE (n     )
          (string)
          (BEGIN )
          (END   )
```

Description

The FOOTNOTE command is used both to assign the string used to reference a footnote, and to denote the beginning and end of the material to be printed at the bottom of the page.

You can assign a footnote string by using \FOOTNOTE = "string" in which the string may contain between one and 20 characters. If you want numbers used to reference the footnotes, you can use FOOTNOTE = n, in which the number may be one to four digits long, or you can identify a numeric string containing up to 20 digits, as in FOOTNOTE = " 1 ". If the footnote identifier is to be printed as a superscript, you can use the string "^>1^<". TDP will automatically increment the number.

To enter the footnote itself, start a new line after typing the word that the footnote refers to, regardless of whether the line was full. Insert the command \FOOTNOTE BEGIN, then type the footnote material on the next line(s), ending with the command \FOOTNOTE END.

The FOOTNOTE is formatted to occupy the full width of the page without any hyphenation. No control is allowed over the formatting of the footnote. However, simple text enhancements may be included by means of the intraline escape sequences. The indentation of the first line of the footnote can be controlled by means of leading blanks on the first line. The footnote string included at the beginning of the footnote is placed before the first non-blank character.

TDP places the footnote(s) at the bottom of the page, after keeping track of how much room is needed. The footnote character is placed at the place the original line was broken.

Limitations

The footnote identifier will only be inserted into your text if you are in format mode. To include the footnote identifier in any other mode the ^#F command should be included in the text.

Options

[BEGIN] Marks the start of the footnote material.

[END] Marks the end of the footnote material.

FOOTNOTE

FOOTN

Example

```
\FOOTNOTE = "*"
This paragraph demonstrates the use of the
FOOTNOTE command.  When you want to
insert a footnote,
\FOOTNOTE BEGIN
A footnote explains a word or phrase
that appears in a document that might
not be understood by the reader.
\FOOTNOTE END
you merely use the FOOTNOTE command
as demonstrated in this example.
```

This example assigns the asterisk as the character used to reference footnoted material, then marks the point of insertion by a carriage return, then enters the footnote material (bounded by the FOOTNOTE BEGIN and FOOTNOTE END commands), and finally continues with the text. The result is shown below:

This paragraph demonstrates the use of the FOOTNOTE command. When you want to insert a footnote,* you merely use the FOOTNOTE command as demonstrated in this example.

(the rest of the page is printed as
usual, until the bottom of the page)

Related commands

FOOT
^#F

*A footnote explains a word or phrase that appears in a document that might not be understood by the reader.

FORMAT

Purpose

To fill each output line between the left and right margins.

Form

```
\FORMAT [n]
        [B]
```

Description

Format mode is the default operation for TDP. In this mode, lines of the printed output file do not correspond directly to lines of the workfile. After a DRAFT or FINAL command, the workfile is read as a continuous text stream and the text formatted, in the output file, so that lines in the output file are filled between the left and right margins defined for that document.

The FORMAT command must be entered to return to format mode after invoking one of the other output modes (IMAGE, RIGHT, and CENTER), unless they were specified for only a certain number of lines.

Lines containing only special characters are treated as image mode. Specifying the parameter [n] causes the next n lines of input to be processed in format mode. Processing then reverts to the previous mode. FORMAT with no parameter processes the lines following the command in format mode, until one of the other modes is specified, and also overrides any other output mode previously set.

Limitations

The FORMAT command is not allowed in headers or footers.

Options

[B] FORMAT B processes the next block of lines in format mode; a formatting command line (normally signified by the \ character) or blank line signals the end of the block.

Example

```
\FORMAT 3
```

This example formats the next three input lines of text.

Related commands

CENTER
IMAGE
INDENT
INFORMAT
HYPHEN
JUSTIFY
RIGHT

GHOST

Purpose

To print one or more lines in ghost face type.

Form

```
\GHOST [n]
```

Description

The GHOST command prints the specified number of lines in ghost face. In the case of the laser printer, this is achieved by using a special font (this may, of course, be any font; not one corresponding to a traditional ghost typeface). In the case of impact printers, characters are printed twice with a very slight offset on the second printing to make the letter appear darker. The entire line is printed in ghost face. The [n] parameter specifies the next n output lines to be printed in ghost typeface. GHOST without a parameter causes the following output line only to be printed in ghost typeface.

Limitations

This command can only be used on laser printers or those impact printers that support ghost, or shadow, printing.

Options

None.

Examples

```
\GHOST 3
```

Will print the three lines following the command with a type offset that makes the characters wider and darker than normal.

Related commands

BOLD
FONTEQ
RED
^B
^G
^R

GO

G

Purpose

To initiate formatting by TDP after entering commands at a terminal.

Form

```
\GO
```

Description

The GO command is only used when the ASK option is included in a FINAL or DRAFT command. GO must be the last command before formatting. TDP will produce the requested output and then return to the TDP editor.

Limitations

GO will only execute if it is the last formatting command entered under the ASK option.

Options

None.

Examples

```
/FINAL ASK
\PAGE 1/3; GO
```

This example prints pages 1 through 3 of your document. The GO command signals the end of the commands entered under the ASK option, and initiates the final output.

Related commands

COPIES
/DRAFT
EXIT
/FINAL
MARGIN
PAGE
PAUSE
TRY

HEAD

Purpose

To define a heading to be printed at the top of each page.

Form

```
\HEAD {string} [,A [,REV]]
      {BEGIN } [,REV     ]
      {END   } [,I       ]
      {OFF   } [,L       ]
      {ON    } [,C       ]
               [,R       ]
```

Description

The HEAD command defines a line or several lines to be printed at the top of each page. A single-line heading is defined by including the heading as the "string" parameter. More than one line is defined by entering the command \HEAD BEGIN, then the heading on as many lines as desired, and finally \HEAD END. The options allow you to determine the placement of the heading.

The heading is placed centrally within the top margin. However, the first line the heading is printed on can be controlled by HEADLINE.

If any positional option is supplied, then it will override any value entered via HEADLINE or ALTERNATE. If no option is supplied the placing of the headings will be determined by the last setting supplied to any of these commands. Also if a positional option is supplied on any of these commands later in the input, then that value will override the value supplied by this command.

Limitations

If a multiple-line heading is used, the TOP margin must be at least one greater than the number of lines in the heading. (See TOP command for how to reset the TOP margin.) HEAD BEGIN must be the last entry on a command line.

No formatting of the heading is possible, other then the positioning of the text on the line. Simple text enhancements are available by use of the intraline escape sequences. Any formatting command encountered in the heading will be saved, then executed when the heading is printed. However, the command will not affect the formatting of the heading, but rather the text that follows the heading.

The following commands are not allowed in headers:

BOX/FIGURE, CENTER/IMAGE/RIGHT/FORMAT/SKIP, ILLUSTRATION, INCLUDE, NEW, NEED, PARAGRAPH and SPACE.

HEAD

Options

[L] or [l] The heading is printed on the right-hand side of the page by default. Use the L or l option to print it on the left side.

[C] Use the C option to print it in the center.

[R] The R option resets positioning to the default condition (ie. the right-hand side of the page) when one of the options has been used.

[A[,REV]] It is also possible to have the heading alternate between the right-hand and left-hand sides of the page. For even numbered pages (if the page number is alphabetic then B,D,F,H,. are even) the heading is placed on the left-hand side. This can be achieved by use of the A option. The alternation can be reversed by the REV option, either alone or following the A option. The ALTERNATE command can also set the position of the heading if no option is supplied.

[OFF]/[ON] It is possible to stop the printing of the heading by using the HEAD OFF command. The heading will be printed on this page but not on subsequent pages. All processing involved in printing the heading will still be performed, but no output will appear. To restart printing the heading use the HEAD ON command.

Examples

Example 1:

```
\HEAD "Introduction"
```

This example prints the heading "Introduction" on the top right of each page.

Example 2:

```
\HEAD BEGIN,LEFT
Introduction to
TDP
\HEAD END
```

This example prints the two-line heading on the left side of each page.

Example 3:

```
\HEAD "Introduction",ALTERNATE
```

This example prints the heading on the top of each page, on the right side of odd-numbered pages, and on the left of even-numbered pages.

Related commands

ALTERNATE
FOOT
HEADLINE
TOP

HEADLINE

HEADL

Purpose

To establish the vertical placement of the page heading.

Form

```
\HEADLINE n [,A [,REV]]
            [,REV     ]
            [,I       ]
            [,L       ]
            [,C       ]
            [,R       ]
```

Description

The HEADLINE command determines how many lines from the top of the page the heading is printed. The command can also be used to determine where on the line the heading is printed.

The default positioning of the heading varies according to the way in which the heading is specified. If the heading is specified in the form: `\HEAD (string)`, the default position is line 4. However, if the heading is specified in the form:

```
\HEAD BEGIN
string
\HEAD END
```

then the position of the first line of the heading is calculated by the formula:

$$\frac{\text{TOP - Number of lines in heading}}{2} + 1$$

HEADLINE explicitly overrides this default positioning.

Limitations

The headline parameter n must be less than or equal to the size of the TOP margin and greater than 1.

If any positional option is supplied, then it will override any value entered via HEAD or ALTERNATE. If no option is supplied the placing of the headings will be determined by the last setting supplied to any of these commands. Also if a positional option is supplied on any of these commands later in the input, then that value will override the value supplied by this command.

Options

`[L] or [I]` Use the I or L option to print the heading on the left.

`[C]` Use C to print in the center.

`[R]` Use R to print on the right.

`[A[,REV]]` The A option can be used to print the heading on the left for even numbered pages, and on the right for odd numbered pages. The REV option either alone or appended to the A option will reverse the alternation.

Examples

Example 1: `\HEADLINE 3`

This example prints the page heading three lines down from the top.

Example 2:

```
\TOP 4
\HEAD BEGIN
  Line 1
  Line 2
  Line 3
\HEAD END
\HEADLINE 3
```

This example illustrates HEADLINE overriding the normal default positioning of the heading. The default position for the first line of the heading, given by the formula shown above, would be:

$$\frac{\text{TOP (4) - Number of lines of heading (3)}}{2} + 1 = 1$$

HEADLINE overrides this and sets the first line of the heading to line 3; so that the heading extends into the document area.

Related commands

ALTERNATE
HEAD
TOP

HYPHALLCAPS
HYPHALLC

Purpose

To selectively hyphenate words all in capitals.

Form

```
\HYPHALLCAPS [ON ]
             [OFF]
```

Description

It is not usual to hyphenate words comprised of all capital letters. In the case that it is desirable, however, the HYPHALLCAPS command allows you to do so. The default setting is off.

Limitations

None.

Options

[ON] Turns on the command. The command remains in effect until turned off.

[OFF] Turns off the command.

Related commands

HYPHCHAR
HYPHDBL
HYPHEN
HYPHFIRSTCAP
HYPHLAST

HYPHCHAR

Purpose

To assign a character as a discretionary hyphen; this allows you to dictate the hyphenation of any words you wish.

Form

```
\HYPHCHAR [char]
```

Description

The HYPHCHAR command assigns a character to be used to indicate possible hyphenation of a word. Called a discretionary hyphen, the character is then inserted at those places in the word where hyphenation would be allowed. This command is useful for non-English words, or unusual words that would not be covered by ordinary rules of hyphenation or TDP's exception dictionary.

As the text is formatted, discretionary hyphens within the line are deleted; if a word with discretionary hyphens occurs as the last word in the line and needs to be hyphenated, the word is broken at one of the places indicated by a discretionary hyphen and all other discretionary hyphens are deleted. The word will be hyphenated if necessary whether or not hyphenation is turned on.

You may use a discretionary hyphen immediately in front of a word to prevent it from ever being hyphenated.

This feature can be used even when automatic hyphenation is not being used, since the discretionary hyphens are used to hyphenate at all times.

Limitations

The character defined as the discretionary hyphen will always be interpreted in that manner. Therefore, it cannot be used in the text in its normal usage. The discretionary hyphen character cannot be escape, blank, space or hyphen.

The character you define must be a seven-bit character (a character from the ASCII character set).

Options

None.

HYPHCHAR

Examples

Example 1: `\HYPHCHAR "="`

This example sets the equals sign as the discretionary hyphen. It would be used to indicate acceptable hyphen points as follows:

```
1 The hyphenation algorithm might not work
2 for the French word sin=gu=li=ere, thus
3 sin=gu=li=ere contains discretionary hyphens.
```

Example 2: `\HYPHCHAR`

This example resets the character previously set as a discretionary hyphen to its normal state.

Related commands

HYPHALLCAPS
HYPHDBL
HYPHEN
HYPHFIRSTCAP
HYPHLAST

HYPHDBL

Purpose

To add a hyphen when automatic hyphenation divides a hyphenated word.

Form

```
\HYPHDBL [ON ]
         [OFF]
```

Description

The HYPHDBL adds a hyphen when a hyphenated word occurs at the end of the line and must be divided where the hyphen occurs. This is a convention used in some special documents and aids the reader in distinguishing compound words which the system has broken across lines.

Limitations

None.

Options

[ON] Turns the command on. The command remains in effect until turned off.

[OFF] Turns the command off.

Examples

```
\HYPHDBL
```

This example sets the double-hyphen feature on, such that if the last word on a line was carbon-tetrachloride, it would be hyphenated as follows:

> The chemical found most often was carbon-
> -tetrachloride

Related commands

HYPHALLCAPS
HYPHCHAR
HYPHEN
HYPHFIRSTCAP
HYPHLAST

HYPHEN
HYPH

Purpose

To set the automatic or interactive hyphen modes on or off.

Form

```
\HYPHEN [n] [ON           ]
            [OFF          ]
            [AUTO [,ASK]  ]
            [string [,ASK]]
```

Description

Two hyphenation modes are available; interactive hyphenation and automatic hyphenation.

In interactive hyphenation, if a word will not fit at the end of the line, TDP displays the word on the terminal, showing you how much of the word will fit on the line, and provides the option of hyphenating the word, where possible.

In automatic hyphenation, TDP hyphenates according to a list of rules, consulting an internal exception dictionary where necessary.

The option to include a user supplied hyphenation routine is provided, as is the ability to modify or add to the standard exception dictionary (see Chapter 7).

Limitations

HYPHEN ON or HYPHEN AUTO,ASK should be given prior to printing the first line, so a hold file will be generated in order to prevent the interactive dialogue from being mixed in with the final output.

Options

[ON] HYPHEN ON initiates interactive hyphenation. If a word will not fit at the end of the line, TDP displays the word on the terminal, showing you how much of the word will fit on the line. If the word can be hyphenated, you space across to the place where the hyphen should be inserted. Otherwise, press (Return) and the entire word will be put on the next line.

[AUTO[,ASK]] HYPHEN AUTO is used to initiate automatic hyphenation. TDP hyphenates according to a list of rules, consulting an internal exception dictionary when necessary. (To see how TDP would hyphenate a particular word, see the editing command HYPHEN in Chapter 3.) The ASK option with \HYPHEN AUTO causes TDP to print each word that needs hyphenation, marking the chosen hyphen point and asking for your approval. It also shows how much of the word fits on the current line (a caret mark points to the last character that would fit on the line). If you approve the proposed hyphenation, answer "yes". If not, answer "no" and you can indicate the hyphen point as you would in interactive mode (see Description above). If you answer with a double backslash (\\) the ASK option is disabled. However, TDP still prints each word it hyphenates.

[OFF] HYPHEN OFF terminates an interactive or automatic hyphenation session.

[string] The form HYPHEN "string", in which "string" is the name of a user-supplied hyphenation routine, is for those installations with unusual hyphenation requirements such as non-english text or highly specialized vocabularies. The use of this form requires that a special hyphenation routine be written for the particular application. To return to the standard TDP hyphenation routine, you may issue the command \HYPHENATE AUTO and to stop hyphenation, use \HYPHENATE OFF. Note that if you use this form of the command and the specified routine does not exist, TDP will identify the command as an invalid command (see Chapter 7 for further information).

[n] The parameter [n] allows you to specify the minimum size of word that can be considered for hyphenation. This value must be greater than 2.

Examples

Example 1: `\HYPHEN ON`

This example sets interactive hyphenation mode on.

Example 2: `\HYPHEN AUTO,ASK`

This example sets automatic hyphenation mode on, but prompts you for approval of each hyphenation.

Example 3: `\HYPHEN "USERHYPH",ASK`

This example will result in TDP accessing the user-written hyphenation routine USERHYPH whenever a word needs to be hyphenated. As a result of the ASK option, for each word to be hyphenated, you will be asked to confirm the hyphenation point.

Related commands

HYPHALLCAPS
HYPHCHAR
HYPHDBL
HYPHFIRSTCAP
HYPHLAST

HYPHFIRSTCAP
HYPHFIRSTC

Purpose

To selectively hyphenate words beginning with a capital letter.

Form

```
\HYPHFIRSTCAP [ON ]
              [OFF]
```

Description

It is not usual to hyphenate words beginning with a capital letter (such as a person's name). When it is desirable, however, the HYPHFIRSTCAP command allows you to do so. The default setting is off.

Limitation

None.

Options

[ON] Turns on the command. The command remains in effect until turned off.

[OFF] Turns the command off.

Related commands

HYPHALLCAPS
HYPHCHAR
HYPHDBL
HYPHEN
HYPHLAST

HYPHFLAGS

Purpose To set a parameter to be passed to the user hyphenation procedure.

Form

```
\HYPHFLAGS n
```

Description The HYPHFLAGS command is used to set the top eight bits of the flag word that is passed to the user hyphenation procedure. (For further information on the user hyphenation procedure, see Chapter 7).

Limitations The maximum setting is 255.

Options None.

Related commands HYPHEN

HYPHLAST

Purpose

To selectively hyphenate the last word of a paragraph, or column.

Form

```
\HYPHLAST [ON ]
          [OFF]
```

Description

The HYPHLAST command allows you to hyphenate words which fall at the end of a paragraph, or column, provided that the word will not be hyphenated across a page. The normal setting is off.

Limitations

None.

Options

[ON] Turns the command on. The command remains in effect until turned off.

[OFF] Turns the command off.

Related commands

HYPHALLCAPS
HYPHCHAR
HYPHDBL
HYPHEN
HYPHFIRSTCAP

Purpose

To provide for conditional execution of a TDP formatting command.

Form

```
\IF [NOT] (MAIN    ) (commandlist)
          (INFILE  )
          (termtype)
          (Y       )
          (N       )
          (SF      )
```

Description

The IF command precedes (on the same line) one or more formatting commands that you want executed only if a certain condition exists. The possible conditions are:

1 MAIN - the file being processed is the main file (ie. the FINAL command was issued for this filename).

2 INFILE - the file being processed has already been included with an IN command.

3 termtype - a certain terminal or output device is being used. Terminals and output devices are referred to using the identifiers given in Appendix F. (The output device name is entered as the termtype parameter. The device name must be in upper case.)

4 Y (YES) - the test condition is "Y". This is useful if the test condition is a macro which is defined when the text is to be formatted. Refer to the discussion of the PROMPT option in the macro definition description.

5 N (NO) - the test condition is "N". This is useful if the test condition is a macro which is defined when the text is to be formatted. Refer to the discussion of the PROMPT option in the macro definition description.

6 SF (SHEET FEEDER) - a sheet feeder is used on the output terminal as identified in the FINAL command.

When the specified condition is true, other commands on the same line are executed. When it is not true, the rest of the line is ignored.

This command is very useful when combining different files to produce custom documents, or if you wish to vary formatting according to the printer to be used.

Limitations

None.

Options

[NOT] The NOT option executes other commands on the line when the condition is false.

Examples

Example 1:

```
\IF MAIN pageno 1
```

This command sets the page number to page 1 if this file is the main file.

Example 2:

```
\IF INFILE INLFT 10; INRHT 10
```

This command, on the other hand, moves the margins in ten spaces if this file is being printed as part of another file. If this file is being printed independently, as the main file, the margins are left at their previous settings.

Example 3:

```
\IF NOT MAIN INLFT 10; INRHT 10
```

This command is equivalent to Example 2.

Example 4:

```
\IF *LP LFT 1; RHT 132
```

This command sets the right margin to 132, providing a longer output line only when the output device is the system line printer.

Example 5:

```
\MA=PROMPT "DOUBLE SPACE?"
\IF ^MA DOUBLE ON
```

This example prompts the user by asking if double spacing is to be used. If the answer is "Y", the IF condition initiates double spacing.

ILLUSTRATION
ILLUST

Purpose

To include an illustration in the output.

Form

```
\ILLUSTRATION filename[:figure] [n] [,rotate  ] [,A [,REV]]
                                    [,P [,REV]] [,REV     ]
                                                [,I       ]
                                                [,L       ]
                                                [,C       ]
                                                [,R       ]
```

Description

The ILLUSTRATION command prints information from a raster or figure file, placing the result between the current margins as requested, rotating and scaling the information as required. It can also output a previously formatted file directly to the device. If insufficient space exists on the current page, then the illustration will be placed at the top of the next page.

The files hold information in either vector or raster format, and if held in vector format, a number of figures may be held in a single figure file. Files held in vector format will be scaled to fit the page. Files produced by DSG/3000, HP DRAW, HP EASYCHART and Drawing Gallery are normally held in vector form. The final size of an illustration can be set by the user (the first 'n' parameter indicates the number of lines the illustration should occupy). Illustrations may be rotated in increments of 90 degrees to any orientation relative to the page. 0 degrees indicates the illustration is the right way up and 180 degrees indicates the illustration is upside down.

The illustration will be scaled so that it fills the current margins, the supplied length, or the current page (PAGELENGTH-TOP-BOTTOM). If scaling produces an illustration that does not fully fill the current margins, then any positional parameter supplied will be used to determine where the illustration will be placed. By default the illustration is placed in the center. If more lines are specified than needed to scale the illustration, then those lines will be left blank after printing the illustration.

Since raster format files cannot be scaled or rotated, their size and orientation will be checked before the illustration is printed. Again if the raster does not fill the current margins then any positional parameter will determine where the illustration is printed; also any excess lines will be left blank.

If output is not to a graphical device (HP 2680 or HP 2688) then the figure file is queried for the size of the illustration, and a space of the appropriate size is reserved.

The [figure] parameter is used when a number of different figures are stored within one figure file. The figure is a string of up to 20 characters, exactly as defined within the sub-system where the figure file was created.

ILLUSTRATION
ILLUST

The parameter [n] is used to specify the maximum number of lines an illustration is to occupy. If the illustration is held in raster format and occupies more lines than this, then an error will be reported. If more lines are specified than will fit on the page, then the page length will be used (PAGELENGTH-TOP-BOTTOM).

The number of lines an illustration occupies if it cannot fill the number of lines requested, or if the 'n' parameter was omitted, is determined by a complicated set of rules. These rules are as follows: the desired number of lines is calculated (x) as either the value of 'n' or the number of lines the illustration takes if it fills the current margins when 'n' was not specified. 65% of this number is calculated (y). A space is looked for on the current page which is at least y lines long. If this space is found the illustration is printed on the current page. If this space is not found the position is reviewed in multi-column output which could involve starting a new page and the calculations are performed again. The illustration is then printed in the biggest available space.

The illustration is printed with the same orientation as the logical page it is being printed on. Use the [rotate] parameter to rotate the illustration, where [rotate] is 90, 180 or 270 and indicates the number of degrees the illustration is rotated. A positive value will rotate the illustration clockwise relative to the current page, while a negative value will rotate it anti-clockwise.

Limitations

This command can be used in normal and multi-column output; however, it should be noted that when placed within multi-column output it will stop the balancing of columns if insufficient data is output to fill the page. It is advised that this command be used within controlled columns and appear as the first command in the first column. This will ensure that the illustration is printed and that the text appears along side it.

Generally if no size is specified for the illustration and a scaled illustration would occupy less than 65% of the current margins then the illustration will appear on the next page with space, or the next blank page, whichever comes first.

In multi-column mode the illustration will be placed on the next page with space if the illustration is too big for the current margins and it appears at the top of the first column or in the last column of uncontrolled columns.

Note that the figure name must be followed by at least one space to act as a delimiter.

ILLUSTRATIONS are not allowed in headers or footers.

Options

[P[,REV]] It is possible to have the illustration rotation determined by the value of the current page number, so that the illustration will be rotated 90 degrees clockwise for even pages, and 90 degrees anti-clockwise for odd pages. This is done by using the P option instead of the [rotate] parameter. This rotation may be reversed by including the 'REV' option following the P option.

[L] or [I] The illustration is printed centrally between the margins by default. Use the L or I options to print it against the left-hand margin.

[C] The C option (default case) prints it centrally between the current margins.

[R] Use the R option to print against the current right-hand margin.

[A[,REV]] It is also possible to have the illustration placed against the left-hand margin for even numbered pages (if the page number ends in a letter then B,D,F,H, . . are considered as even), and against the right-hand margin for odd numbered pages by using the A option. This placement may be reversed by using the REV option alone, or by appending it to the A option.

It should be noted that if the illustration is to be rotated according to the page number, and then positioned according to the page number, but the positioning reversed, then this must be specified as P,A,REV and not P,REV as this would just reverse the sense of the P.

Examples

Example 1: `\ILLUSTRATION PICTFILE:EXCLUSIVE_OR 20,C`

This example will print the figure EXCLUSIVE_OR from the figure file PICTFILE. If, when scaled the illustration occupies 20 lines, but does not fill the current margins, then it will be placed centrally between the margins. The illustration will always occupy 20 lines, regardless of its scaled size.

Example 2: `\ILLUSTRATION RASTFILE,A`

This example will print the pre-formatted (raster format) illustration from the file RASTFILE. It can neither be scaled nor rotated. If the raster image occupies less space than the current margin, then it will be printed on the left for even pages and on the right for odd pages. The number of lines the illustration occupies will be determined by the raster image.

Related commands

ACTIVATE
DEACTIVATE
ENVIRONMENT
FONT
FONTEQ
FONTID
LAYOUT NEW
NAME
^A
^F

IMAGE

Purpose

To output text exactly as it appears in the input file.

Form

```
\IMAGE [n]
       [B]
```

Description

The IMAGE command is used to initiate image mode, where text is printed in exactly the format that appears in the input file. The format of the line is unchanged (the line is not filled, justified, centered, nor extra blanks removed). You could use image mode when entering the inside address on a business letter, for example.

Limitations

If a line is too long to fit between the current margins, an error message is displayed.

The IMAGE command is not allowed in headers or footers.

Options

`[B]` IMAGE B processes the next block of lines in image mode; a formatting command line (normally signified by the \ character) or blank line signals the end of the block.

`[n]` The [n] option specifies that the next n lines are to be processed in image mode. IMAGE with no parameter will cause all following lines to be processed in image mode until another mode command is encountered (CENTER, FORMAT, RIGHT).

Examples

Example 1: `\IMAGE 3`

In this example, the next three lines of text are printed exactly as they appear in your file.

Example 2: `\IMAGE B`

This example prints text exactly as is from this point until a blank line or line with a backslash is found, at which point the mode reverts to whatever was in effect before this command.

Example 3: `\IMAGE`

This command prints all lines following exactly as they appear until another mode (FORMAT, CENTER, RIGHT) is encountered.

Related commands

CENTER
FORMAT
INFORMAT
RIGHT

INCLUDE
IN

Purpose

To change the TDP input file in the middle of formatting another file.

Form

```
\INCLUDE filename [(first line [/ last line])]
```

Description

The INCLUDE command is used to insert, at output time only, another file somewhere in the file you are currently formatting and printing. TDP suspends processing on the main file, processes the file specified in the IN command, then returns to the main file and continues at the point it left off. This procedure is most commonly used to insert "boilerplate" material in documents being prepared. (An example of boilerplate might be a company's history and business philosophy that is included as standard material at some point in each proposal that is prepared.)

Nested IN commands are allowed up to 16 levels deep (i.e., a file that is brought in by an IN command may itself contain IN commands leading to other files).

Note that page width commands (such as LFT, RHT, MARGIN, etc.) in an IN file are ignored; thus, only the outer or main file's page width settings apply. All other commands take effect and can be controlled by \IF commands.

If the file to be included is numbered, then a small section of the file can be included by specifying the [first line/last line] parameters. If, however, the file to be included is not numbered, then the whole file will be included (record numbers may not be specified).

Limitations

No more than sixteen nested IN files are allowed.

Any commands following IN on the same line will be ignored and an error message displayed to say so. Note that the first line/last line must be enclosed by parentheses, and that the line number must be specified by number: FIRST and LAST are not acceptable values.

You can not INCLUDE a file in a header or a footer.

Options

None.

Example

```
\INCLUDE PROJMGRS
```

This example switches processing to a file named PROJMGRS that contains a brief summary of the individual project managers within the department or company. When that file has been output, TDP returns to the departure point in the main file.

Related commands

CONTENTS
INDEX
TABLE FIGURE

INDENT

IND

Purpose

To control indentation of individual lines within paragraphs.

Form

```
\INDENT ({(] n [,n [,n ... ] ] [)})
        (ON )
        (OFF)
```

Description

The INDENT command is used to specify the number of spaces [n] which certain lines are to be indented within a paragraph. (For laser printers this means spaces measured in the current font.) The command remains in effect until cancelled with a \INDENT OFF command. It does not cause a paragraph break. It should be noted that leading blanks on paragraphs are ignored when this command is in effect.

Limitations

A maximum of six values may be specified, the sixth value applies to the sixth through last lines of the paragraph. Note that with proportional typefaces blank character spacing is much less then with non-proportional typefaces, and spacing may need to be adjusted accordingly.

Options

`[OFF]` The OFF option allows you to do paragraph indentation yourself. Leading blanks on the first two lines of the paragraph are used to determine the indentation of the paragraph. Where the paragraph is one line long but must be split, then TDP will look for a Stepped Paragraph label and indent the second line from there. See Appendix H for more information on Stepped Paragraphs.

`[ON]` The ON option re-establishes the control over indentation by the last INDENT command.

Examples

Example 1: `\INDENT 5,10,5`

This command indents the following paragraphs according to this scheme: the first line is indented five spaces, the second 10, and the third and all subsequent lines 5 spaces.

Example 2: `\INDENT 5,0`

The first line of each paragraph is indented five spaces; following lines in the paragraph are not indented.

Example 3: `\INDENT 0,5`

This example produces a "hanging" first line with no indentation. Following lines in the paragraph are indented five spaces.

Related commands

INFORMAT
INLFT
INRHT

INDEX

Purpose

To write information to a file for later production of an index.

Form

```
\INDEX [n       ] [,IN n    ]
       [string] [,NO[DOTS]]
       [BEGIN ]
       [END   ]
       [ON    ]
       [OFF   ]
```

Description

The INDEX command causes the next [n] line(s) of output to be included in the Index file. When the document is printed, the Index file is constructed and stored in a temporary file called INDEX. When printing of the document is complete, the Index file is saved as a permanent file. The Index file is not sorted or printed with the document to allow you to do your own sorting, or editing before final output.

The [string] parameter allows you to insert the specified string in the Index file, referenced to the current page, rather than using a line already in the page of the text.

Limitations

The string option is limited to 72 characters.

Options

[BEGIN/END] Formatting commands to alter the margins, spacing, underlining and so forth can be entered beginning on the line following the \INDEX BEGIN command, and must be followed by \INDEX END. This is particularly useful when printing a document in another language.

Subsequent BEGIN/ENDs are used to insert information into the Index that does not appear in the document. Any lines between BEGIN and END are printed in the Index, but they are not printed in the document. No reference to this page is included.

[ON/OFF] The ON/OFF option can be used to turn on and off the Index when there are \INDEX commands in the file but you do not want or need an Index generated (such as when you are printing multiple copies of a portion of the document).

[IN n] The IN n option causes the specified line to be indented n spaces from the left margin. (for laser printers spaces are measured in the font current at the time of printing the Index.)

[NO[DOTS]] The NO[DOTS] option suppresses the printing of the dots between the entry and the page numbers (NODOTS), or both the dots and the page number (NO).

INDEX

Examples

Example 1: `\INDEX`

This example will add the next line of output to the Index file.

Example 2: `\INDEX "Summary"`

This example adds the string "Summary" to the Index, keyed to the current page number.

Example 3: `\INDEX 3, IN 5, NODOTS`

This example adds the next 3 line of output to the Index file, indented 5 spaces from the normal margin. No dots will be added to connect the last line with the page number.

Related commands

CONTENTS
NAME
TABLE FIGURE

INFORMAT

INFORM or INFMT

Purpose

To restrict format mode to a certain portion of each line.

Form

```
\INFORMAT n
```

Description

The INFORMAT command is used to fill only part of each output line. The portion of the line to the left of the specified column number is processed in image mode (for laser printers columns are defined in terms of the base character font).

The command is terminated by entering the command INFORMAT 0.

Limitations

The informat area cannot be greater than the current margins including INRHT and INLFT.

Options

None.

Example

```
\INFORMAT 21
```

Suppose you are preparing a glossary, with the words listed in the first 20 columns of each line, and their definitions on the rest of the line. This example sets format mode to operate on columns 21 to the right margin; columns 1-20 are processed in image mode. For example:

```
20  PEACOCK                 A large bird having a small head
21                          and an even smaller brain that makes
22                          a lot of noise to little effect.
23
24  BOULDER                 Inanimate object that may be of
25                          any size; doesn't have a brain
26                          but doesn't make a noise either -
27                          smarter than a peacock.
```

Is formatted as:

PEACOCK	A large bird having a small head and an even smaller brain that makes a lot of noise to little effect.
BOULDER	Inanimate object that may be of any size; doesn't have a brain but doesn't make a noise either - smarter than a peacock.

Related commands

CENTER
FORMAT
IMAGE
RIGHT

INLFT
INL

Purpose

To temporarily move the left margin in towards the center of the page.

Form

```
\INLFT n
```

Description

The INLFT command is used to move the left margin a specified number of columns to the right of LFT. It can be changed repeatedly if appropriate. However, INLFT always sets the margin relative to the value of LFT; any preceding use of INLFT is ignored in setting the new margin. To reset enter the command INLFT 0. (For laser printers, columns are defined in terms of the base character font.)

Limitations

The INLFT command cannot be used to move the left margin to the left of the current setting of LFT, or to the right of the current setting of the right margin.

If INLFT is set when a COLUMN command is executed then the setting will be ignored in calculating the widths of the columns. However, the setting will be applied to the column itself.

Options

None.

Example

```
\INLFT 10
```

This command moves the left margin ten columns to the right of the current setting of LFT.

Related commands

INDENT
INRHT
LFT
RHT

INRHT

INR

Purpose

To temporarily move the right margin in towards the center of the page.

Form

```
\INRHT n
```

Description

The INRHT command is used to move the right margin a specified number of columns to the left of RHT. It can be changed repeatedly if appropriate. However, INRHT always sets the margin relative to the value of RHT; any preceding use of INRHT is ignored in in setting the new margin. To reset enter the command INRHT 0. (For laser printers, columns are defined in terms of the base character font.)

Limitations

The INRHT command cannot be used to move the right margin to the right of the current setting of RHT, or to the left of the current setting of the left margin.

If INRHT is set when a COLUMN command is executed then the setting will be ignored in calculating the widths of the columns. However, the setting will be applied to the column itself.

Options

None.

Example

```
\INRHT 10
```

This command moves the right margin ten columns to the left of the current setting of RHT.

Related commands

INDENT
INLFT
LFT
RHT

JUSTIFY

J

Purpose

To square off the right margin.

Form

```
\JUSTIFY [ON ]
         [OFF]
```

Description

The JUSTIFY command squares off the right margin by adding extra blanks as necessary to even the line lengths from line to line. The default condition is OFF.

Limitations

JUSTIFY works only in format mode. If specified while in one of the other modes, it is ignored until format mode is reinstated. Thereafter, it applies each time format mode is entered, until a \JUSTIFY OFF command appears.

Options

[ON] The ON option starts justification. The command remains in effect until turned off.

[OFF] The OFF option terminates justification.

Examples

```
\JUSTIFY ON
```

This command causes TDP to begin right-justifying all output lines processed in format mode, as illustrated by this sentence, which has been justified to the left and right margins of the current column.

Related commands

CENTER
FORMAT
HYPHEN
IMAGE
INFORMAT
RIGHT

LAYOUT

Purpose

To allow the re-definition of the page parameters in the middle of a document.

Form

```
\LAYOUT [NEW] [n]
```

Description

The page definition commands LFT, RHT and WIDTH, have restrictive usage within TDP, since they define the constraints within which the text is printed. The LAYOUT command allows you to use these commands to change the page boundaries in the middle of a document. This command will cause any outstanding text and boxes to be printed. Thus all information concerning the current page is cleared out before starting the new layout. The LFT, RHT and WIDTH commands immediately following the LAYOUT command will result in a new page definition to begin on the next page.

Limitations

The options only work with a HP 2680 or HP 2688 laser printer. Any \NEW commands following a LAYOUT command will be ignored until output is restarted.

Options

[NEW] The NEW option will cause a physical page eject as well as a logical page eject on a HP 2680 or HP 2688 laser printer.

[n] The parameter n specifies the number of the HP 2680 or HP 2688 logical page to be used with the new layout. If the selected logical page definition is not active when text is about to be output, it will be activated, then immediately deactivated. Thus the logical page definition will only be used for the printing of the first page of output following the Layout command. The normal sequence of selecting the next highest numbered active logical page (from n) will be resumed.

Related commands

ACTIVATE
DEACTIVATE
ENVIRONMENT
LFT
FONT
FONTEQ
FONTID
ILLUSTRATION
RHT
WIDTH
^A
^F

LFT
LF

Purpose

To set the left margin.

Form

```
\LFT n
```

Description

The LFT command sets the value [n] of the left margin to be used when formatting the file for output. The default margin is 11 for a ten pitch output device, 13 for a twelve pitch device. (Note that for laser printers the value specified for LFT will be translated into character spaces in the base font; any left margin specified in the logical page definition in the environment file is ignored.) LFT must be defined before printing the first line of output, or the default values will be assumed. To change the margin later in the file, use INLFT.

Limitations

The LFT command must precede any output lines (including blank lines), or a \SPACE command, either at the beginning of the file or following a \LAYOUT command.

The maximum setting of LFT is 158.

Options

None.

Related commands

INLFT
INDENT
INRHT
LAYOUT
PITCH
RHT
WIDTH

LINESPACE

LINES

Purpose

To to set the spacing between lines when outputting text.

Form

```
\LINESPACE {n}
           {n.5}
```

Description

The LINESPACE command is used to indicate the desired spacing between lines. The default is 1 (single spacing). The spacing set with this command remains in effect for the rest of the document, unless reset by issuing the command again.

Limitations

The n.5 setting can only be used on devices which support half line feeds. All other devices will round the value up.

[n] must be in the range 1 to 4.

Line spacing of n.5 is not allowed in multi-column formats and the value will be rounded up or down.

Options

None.

Example

```
\LINESPACE 2
```

This example causes double-spaced output. (Equivalent to \DBL.)

Related commands

DOUBLE

LPTOP

Purpose

To adjust the placement of printing on line printer output.

Form

```
\LPTOP n
```

Description

TDP assumes that the line printer prints its first line 4 lines down from the perforation. LPTOP allows you to notify TDP that the line printer at your installation has some other spacing as its default condition.

Limitations

The LPTOP command must be given before output begins.

Options

None.

Examples

```
\LPTOP 0
```

This command informs TDP that the line printer at your installation is set to begin printing on the line immediately following the perforation.

Related commands

COPIES
PITCH
PROPORTIONAL
WIDTH

M(Macro)

Purpose

To define a two letter code (macro) to represent a string or to prompt for input.

Form

```
\M{id} {[PROMPT] {string}}
       {SET text         }
```

Description

TDP allows you to define up to 36 (id=a .. z,0 .. 9) simple macros. To define a macro string use M followed by a letter or a digit, and the string. Then, when you wish to enter that string in the text, you enter a caret (or the current ESCAPE character) followed by the macro, and TDP replaces it with the string it represents.

Alternatively, immediately after the macro name, give the keyword SET followed by the macro definition. In this case, the definition is not a string in the TDP sense (ie. enclosed by special characters). Hence, other formatting commands cannot follow a macro definition on the same line since they would be treated as part of the macro definition.

Limitations

A macro definition is limited to a maximum of 158 characters. All macro strings in a file combined may not exceed 1024 characters. They may, however, be redefined as many times as necessary in the file.

All macros on the line are expanded immediately upon reading the line thus it is not possible to define and reference a macro on the same line.

If a macro string causes the line to be extended beyond the right margin in any mode but format, then the line will be truncated.

Options

[PROMPT] The PROMPT option allows you to alter the definition of the macro each time the text is formatted. If you issue a FINAL (or DRAFT) command, and TDP encounters the macro definition, it will prompt you for input with the string given in the macro definition. You can then define the value of the macro through your terminal.

Examples

Example 1: `\MW "text processing"`

This example defines MW as a macro equal to the string "text processing". Suppose line 1, then, reads like this:

```
1  The ^MW system allows the use of macros.
```

Then, at the time of output, the macro is replaced by the string:

The text processing system allows the use of macros.

M(Macro)

Any time ^MW is encountered in the file, "text processing" will replace it, until MW is assigned to some other string.

Example 2: `\MW SET text processing`

This example will provide exactly the same results as in Example 1.

Example 3: `\MC=PROMPT "Enter customer name?"`

In this example, MC is defined as a macro to be solicited from the user. When output is requested, the user is asked "Enter customer name?" Thereafter, when ^MC is encountered, the name that was supplied is substituted for the macro. Thus,

```
 1  \MC   PROMPT "Enter customer name?"
17  Part number 30012 was ordered by ^MC.
```

would result in the prompt:

Enter customer name?

to which the user might respond:

James Smith

The resulting line in formatted form would then be:

Part number 30012 was ordered by James Smith.

Any further references to ^MC in the document would be replaced by "James Smith", until such time as the meaning of ^MC was redefined by means of a \MC assignment command.

Related commands

IF

MARGIN
MAR

Purpose

To move the entire text a specified distance to the right.

Form

```
\MARGIN n
```

Description

The MARGIN command moves the entire text the indicated number of spaces to the right. (For laser printers this will be character spaces in the base font.) This allows you to center material conveniently on any size page. It can be used with a FINAL,ASK command, so that you can input the desired margin from the terminal at the start of output. The default is 0.

Limitations

The MARGIN command must precede any output lines (including blank lines), or a \SPACE command, either at the start of formatting or following a \LAYOUT command.

The value of MARGIN must be in the range 0 to 158.

Options

None.

Examples

```
/FINAL, ASK
\MARGIN 7; GO
```

This example moves the entire text that follows seven spaces to the right, for this printing only.

Related commands

/DRAFT
/FINAL
GO
TRY

NAME

Purpose

To allow the user to name TDP generated files.

Form

```
\NAME [CONTENTS    ] filename
      [INDEX       ]
      [TABLE FIGURE]
      [RASTER      ]
```

Description

The NAME command allows you to set the name of various files generated by TDP. The files which TDP may generate during formatting are: the Table of Contents, the Table of Figures and the Index. In addition, when a vector format file is printed on the HP 2680 or HP 2688, then TDP will use an intermediary raster file; the NAME RASTER form of this command allows you to specify a general file name into which each conversion is done.

The temporary files generated to hold the Table of Contents and the Table of Figures are purged after the document is printed, unless a NAME command for them has been processed. The temporary file generated to hold the Index entries will be saved as a permanent file named INDEX, unless a NAME INDEX has been issued.

The raster file used by TDP for holding the intermediary conversion is named OUT2680 and is neither saved or purged when the formatter terminates. However, it is possible to save the raster intermediary form of all vector converted pictures with the NAME RASTER command; TDP will use these files in preference to the original vector files, provided they agree in source, size and orientation. Thus specifying a name for the raster conversions will increase the speed of printing of future copies of the document. Once the NAME RASTER command has been used TDP will use the specified filename to generate a sequence of raster files to hold the converted figures (for example, if the filename specified is RAST000, and there are three figures in the document, TDP will generate raster files: RAST000, RAST001 and RAST002).

Note that use of the NAME command will result in a table being generated (CONTENTS, INDEX or FIGURE) even if that table has previously been disabled. The table can subsequently be disabled if required.

Limitations

No account name can be included on the filename. Any groupname should only be included with care, because of security restrictions.

Options

[CONTENTS] Use this option to name the generated Table of Contents file.

[INDEX] Use this option to name the generated Index file.

[TABLE FIGURE] Use this option to name the generated Table of Figures file.

[RASTER] Use this option to name a generated raster file.

Examples

Example 1: `\NAME CONTENTS CONTABLE`

This command will, if necessary, build a file named CONTABLE and, during the formatting process, write the Table of Contents of the document to that file. The result will be a TDP file which contains formatting command and text representing the Table of Contents. In order to print this as a document, issue a FINAL FROM CONTABLE command.

Example 2: `\NAME INDEX DOCINDEX`

This command will, if necessary, build a file named DOCINDEX and, during the formatting process, write the Index entries to that file. The resultant file will contain just the index entries, along with page references where specified. This file can then be edited, sorted and printed by the user.

Example 3: `\NAME RASTER RAST000`

This command will cause the next ILLUSTRATION command which does a vector conversion to use the file RAST000 as its temporary file. The next raster file after that will be RAST001 and so on. If any of these files exist already, and contain the raster form of the vector file being converted, then the raster file will be used. If for example a TDP file contains 3 ILLUSTRATION commands, then the first vector file picture will generate raster file RAST000. The second ILLUSTRATION will generate raster file RAST001 and the last ILLUSTRATION will generate raster file RAST002.

Related commands

CONTENTS
ILLUSTRATION
INDEX
TABLE FIGURE

NEED

Purpose

To force a new page if a certain block of material will not fit on the current page.

Form

```
\NEED  {n}
       {B}
```

Description

The NEED command is used to ensure that a certain block of material is not broken between two pages. For instance, you might want a chart to appear all on the same page.

When NEED n is used, TDP checks to see if n lines will fit on the current page; if not, a new page is started.

When NEED B is used, TDP checks to see if the next block of lines will fit, and if it will not a new page is started. (A block is considered to start with the next line, and continue until a blank line or formatting command line is encountered.)

Limitations

In format mode, the \NEED B option may not be accurate. The formatter will "look ahead" to find the number of lines before a \command or blank line is found. These lines, however, have not yet been formatted, so the count may not be correct. \NEED n is more accurate.

NEED is not allowed in headers or footers.

Options

None.

Example

```
\NEED 15
```

If fewer than 15 lines are left on the current page, the next line in the file will be placed on a new page. If 15 lines are left, processing continues as usual.

Related commands

NEW
SPACE
WIDOW

NEW

Purpose

To force the start of a new page of output.

Form

```
\NEW [EVEN [BLANK ]]
     [ODD  [string]]
```

Description

The NEW command is used to start a new page of output. The next line in the file starts a new page. (Footings and the page number, if they have previously been specified, will be printed before the new page is started.) If TDP is already at the top of a new page, the command is ignored. (In the case of laser printers, NEW causes a new logical page to be started.)

The [string] parameter specifies a string to be printed centrally on a skipped page.

Limitations

NEW is not allowed in headers or footers.

Options

`[ODD]/[EVEN]` The ODD/EVEN options will force output to start on the next odd/even numbered page. If no other option is supplied and the current page is odd/even numbered then the next page will be skipped.

`[BLANK]` The BLANK option allows the skipped page to be printed with no data. The string parameter would print the string centrally on the skipped page.

Examples

Example 1: `\NEW`

Output will begin on the next page, unless we are already at the top of the page.

Example 2:

```
\HEAD " "
\FOOT " "
\NEW ODD "This page left intentionally blank"
```

This will print the string "This page left intentionally blank" centrally if we are currently printing on an even numbered page and not at the top of the page.

Related commands

LAYOUT NEW
PAGENO
^#P

NEXT

Purpose

To start a new column of output when controlled multi-column output is being produced.

Form

```
\NEXT
```

Description

When the CONTROLLED option is used with the COLUMN command, the NEXT command is used to indicate the start of each new column. In CONTROLLED columns, NEXT in the last column starts output in column one on same page.

In UNCONTROLLED columns, NEXT in the last column causes a new page to begin. Output starts in column one of the new page.

Limitations

None.

Options

None.

Example

```
\COLUMN 2, CONTROLLED
.
.
.
(column 1 material)
.
.
.
\NEXT
.
.
.
(column 2 material)
.
.
.
\COL 1
```

The output for this example is in two columns:

```
Column 1 material        Column 2 material
.................        .................
.................        .................
.................        .................
```

At this point, the formatter returns to one-column output.

Related commands

COLUMN

PAGE

Purpose

To print selected pages of the document.

Form

```
\PAGE [#]startingpage [/[#]endingpage]
```

Description

The PAGE command allows printing of selected pages when the ASK option has been specified with the FINAL or DRAFT commands. The pages to be printed may be selected on the basis of page number or their location in the document. To specify page numbers, use starting page/ending page; to specify relative location in the document, use #starting page/#ending page. To begin printing, issue a GO or TRY command.

Limitations

If pages were numbered with strings (A,B,C...), you must use the #starting page option. PAGE A or PAGE "A" will not work.

Options

None.

Examples

Example 1:

```
/FINAL,ASK
\PAGE 1/3; GO
```

This example prints pages numbered 1 through 3, not necessarily the first three pages in the document.

Example 2:

```
\PAGE #1/#3; GO
```

This example, after a FINAL, ASK command, prints the first, second, and third pages in the document. Their page numbers might be i, ii, and iii; or 1A, 1B, 1C; or III-1, III-2, III-3; or they may be unnumbered.

Example 3:

```
\PAGE 6; GO
```

This example, after a FINAL, ASK command, prints page 6 only.

Related commands

/DRAFT (Chapter 3)
EXIT
/FINAL (Chapter 3)
GO
MARGIN
PAUSE
TRY

PAGELENGTH
PAGEL

Purpose

To define the total number of lines available per page.

Form

```
\PAGELENGTH n
```

Description

The PAGELENGTH command defines the number of lines per output page, including the top and bottom margin. The default pagelength is normally 66 lines (assuming six lines per inch on an eleven inch piece of paper). Subtracting the default top and bottom margins (6 lines each), that means 54 lines are available for text in the default condition.

Limitations

There must be at least 2 print lines on the page, and the maximum value is 256.

Options

None.

Example

```
\PAGELENGTH  84
```

This command sets the pagelength to 84 lines, the size of standard legal size paper.

Related commands

BOTTOM
TOP

PAGENO

Purpose

To set automatic page numbering on, and indicate the starting page number.

Form

```
\PAGENO {n      } [,A [,REV]}
        {string} [,REV     }
                 [,I       }
                 [,L       }
                 [,C       }
                 [,R       }
```

Description

The PAGENO command is used to initiate TDP's automatic page numbering feature, and to specify the beginning page number. The command can also be used to manually insert page numbers on each page in the text. The page numbering can be integer (use parameter n) or character (use the string parameter).

The page number is placed on the bottom of the page, on the right-hand side unless an option is selected that changes that position. (Vertical placement of the page number is controlled with the PAGENOLINE command.)

To terminate page numbers, enter \PAGENO -1.

To set a page number and place it somewhere other than the bottom of the page, see the #P command.

If any positional option is supplied, then it will override any value entered via FOOT, PAGENOLINE, SECTION or ALTERNATE. If no option is supplied the placing of the page numbers will be determined by the last setting supplied to any of these commands. Also if a positional option is supplied on any of these commands later in the input, then that value will override the value supplied by this command.

Limitations

The [string] parameter must be 20 characters or less. The parameter [n] is limited to integers. Note that if the string parameter is specified you must ensure that the string (and hence the numbering sequence) is long enough to cater for the the length of the document, since there is no overflow provided. (For example, if the page number is specified as "A", then after page "Z" page numbering will revert to page "A". However, if " A" had been specified as the start of page numbering, then after page "Z", numbering would continue with page "AA".This point is illustrated by Examples 3 and 4.)

If a #P command is issued after a PAGENO command, automatic page numbering will be turned off and the page value will assume that given in the #P command.

PAGENO

Options

[L] or [I] The page number is printed on the right-hand side of the page by default. Use the L or I options to print it on the left-hand side.

[C] Use the C option to print it in the center.

[R] Use the R option to reset it to the default condition when one of the options has been used.

[A[,REV]] It is also possible to have the page number alternate between the right- and left-hand sides of the page. For even numbered pages (if the page number is alphabetic then B,D,F,H,.. are even) the page number is placed on the left-hand side. This can be achieved by use of the A option. The alternation can be reversed by the REV option alone or following the A option.

The ALTERNATE command can also set the position of the page number if no option is supplied.

Examples

Example 1: `\PAGENO 1`

This example starts numbering the pages, beginning with 1 and incrementing by 1 for each new page. The page numbers are placed on the right side of the page by default.

Example 2: `\PAGENO 22, CENTER`

This example starts numbering the pages, beginning with 22 and incrementing by 1 for each new page. The page numbers are centered between the left and right margins.

Example 3: `\PAGENO "A"`

This example numbers the pages A,B,C, . . . ,Z,A,B, etc.

Example 4: `\PAGENO "  A"`

In this example, pages will be numbered A,B, . . . ,Y,Z, . . . ,AAA,AAB, . . . ,ZZY,ZZZ.

Related commands

ALTERNATE
NEW
PAGENOLINE
^#P
#P

PAGENOLINE

PAGENOL

Purpose

To specify the vertical placement of the page number.

Form

```
\PAGENOLINE n [,A [,REV]]
              [,REV     ]
              [,I       ]
              [,L       ]
              [,C       ]
              [,R       ]
```

Description

The PAGENOLINE command specifies either the exact line within the bottom margin, or the number of lines from the bottom the page number is to be printed.

If PAGELENGTH=66 (default value), then \PAGENOLINE 62 is equivalent to \PAGENOLINE -4. A positive number indicates lines from the top of the page; a negative number indicates lines up from the bottom edge of the page. The default value is -4.

If any positional option is supplied, then it will override any value entered via FOOT, PAGENO, SECTION or ALTERNATE. If no option is supplied the placing of the page numbers, footings and section will be determined by the last setting supplied to any of these commands. Also if a positional option is supplied on any of these commands later in the input, then that value will override the value supplied by this command.

Limitations

The value of n must not be zero.

Options

[L] or [I] The page number is printed on the right-hand side of the page by default. Use the L or I options to print it on the left-hand side.

[C] Use the C option to print it in the center.

[R] Use the R option to reset it to the default condition when one of the options has been used.

[A[,REV]] It is also possible to have the page number alternate between the right- and left-hand sides of the page. For even numbered pages (if the page number is alphabetic then B,D,F,H,.. are even) the page number is placed on the left-hand side. This can be achieved by use of the A option. The alternation can be reversed by the REV option alone or following the A option.

The ALTERNATE command can also set the position of the page number if no option is supplied.

PAGENOLINE
PAGENOL

Examples

Example 1: `\PAGENOLINE -2`

This example prints the page numbers two lines from the bottom of the page on the right-hand side.

Example 2: `\PAGENOLINE -3,ALT`

This example prints the page numbers three lines from the bottom of the page, on the right-hand side for odd-numbered pages, and on the left-hand side for even-numbered pages.

Related commands

ALTERNATE
NEW
PAGENO
^#P
#P

PARAGRAPH

PARA or P

Purpose

To cause a paragraph break without a blank line.

Form

```
\PARAGRAPH
```

Description

The PARAGRAPH command allows the user to cause a paragraph break without any blank line. In format mode TDP will start a new paragraph whenever a blank line is found, however, if no blank line is required then this command can be used. The same effect can be produced with just a PROMPT character (see the PROMPT command).

Limitations

This command is not allowed in headers and footers.

Options

None.

Examples

```
This is an example in which a paragraph
is formatted in two different sections
\PARAGRAPH
without a space between the two sections.
```

would be formatted as:

> This is an example in which a paragraph is formatted in
> two different sections
> without a space between the two sections.

Related commands

SPACE

PAUSE

PAU

Purpose

To suspend output at the end of each page.

Form

```
\PAUSE [ON ]
       [OFF]
```

Description

The PAUSE or PAUSE ON command suspends printing at the end of each page of output until you signal TDP to continue. This allows you to adjust the paper, put a fresh sheet of paper in the printer (when using separate sheets of letterhead, for example), and so forth. When you are ready to print the next page, press the space bar or a carriage return to cause processing to re-commence. On the HP 2601 press the (CONTINUE) button.

PAUSE is ignored if output is going to a line printer or a printer using a sheet feeder, or a printer with automatic form feed set.

Limitations

PAUSE will be ignored if it is issued in an INCLUDE file.

Options

[ON] The ON option turns the command on, causing TDP to pause after every page is printed. The command remains in effect until turned off.

[OFF] Use to turn the command off.

Related commands

/DRAFT
/FINAL
GO
MARGIN
PAGE
TRY

PITCH

PI

Purpose

To set the pitch for the output device.

Form

```
\PITCH (10)
       (12)
       (15)
```

Description

Some printers allow the pitch to be changed between various values, between 10, 12 and 15, for example. The PITCH command sets this value to the desired setting. For printers that can have their pitch set TDP will reset the value to that supplied. If no value is given then 10 pitch is assumed.

Limitations

A PITCH command must be given before any output lines (including blank lines) or SPACE command.

This command is effective only for those printers that support multiple pitch settings, for all other devices the line will be extended to simulate the appropriate pitch setting.

Options

None.

Examples

```
\PITCH 12
```

This command sets the pitch to 12.

Related commands

LFT
PROPORTIONAL
RHT
WIDTH

PROMPT
PR

Purpose

To change the prompt character for formatting commands.

Form

```
\PROMPT [char]
```

Description

A PROMPT character in column 1 indicates that the remainder of the line is a command string rather than text. It is also used to prompt you for input when using the ASK option under the FINAL or DRAFT commands.

The PROMPT character may also be applied by itself to indicate a formatting break. If in format mode, the effect will be that formatting is terminated at the line preceding the PROMPT character and resumed at the line following the PROMPT character without providing a blank line between the two sets of text.

The default prompt character is a backslash (\). This command sets that prompt to any character you choose.

Limitations

None.

Options

None.

Example

```
\PROMPT = "?"
```

This example sets the prompt character to the question mark sign. The next command line might look like this:

```
 37  ?Center 1
```

where ? designates the line as a command line.

Related commands

BACKSPACE
BLANK
ESCAPE

PROPORTIONAL
PROP

Purpose

To force the characters in the output to be proportionally spaced.

Form

```
\PROPORTIONAL [METAL  ]
              [PLASTIC]
```

Description

The PROPORTIONAL command allows you to state that the characters are to be printed proportionally. This means that the characters will only occupy the space they need to.

Limitations

This command can only be used on those printers that support proportionally spaced characters.

If the 2601 printer is to be used in proportional mode then the type of wheel being used must be specified before any output is generated.

Options

`[METAL]/[PLASTIC]` The METAL/PLASTIC option allows you to specify the type of wheel being used on the 2601 printer.

Example

```
 \PROPORTIONAL METAL
```

This states that the output will be proportionally spaced and that a metal wheel is being used on the 2601

Related commands

PITCH
WIDTH

RED

Purpose

To print one or more lines on red ribbon.

Form

```
\RED [n]
```

Description

The RED command prints the specified number of lines in red. The entire line is printed in red. If no value for [n] is specified, only the following line is printed in red. See also the ^R command.

Limitations

This command can only be used on those printers which support two-colored ribbons (normally black and red).

Options

None.

Examples

```
\RED 3
```

Will print the three lines following the command in red if a black and red ribbon is on the printer.

Related commands

BOLD
GHOST
^B
^G
^R

REVISION
REV

Purpose

To flag revised lines in the final printed copy.

Form

```
\REVISION {MARK [char] }
          {OFF        }
          {ON         }
```

Description

Ordinarily, proof-marked lines are not noted on output printed with the FINAL command. The REVISION command is used to flag revised lines in the final output; this will only be output if PMARK has been set and the file has been updated since PMARK was set. The character to be used as the revision mark is set using the REVISION MARK command and is printed at the end of of the revised line. A single character only is allowed.

There are two kinds of revision marking available. TDP revision marking is set using the /SET PMARK command. Any lines which have been updated since PMARK was set are flagged as revised. User revision marking is controlled through the REVISION ON and REVISION OFF commands. REVISION ON initiates revision marking, and all lines are marked as revised until the REVISION OFF command is given.

In draft output, proof marks are printed at the end of changed lines if TDP revision marking is being used, and "REVISED" is printed if user revision marking is set.

Limitations

An empty string "" is not an acceptable value. However, space " " is allowed.

In draft multiple column output, TDP proof marks are not printed, by lines are marked as "REVISED".

Options

None.

Examples

```
\REV MARK = "&"
```

This example sets the ampersand as the revision mark used when final output is printed. A revised line will be printed as follows:

```
The next line was revised after PMARK was set.
So it is marked with an & in the right margin.    &
```

Related commands

RMARGIN
/SET PMARK (Chapter 3)

RHT
RH

Purpose

To set the right margin.

Form

```
\RHT n
```

Description

The RHT command sets the right margin to be used when formatting the file for output. The default margin is 75 for a 10 pitch output device; 90 for a 12 pitch device. Setting RHT without setting LFT causes the right margin to be set to the value specified by RHT plus the default value of LFT.

To change the margin later in the file, use INLFT or INRHT.

Limitations

The RHT margin must be set prior to the first line of output (including blank lines) or a \SPACE command, either at the start of formatting or following a LAYOUT command. Note that the maximum value for n is 158.

Options

None.

Example

```
\RHT 65
```

This example sets the right margin at column 65.

Related commands

INLFT
INRHT
LAYOUT
LFT

RIGHT

Purpose

To shift all lines to the right so the last character in each line is at the right margin.

Form

```
\RIGHT [n]
       [B]
```

Description

The RIGHT command moves a specified [n] number of lines to the right so that the last character in each line is at the right margin. The command moves the entire line as many spaces to the right as is necessary to justify the right margin. In this respect, it is different from the JUSTIFY command used within format mode, where lines are filled and spaces inserted or words hyphenated to justify the left and right margins.

If no parameter is specified, all lines are right justified until another mode command is encountered (IMAGE, CENTER or FORMAT).

Limitations

None.

Options

[B] The B option right-justifies the next block of lines, continuing until a formatting command line (normally signified by a \ character) or a blank line is encountered.

Example

Suppose your workfile looks like this:

```
\center 2
1988 ANNUAL REPORT
SAN JOSE WATER RESOURCES BOARD
\SPACE 2;right b
Prepared by:
Smith and Jones, Inc.
Certified Public Accountants
```

The output would be as follows:

```
                1988 ANNUAL REPORT
          SAN JOSE WATER RESOURCES BOARD



                                            Prepared By:
                                   Smith and Jones, Inc.
                            Certified Public Accountants
```

Related commands

CENTER
FORMAT
JUSTIFY

RMARGIN
RM

Purpose

To set the size of the right margin. In draft copies RMARGIN sets the margin between the end of the line of text, and line numbers and proofmarks. In final copies it sets the margin for placing revision marks.

Form

```
\RMARGIN n
```

Description

The RMARGIN command sets the right margin between the end of the line of text and the line numbers and proofmarks printed on draft copies of the document. Marks are printed one space to the right of the right margin. So, if RMARGIN is zero, revision marks will be printed one space to the right of the text.

The default right margin is roughly 10% of the line width. The RMARGIN command can be used after a \LAYOUT command.

Limitations

The RMARGIN command must precede any output lines (including blank lines), or a \SPACE command, either at the start of formatting or following a LAYOUT command.

Options

None.

Examples

```
\RMARGIN 10
```

This example sets the margin between the end of the line of text and the line numbers or proof marks to 10 spaces.

Related commands

REVISION MARK

SECTION

SE

Purpose

To specify a section title to be printed immediately before the page number.

Form

```
\SECTION string [,A [,REV]]
                [,REV     ]
                [,I       ]
                [,L       ]
                [,C       ]
                [,R       ]
```

Description

The SECTION command is used to specify a title to be printed just before the page number. This might be used to identify an appendix, or to force numbering within each chapter or section. The string also appears in the table of contents next to the page number.

If any positional option is supplied, then it will override any value entered via FOOT, PAGENO, PAGENOLINE or ALTERNATE. If no option is supplied the placing of the section title will be determine by the last setting supplied to any of these commands. Also if a positional option is supplied on any of these commands later in the input, then that value will override the value supplied by this command.

Limitations

The maximum length allowed for the string is 158 characters.

The section title is only printed if pages are being numbered with the PAGENO command, and a footing of one line or less is specified.

Options

[L] or [I] The section title is printed on the right-hand side of the page by default. Use the L or I options to print it on the left-hand side.

[C] Use the C option to print the section title in the center.

[R] Use the R option to reset the section title to the default condition when one of the options has been used.

[A[,REV]] It is also possible to have the section title alternate between the right- and left-hand sides of the page. For even numbered pages (if the page number is alphabetic then B,D,F,H,.. are even) the section title is placed on the left-hand side. This can be achieved by use of the A option. The alternation can be reversed by REV or A,REV.

The ALTERNATE command can also set the position of the section title if no option is supplied.

SECTION
SE

Examples

Example 1: `\SECTION "Appendix A- "`

This example prints the title Appendix A immediately in front of the page number. The first page would have Appendix A-1 on the right side of the bottom of the page.

Example 2: `\SECT "III-",ALT`

This example would be used to force independent page numbering within each section. The page number must be reset to one at the start of each section, and a SECTION command used to insert the correct numeral. The first page would have III-1 on the right hand side of the bottom of the page; the second page would have III-2 on the left-hand side, and so forth.

Related commands

ALTERNATE
FOOT
FOOTNOTE
HEAD
HEADLINE
^#F
^#S

SKIP

Purpose

To skip over lines in the file to be formatted.

Form

```
\SKIP (n)
```

Description

The SKIP command causes the next n lines in the file to be ignored. The command can be used to suppress listing of comments that have been included in the file, or with the IF command to conditionally ignore lines of the file.

Use of SKIP causes a paragraph break.

Limitations

Nothing can follow SKIP on the same line. If you wish to make a comment following a SKIP command, it should be placed on the following line.

SKIP is not allowed in headers or footers.

Options

None.

Example

```
\SKIP 3
```

This command effectively ignores the next three lines of text. The lines are not printed as output, nor "read" as commands; they are completely ignored.

Related commands

EOD
IF

SPACE
SP

Purpose

To leave a specified number of blank lines in the output.

Form

```
\SPACE [EVERY] n
```

Description

The SPACE command inserts a specified number of blank lines in the output. When the end of the page is reached before the specified number of lines has been inserted, the command is terminated and output continues at the top of the next page.

If a \SPACE n command occurs as the first line of a new page after automatic pagination, the command is ignored. However, if the new page was forced by a NEW command (or a column started by \NEXT), then the spaces will be inserted.

Limitations

SPACE is not allowed in headers or footers.

Options

[EVERY] The option EVERY n causes a blank line to be inserted in the output every n lines. To terminate SPACE EVERY use SPACE EVERY 0.

Examples

Example 1: `\SPACE 10`

This example leaves 10 blank lines, or starts a new page if fewer than 10 lines remain on the current page.

Example 2: `\SPACE EVERY 5`

A blank line is inserted in the output after every five lines of output. The count is reset whenever there is a blank line or page break.

Related commands

BOX
DOUBLE
FIGURE
LINESPACE

TABLE FIGURE

Purpose

To control the inclusion of data in the Table of Figures file.

Form

```
\TABLE (FIGURE) (string) [,IN n            ]
                (BEGIN ) [,NEW             ]
                (ON    ) [,SPACE n [AFTER]]
                (OFF   ) [,NEED n          ]
                         [,NO[DOTS]        ]
    (END)
```

Description

The TABLE FIGURE command allows you to control the inclusion of data in the Table of Figures file. When the document is printed, the Table of Figures is constructed and stored in a temporary file, then output after the last page of the document, or the Table of Contents. The Table of Figures is printed without a heading or page number. The temporary file can be saved permanently using the "NAME" command. Entries in the table of figures are as a result of "TABLE FIGURE" and "FIGURE" commands. Options allow you to control the format of the Table of Figures, and to enter new strings that will appear in the Table of Figures instead of existing titles of figures.

The [string] parameter allows you to insert the specific string in the Table of Figures file, referenced to the current page, rather than using a figure command with a title. The page number used will be that on which the command was executed, not the one on which the figure was placed.

Limitations

None.

Options

Each TABLE FIGURE command resets all the options which are not specified in the particular command.

(BEGIN)/(END) The first BEGIN/END option is used to supply a Table of Figures title other than "TABLE of FIGURES", provided it precedes the first \FIGURE command. Formatter commands to alter the margins, spacing, underlining and so forth can be entered beginning on the line following the \TABLE FIGURE BEGIN command, and must be followed by \TABLE END. This is particularly useful when printing a document in another language.

Subsequent BEGIN/ENDs are used to insert information into the Table of Figures that does not appear in the document. Any lines between BEGIN and END are printed in the Table of Figures, but not printed in the document. No reference to the page is included.

TABLE FIGURE

(ON)/(OFF) The ON/OFF options can be used to turn off the Table of Figures when there are \FIGURE commands in the file, but you do not want or need a Table of Figures. This is useful when printing multiple copies of a portion of the document. Note initially the Table of Figures is off and so will need to be turned on before TABLE FIGURE commands are actioned.

[NEW] The NEW option allows you to force a new page in the Table of Figures before the next entry.

[SPACE n [AFTER]] The SPACE n [AFTER] allows you to insert blank lines before (SPACE n) or after (SPACE n AFTER) the next entry.

[IN] The IN n option causes the next entry to be indented n spaces from the left margin.

[NO[DOTS]] The NO[DOTS] option suppresses the printing of the dots between the entry and the page number (NODOTS), or the dots and the page number (NO).

[NEED] The NEED option functions the same as the NEED command, however, it is limited to numeric values.

Examples

Example 1:

```
\TABLE FIGURE BEGIN
\CENTER 1;UL 1
List of Illustrations
\SPACE 2
\HEAD "List of Illustrations (Continued)"
\SPACE EVERY 5
\TABLE END
```

This example sets the title for the Table of Figures to be "List of Illustrations": succeeding pages will be headed "List of Illustrations (Continued)". After the \TABLE END command, you use the \FIGURE command to specify the entries in the Table of Figures.

Example 2: `\TABLE FIGURE ON,NEW,IN 5`

This example will cause a new page in the Table of Figures and set the next entry in by 5 spaces. The next entry comes from the next \FIGURE command encountered.

Related commands

CONTENTS
INDEX
NAME

TOP

Purpose

To set the top margin size.

Form

```
\TOP n
```

Description

The TOP command defines the number of lines to be left from the top of the page to the first line of formatted text. The default setting is six lines.

The heading must be included in the space set by TOP, so make certain you leave enough room if it is a multiple line heading. Otherwise, you will have an error message in the output.

Limitations

The size of the top margin, n, must be greater than or equal to 1, and less than or equal to the pagelength minus the size of the bottom margin minus two. Note that the first line of a page may not be used for any purpose; thus, the minimum size for TOP is 1.

Options

None.

Examples

```
\TOP 10
```

This example sets the top margin to 10 lines.

Related commands

BOTTOM
HEAD
PAGELENGTH

TRY

Purpose

To test various formatting values through FINAL,ASK or DRAFT,ASK until the results are satisfactory.

Form

```
\TRY
```

Description

The TRY command is only used when the ASK option is included in a FINAL or DRAFT command. If formatting is started with TRY, TDP will produce the requested output and will then restart the sequence as though the FINAL,ASK or DRAFT,ASK command was just entered. This is to allow you to see what the output is like with one particular set of parameters before producing a final copy.

Limitations

TRY will only execute if it is the last command entered after the ASK option.

Options

None.

Example

```
\LFT 10;RHT 70;TRY
```

This example will set the left margin to 10, the right margin to 70, then format the text. Upon completion, it will return with a formatter prompt character to allow you to change the left or right margins if they did not produce the desired result. If the result was satisfactory, you can enter EXIT to return to the Editor.

Related commands

COPIES
/DRAFT
EXIT
/FINAL
GO
MARGIN
PAGE
PAUSE

UL
U

Purpose

To underline entire lines of text, including spaces between words.

Form

```
\UL [n  ]
    [ON ]
    [OFF]
```

Description

The UL command underlines the line(s) following the command from the first non-blank character to the last non-blank character, including the spaces between words. Thus, if a short line is centered on the page, the leading and trailing blanks are not underlined. The parameter [n] specifies the number of output lines to be underlined. If the parameter is not specified, only the following output line is underlined.

Limitations

None.

Options

[ON] The ON option enables all following ^U commands and produces underlining in lines containing ^U commands.

[OFF] Disables all following ^U commands.

Example

```
16 \UL; CENTER 1
17 This is the title
```

This example centers and underlines line 17 as follows:

This is the title

Related commands

UW
^U
^W
^_

UW

Purpose

To underline all words (non-blank characters) in the following line(s).

Form

```
\UW [n  ]
    [ON ]
    [OFF]
```

Description

The UW command underlines only the non-blank characters on the specified [n] number of output lines (spaces between words are not underlined). If the parameter is not specified, only the following output line is processed.

Limitations

None.

Options

[ON] The ON option enables all following ^W commands and will produce underlining of words in lines containing ^W commands.

[OFF] Disables all following ^W commands.

Examples

```
16 \UW;CENTER 1
17 This is the title
```

This example centers and underlines line 17 as follows:

This is the title

Related commands

UL
^_
^U
^W

VERIFY

Purpose

To display the setting of various parameters so as to aid the debugging of any problems encountered while formatting a document.

Form

```
\VERIFY [PAGES      ]
        [FONTS      ]
        [ENVIRONMENT]
        [ALL        ]
```

Description

It is sometimes useful to check what the TDP formatter considers the current setting of various parameters is; e.g. LFT, RHT etc. These values can be displayed using the VERIFY command.

It should be noted that the current setting of the parameters is not necessarily the values actually entered. This is particularly true of the values of LFT and MARGIN.

Limitations

The [PAGES], [FONTS] and [ENVIRONMENT] options are only valid if output is being sent to the HP 2680 or the HP 2688 laser printers.

Options

[PAGES] The PAGES option displays the size and orientation of the logical pages in the environment file; whether they are active or not; and whether they are wide enough for the current line of output.

[FONTS] The FONTS option displays the size, orientation and name of all available fonts in the environment file.

[ENVIRONMENT] The ENVIRONMENT option displays the environment filename; the current character width; the current line height; and all the information available under the PAGES and FONTS options.

[ALL] The ALL option displays all the information that can be verified. It displays the same information as VERIFY ENVIRONMENT and VERIFY. However, it also provides details on the heading, the footing, hyphenation and the setting of flags such as WIDOW, PAUSE and UNDERLINE.

WIDOW

Purpose

To force single lines of a paragraph to be printed at the top or bottom of a page if they occur in the course of formatting.

Form

```
\WIDOW [ON ]
       [OFF]
```

Description

When formatting paragraphs of text, the situation often occurs in which the first line of a paragraph would be printed at the bottom of a page with the remainder of the paragraph on the next page (referred to as a widow line), or all but the last line of a paragraph might be printed at the bottom of a page with the last line printed on the following page (referred to as an orphan line). Normally, TDP looks for these situations and never allows fewer than two lines of a paragraph to be printed on a page. However, in some cases it is required that printing be continuous across a page boundary without regard for widow or orphan lines. In such a case, the WIDOW OFF command allows the user of TDP to print the text exactly as formatted. If no option is given in the command, ON is assumed.

Limitations

None.

Options

[ON] The ON option invokes Widow/orphan control. All lines following are under Widow/Orphan control until the command is turned off.

[OFF] Turns the command off, and terminates Widow/Orphan control.

Related commands

NEED

WIDTH

Purpose

To specify the width of the output device to be used.

Form

```
\WIDTH n [INCHES]
```

Description

The WIDTH command is used to specify the width of the output device either in terms of characters or inches. This command will set the outer limits that can be used with the RHT and LFT commands. If the INCHES option is used then the pitch of the device must be specified before this command. The command should also precede any text output.

Limitations

The WIDTH command must precede any output lines or SPACE command. It should also precede the LFT or RHT commands.

For a laser printer, the maximum width depends upon the environment.

Options

`[INCHES]` The INCHES option allows the width [n] of the output device to be specified in inches.

Examples

Example 1: `\WIDTH 60`

The result of this example is that the width of the output device is set to 60 characters.

Example 2: `\WIDTH 7 INCHES`

This example sets the width to 7 inches which represents 70 characters on a 10 pitch device or 84 characters on a 12 pitch device.

Related commands

LFT
MARGIN
LPTOP
PITCH
PROPORTIONAL
RHT
LAYOUT

*

Purpose

To allow comments to be included in a TDP formatter file.

Form

```
\* any text
```

Description

The * command allows comments to be included in a TDP formatter file. The text following the * will be ignored when the file is formatted. The * command provides a useful means of causing formatter commands to be ignored, by turning them into comments. For example, you might turn off the command \SKIP 14, in your file, by changing it to *SKIP 14. This way the command can be left in the file for future use, without later having to recount the number of lines you wish to skip. Similarly, you can selectively turn off commands in lines containing more than one command; putting an asterisk in front of the first command in the line affects only that one command.

Limitations

This is a formatter command and appears on a command line. If there is a semi-colon in the text following *, the text following the semi-colon will be treated as a new command line.

Options

None.

Example

```
\* The settings of RHT, LFT, etc. should follow now:
```

This tells you where you should include any values of RHT, LFT in the file.

Purpose

To shift to the alternate character set.

Form

```
^A(text)^N
```

Description

The ^A command switches to the alternate character set. (In the case of laser printers, the alternate character set will be the font identified as the alternate character set by a FONTEQ ALTERNATE command or the second of a pair of fonts identified in a FONTID command. In the case of the Agile Twin-Track printer, the alternate character set will be the second print wheel.) To shift back to the normal character set, use the ^N command.

Limitations

This command is only effective on devices that support the alternate character set.

Options

None.

Example

Assuming that the alternate character set is italics, the following line shows the use of the ^A.

```
    26  Print just one ^Aword^N in italics.
```

The word *word* is printed in italics, the rest of the sentence in normal type.

Related commands

FONT
FONTEQ
FONTID
^B
^G
^N
^F

^B

Purpose

To print in bold face type.

Form

```
^B(text)^S
```

Description

The ^B prints characters in bold face type on printers which support bold face type (each character is actually printed twice). In the case of laser printers, the characters are printed in the font identified by means of the FONTEQ BOLD command. To return to regular type, use the ^S command.

Limitations

This command is effective only for those printers which support bold face output.

Options

None.

Example

```
16      One word in the sentence must be ^Bemphasized^S.
```

This example prints the word **emphasized** in bold type.

Related commands

GHOST
BOLD
FONTEQ
RED
^A
^N
^R
^G
^S

^C

Purpose

To send a control character to the printer.

Form

`^C{char}`

Description

The ^C command is used to send a control character to the printer. This is used with some printers to set special capabilities available on them; or it could be used to ring the bell, and so forth.

Limitations

None.

Options

None.

Examples

Example 1: `^CG`

This command will ring the bell on the printer whenever it is encountered by TDP during output.

Example 2: `^C[\`

This sequence will set unidirectional printing on a HP 2601 printer.

^D

Purpose

To automatically insert the current date.

Form

```
^D{0}     6/10/88
  {1}     June 10, 1988
  {2}     Friday, June 10, 1988
  {3}     Friday, June 10, 1988, 2:54 PM
  {4}     June 10
  {5}     1988
  {6}     2:54 PM
  {7}     88
  {8}     Friday
  {9}     10/6/88
  {A}     10 June, 1988
  {B}     Friday, 10 June, 1988
  {C}     Friday, 10 June, 1988, 2:54 PM
  {D}     10 June
  {E}     10.6.88
  {F}     10/6/1988
  {G}     6/10/1988
  {H}     6.10.1988
  {I}     06.10.1988
  {J}     Fri
  {K}     FRI
  {L}     10
  {M}     June
  {N}     Jun
  {O}     JUN
  {P}     6
  {Q}     2:54
  {R}     14:54
  {S}     PM
  {T}     161
```

Description

The ^D command is used to insert the current date or time in your output. The options allow you to choose the form in which the date is presented as shown above.

Limitations

None.

Options

The optional date formats are illustrated above.

Examples

Example 1: `It is now ^DB and I have not yet received a reply to`

This example would result in the following:

It is now Friday, 10 June, 1988 and I have not yet received a reply to

Example 2: `^DM ^DL ^D5`

This example shows how you can combine date elements to produce a date in any format you choose. Here the date produced is: June 10 1988

Related commands

^#
^#F
^#P
^#S
^M

^E

Purpose

To expand a line by moving the right-hand portion to the right-hand margin.

Form

```
^E(char)
```

Description

The ^E command effectively splits a line into two parts, leaving the left part where it is, and moving the right part to the right margin. Any blank spaces in the line are filled by the character immediately following ^E. Any character (including blanks) can be specified to fill the space between the two parts of the line.

When the ^E command is used in format mode and there is text on the preceding or following lines, the text is first formatted in the normal way and only then is the line expanded with the character immediately following the ^E.

Limitations

None.

Options

None.

Example

Suppose your input line is as follows:

```
l7 Left half^E*Right half.
```

Then the output is:

Left half**Right half.

Purpose

To cause a font change.

Form

```
^F(id) text ^S
```

Description

The ^F command is used to switch to another font, which has been set up via the FONTID command. To switch back to normal font use the ^S command. This command remains in effect until turned off by the ^S command, or another font is selected by ^F.

Limitations

This command is only valid when used with the HP 2680 or HP 2688 laser printer. If no font has been associated with that digit or character, an error is generated.

Options

None.

Example

```
^fnTHIS^s  ^fmIS^s  ^fkAN^s  ^ffEXAMPLE^s
of some of the fonts used in this manual.
```

This prints as follows:

THIS IS AN *EXAMPLE* of some of the fonts used in this manual.

Related commands

FONT
FONTEQ
FONTID
^A
^S
^N
^B
^G

^G

Purpose

To print in ghost typeface.

Form

```
^G(text)^S
```

Description

The ^G prints the characters in ghost face type (each character is actually typed twice with a slight offset on the second printing). (For laser printers the font used will be the font specified in a FONTEQ GHOST command.) To return to regular type, use the ^S command. The effect is similar to bold printing.

Limitations

This command can only be used on those printers that support ghost, or shadow, printing.

Options

None.

Example

```
  1   In this example, ^Gghost^S is printed with shadow printing.
```

In this example, the word ghost is printed in the font specified in a FONTEQ GHOST command.

Related commands

BOLD
GHOST
FONTEQ
RED
^A
^N
^R
^S
^B

Purpose

To reference a macro set with the Macro command.

Form

```
^M(id)
```

Description

The ^M command allows you to reference a macro string set with the Macro command. When TDP encounters a ^M command it will replace it with the string it represents.

Limitations

If a macro string causes the line to be extended beyond the right margin in any mode other than format, then the line will be truncated.

Options

None.

Example

See the \M command for examples of the \M and ^M commands used together.

Related commands

M(Macro)

^N

Purpose To switch from the alternate character set to the normal character set.

Form

```
^N
```

Description The ^N command is used to switch back to the normal character set after having switched to the alternate character set with the ^A command.

Limitations None.

Options None.

Related commands ^A
^S

^R

Purpose

To print characters in red.

Form

```
^R{text}^S
```

Description

On those printers which support two-colored ribbons, the intraline ^R command causes characters to be printed in red. To return to standard printing, use the ^S intraline command.

Limitations

This command can only be used on those printers that support two-colored ribbons (normally black and red).

Options

None.

Example

```
   1    In this example, the word ^Rred^S is printed in red.
```

As indicated, the first example of the word "red" will be printed in red ink.

Related commands

RED
BOLD
GHOST
^B
^G
^S

^S

Purpose To stop all intraline enhancements.

Form ^s

Description The ^S command is used to stop the effect of any outstanding intraline commands, with the exception of ^A. All active intraline commands are turned off.

Limitations None.

Options None.

Related commands ^N

^U

Purpose

To underline a section of text.

Form

```
^U(text)^S
```

Description

The ^U command is used to start intraline underlining. Everything in the lines, blanks included, will be underlined until a ^S command is encountered. Note that the underlining will continue indefinitely if none is encountered.

Limitations

None.

Options

None.

Example

The following input line shows the use of ^U:

```
23  When certain words are ^Uto be underlined^S.
```

The output appears like this:

When certain words are to be underlined.

Related commands

UL
UW
^W
^S
^_

^W

Purpose

To underline non-blanks within a section of text.

Form

```
^W(text)^S
```

Description

The ^W command is used to start intraline underlining of words. Unlike ^U, however, it does not underline a special character immediately preceding the ^S. The command remains in effect until a ^S command is encountered.

Limitations

None.

Options

None.

Example

The following input line shows the use of ^W:

```
  23  When certain words are ^W to be underlined !^S
```

The output appears like this:

When certain words are to be underlined !

Related commands

UL
UW
^U
^S
^_

Purpose

To underline a single word within a line.

Form

^_

Description

The ^_ command indicates that the next word in the line is to be underlined. Note that if the word is terminated with a special character, that character will not be underlined.

Limitations

None.

Options

None.

Example

This example underlines the word "word".

```
Let's underline one ^_word.
```

The output is:

Let's underline one word.

Related commands

UL
UW
^U
^W

^+

Purpose

To position characters half a line above the normal line of printing, or to return to the normal line after a ^- has been issued.

Form

```
^+
```

Description

This command provides for the printing of characters one half of a line higher than the normal line position until a ^- is encountered to return printing to the normal position.

Note that if the line spacing is left one half line off the normal line position, when TDP reaches the end of the page (or a footing) it adjusts the spacing to the original position.

Limitations

This command is effective only on those printers that support negative half-line feed.

Options

None.

Example

```
16  The proposal is due the 22^+nd^- of June.
```

This command raises the "nd" one half line.

The proposal is due the 22$^{\text{nd}}$ of June.

Related commands

EQUATION
^-
^>
^<

Purpose

To position characters half a line below the normal line of printing, or to return to the normal line after a ^+ has been issued.

Form

^-

Description

This command provides for the printing of characters one half of a line lower than the normal line position until a ^+ is encountered to return printing to the normal position.

Limitations

This command is effective only on those printers that support half-line feed.

Options

None.

Example

```
    34  The value of X^-i^+ is calculated.
```

This command lowers the "i" one half of a line position.

The value of X_i is calculated.

Related commands

EQUATION
^+
^>
^<

^>

Purpose

To allow the printing of superscripts in a regular input line; or to return to the normal line after a ^< has been issued.

Form

```
^>
```

Description

On printers that offer a superscript capability, you use ^> to indicate a superscript within a line of text. The characters are printed in superscript position until a ^< command returns it to the normal line.

Limitations

This command is effective only on those printers that support superscript capability.

Options

None.

Example

```
13  The company is located on East 88^>th^< street.
```

This command prints the "th" as a superscript.

The company is located on East 88th street

Related commands

EQUATION
^+
^-
^<

^<

Purpose

To allow the printing of subscripts in a regular input line; or to return to the normal line after a ^> has been issued.

Form

```
^<
```

Description

On printers that offer a subscript capability, you use ^< to indicate a subscript within a line of text. The characters are subscripted until a ^> command returns it to the normal line.

Limitations

This command is effective only on those printers that support subscript capability.

Options

None.

Example

```
    72  The equivalent expression is X^<1^>+Y^<2^>.
```

This command prints the 1 and 2 as subscripts of X and Y, respectively.

The equivalent expression is X_1+Y_2.

Related commands

EQUATION
^+
^-
^>

^#n

Purpose

To reference and increment paragraph and section labels.

Form

```
^#n
```

Description

The command is used to reference and increment paragraph or section labels. The ^#n command places the label/number in the text, and automatically increments it.

Numbers, and upper and lower case letters are incremented. The rightmost character is incremented, and if that causes an overflow the next character that can be incremented is increased accordingly.

A "1", "a", or "A" will be placed in the next position to the left, if a blank was left in the quoted string. Otherwise, a new character is placed to the right of the original.

Limitations

None.

Options

None.

Example

```
\image
\#1 = " 1."
\#2 = "    a."

^#1 Objectives

      This paragraph defines short and long-term goals.

^#2 Short-Term Objectives

      The next material would outline short-term objectives.

^#2 Long-Term Objectives

      The next material would outline long-term objectives.

\#2 = "    a."
^#1 Strategies

      This paragraph defines the two different strategies.

^#2 Short-Term Strategies
```

The output looks like this:

```
1. Objectives

    This paragraph defines short and long-term goals.

  a. Short-Term Objectives

    The next material would outline short-term objectives.

  b. Long-Term Objectives

    The next material would outline long-term objectives.

2. Strategies

    This paragraph defines the two different strategies.

  a. Short-Term Strategies
```

Related commands

^#F
#n
^#(n)
^#P
#P
^#S

^#(n)

Purpose

To reference an automatic label/number without incrementing it.

Form

```
^#(n)
```

Description

The ^#(n) references an automatic label/number and places it in the text, but does not increment it.

Limitations

None.

Options

None.

Example

This example shows how numbers and letters are incremented, how to refer to a reference without causing it to increment, and how to reset a reference when needed. Note the result in Section 2 of the example if the paragraph reference is not reset.

```
100     \#1 = " 1"
101     \#2 = "Paragraph a :  "
102     \image
103     ^#1. Section ^#(1)
104         ^#2
105         ^#2
106     ^#1. Section ^#(1)
107         ^#2
108         ^#2
109     \#2 = "Paragraph a :  "
110     ^#1. Section ^#(1)
111         ^#2
112     \#2 = "Paragraph a :  "
113     ^#1. Section ^#(1)
114         ^#2
115         ^#2
116     \#2 = "Paragraph a :  "
117     ^#1. Section ^#(1)
118         ^#2
119         ^#2
120         ^#2
121     \#2 = "Paragraph a :  "
122     ^#1. Section ^#(1)
123         ^#2
124         ^#2
125     ^#1. Section ^#(1)
126     ^#1. Section ^#(1)
127     ^#1. Section ^#(1)
128     ^#1. Section ^#(1)
129     ^#1. Section ^#(1)
130     ^#1. Section ^#(1)
```

The formatted result of this is:

```
 1. Section  1
    Paragraph a :
    Paragraph b :
 2. Section  2
    Paragraph c :
    Paragraph d :
 3. Section  3
    Paragraph a :
 4. Section  4
    Paragraph a :
    Paragraph b :
 5. Section  5
    Paragraph a :
    Paragraph b :
    Paragraph c :
 6. Section  6
    Paragraph a :
    Paragraph b :
 7. Section  7
 8. Section  8
 9. Section  9
10. Section 10
11. Section 11
12. Section 12
```

Related commands

^#n
#n
^#F
^#P
#P
^#S

^#F

Purpose

To access the current footnote number.

Form

```
^#F
```

Description

The ^#F command will access the current value of the footnote number. This command allows the user to save the footnote number which has just been used.

Limitations

None.

Options

None.

Example

```
\SECTION "III"
\FOOTNOTE "^#S.01"
This line of text will contain a footnote now
\FOOTNOTE BEGIN
 This is a footnote which references the section and should
be reset from section to section.
\FOOTNOTE END
\MA="^#F"
.  This follows the footnote.

We can now make a reference to the last footnote by using
the macro a; such as `ma was the last footnote referenced.
```

This line of text will contain a footnote nowIII.01 . This follows the footnote.

We can now make a reference to the last footnote by using the macro a: such as III.01 was the last footnote referenced.

Related commands

^#
^#P
^#S

III.01This is a footnote which references the section and should be reset from section to section.

^#P

Purpose

To reference the page number so that it may be placed within the text.

Form

```
^#P or ^#(P)
```

Description

The ^#P command is used to insert page numbers into the text. It can also be used to number pages within sections (i.e., III-1, III-2, etc.).

The page number is incremented for each new page; if the "string" is defined as "1", there is no overflow provided (after page 9, the number goes back to 0), whereas a definition of " 1" would allow for a value of 99. Both digits and letters can be incremented - special characters are left alone.

^#P may also be referred to in the document, or can be saved in a macro for automatic back references (back only). For example, if you want to reference a certain page later in the document, set \mp="^#(P)" while entering text for that page; anytime later in the document, you can insert that page number as follows:

> Please see example on page ^mp.

A ^#P reference will take the value from a \PAGENO command, if one is given, as well as from a \#P command.

Limitations

None.

Options

None.

Example

```
\#P="IV-1"
\HEAD BEGIN
Introduction, page ^#P
Revised 6/10/88
\HEAD END
```

This example prints a two line heading on each page. The first page looks like this:

Introduction, page IV-1
Revised 6/10/88

Related commands

PAGENO
#P
^#
^#F
^#S

^#S

Purpose

To include the current section into the text.

Form

```
^#S
```

Description

The ^#S command allows you to include the text currently associated with the SECTION command into the middle of your text.

Limitations

None.

Options

None.

Example

```
\SECTION "Cakes and Biscuits"
For more information on chocolate brownies
see the next section ^#S.
```

This produces the following formatted output:

For more information on chocolate brownies see the next section Cakes and Biscuits.

Related commands

SECTION

#n

Purpose

To define the starting value for the automatic labeling/numbering of paragraphs or sections.

Form

```
\#n (string)
    (n      )
```

Description

The #n command is used to specify the starting value for the automatic labeling/numbering of paragraphs or sections. To number the paragraph or section, see the ^#n command. Up to ten automatic labels/numbers can be defined (ie. 0 to 9).

Limitations

If the starting value is specified by string, then it is limited to twenty characters. If it is specified by n, then it is limited to a four digit number.

Options

None.

Related commands

^#n
^#(n)

#P

Purpose

To specify the starting value for page numbering.

Form

```
\#P {string}
    {n     }
```

Description

The #P command is used to specify the initial value for page numbering; the page number is not automatically printed. To print the page number somewhere on the page, use the ^#P command. To automatically print the page number in the footing, see the PAGENO command.

Page numbering is terminated by the #P -1 or PAGENO -1 commands.

Limitations

If the starting value is specified by string, then it is limited to twenty characters. If it is specified by n, then it is limited to a four digit integer.

Options

None.

Related commands

PAGENO
#P

The Spooler

Spooler overview

What is the spooler

TDP has its own spooler software, which is completely independent of the MPE spooler. If you are familiar with the MPE operating system, or you have used other computer systems, then you are probably familiar with the concept of a spooler. If you are not used to the idea of a spooler, then the simplest way of explaining what a spooler does is describe briefly how the TDP spooler works.

When you issue a FINAL or DRAFT command the following actions take place:

1 TDP editor creates a disc file (the spoolfile) to take the output from the formatter. (The spoolfile is created in the user's group and account, with a filename of the form Snnnhhmm.)

2 TDP formats your textfile and sends the formatted output to the spoolfile (rather than to the printer). This is a special TDP spoolfile ready to be output to the specified output device. Any error messages or prompts are still directed to your terminal in the normal way.

3 Once formatting is finished, TDP editor passes a print request to the spooler, and you may then continue editing a document. No further action is required to get your file printed, since the spooler runs as a batch job.

4 The spooler receives the print requests and schedules them on the available printers, on a first come first served basis. The spooler prints identification information with each spoolfile, identifying the name of the textfile, the spoolfile identity number, the username, group and account details.

  If for any reason the spooler stops before the file has been printed, the spoolfile is saved and printing will be rescheduled once the spooler has been restarted.

  Once your document has been safely printed, but not before, the spoolfile is deleted.

The process of spooling is illustrated in Figure 5-1.

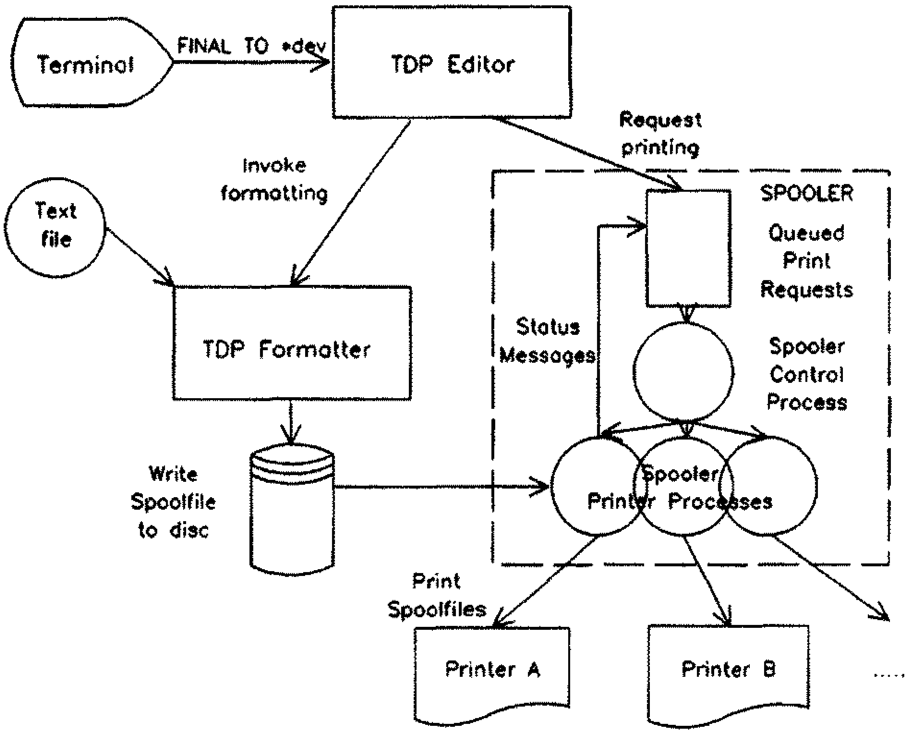


Figure 5-1. Spooling

The benefits of a Spooler

Using a spooler to take care of printing removes the dependence on the availability of (relatively) slow peripherals. For example, if you request output to a printer that is currently in use, your request will be rejected with an error message. You will then have to re-submit your printing request later. The spooler would simply queue your request, printing your document when the printer is available, allowing you to get on with other work.

TDP or MPE Spooler?

Since the MPE operating system supports a general system spooler that can be used by any system user, you might wonder if it's worth having a distinct TDP spooler.

There are two general reasons why you might wish to use the TDP spooler at your installation. Firstly, the TDP spooler and the MPE system spooler support different types of printer. In practice, the TDP spooler tends to support many of the smaller office printers that are likely to be needed by the user of TDP. For example, the HP 2601 printer is supported by the TDP spooler but not by the MPE system spooler. The types of printer supported by TDP are listed in Appendix F (as well as being available on the HELP system).

Secondly, the MPE system spooler handles the printing requests for all users of the system, whilst TDP's spooler is dedicated to the work of TDP users. That's fairly obvious, but in practice it means that it is generally easier for the group of TDP users to control their own printing workload if they have a dedicated spooler.

Spooler operation

The TDP spooler can support up to 16 printers. However, before any work can be done by the spooler, every printer that is to be controlled by the spooler must be defined in the configuration file. The configuration file supplies to the spooler information describing the name, type and characteristics of each printer, and the port to which each printer is attached.

Whenever the TDP editor receives a FINAL (or DRAFT) command with a printer specified, the configuration file is searched for the named printer. If the printer is found in the configuration file the action taken depends upon the way you are using the spooler.

If the spooler is not in use, TDP tries to access the printer directly. If the printer is available, then the printer number and type are passed to the formatter and output is directed to the printer. If the printer is not available an error is returned and the formatter is not called.

If you are using the spooler, the editor tells the formatter to direct its output to a spoolfile and on completion of formatting the editor sends a print request to the TDP spooler and you may then continue editing.

Setting up the Spooler

Steps to using the Spooler

It is recognized that some installations will not wish to use a separate TDP spooler, and for that reason the TDP spooler is not automatically initiated when the rest of the TDP software is installed. To initiate the TDP spooler three explicit steps must be performed:

1 Configure the spooler

Normally performed by the System Manager or Office Products Coordinator, this will be done at installation by building the configuration file, and thereafter by modifying the configuration file to add or delete a device.

2 Start the spooler running

This will normally be performed by the System Manager issuing the STARTSPOOL command. This will stream the SPOOLMAN job. This only needs to be done once.

3 Set the spooler ON

This may be performed on behalf of users by the System Manager or Office Products Coordinator by including SET SPOOLER ON in the TDPPARMS file (or in PARMSET for group level) in which case it need be done only once. Alternatively, it may be performed by users on their own behalf at the start of each session.

Building the configuration file

To build the configuration file, TDPCONFG:

1. Log on as MGR.HPOFFICE.TDPDATA.
2. Run TDP.
3. Set LENGTH and RIGHT to 32.
4. Enter the necessary configuration records as defined below.
5. KEEP the file as TDPCONFG.

There are two types of records in TDPCONFG - a Port Record and a Printer Record. The Port Record specifies characteristics of the device (speed, receive only, continuous forms, etc.). The Printer Record associates a printer name with a specific port or ports. There should be one Port Record for each printer in the system. Thus there may be up to sixteen such records. However, there may be any number of Printer Records corresponding to each physical printer. The format and content of the two record types are described below. It is essential that the information in the records is in the proper column positions and in the proper format.

Port Record

<u>Column</u> <u>Content</u>

1 #: for the record identifier.

2-4 Port number: a three digit number identifying a port to which a printer is to be attached. Leading zeros must be given.

6-8 Speed : the speed (in characters per second) at which the printer is to operate.

10 Receive-Only flag : indicating whether or not the printer is receive-only. "Y" if so, "N" otherwise.

12 Continuous Forms flag : indicating whether or not continuous forms are on the printer. "Y" if so, "N" otherwise.

14-15 Termtype : determines device dependent characteristics (see the **MPE Commands Reference** manual for further information).

Printer Record

<u>Column</u> <u>Content</u>

1-12 Printer name : comprised only of alphanumeric characters.

13-18 Printer type : one of the printer codes recognized by TDP (see Appendix F.)

19-21
22-24 Port number : a three digit number identifying the port to be associated with the printer name. Leading zeros must be given. At

25-27 least one port must be specified, and up to four ports can be
28-30 specified.

31 Sheet Feeder Flag: "Y" indicates that all the printers referenced by this record have sheet feeders. "N" indicates that the referenced printers do not have sheet feeders.

32 Paper Saving Flag: "Y" indicates that for all the printers referenced by this record the TDP formatter will not print end of formatting messages nor the automatic top of page messages. "N" gives normal operation. (This only applies when using the TDP spooler.) If paper save mode is set, the printing of banners, top of page messages and end of formatting messages is suppressed. In general, this mode should be used only if spooler is to have sole use of a printer, since use by other users may leave the paper in an unknown position with respect to top of form, with unpredictable placement of the printed text.

After keeping the configuration file, the VERIFY CONFIG command should be used to check that the record details are correct. The lines below show an example of a configuration file.

```
#025 960 Y Y 13
#034 120 Y Y 13
PRTSF       HP2601034         YY
PRT01       HP2601034         NN
BONSAI      HP2687025         NY
```

In the example above, the first two lines represent Port Records defining two printers, a HP 2601 daisywheel printer and a HP 2687 laser printer. The HP 2601 is connected to port 34, which runs at 1200 baud, whilst the HP 2687 laser printer is connected to port 25, which runs at 9600 baud.

The last three lines in the example represent Printer Records for these printers. The last of these associates the HP 2687 laser printer with the name BONSAI. To request formatted output to this device you would FINAL TO *BONSAI. The printer is not using a sheet feeder and is set to paper saving mode. The first two Printer Records provide for different modes of use of the same printer, the HP 2601. The first record defines the printer for use with a sheet feeder and the second record defines the printer for use without a sheet feeder. To print to the HP 2601 using a sheet feeder you would FINAL TO *PRTSF, and without the sheet feeder FINAL TO *PRT01.

Using the Spooler

Getting started

After specifying and verifying the configuration details, the spooler must be set on and started before use. Issue the SET SPOOLER ON command and STARTSPOOL. If VERIFY SPOOLER is now issued it should show:

SPOOLER is SET
SPOOLER is RUNNING

The spooler must be SET ON for each user wishing to use spooler. This may be done on behalf of users at the system or group level using the STOREPARMS command. Alternatively, it may be done by users individually during a TDP session.

Controlling the Spooler

Once the spooler is running it can be used simply by issuing normal print requests (DRAFT, FINAL etc) specifying a printer controlled by the spooler. However, spooler provides a number of commands to control its operation:

SHOWSPOOL

Use this command to display the status of any outstanding print requests. For each print request the display will show the file identification number, textfile name, creator, status, and the port number of the printer on which it is printing or will be printed. The print requests are listed in the order in which they will be considered for printing. The status of a spoolfile can be READY, STOPPED or PRINTING. A READY file will be printed as soon as its printer is free. A STOPPED file will not be printed until a user changes its status to READY. A PRINTING file is currently being printed.

ALTERSPOOL

Use this command to alter the status of your spoolfiles in the following ways:

A READY spoolfile can be made STOPPED.

A STOPPED spoolfile can be made READY.

A PRINTING spoolfile can be made STOPPED.

Any spoolfile can be CANCELLED.

Users with Account Manager capability can alter the status of any spoolfile in their account. Users with System Manager capability can alter the status of any spoolfile.

STOPSPOOL

Use this command to stop the spooler. The spooler will stop after all the currently printing spoolfiles have finished printing. (This command is analogous to the MPE command SUSPENDSPOOL, and should not be confused with the MPE command STOPSPOOL.)

ABORTSPOOL. This command stops the spooler immediately with the suspension of any jobs currently being printed. Suspended print jobs will request printing when the spooler is restarted. (This command is analogous to the MPE command STOPSPOOL.)

Full descriptions of these commands are given in Chapter 3.

Notes for the Office Products Coordinator

The two programs that make up the TDP spooler are TDPSP1.PUB.SYS (the spooler Control program) and TDPSP2.PUB.SYS (the spooler Print program).

While the TDP spooler is running, a number of files are created in the group TDPDATA in the HPOFFICE account. The following provides an overview of the way the TDP spooler works.

Once the TDP spooler is started, the Spooler Control Process is created. This process reads the TDP configuration file (TDPCONFG) and creates a Print Process for each port identified.

The Spooler Control Process is driven by an Interprocess Communication file, TDPMSG. It waits on a read on this file and is, therefore, event driven rather than having to scan every n seconds for a message. This approach requires less resources.

When a print request is written to the TDPMSG file (by the TDP editor), the Spooler Control Process queues a request in the Spooler Queue file, TDPQFILE. It is this file that holds all the information concerning print requests.

When a printer becomes free, the print request is passed to the appropriate Print Process which reads the data from the spoolfile and writes it to the printer. The Spooler Control Process communicates with each Print Process through IPC files. These IPC files are PRnn.TDPDATA.HPOFFICE where "nn" is an internal two-digit identifier for the Print Process.

When a Print Process finishes outputting a file, it writes a print completion record to the TDPMSG file. This ensures that no print requests are lost in the event of a system failure.

In summary, the files created by the TDP spooler are:

1 The spoolfile, Snnnhhmm, in the user's group and account.

2 TDPMSG.TDPDATA.HPOFFICE - an IPC file for the spooler Control Process.

3 TDPQFILE.TDPDATA.HPOFFICE - a file to record the status of the print requests.

4 TDPINDEX.TDPDATA.HPOFFICE - an index to the TDPQFILE. This allows a faster means of access to the information in the TDPQFILE.

5 PRnn.TDPDATA.HPOFFICE - IPC files for the Print Processes.

The TDP spooler job, TDPSPOOL,SPOOLMAN.HPOFFICE is started by the STARTSPOOL command. This STREAMs the job TDPSPJOB. If your installation has the requirement to leave a lot of spoolfiles in the TDP spooler queue, you may get the message:

```
(100)OUTPUT NOT SPOOLED, QUEUEFILE FULL.
```

If this happens too often, you may wish to configure a bigger queuefile. The following steps should be taken:

1 Log on as MGR.HPOFFICE.TDPDATA.

2 Run TDP.

3 If the TDP spooler is running, issue a STOPSPOOL and wait for the job to finish.

4 Purge TDPQFILE.TDPDATA.HPOFFICE.

5 Edit the file TDPSPJOB.TDPDATA.HPOFFICE and change the DISC parameter on the BUILD command for the file TDPQFILE to the required value. This should not be less than 50. To optimize the TDP spooler performance, the number of entries configured should be about 40 more than the peak number of outstanding spoolfiles you expect to have.

6 Keep the file, overwriting the old version.

Next time the TDP spooler is started, the queuefile will be built to the new size.

Note: This reconfiguration will result in the loss of any outstanding entries in the queuefile.

Mailer

Mass mailing

Mailer is a general purpose mass mailing facility in TDP. Using Mailer you can take a standard form letter and automatically add names and addresses and insert variable information into the body of the letter to produce personalized letters and/or print addressed envelopes and/or produce mailing labels.

How Mailer works

In the simplest case Mailer takes as its input two files, prepared in advance, one a list of names and addresses for the circulation of your letter, and the other a standard form letter you wish to circulate to all your addressees. Mailer effectively merges these two files to produce a new file containing the name and address of each addressee merged with a copy of the standard letter.

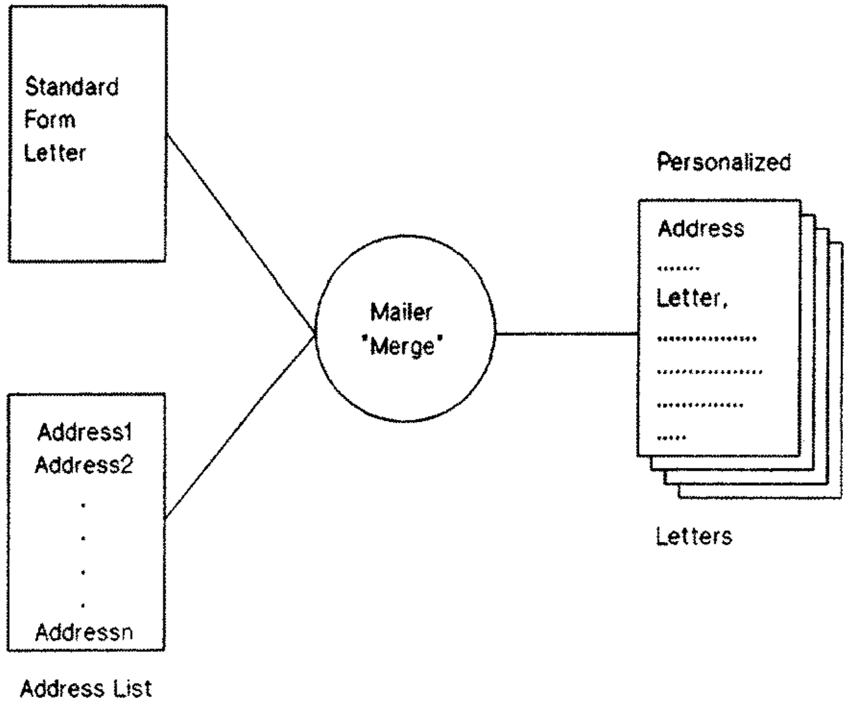


Figure 6-1 Using Mailer to produce a simple merged letter.

Merging standard letters with names and addresses is a very simple case; Mailer can handle much more complicated mailing requirements. For example, on input you can specify a third or even a fourth file containing formatting instructions to control the layout of address and date information in the letter and envelope addresses, rather than use the standard layout. In the address file you can include more information that can be merged into the body of the letter to further personalize it. As part of its standard output Mailer can produce another file containing all of the names and addresses and which can be formatted to print addressed envelopes. As a further option Mailer can produce a similar address file for output, but formatted to produce mailing labels.

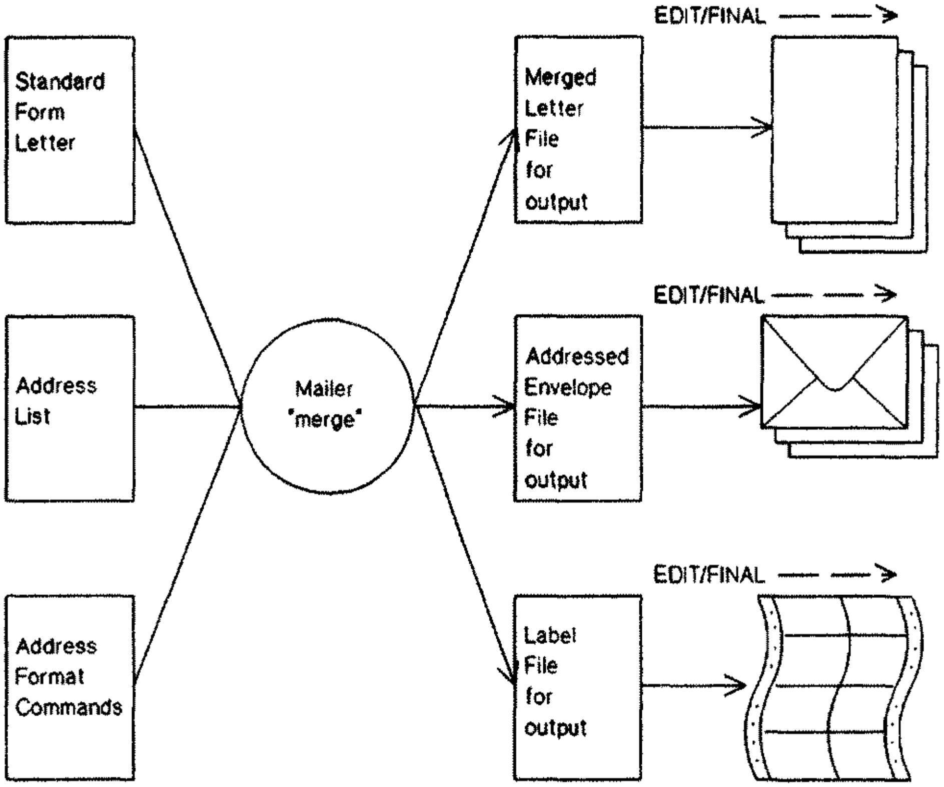


Figure 6-2 Input and output files used by Mailer.

The main steps in using Mailer

To use Mailer the main steps are as follows:

1 Produce the address file. The address file may be produced by manually creating a separate file or, in appropriate cases, a program may be written to extract the information from a larger database and create an address file.

2 Produce the textfile for the standard form letter.

3 If a non-standard layout is to be used for the address information in the letter or envelopes, produce a Header Format File containing the formatting commands.

4 Use the /MAILER command to enter the Mailer dialogue to produce the appropriate merged output file(s).

5 You may then use the TDP Editor to further edit the output files and then use the FINAL command to print the letters, envelopes and mailing labels from the output files.

In the case of the letter output file little editing, if any, should be required. With envelope output, it is worth checking the formatting to ensure that the addresses are properly placed for your chosen envelopes. However, if are you producing labels, the output file produced by Mailer will almost certainly require further editing before printing. Typically, the file will need the top, bottom, left and right margins adjusting to ensure proper positioning of the addresses on the labels. Most importantly, for sheets of multi-column labels, multi-column formatting commands will need to be added to the label output file.

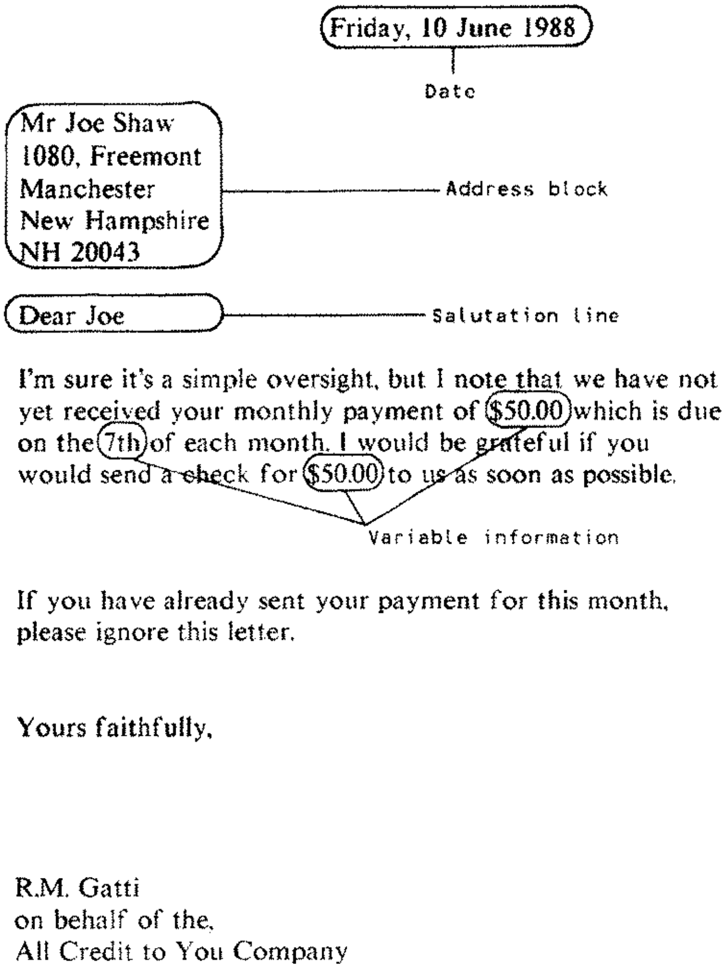
These steps are described in more detail below.

Terminology

At this point it is worth introducing some terminology used to refer to parts of a letter in describing Mailer facilities.

In Example 1 a letter is annotated to illustrate the use of this terminology.

Example 1: Example letter


Date
Address block
Salutation line
Variable information

Friday, 10 June 1988

Mr Joe Shaw
1080, Freemont
Manchester
New Hampshire
NH 20043

Dear Joe

I'm sure it's a simple oversight, but I note that we have not yet received your monthly payment of $50.00 which is due on the 7th of each month. I would be grateful if you would send a check for $50.00 to us as soon as possible.

If you have already sent your payment for this month, please ignore this letter.

Yours faithfully,

R.M. Gatti
on behalf of the,
All Credit to You Company

Address file

Since this file contains not only addresses but also other information about each addressee, such as the salutation line and variable information to be inserted into the body of the letter, it is more properly called the **Addressee Information File**. The file contains a group of records for each addressee, where a record corresponds to one line in the file.

The address block

In the simplest case the file contains just a series of address blocks (see Example 1 for an illustration of an address block). Each line of the address block (one record) corresponds to one address line in the output letter. There are no separators between address blocks in the file; Mailer knows where one address ends and another starts because it asks in the dialogue for the length of the address block and expects all blocks to be of this length. Clearly, the length of the address block is set by the length of the longest address. To deal with different address lengths you could simply fill out shorter addresses with blank lines in the address block. However, you can avoid extraneous blank lines in the printed output by inserting a "*" formatter command in place of blank line in the address block.

In this simple case (Example 1) it assumed that the standard layout for the date and address heading is being used. Non-standard or customized formatting is described later. With the standard layout you can specify, during the Mailer dialogue, the format of the date (based on the ^D formatting command), the number of blank lines between the top of page and the date, and the number of blank lines between the date and the first line of the address block.

Salutation line

A simple variant on straightforward addresses is to add a salutation line (see Example 1) after each address block. The salutation line is included immediately after the last line of the address block, without a separator. Mailer asks you in the dialogue if a salutation line is included, and if the answer is yes, places it appropriately. Note that the salutation line is not counted as a line of the address block in specifying the length of the block. If you require a blank line to appear after the salutation line, before starting the body of the letter, then a blank line should be included as the first line of your standard letter.

Variable information

Standard form letters can be further personalized by including variable information in the body of the letter (see Example 1). This is achieved by using the TDP macro facility. In the body of the standard form letter, a macro reference is included in each place where a variable is to be inserted. The variable information for each addressee is included in the Addressee Information File after each address block, following the salutation line. Each item of variable information, known as the **text macro line**, must not occupy more than one line of the file. Using Example 1, Examples 2 and 3 illustrate the layout of the address block in the Addressee Information File and the standard form letter to achieve that result.

Example 2: Addressee Information File

```
Mr Joe Shaw
1080, Freemont
Manchester                  Address block
New Hampshire
NH 20043
Dear Joe                    Salutation line
$50.00                      Text macro line 1
7th                         Text macro line 2

Ms Sam Brown
Circus Circus
Reno                        Address block
Nevada
NV 56478
Dear Sam                    Salutation line
$75.00                      Text macro line 1
21st                        Text macro line 2
```

Example 3: Standard form letter

> I'm sure it's a simple oversight, but I note that we have not yet received your monthly payment of ^MA which is due on the ^MB of each month. I would be grateful if you would send a check for ^MA to us as soon as possible.
>
> If you have already sent your payment for this month, please ignore this letter.
>
> Yours faithfully,
>
> R.M. Gatti
> on behalf of the,
> All Credit to You Company

Note that there is no need to use a formal macro declaration in the text macro line (for example, \MA="$50.00"), since Mailer automatically associates each macro reference in the body of the letter with the appropriate text macro line (for example, ^MA is associated with text macro line 1, ^MB is associated with text macro line 2, etc). There may be up to 36 text macro lines and 36 associated macro references (^MA to ^MZ and ^M1 to ^M9).

Macro assignment and referencing

The ability to handle variable information by means of text macro lines is one of Mailer's most powerful features. To get the most out of it it is necessary to understand how Mailer assigns the text macro lines and how they may be referenced.

The default assignment of macro references in the standard form letter file, the Header Format File and the Envelope Format to the text macro lines in the Addressee Information File is alphabetic. If, in the form letter, you use ^MA as the first macro, ^MB as the second macro reference ,^M1 as the 28th macro reference up to ^M9 as the 36th and last macro reference, Mailer would assign ^MA to the first text macro line in the Addressee Information File, ^MB to the second and so on. This is referred to as alphabetic assignment. However, you are not restricted to this method of assignment. You might want to use ^M1 as your first macro reference, ^MC as your second reference and so on, but Mailer needs to be told of your assignment. If you answer "no" to the prompt "Is your macro assignment alphabetic?" in the Mailer dialogue, Mailer will ask what identifier you have used to refer to the first text macro line, the second text macro line and so on until all the text macro lines have been assigned to macro references.

The second important point about macro assignment is to remember that macro assignment and macro referencing are distinct; just as they are in any TDP document. As long as you tell Mailer how many text macro lines there are in total in each record of the Addressee Information File and make clear the method it should use in assigning macros, you can reference any or all of those macros and do so in any order. This means that you can build an Addressee Information File that contains many text macro lines that will be used to generate many different form letters. Building on the earlier example, Example 4 shows the Addressee Information File expanded to include further text macro lines. Example 5 shows a different form letter based upon this file but referencing different macros and in a different order. Examples 6 and 7 show the printed results. To make sure you are clear about the concepts, it is worth experimenting with a few simple files.

Example 4: Revised Addressee Information File

Mr Joe Shaw	Address block
1080, Freemont	
Manchester	
New Hampshire	
NH 20034	
Dear Joe	Salutation line
$50.00	Text macro line 1
7th	Text macro line 2
Mr Shaw	Text macro line 3
car	Text macro line 4
Ms Sam Brown	Address block
Circus Circus	
Reno	
Nevada	
NV 56478	
Dear Sam	Salutation line
$75.00	Text macro line 1
21st	Text macro line 2
Ms Brown	Text macro line 3
house	Text macro line 4

Example 5: New form letter

^MC

I note that we have still not received your payment since I last wrote to you. We will have no option but to repossess your ^MD unless we receive payment of ^MA within three days.

Yours,

R.M. Gatti
on behalf of the,
All Credit to You Company

Example 6: Example letter 1

Friday, 10 June 1988

Mr Joe Shaw
1080, Freemont
Manchester
New Hampshire
NH 20043

Dear Joe

I'm sure its a simple oversight, but I note that we have not yet received your monthly payment of $50.00 which is due on the 7th of each month. I would be grateful if you could send a check for $50.00 to us as soon as possible.

If you have already sent your payment for this month, please ignore this letter.

Yours faithfully,

R.M. Gatti
on behalf of the,
All Credit to You Company

Example 7: Example letter 2

Monday, 1 August 1988

Mr Joe Shaw
1080, Freemont
Manchester
New Hampshire
NH 20043

Mr Shaw

I note that we have still not received your payment since I last wrote to you. We will have no option but to repossess your car unless we receive payment of $50.00 within three days.

Yours,

R.M. Gatti,
on behalf of the,
All Credit to You Company

Formatting addresses

Mailer allows you to explicitly control the layout of envelope and label addresses and letter headers (address, salutation line and date). To achieve this formatting each line of the address block is treated as a special text macro line, with the layout specified in a separate **Header Format File** containing formatting commands and macro references to the address lines.

It is important to note that if this mode of working is used, you must include the macro reference to the salutation line, otherwise Mailer will get out of step as it processes the Addressee Information File and will confuse the salutation line with an address line or a text macro line to be included in the body of the letter. The macro references to the header are treated by Mailer as completely independent of any macro references in the body of the letter. Thus, it is perfectly feasible to have an address line referenced as ^MA in the Header Format File and a text macro line referenced as ^MA from the body of the letter. Examples 8 to 10 illustrate some of these points with a Header Format File for use with the earlier example and the output produced.

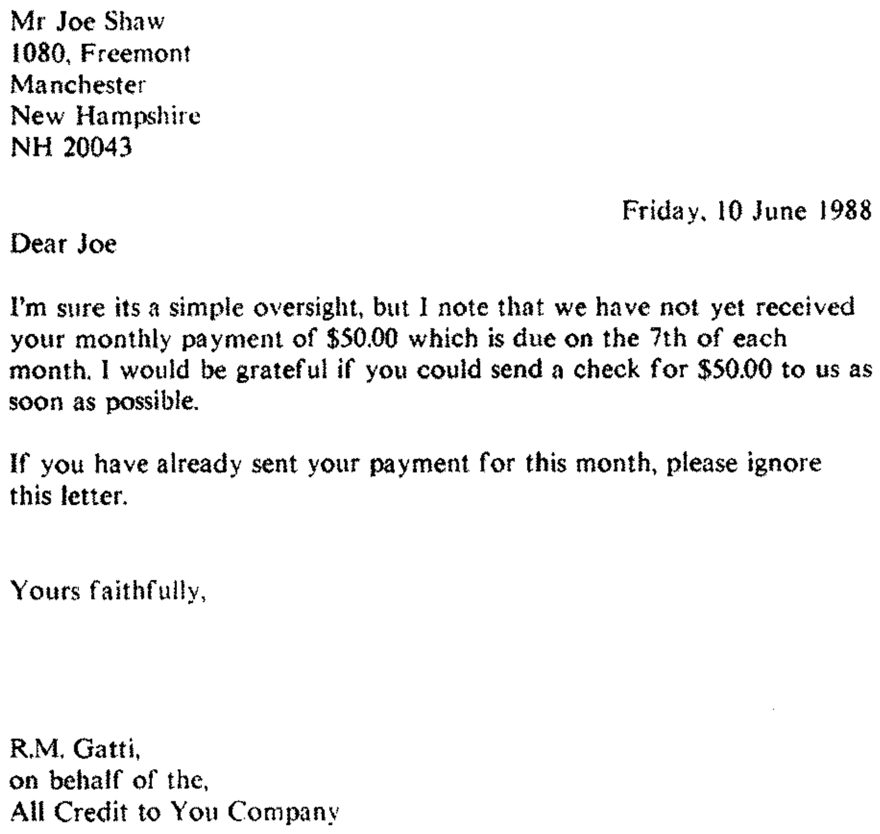
Example 8: Section of Addressee Information File

Mr Joe Shaw	Header macro reference ^MA
1080, Freemont	Header macro reference ^MB
Manchester	Header macro reference ^MC
New Hampshire	Header macro reference ^MD
NH 20043	Header macro reference ^ME
Dear Joe	Header macro reference ^MF
$50.00	Text macro line 1
7th	Text macro line 2

Example 9: Header Format File

```
^MA
^MB
^MC
^MD
^ME
\right
^DB
\format
^MF
```

Example 10: Mailer output with formatted header



Mr Joe Shaw
1080, Freemont
Manchester
New Hampshire
NH 20043

Friday, 10 June 1988

Dear Joe

I'm sure its a simple oversight, but I note that we have not yet received your monthly payment of $50.00 which is due on the 7th of each month. I would be grateful if you could send a check for $50.00 to us as soon as possible.

If you have already sent your payment for this month, please ignore this letter.

Yours faithfully,

R.M. Gatti,
on behalf of the,
All Credit to You Company

Envelope formatting

You can also specify the format of the address to be printed on the envelope. The principles for creating an **Envelope Format File** are the same as for the Header Format File, but there are a couple of points to note.

Firstly, the macro references in the Header Format File and the Envelope Format File are common (they reference the same text macro lines). For example, suppose in the Header Format File your first macro reference is ^MA, this will reference the first line of your header (normally the first line of the address). To reference that same line of the Header in the Envelope Format File you should also use ^MA.

Secondly, whilst the macro references in the Header Format File must match the number of lines in the header, including the salutation line, this is not true in the Envelope Format File. This is obviously important, since you don't normally wish a salutation line to appear as part of the envelope address. To avoid printing the salutation line you simply don't include the reference to that line in the Envelope Formatting File.

Mailer dialogue

Once all of the input files are created, you can issue the Mailer command. The Mailer command invokes a dialogue with Mailer, which is described below. The description of the dialogue includes all of the questions asked by Mailer. In practice, depending on your response at certain points, Mailer will branch to different places in the dialogue. The dialogue questions below are numbered to simplify cross-referencing.

```
/MAILER
```

```
TDP Mass Mailing Utility
************************
```

1
```
Enter a filename if you wish to store your
commands for future input:
```

Mailer checks if the filename you have given already exists and if so prompts for another. Mailer stores your responses to each of the prompts during the dialogue in the file named. This facility allows a standard form letter job to be run at regular intervals without the need to run the dialogue. The output files must first be deleted to allow new files to be created. The facility is invoked by entering /MAILER filename, where filename is the name of the file specified here.

2
```
Enter Addressee Information Filename:
```

If the filename you enter cannot be found, you will be prompted for another.

3
```
Do you want to print letters/documents?
```

Answer Y or N. If you answer NO you will skip that part of the dialogue that deals with producing letter output. If you answer YES you will be prompted later in the dialogue for the name of an output file.

4
```
Do you want to print envelopes?
```

Answer Y or N. If you answer YES you will be prompted later in the dialogue for the name of a file to hold the output. If you answer NO you will skip that part of the dialogue dealing with envelope output.

5
```
Do you want to print labels?
```

Answer Y or N. If you answer YES you will be prompted later in the dialogue for the name of a file to hold the output. You should not normally answer YES to this question if you have answered NO to questions 3 and 4.

6
```
Do you want a letter heading?
```

Answer Y or N.

7

```
If you require the standard heading press RETURN
Else enter your letter/document heading filename:
```

If you press return you get the standard letter heading layout as described above. With the standard layout Mailer prompts you for information on spacing and date format. If you specify a filename Mailer prompts you for information about the macros used in the file.

8

```
How many lines in the address block?
```

Respond with the number of lines in the address block, but remember not to count the salutation line.

9

```
Is there a salutation line?
```

Answer Y or N

10

```
Enter TDP date format:
```

The response should be one of the permissible formats as described in the ^D command (Chapter 4). For example, ^DB.

11

```
Number of blank lines before the date?
```

Specify the number of blank lines between the top of page and the date. This must be greater than four.

12

```
Number of blank lines between date & address?
```

The minimum value is one.

13

```
How many different macros in your letter/doc Heading?
```

If you are using a Header Format File, specify the number of macro text lines referenced. You must count the salutation line, if you are using one.

14

```
Is your macro assignment alphabetic?
```

Answer Y or N. If you answer N, Mailer will prompt you for the macro to be assigned to each line of the Addressee Information File.

15

```
Enter filename holding the text of the letter/document:
```

Enter the name of your standard form letter file. Mailer does not check until the dialogue is complete to see if the file exists. Therefore if you specify the wrong name, an error will not occur until after the dialogue is complete.

16
```
How many different macros in your letter/document?
```

Enter the number of macro text lines referenced in the body of the letter, if any.

17
```
Is your macro assignment alphabetic?
```

Answer Y or N. The same comments apply as for question 14.

18
```
Specify Left Margin column number:
```

Enter a column number.

19
```
Specify Right Margin column number:
```

Enter a column number less than 90 and at least 30 greater than the left margin. The left and right column numbers are used to implement the TDP LFT and RHT commands.

20
```
Enter filename for letter/document output:
```

Enter the name of a file to hold the output. If you specify the name of an existing file, Mailer will prompt you for another.

21
```
Do you want to pause between printing each letter?
```

Answer Y or N.

22
```
If you require the standard envelope press RETURN
Else enter your envelope layout filename:
```

If you are using an Envelope Formatting File, enter the filename.

23
```
How many different macros in your envelope:
```

Note that if you are not producing letter output in this run of Mailer, ie. you answered NO to question 3, then you must count all of the lines of the record in the Addressee Information File including salutation lines and text macro lines in response to this question. This is so that Mailer can distinguish the end of one record and the start of the next. Your formatting file will not make reference to these other lines, so they will not appear in the printed output.

24
```
Is your macro assignment alphabetic?
```

Answer Y or N. The same comments apply as for question 14.

25

```
Enter filename for envelope output. If you specify
the same file as for letter output, letters and
envelopes will be interleaved in that file.
          Envelope Output Filename ?
```

Enter the name of the file to hold envelope output.

26

```
Do you want to pause between printing each envelope?
```

Answer Y or N.

27

```
How many lines in the address block?
```

Enter the number of lines in the address block. This prompt appears if you are using the standard envelope output.

28

```
How many lines on a label?
```

Enter the number lines physically occupied by each label. There must be at least as many lines on a label as there are lines in the address block. For example, a sheet of labels 1.5" from TOP to TOP, on a printer with 6 lines per inch spacing would occupy 9 lines. (Remember that the file holding label output almost certainly needs further formatting commands adding before it is suitable for printing).

29

```
Enter filename for label output or
press RETURN for further label output options:
```

Enter the name of the file to hold the label output. If you press RETURN you will be given the choice of having the labels printed at the terminal or on a printer.

```
MAILER has finished. Use TDP FINAL command to produce printed output from
your files.
```

Mailer dialogue examples

The following examples of the Mailer dialogue are based upon the examples described earlier in this chapter.

Example 11:

In this example letters and envelopes are to be produced. This example uses an Addressee Information File called ADFILE, the content of which is as described in Example 4. The form letter to be used for this run of Mailer is that shown in example 3. A Header Format File and an Envelope Format File will be used called LETHED and ENVFORM, respectively. Letter output will be held in a file called LETOUT and the envelope output will be held in a file called ENVOUT.

```
/MAILER

TDP Mass Mailing Utility
************************

Enter a filename if you wish to store your
commands for future input:

Enter Addressee Information Filename: ADFILE

Do you want to print letters/documents? Y

Do you want to print envelopes? Y

Do you want to print labels? N

Do you want a letter heading? Y

If you require the standard heading press RETURN
Else enter your letter/document heading filename:  LETHED

How many different macros in your letter/doc Heading? 6

Is your macro assignment alphabetic? Y

Enter filename holding the text of the letter/document: LETTER1

How many different macros in your letter/document? 4 (see Note 1)

Is your macro assignment alphabetic? Y

Specify Left Margin column number: 10

Specify Right Margin column number: 80
```

```
Enter filename for letter/document output: LETOUT

Do you want to pause between printing each letter? N

If you require the standard envelope press RETURN
Else enter your envelope layout filename: ENVFORM

Enter filename for envelope output. If you specify
the same file as for letter output, letters and
envelopes will be interleaved in that file.
              Envelope Output Filename ? ENVOUT

Do you want to pause between printing each envelope? N

MAILER has finished. Use TDP FINAL command to
produce printed output from your files.
```

Note 1 When responding to the question "How many macros in your letter", the number must include all of the text macro lines in the Addressee Information File (even if not all of those text macro lines are referenced in the standard form letter). In this example, there are four text macro lines in the file, though only two of those lines are referenced in the letter (see Example 3). This is because Mailer distinguishes between text macro lines and lines of the address block by counting the number of lines in each block of the file (address lines, salutation line and text macro lines). Had the response here been 2 (the number of macros referenced) the run would have failed with the message **Encountered end of addressee information file in the middle of a group**.

Example 2:

This example run of Mailer is similar to the previous example, however only envelope output is to be produced.

```
/MAILER

TDP Mass Mailing Utility
************************

Enter a filename if you wish to store your
commands for future input:

Enter Addressee Information Filename: ADFILE

Do you want to print letters/documents? N

Do you want to print envelopes? Y

Do you want to print labels? N
```

```
If you require the standard envelope press RETURN
Else enter your envelope layout filename: ENVFORM

How many different macros in your envelope: 10 (see Note 1)

Is your macro assignment alphabetic? Y

Enter filename for envelope output. If you specify
the same file as for letter output, letters and
envelopes will be interleaved in that file.
              Envelope Output Filename ? ENVOUT

Do you want to pause between printing each envelope? N

MAILER has finished. Use TDP FINAL command to
produce printed output from your files.
```

Note 1 Both the Addressee Information File and the Envelope Formatting File specified in this example are the same as those specified in the previous example. However, because no letter output is being produced in this run, Mailer must be told explicitly how many macros there are in each block of the file (address lines, salutation line and text macro lines) since there are no prompts dealing with the number of macros in the letter heading or the letter. Thus in this case the number of macros specified is the total number of lines in the file block, ie.10. Had the response been 5 (the number of macros referenced in the Envelope Formatting File) the run would have failed with the message **Encountered end of addressee information file in the middle of a group.**

Very large Mailings

The default size for the output file in TDP is 3000 records. So if there are 6 lines in each address, you can have 500 addresses, and 5 lines in each address gives 600 addresses. If you want to produce envelopes or labels from a file of more than 3000 records, then you can use a file equation to increase the maximum record size, for example

```
:FILE envout; DISC=5000
```

Here the output file is envout, and the number of records has been increased to 5000.

Mailing applications without Mailer

Mailer is a very flexible general purpose mass mailing facility. Nevertheless there may be times when the facilities of Mailer are not suitable to a particular application. However, it is worth noting that all of the facilities of Mailer are provided by means of the standard features of the TDP Editor and Formatter. Where a special application is needed it is recommended that careful study is made of some typical Mailer output files, particularly the use of INCLUDE and macro commands to provide a guide in building the application.

Hyphenation

The commands that control the hyphenation of words have been described in Chapter 4. They are HYPHEN, HYPHENALLCAPS, HYPHENFIRSTCAP, HYPHLAST and HYPHDBL. This chapter describes the hyphenation exception dictionary and how to supply a user hyphenation procedure.

Adding words to the hyphenation exception dictionary

In order to accommodate words which are not properly hyphenated by the standard TDP hyphenation algorithm, you have access to a hyphenation exception dictionary named TDPEXCPT in the group TDPDATA in the HPOFFICE account. The TDP formatter accesses this dictionary before applying the hyphenation rules of the algorithm. To modify or add to the exception dictionary, complete the following steps:

1 Log on as MGR.HPOFFICE,TDPDATA.

2 Run TDP.

3 Type TDPEX and when you are asked if you want to list or expand the file, answer "expand". TDPEX is a program used to format the exception dictionary for access efficiency.

4 TEXT in the file TDPEXCPT,UNN.

5 Edit the file adding new words and changing existing words. Indicate all acceptable hyphenation points with a hyphen. You can enter words with a maximum of five hyphenation points. Each word must not exceed 26 characters with hyphenation marks, and must appear on separate lines in the workfile.

6 KEEP the file, overwriting the old version.

7 Type TDPEX and the added words will be sorted into alphabetical order.

8 To check the hyphenation points for a particular word, issue the TDP command HYPHEN and type in the word. TDP will display the word with the hyphenation points indicated.

User hyphenation procedures

If you have unusual hyphenation requirements, such as highly specialized vocabularies or non-English text, you can write your own hyphenation procedure and get TDP to use it instead of, or as well as, the standard TDP hyphenation procedure.

The user hyphenation procedure must be a separately compiled Integer procedure, and may be written in any language which can be combined with an SPL program (such as SPL and FORTRAN). The procedure must be placed in a Segment Library. When TDP is called upon to access the user hyphenation procedure, the order of access of segmented libraries is, first the Group Library, then the Account Library, and finally the System Library. If the hyphenation routine is not found in any of these libraries (or they do not exist), an error message is returned.

When the TDP formatter needs to hyphenate a word it calls the user hyphenation procedure and passes it the word to be hyphenated. Among the other parameters passed to the routine is a word containing 1-bit flags to be used for control of hyphenation within the routine. The first eight bits in this word may be set through the \HYPHFLAGS command (See section 4). This way you may construct special controls which may then be interpreted by the user hyphenation procedure.

The procedure will return an integer value to the TDP formatter to indicate the action to be taken.

A value of +1 indicates that the word has been processed and all valid hyphenation points are indicated by the hyphenation character. In this case, TDP will chose the rightmost hyphenation point which will allow the partial word to fit on the current line, and break the word accordingly. If the ASK option has been supplied with the procedure name on the HYPHEN command, TDP will act in the same way as HYPHEN AUTO,ASK but with the hyphenation points chosen by the user hyphenation procedure.

A value of 0 (zero) indicates that TDP is not to hyphenate the word at all.

A value of -1 indicates that TDP is to call its own hyphenation routine to hyphenate the word. This allows you to construct a customized hyphenation routine to look up a set of words in order to hyphenate them, and if a word is not found, to pass it to TDP to hyphenate.

The word passed to the user hyphenation procedure is taken directly from the text file - that is, it may have contain special characters, numbers etc. It is simply that sequence of non-blank characters which extends beyond the end of the current line. Therefore, the user hyphenation procedure must test for compound words (words with embedded hyphenation character), upper and lower case words, etc. The programmer writing this procedure should bear in mind that any occurrence of a surrogate blank character will be represented by a byte with a value of 0 (null). All other characters will be exactly as they appear in the text file, including the hyphenation character if a discretionary hyphen character was specified.

The following is an example of an SPL procedure header with each of the parameters and return values defined:

```
INTEGER PROCEDURE USERHYPH ( WORD,MYINT,HYPHCHAR,MYARRAY,FLAGS);
   BYTE ARRAY WORD;
   INTEGER MYINT;
   BYTE HYPHCHAR;
   ARRAY MYARRAY;
   LOGICAL FLAGS;

<< ----------------------------------------------------------- >>
<< WORD - A byte array that will contain the string which is to  >>
<<   be hyphenated.  It will be a maximum of 32 characters in    >>
<<   length, left justified in the array, and padded with trailing>>
<<   blanks.  The string is returned to TDP in this array with   >>
<<   the valid hyphenation points identified by the character    >>
<<   supplied through HYPHCHAR.                                  >>
<<                                                               >>
<< MYINT - An integer purely for use by this procedure as a      >>
<<   "global" variable.  It will be set to zero prior to the     >>
<<   first call to the procedure, and not modified by the TDP    >>
<<   formatter thereafter.  A typical use for this is to detect  >>
<<   the first call by MYINT=0, and placing the file number of a >>
<<   file used by this procedure into MYINT. (For example, a     >>
<<   file of words with their correct hyphenation specifications).>>
<<                                                               >>
<< HYPHCHAR - A character to be used in this procedure to        >>
<<   indicate hyphenation points within the byte array WORD.  It >>
<<   is the discretionary hyphen character.                      >>
<<                                                               >>
<< MYARRAY  - An 8-word array strictly for the use of this       >>
<<   procedure, is initialized to all zeros, and not altered    >>
<<   between calls.  A typical use is for record pointers into a >>
<<   dictionary to shorten look-up times.                        >>
<<                                                               >>
<< FLAGS - A 16-bit set of flags to be used in the hyphenation   >>
<<   procedure.  The sub-fields of FLAGS are:                    >>
<<                                                               >>
<<   Bits 0-7 - The 8 bits set by the HYPHFLAG command.          >>
<<   Bit 8   - 1 => the string passed is not the last in a       >>
<<                 paragraph, column or page.                    >>
<<   Bit 9   - 1 => HYPHLAST is ON.                              >>
<<   Bit 10  - 1 => HYPHALLCAPS is ON.                           >>
<<   Bit 11  - 1 => HYPHFIRSTCAP is ON.                          >>
<< ----------------------------------------------------------- >>
<< The returned value of the procedure identifies the action to be>>
<< taken by TDP.                                                 >>
<<                                                               >>
<<  USERHYPH = 1 => hyphenate the string as indicated.           >>
<<  USERHYPH = 0 => do not hyphenate the string.                 >>
<<  USERHYPH = -1 => hyphenate the string using the standard TDP >>
<<                   hyphenation routine.                        >>
<< ----------------------------------------------------------- >>
```

Error messages

Editor error messages

This section describes the error messages that may be produced whilst you are using the TDP editor.

Note that all error messages with a (File Error) noted after the message will be displayed with a file error number and file error message when the error actually occurs.

(1) INVALID COMMAND

The command you entered either does not exist, or is spelled incorrectly. Check the appropriate page in this manual for the correct syntax.

(2) INVALID OPTION

One of the options you listed on the command line does not exist for this command. Check the description of the command in the appropriate section of this manual.

(3) MISSING PARAMETER

You have failed to include a required parameter for this command. Check the appropriate page in this manual for the correct syntax.

(4) DELIMITED STRING TOO LONG

The maximum number of characters allowed in a delimited string is 72.

(5) TOO MANY DISTINCT WORDS

When using a wordlist within parentheses for a command, you are limited to 15 words in the list. You will need to split your wordlist into two segments to comply with this restriction.

(6) BAD OCTAL STRING

An octal string can consist of digits only, with each digit less than or equal to 7. Check your string for a typographical error.

(7) READ ERROR ON TEXT FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(8) CLOSE ERROR ON OUTPUT FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(9) ERROR IN POSITION EXPRESSION

You have used an invalid column indicator in a string(column) position expression. Example: "The"(+93), when the line length is set at 72. Check the command line for a mistyped number.

(10) COLUMN POSITION OUT OF RANGE

The column position you have listed is greater than the currently set line length. Check the command line for a typographical error.

(11) NULL STRING NOT ALLOWED

Certain commands will not allow the null string ("") to be used as a string parameter. For example, CHANGE will not accept the null string as the string to be changed.

(12) HOLD FILE EMPTY

If you issue a LIST HOLD command and the hold file is empty, you will receive this error message.

(13) IMPOSSIBLE WITH LINES IN WORKFILE

The STOREPARMS and GETPARMS commands can only be issued when the work file is empty. This error message will be issued if the work file is not empty.

(14) INVALID LINE NUMBER

Line numbers consist of up to eight digits, with five digits maximum before the decimal point, and three digits maximum after it. Letters and special characters are not allowed in a line number.

(15) NO SPACE HERE FOR ANOTHER LINE

The smallest increment available for line numbering is .001. Therefore, if you are adding lines between line 3 and line 4, you go along in this sequence: 3.1,3.2,...3.9,3.91,3.92,... 3.99,3.991,3.992,...3.999. In this case, 3.999 is the last available line number before line number 4. If you attempt to add another line, you will receive this error message. If you want to add more lines issue a RESEQUENCE command first.

(16) LINE WOULD BE TOO LONG

You have attempted an insertion or change that would make the line longer than the currently set line length. Use the split sub-command of MODIFY to split the line at the point of insertion, then try again.

(17) A, B, & C ARE THE ONLY COMMAND BUFFERS

This error message will be issued if you attempt to reference a command buffer by a name other than A, B or C.

(18) INVALID WITHOUT AM CAPABILITY

This error message will be issued if a command is issued which requires either Account Manager or System Manager capability (such as FLUSH) and you do not have that capability.

(19) FALSE RETURN FROM EDITOR PROCEDURE

A problem has occurred in the user procedure you are using. Please make the necessary corrections in that procedure.

(20) UNDELIMITED STRING

The string used as a parameter in this command is not delimited i.e., enclosed in quotes or other allowable special characters. Or, the delimiters on either end of the string do not match.

(21) STRING NOT FOUND

The specified search string was not found in the indicated range.

(22) LINE DOES NOT EXIST

You have specified a non existent line number for a file other than your work file. (PRINT, QT, etc. commands)

(23) CAN'T OPEN TEXT FILE (File Error)

The file you are trying to text in cannot be accessed. Check the spelling of the name first. Then, enter a LISTF command to see if the file can be found in the list of all files presently in your account.

(24) WON'T EDIT FILES OF TYPE 0

You have attempted to text in a file that cannot be edited by TDP, such as the object code for a program. The filetype is substituted for the 0 when the error message is printed.

(25) CAN'T OPEN WORK FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(26) GET DSEG FAILURE. SEG. ALREADY THERE

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(27) INVALID STRING DELIMITER

You have used an illegal character to delimit a string. The legal delimiters are all non-blank characters except letters, numbers, and the special characters #, *, +, -, ., ;, and comma.

(28) READ ERROR ON WORK FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(29) INVALID LINENUMBER IN RECORD #0

There is a bad or missing linenumber in the file you are TEXTing into the work file. TDP will assign a good linenumber to that line, and it will become a permanent part of the file. This message is just to inform you of the linenumber substitution.

(30) DMOVOUT ERROR

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(31) WRITE ERROR ON WORK FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(32) DMOVIN ERROR

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(33) INVALID COMMAND FOR ADD MODE

While in ADD mode, you can temporarily exit to execute other commands by typing // and then the command you wish to execute. When the command has been completed, you return to ADD mode at the point of departure. ADD and USE are not allowed while in ADD mode.

(34) ONLY 20 RANGES ALLOWED IN LIST

You are allowed to list a series of line ranges in one command; however, no more than 20 may be included in a single command. You will need to split the rangelist in two parts, and enter the command twice, to stay within the limit of the number of ranges.

(35) FAILURE TO LOAD LIBRARY PROCEDURE

The procedure you are attempting to use has a problem that prevents it from being loaded. Make the necessary corrections in the procedure.

(36) TEXT FILE FULL; KEEP AND RE-TEXT

The size of your work file is determined by the size of the file that you text into it. If, when adding to the file, you overrun the work file, the easiest way to increase the size of the work file is to keep the file, then re-text it. The work file size will be recomputed, based on the present size of the file you are texting in.

(37) INVALID INTEGER

You have probably made a typing error when entering an offset value for a line number. Example: LIST 23+ABC would yield this error.

(38) INVALID RECORD-LINE PAIR

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(39) RANGE LIST EMPTY

You have attempted to execute a command on a line or lines that do not exist in the work file. Check your command line for typographical errors; if that isn't the problem, list the range you are interested in to see what is actually there.

(40) UNDEFINED TEXT

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(41) CAN'T OPEN KEEP FILE (File Error)

You are not able to get exclusive access to the file, perhaps because someone else is accessing it to print final output. Wait a few moments and try again.

(42) 2000 LINE MAX FOR MOVE, COPY, OR HOLD

The maximum number of lines that can be moved, copied, or held with a single command is 2000. If you really want to modify the placement of that many lines, you will have to issue more than one command.

(43) TOO MANY ERRORS IN SEQ. FIELD

When TDP attempts to text in a numbered file, if the first three line numbers are bad, it texts the file as unnumbered. However, if the first few line numbers are acceptable, but succeeding lines are consistently bad, this error occurs.

(44) NOT ENOUGH ROOM HERE TO MOVE THOSE LINES

Lines are put at the end of the file in this case.

(45) READ ERROR ON COMMAND FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(46) COMMAND LINE OVER 127 CHARACTERS.

TDP allows you to enter more than one command on a single line; however, the maximum number of characters in the line is the current line length setting, or 127 maximum.

(47) CAN'T OPEN USE FILE (File Error)

The Use file you have specified in your USE command does not exist as a permanent file on the disc.

(48) ILLEGAL SPEC. FOR GOTO IN USE FILE

The @GO command branches unconditionally to a record number or a label. Any other parameter is illegal.

(49) MISSING LABEL IN USE FILE

If you specify a label as the parameter in a @GO command, that label must have been assigned to a command line in the Use file. This error message signals that a label that is referenced does not exist.

(50) KEEP OR DELETE WORK FILE

TDP will not allow you to exit from the session, or to text in or make another file without keeping or deleting the current contents of the work file. Determine the appropriate action in your situation. To keep, type "KEEP (filename)". To delete, type "DELETE ALL".

(51) CAN'T OPEN JOIN FILE (File Error)

The file you have specified in a JOIN command does not exist on the disc.

(52) READ ERROR ON JOIN FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(53) NO WORK FILE

You have attempted an editing function on an empty work file.

(54) CAN'T OPEN INPUT FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(55) READ ERROR ON HOLD FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(56) READ ERROR ON INPUT FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(57) END OF INPUT FILE

This is often the normal ending for a batch run of TDP. If the message occurs in interactive mode, there is some problem that will need to be unraveled by a knowledgeable person. Report the error to a responsible person, such as the system manager, so the problem can be tracked down and corrected.

(58) CAN'T OPEN HOLD FILE

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(59) WRITE ERROR ON HOLD FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(60) CLOSE ERROR ON WORK FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(61) MAX. LINE NUM. EXCEEDED. RENUMBER!

The maximum line number allowed is 99999.999. You have exceeded that. If space does exist in your work file, use the resequence option with the KEEP command, and renumber the lines with a smaller increment.

(62) SPECIFY OFF OR ON

The parameter you set must be set OFF or ON.

(63) WRITE ERROR ON OUTPUT FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(64) CAN'T USE (WORD LIST) WITH LITERAL SET

(65) TOO MANY STOPS SET (15 IS MAX.)

This error message will be issued if you try to set more column stops or tab stops than is possible.

(66) FREEDSEG REQUEST DENIED

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(67) Reserved

(68) INVALID VALUE FOR PARAMETER

You tried to assign an invalid value to a parameter with the SET command. Consult the appropriate section of this manual for the correct syntax.

(69) CAN'T UNLOAD PREVIOUS PROCEDURE

The procedure that you are attempting to use has something wrong with it that prevents unloading. Please make the necessary corrections.

(70) Reserved

(71) CLOSE ERROR ON KEEP FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(72) NO EXPRESSION STORED

You have attempted to recall an expression from one of the three registers (X,Y, and Z) in the calculator. However, no expression is currently stored in the specified register.

(73) IMPOSSIBLE WITH PMARK SET

Certain operations are not allowed when PMARK is set. For example, you are not allowed to change the line length. If reasonable, set PMARK off to make the change.

(74) Reserved

(75) ARITHMETIC ERROR

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(76) CAN'T HANDLE FILE WITH REC. SIZE OF 0

The file you are attempting to text in has a wider line length than TDP's maximum of 168 characters. The record size of the file is substituted for the 0 in the error message above.

(77) CAN'T OPEN OUTPUT FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(78) FILE TO RUN NOT FOUND

The filename you have included in your RUN command is not found in your account, in PUB.SYS or in UTIL.SYS. Check your spelling, then use a LISTF to see if the program is stored under a different filename.

(79) MAX. LINE NUM. WOULD BE EXCEEDED

This error message will be issued if a RESEQUENCE operation would result in a line number greater than the maximum allowed.

(80) Reserved

(81) READ ERROR AFTER RECORD #0

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(82) NO COMMAND ALLOWED IN ADDSINGLE

This message is issued if you attempt to execute another command after issuing an ADDSINGLE command but before completing the addition.

(83) ILLEGAL FILENAME

Filenames must be a maximum of 8 characters, beginning with an alphabetic character and not including any special characters.

(84) JOIN/MERGE RECORD SIZE MUST MATCH LENGTH

When a file is joined or merged with the work file, the record size of the two files must be equal. Change the LENGTH parameter to force a match.

(85) Reserved

(86) Reserved

(87) Reserved

(88) Reserved

(89) Reserved

(90) CAN'T SCREEN MORE THAN n LINES

This error message will be issued if you attempt to screen more lines than the terminal can handle.

(91) CAN'T OPEN HELP FILE (File Error)

This error will not occur under normal circumstances. It is most likely due to faulty installation of the product , the file TDPHELP.TDPDATA.HPOFFICE not being present. PLease bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(92) READ ERROR ON HOLD FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation , such as the system manager, so that the problem can be tracked down and corrected.

(93) CLOSE ERROR ON HOLD FILE (File Error)

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation , such as the system manager, so that the problem can be tracked down and corrected.

(94) INCORRECT HELP FILE STRUCTURE

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation , such as the system manager, so that the problem can be tracked down and corrected.

(95) MPE Createprocess Intrinsic Error Message

One of the MPE Createprocess Intrinsic Error Messages (MPE Intrinsics Manual table E-14) is displayed as the result of a problem in execution of the RUN command.

(96) CAN'T OPEN MESSAGE FILE

The MESSAGE file is used by the TDP spooler. This error is most likely due to the MESSAGE file having been purged off or never built. Use the TDP command STARTSPOOL to start the spooler. One of the actions taken by the STARTSPOOL command is to build the message file.

(97) CAN'T OPEN QUEUE FILE

The QUEUE file (TDPQFILE.TDPDATA.HPOFFICE) holds the spoolqueue. This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation , such as the system manager, so that the problem can be tracked down and corrected.

(98) THERE IS NO FILE WITH THIS ID#

TDP could not find a spoolfile with this ID# in the spool queue.

(99) REQUESTED STATUS CHANGE NOT ALLOWED

TDP will only allow certain spoolfile status changes. See description of ALTERSPOOL command.

(100) OUTPUT NOT SPOOLED. QUEUEFILE FULL

The queuefile stores the details of printing requests to the TDP Spooler from the time the FINAL command is executed until the printing is completed. If there are more outstanding print requests than the queuefile can hold the spooler will not accept any new printing requests until three of the outstanding print requests have either been printed or canceled using the ALTERSPOOL command. If this error occurs often the size of the queue file will need to be increased.(See APPENDIX F for details)

(101) RANGE NOT FILLed.

This error will be accompanied by a display of the linenumber at which there was not room to add a new line generated by the FILLing process. Space for the new line can be made by using the MOVE or RESEQ commands and then the FILL command reentered. Alternatively, the FILL command can be used with the RESEQ option.

(102) TOO MANY LINES, RANGE NOT FILLed

This error occurs when using the FILL command with the RESEQ option on a range of file and there are new lines being generated by the filling process which cannot be fitted in the given range, even with a DELTA of .001. The MOVE command or RESEQ command can be used to redistribute the linenumbers to allow filling to take place.

(103) OUTPUT NOT SPOOLED-SPOOLER JOB NOT RUNNING

Output can only be spooled to a printer if the spooler job is running. The STARTSPOOL command can be used to start the spooler.

(104) COMMAND NOT EXECUTED-SPOOLER JOB NOT RUNNING

The ALTERSPOOL command and STOPSPOOL commands can only be executed when the spooler is running.

(105) SPOOLER ALREADY RUNNING

This error occurs when a STARTSPOOL command is issued when the spooler is already running.

(106) STREAM OF SPOOLER JOB FAILED

This error will not occur under normal circumstances. When it does, it is a symptom of some problem outside of your control, such as a malfunction of the computer hardware or software. Please bring this error to the attention of a responsible person at your installation, such as the system manager, so that the problem can be tracked down and corrected.

(107) GROUP SECURITY PREVENTS CREATION OF KEEP FILE (File Error)

When TDP keeps a file it first purges the old file of the same name (if it exists) and then builds a new file. If you are keeping to a group to which you do not have SAVE access or to which you do not WRITE access this error will occur and the KEEP is not done.

(108) KEEPING ACROSS ACCOUNTS NOT ALLOWED

When TDP keeps a file it first purges the old file of the same name (if it exists) and then builds a new file. MPE does not allow a user to build a file in an account to which she/he is not logged on.

(109) INSUFFICIENT CAPABILITY TO ALTER A SPOOLFILE NOT IN YOUR ACCOUNT

To alter a spoolfile not in the user's account requires System Manager Capability or to be logged onto the HPOFFICE account with Account Manager Capability.

(110) INSUFFICIENT CAPABILITY TO ALTER A SPOOLFILE YOU DIDN'T CREATE

To alter a spoolfile you didn't create requires at least Account Manager Capability. See ERROR (109) above.

(111) LONG AND NUM CANNOT BE SPECIFIED TOGETHER

These options cannot be used at the same time with this command.

(112) Reserved

(113) Reserved

(114) UNEXPECTED ERROR RETURNED FROM HP SPELL

This error will not occur under normal circumstances. If it does, it is a symptom of some problem with HP SPELL. One of the HP SPELL error messages will be displayed. Please bring this error to the attention of your system manager so that the error can be tracked down and corrected.

(115) HP SPELL IS NOT INSTALLED

This error is displayed if a spelling function is requested but HP SPELL is not installed on the system. No spelling function can be performed without HP SPELL.

ERRORExtra pair of brackets

This error occurs when using the EQN command. Count the brackets to make sure there are equal numbers of openings and closing brackets; also check their placement. This message also occurs when there are blanks surrounding a "/".

Formatter error messages

Formatter error messages are outlined below.

The error messages are unnumbered, and so are grouped according to aspects of formatting. The categories are:

running the formatter
accessing files
layout
truncation
equation processing
parameters
graphics integration
environment files
devices
others

Running the Formatter

***** FORMATTER WAS NOT RUN FROM TDP*****

The formatter can only be run using the DRAFT and FINAL commands.

Insufficient stack space available for 2680 routines

This is an MPE system problem. Please contact your system administrator.

FORMATTER terminating following an error in batch mode

Check the batch file you have created, as it contains errors which cause TDP to fail. This message is displayed when the number of errors allowed in the job is exceeded. (The number of errors allowed is set using BATCHERROR).

Errors in accessing files

Failed to open an Input file
Failed to open the List file
Failed to open the COPY file
Failed to open the Utility file

The formatter fails to open a file and so cannot begin processing. Check the way you have given the file name, as this type of error is may be caused by mistyping or not giving a file name in full. With all of these errors the file system error number is given in case you need to get help from your system administrator.

Failed to open the Contents file
Failed to open the Index file
Failed to open the Table of Figures file
Failed to open one of the Table files
Failed to close an Input file

These errors should not occur in normal circumstances. Please contact your system administrator so that the error can be tracked down and corrected.

Failed to close the List file
Failed to close the Copy file
Failed to close the Utility file
Failed to close the Contents file
Failed to close the Index file
Failed to close the Table of Figures file
Failed to close one of the Table files
Failed to close the ERRORLOG file

The file system error message is given should any of these problems occur. Get help from your system administrator as these errors are caused by abnormal circumstances.

Invalid file type for input file

The input file specified is not an ASCII file. This could be due to a typing error when specifying the file name. Use a LISTF,2 command to check the filetype of the input file you have specified.

Invalid lockword

You get this message if you try to purge an ERRORLOG, Contents, Table or Index file and the old version has a lockword on it. You need to supply the lockword so that the purge can take place.

Output file was empty, so was not saved

You get this message if an empty spool file is generated, or if the formatted file only contains formatting commands.

The List file is not ASCII

The file you are formatting to is not ASCII. You need to supply an ASCII file name.

End of file before requested starting line on IN file

The starting line number you specified is greater than the last line of the INCLUDE file.

*****WARNING*** Record size of input file exceeds 168 characters**

If records are greater that 168 characters, then formatting may cause data to be lost. Formatting will continue but data loss may mean that the document does not have the text layout you expected.

Input file record size greater than 256 bytes

TDP can not cope with an input file record length greater than 256 bytes. Please get help from your system administrator.

Illegal input filename

The name of the "FROM" file is not valid. Please check the name you have typed in.

Illegal output filename

This error occurs when you are trying to DRAFT or FINAL to a disc file with an invalid name. Examples of invalid names are those which contain more than 8 characters or which have illegal characters within them.

Invalid input file line number

There is an error in the line number sequence in the input file. Resequence the file, and if the problem persists, see your system administrator.

Illegal filename

The file name you have given is invalid, for example it has too many characters, or includes characters which are not allowed.

File not found or Illegal file type

The file you have specified does not exist, or the filetype is not valid for the specific command, for example giving a file name which is not a figure file in an ILLUSTRATION command.

File already opened

CONTENTS, INDEX, FIGURE, RASTER and TABLE files can all be named. However, if you try to name the same file twice, the error message is displayed. Please remove the second NAME command from your file.

Layout Problems

Illegal page layout values

You have given layout values which are not allowed, for example a RHT value which is smaller than the LFT value. Please check all the layout commands in your document.

No page big enough for the output

If the current logical page is not wide enough for the output, then TDP will try to find another active page which is large enough. If none is found then this error is displayed. You may need to reduce the width of your output to avoid this problem.

Command is not allowed in a footnote

This error is displayed if any of the following are found in footnotes: CENTER, FORMAT, IMAGE, RIGHT. None of these commands are allowed in footnotes.

Footing exceeds the size of bottom
Heading exceeds the size of top

The number of lines between the beginning and end of the footing or heading exceeds the space available at the top or bottom of the page.

Insufficient space to save this heading or footing

You get this message when you have exceeded 66 lines of heading and footing, the maximum TDP allows.

\LINESPACE n.5 is not allowed in MULTICOLUMN mode - rounded up

This is a warning message to tell you that the line spacing has been increased from .5 to 1.0

Truncation errors

Output from ! truncated by ! characters
Contents entry from ! truncated by ! characters
Index entry from ! truncated by ! characters
Table of Figures entry from ! truncated by ! characters
Table #1 (to #5) entry from ! truncated by ! characters

The number of characters on a line of text exceeds the space on the line and is truncated to fit. This can occur if you use an \IMAGE command or a font size which causes text to expand beyond the printed page.

Output will be truncated due to file size

The output device is configured with a record size smaller than the width required for formatted output.

Equation Processing

Illegal character detected in Equation
Extra ']' detected in Equation
Too many terms in Equation
Missing ']' detected in Equation
Extra set of brackets detected by Equation

Check the layout of characters within all equations you have used in your file. Up to 63 terms are allowed in an equation, and eight bit characters can not be used.

Parameters

Missing parameter

The command you are using requires at least one parameter.

Too many parameters

The line contains more parameters than the command requires.

Unrecognizable parameter

An invalid parameter has been supplied. This could be due to a spelling error.

WARNING * Extra parameters will be ignored**
WARNING * Extra parameters ignored**

You have specified more parameters than are required. These extra parameters will be ignored.

Illegal string

This message is displayed when no string is specified or the string is too long for the command.

String too long

The maximum length for any string is 158 characters. Please reduce the length of strings in your document.

Illegal number

This message is displayed when the number has NOT been specified when it should be, when the number is greater than 9999 or when there is an alphabetical character within the number.

Number out of range

The number you have given is out of the range for that specific command, for example RHT can only have values between 1 and 158. Check the limitations of the command you are using in this manual.

Errors concerning graphics integration

No graphics intrinsics available

This message is displayed when the PSP graphics interface has not been installed. Please get help from your system administrator.

A Figure file requires a figure name

This message is displayed when an ILLUSTRATION command has a figure file name, but no figure name has been supplied. Please check the syntax of all ILLUSTRATION commands you have used.

Figure file error

This error is caused by a corrupt figure file. Use the graphics package in which the file was created to examine the file and, if necessary, save it again.

Raster image file error

The figure file is corrupt in some way. Use the graphics package to examine the file, and if necessary save the raster image again.

Insufficient space to print this raster image file

The raster image is of a fixed size. This message is displayed when there is not enough room on the page to print the raster image. To correct the problem, you need to allow more lines on the page for the raster image.

Raster file prepared for another device

This message will be displayed when a raster image has been prepared for a 2688 and you are trying to send it to a 2680 (or vice versa). Raster images must be prepared for specific device types.

Environment file Errors

No Environment file currently defined

This message is displayed when a FONT command is specified before an environment file has been specified. Please check that you specify the environment file at the beginning of your document.

Unexpected problem occurred when accessing supplied Environment file

Get help from your system administrator or from someone who knows IFS. (This problem is caused by an unexpected response from PENVINFO. See the IFS manual for more information.)

Could not open the default Environment file
File error on default Environment file (FGETINFO)
Default Environment file could not be read
Environment file could not be opened
File error on Environment file
File error on reading the Environment file
Default Environment file was not binary
Environment file is not binary

If you get any of these messages, first check that you have given the correct file name for the environment file. Then get help from someone who knows IFS, or your system administrator.

Default Environment file had incorrect record size
Environment file has wrong record size

The record size for environment files must be 512 bytes. If you need help, consult your system administrator or someone who knows IFS.

Default Environment file had the wrong file code
Environment file has wrong file code

The file code for environment files is PENV 1112. If you need more help consult your system administrator or an IFS expert.

Default Environment file has not been compiled
Default Environment file has been changed since last compiled
Environment file has been changed since last compiled
Environment file has not been compiled

To compile the environment file, you need to go into IFS and run the compiler. Get help from an IFS expert or from your system administrator if you have never used IFS before.

Default Environment file compiled for wrong device
Environment file was compiled for another device

The environment file needs to be compiled for the type of device you are using (2680 or 2688). You can not use an environment file originally compiled for one type of device on another without recompiling.

Environment file was specified on the file equation

This message is displayed when there is an environment file specified both in the TDP file and in a file equation being used to redirect output. If you need to redirect output, using a new environment file, then take the \ENV command out of your TDP file.

Device Errors

Illegal output filename for SPOOLED output

The filename you have given for spooled output is not valid.

Illegal output device number for directed output

Check that the device number that you have specified for your output refers to a device supported by TDP. Please check the name you have typed in.

No intrinsic support for HP 2688 available
No intrinsic support for HP 2680 available

This is an MPE system problem. Check with your system administrator that the printer intrinsics have been installed.

Illegal output device specification

The output device you have specified is not one of TDP's supported devices. Look at Appendix F of this manual for more information about devices which are supported.

Font not defined

Check that the font you are trying to use is defined both in your TDP file and is available in the environment file you are using. Use the command \FONTID fontname where the fontname is not defined in the environment file. If you continue to have problems, consult your system administrator, or an IFS expert.

Page not defined

The logical page you are trying to use has not been defined in the environment file you are using. Please get help from an IFS expert, or your system administrator.

Others

Identifier has not been set

The font identifier and font change commands have not been set in the file you are using. You need to use a \FONTID command if the font name is not defined in the environment file.

Failed to unload last user hyphenation routine
Failed to load last hyphenation routine

Please get help from your system administrator. (They will need to check LOADPROC and UNLOADPROC in the intrinsics manual and find out why they fail.)

IF statement error

This error message is displayed when there is a syntax error in an IF command. Please check all IF statements in your file.

Insufficient space to save this macro

The total number of macro commands must not exceed 1024 characters. Look up the MACRO command in this manual for more information.

Unrecognizable command

TDP does not recognize this command. Check that it is spelled correctly.

Command not allowed in this context

TDP does not allow the command at this time. For example, some formatting commands are not allowed in headers and footers. Check the limitations of the command you are trying to use in this manual.

Included files nested greater than 16 deep

When you are using INCLUDE files, the maximum number of nested calls allowed is 16.

Invalid escape sequence detected

There is an invalid escape sequence in your file, for example ^X. Check through and delete the sequence.

Output would be reduced below 1 character

You will receive this error message when the sum of the widths of columns and column margins exceeds the width of the page. Please check the total width of columns and column margins.

No commands allowed following a 'BEGIN'
No commands or parameters allowed on an 'END'

The BEGIN and END commands do not allow parameters, or other commands on the same line. Please check all BEGIN and END commands in your file.

'COMMAND' found when looking for 'COMMAND END'

TDP expects to find COMMAND BEGIN and COMMAND END in pairs. If the COMMAND is repeated before COMMAND END appears then this message will be displayed.

'COMMAND xxx' found when looking for 'COMMAND END'

The formatter has found extra parameters in the command which do not fit the syntax, for example TABLE END SALES. Here the name of the table, SALES, is not part of the command syntax.

Identifier is a single alphanumeric character

An identifier of more than one character in length has been specified to a command where only one character is required.

EOF found when looking for 'COMMAND END'

The formatter has reached the end of the file before it has finished a command. A COMMAND END is needed.

Too many escape sequences on one line

This message is displayed when there are more than the maximum of number of 63 intraline commands on one line. Splitting the line of the file may solve this problem if it is necessary to have a large number of intraline commands.

Line too long to expand this escape sequence

When ^M or ^D is expanded, the line length exceeds TDP's limits. Splitting the line may solve this problem.

WARNING * invalid intraline expand - blank assumed**

The file has an invalid option for an expand, for example an escape character. Check through your file and correct the intraline command.

Command summary

ABORTSPOOL

```
ABORTSPOOL
```

ACTIVATE

```
\ACTIVATE [(] n [,n [,n [...]]] [)]
```

ADD

```
ADD[Q]  [linenumber]  [,HOLD[Q] ]
        [string    ]  [,filenamc]
```

ADDLINE

```
ADDLINE[Q] [linenumber] [,text]
           [string    ]
```

ADDSINGLE

```
ADDSINGLE[Q]  [linelist]
```

ALIGN

```
ALIGN(R                ) [Q][rangelist][,ADJUST][,NOTEXT]
     (L                )
     (C                )
     (D                )
     (special character)
```

ALTERNATE

```
\ALTERNATE [ON ]
           [OFF]
           [REV]
```

ALTERSPOOL

```
ALTERSPOOL file id#,(STOPPED   )
                    (CANCELLED )
                    (READY     )
```

BACKSPACE

```
\BACKSPACE [char]
```

BATCHERROR

```
\BATCHERROR n
```

BLANK

```
\BLANK [char]
```

BOLD

```
\BOLD [n]
```

BOTTOM

```
\BOTTOM n
```

BOX

```
\BOX n [string] [,ASAP] [,FULL]
```

CATALOG

```
CATALOG filelist
```

CENTER

```
\CENTER [n]
        [B]
```

CHANGE

```
CHANGE[Q] {string         } TO {string    } [IN rangelist]
          ((wordlist)     }    ((wordlist)) [,ASK        ]
          {column[/column]}    {column    } [,LIT        ]
                                            [,NONLIT     ]
                                            [,SL         ]
```

CHECK

```
CHECK[Q] [rangelist] [,UNN     ]
                     [,OFFLINE ]
                     [,NEW     ]
                     [,LONG    ]
```

CLOSE

```
CLOSE
```

CMARGIN

```
\CMARGIN [(] {string} [, [string] [, [...]]] [)]
             {n     }    [n     ]
```

COLINSERT

```
COLINSERT[Q] {column} TO string  [IN rangelist]
             {string}            [,ASK        ]
                                 [,LIT        ]
                                 [,NONLIT     ]
                                 [,NOTEXT     ]
```

COLMOVE

```
COLMOVE[Q] columnrange,column [,rangelist]
                              [,NOTEXT   ]
```

COLREPLACE

```
COLREPLACE[Q] {column} TO newstring [IN rangelist]
              {string}              [,ASK        ]
                                    [,LIT        ]
                                    [,NONLIT     ]
                                    [,NOTEXT     ]
```

COLUMN

```
\COLUMN {n                         } [,CONTROLLED]
        {(n       [,n       [...]])} [,NEED      ]
        {(n string[,n string[...]])} [,SPACE n   ]
        {END                       }
```

CONTENTS

```
\CONTENTS [n      ] [,IN n           ]
          [string] [,NEW            ]
          [BEGIN ] [,NO[DOTS]       ]
          [END   ] [,SPACE n [AFTER]]
          [ON    ] [,NEED n         ]
          [OFF   ]
```

COPIES

```
\COPIES n
```

COPY

```
COPY[Q](rangelist    ) TO (linenumber) [BY increment]
                                        [,NOTEXT    ]
```

CRUNCH

```
CRUNCH (filename)
```

DEACTIVATE

```
\DEACTIVATE [(] n [,n [,n [...]]] [)]
```

DELAY

```
\DELAY
```

DELETE

```
DELETE[Q] (rangelist    ) [,HOLD  ]
                          [,NOTEXT]
```

DELIM

```
DELIM[Q] "char" [rangelist] [,NOTEXT]
```

DISPLAYPARMS

```
DISPLAYPARMS
```

DOWNSHIFT

```
DOWNSHIFT[Q] (rangelist) [,ADJUST]
                         [,NOTEXT]
```

DRAFT

```
DRAFT[Q]  [FROM filename ]
          [TO filename   ]
          [ASK           ]
          [COPIES=number ]
          [AUTO          ]
          [RO            ]
          [DEFER         ]
          [SP            ]
```

DOUBLE

```
\DOUBLE [ON ]
        [OFF]
```

END

```
\END
```

END

```
END
```

ENVELOPE

```
ENVELOPE [filename]
```

ENVIRONMENT

```
\ENVIRONMENT filename
```

EOD

```
\EOD
```

EQN

```
EQN [linelist] [,HOLD]
```

EQUATION

```
\EQUATION
```

ERROR

```
\ERROR {OFF}
       {ON }
```

ESCAPE

```
\ESCAPE [char]
```

EXIT

```
\EXIT
```

EXIT

```
EXIT
```

FIGURE

```
\FIGURE n [string] [,ASAP] [,FULL]
```

FILL

```
FILL[Q] rangelist [,RESEQ]
```

FINAL

```
FINAL[Q]  [FROM filename ]
          [TO filename   ]
          [ASK           ]
          [COPIES=number ]
          [AUTO          ]
          [RO            ]
          [DEFER         ]
          [SP            ]
```

FIND

```
FIND[Q] {string     } [IN rangelist] [,HOLD[Q]]
        {(wordlist)}                 [,LIT    ]
                                     [,NONLIT ]
                                     [,NOTEXT ]
                                     [,SL     ]
                                     [,UNN    ]
```

FINDNEXT
FINDNUMBER

```
FINDNEXT[Q]   {string           } [,LIT   ]
FINDNUMBER[Q] {line[(column)]} [,NONLIT]
                                  [,NOTEXT]
                                  [,SL    ]
```

FLUSH

```
FLUSH filelist
```

FONT

```
\FONT {("name"/n [,"name"/n])}   n
      {          id          }
```

FONTEQ

```
\FONTEQ  {BOLD  } [(] "name" [, "name"] [)]
         {GHOST }        n   [,     n  ]
         {NORMAL}
         {ALTER }
```

FONTID

```
\FONTID id [(] "name"/n [, "name"/n] [)]
```

FOOT

```
\FOOT {string} [,A [,REV]]
      {BEGIN } [,REV     ]
      {END   } [,I       ]
      {OFF   } [,L       ]
      {ON    } [,C       ]
               [,R       ]
```

FOOTNOTE

```
\FOOTNOTE {n      }
          {string}
          {BEGIN }
          {END   }
```

FORMAT

```
\FORMAT [n]
        [B]
```

GETPARMS

```
GETPARMS
```

GHOST

```
\GHOST [n]
```

GLUE

```
GLUE[Q] rangelist [,NOTEXT]
```

GO

```
\GO
```

HEAD

```
\HEAD {string} [,A [,REV]]
      {BEGIN } [,REV     ]
      {END   } [,I       ]
      {OFF   } [,L       ]
      {ON    } [,C       ]
               [,R       ]
```

HEADLINE

```
\HEADLINE n [,A [,REV]]
            [,REV     ]
            [,I       ]
            [,L       ]
            [,C       ]
            [,R       ]
```

HELP

```
HELP [command [SYNTAX/ALL]]
     [MENU]
     [FURTHER]
     [topic [keyword/ALL]]
```

HOLD

```
HOLD[Q] [rangelist [,APPEND] [,NOTEXT] [,NUM]]
        [filename  [rangelist]                ]
```

HYPHEN

```
HYPHEN
```

HYPHALLCAPS

```
\HYPHALLCAPS [ON ]
             [OFF]
```

HYPHCHAR

```
\HYPHCHAR [char]
```

HYPHDBL

```
\HYPHDBL [ON ]
         [OFF]
```

HYPHEN

```
\HYPHEN [n] [ON            ]
            [OFF           ]
            [AUTO [,ASK]   ]
            [string [,ASK]]
```

HYPHFIRSTCAP

```
\HYPHFIRSTCAP [ON ]
              [OFF]
```

HYPHFLAGS

```
\HYPHFLAGS n
```

HYPHLAST

```
\HYPHLAST [ON ]
          [OFF]
```

IF

```
\IF [NOT] {MAIN    } (commandlist)
          {INFILE  }
          {termtype}
          {Y       }
          {N       }
          {SF      }
```

ILLUSTRATION

```
\ILLUSTRATION filename[:figure] [n] [,rotate  ] [,A [,REV]]
                                    [,P [,REV]] [,REV     ]
                                                [,I       ]
                                                [,L       ]
                                                [,C       ]
                                                [,R       ]
```

IMAGE

```
\IMAGE [n]
       [B]
```

INCLUDE

```
\INCLUDE filename [(first line [/ last line])]
```

INDENT

```
\INDENT {[(] n [,n [,n ... ] ] [)]}
        {ON }
        {OFF}
```

INDEX

```
\INDEX [n     ] [,IN n    ]
       [string] [,NO[DOTS]]
       [BEGIN ]
       [END   ]
       [ON    ]
       [OFF   ]
```

INFORMAT

```
\INFORMAT n
```

INLFT

```
\INLFT n
```

INRHT

```
\INRHT n
```

INSERT

```
INSERT[Q] {line[(column)]} [,HOLD[Q]]
          {string        }
```

JOIN

```
JOIN[Q] filename [filerange] [TO linenumber] [BY increment ]
```

JUSTIFY

```
\JUSTIFY [ON ]
         [OFF]
```

KEEP

```
KEEP [filename] [range] [,ASK  ]
                        [,CODED]
                        [,HOLD ]
                        [,NUM  ]
                        [,RESEQ]
                        [,UNN  ]
```

LAYOUT

```
\LAYOUT [NEW] [n]
```

LFT

```
\LFT n
```

LINESPACE

```
\LINESPACE {n}
           {n.5}
```

LIST

```
LIST[Q] [rangelist] [,EXPAND ]
                    [,HOLD   ]
                    [,LONG   ]
                    [,NEW    ]
                    [,NOTEXT ]
                    [,OFFLINE]
                    [,PMARK  ]
                    [,UNN    ]
                    [,WINDOW ]
```

LPTOP

```
\LPTOP n
```

M

```
\M{id} {[PROMPT] {string}}
       {SET text         }
```

MAKE

```
MAKE[Q] filename
```

MAILER

```
MAILER [filename]
```

MARGIN

```
\MARGIN n
```

MERGE

```
MERGE[Q] filename [filerange]
```

MODIFY

```
MODIFY[Q] [rangelist] [,EXPAND]
```

MONITOR

```
MONITOR
```

MOVE

```
MOVE[Q] rangelist TO {linenumber} [BY increment]
                     {string    } [,NOTEXT]
```

NAME

```
NAME filename
```

NAME

```
\NAME [CONTENTS    ] filename
      [INDEX       ]
      [TABLE FIGURE]
      [RASTER      ]
```

NEED

```
\NEED  {n}
       {B}
```

NEW

```
\NEW [EVEN [BLANK ]]
     [ODD  [string]]
```

NEXT

```
\NEXT
```

OVERLAY

```
OVERLAY[Q] filename [filerange]
```

PAGE

```
\PAGE [#]startingpage [/[#]endingpage]
```

PAGELENGTH

```
\PAGELENGTH n
```

PAGENO

```
\PAGENO {n      } [,A [,REV]]
        {string} [,REV     ]
                 [,I       ]
                 [,L       ]
                 [,C       ]
                 [,R       ]
```

PAGENOLINE

```
\PAGENOLINE n [,A [,REV]]
              [,REV     ]
              [,I       ]
              [,L       ]
              [,C       ]
              [,R       ]
```

PARAGRAPH

```
\PARAGRAPH
```

PAUSE

```
\PAUSE [ON ]
       [OFF]
```

PITCH

```
\PITCH {10}
       {12}
       {15}
```

PRINT

```
PRINT[Q] filename [range    ] [,EXPAND ]
                  {filerange} [,NUM    ]
                              [,OFFLINE]
                              [,RNUM   ]
                              [,UNN    ]
```

PROCEDURE

```
PROCEDURE   [pname [,{G}] [rangelist]]
                     {P}
                     {S}
```

PROMPT

```
PROMPT
```

PROMPT

```
\PROMPT [char]
```

PROPORTIONAL

```
\PROPORTIONAL [METAL  ]
              [PLASTIC]
```

Q

```
Q[Q] string
```

QUICKTEXT

```
QUICKTEXT filename [,UNN]
```

RACK

```
RACK[1]
    [3]
```

RED

```
\RED [n]
```

REDO

```
REDO
```

REPLACE

```
REPLACE[Q] [rangelist ] [,HOLD[Q]]
           [string    ]
           [(wordlist)]
```

RESEQUENCE

```
RESEQUENCE [line] [BY increment] [,SORT]
```

REVISION

```
\REVISION {MARK [char]}
          {OFF        }
          {ON         }
```

RHT

```
\RHT n
```

RIGHT

```
\RIGHT [n]
       [B]
```

RMARGIN

```
\RMARGIN n
```

SCREEN

```
SCREEN [rangelist] [,NUM    ]
       [,LONG   ]
       [,ADJUST]
```

SECTION

```
\SECTION string [,A [,REV]]
                [,REV     ]
                [,I       ]
                [,L       ]
                [,C       ]
                [,R       ]
```

SET

```
SET

      [BATCHERROROK               ]      [NORUN                      ]
      [BIGBLOCK [[=] ON ]         ]      [NOSTREAM                   ]
      [         [[=] OFF]         ]      [NOTAB                      ]
      [BLIT [[=] ON ]             ]      [PERMYES[[=]ON ]            ]
      [     [[=] OFF]             ]      [       [[=]OFF]            ]
      [COLSTOP [[=] n1,n2,...,n14]]      [PMARK                      ]
      [        [[=] SPREAD n     ]]      [PNEWS [[=]ON ]             ]
      [COMP                       ]      [      [[=]OFF]             ]
      [CONTINUE                   ]      [POSTADD [=] {n}            ]
      [CONTROLY                   ]      [         {COMPILE   }      ]
      [DELTA [=] {n}              ]      [         {TDP[/3000]}   {C}]
      [DICTIONARY [=] {filename}  ]      [PRIORITY {RUN       }[=]{D}]
      [DISPLAY                    ]      [         {FORMAT    }   {E}]
      [ERROR [=] {n}              ]      [         {SPOOLER   }      ]
      [ESCAPE [=] char            ]      [PROMPT [=] char            ]
      [FILETYPE [=] {n}           ]      [QUIET                      ]
      [FORMAT[=]{COBOL  }         ]      [RIGHT [=] {n}              ]
      [         {DEFAULT}         ]      [RUN                        ]
      [         {DIARY  }         ]      [SCREENMAX [=] {n}          ]
      [HIGHC [=] {n}              ]      [SESSQUIET [[=] ON/OFF]     ]
      [LANGUAGE [=] {string}      ]      [SHORTERROR                 ]
      [LEFT [=] {n}               ]      [SIZE [=] {n}               ]
      [LENGTH [=] {n}             ]      [SPOOLER [[=] ON/OFF]       ]
      [LINE [=] {n}               ]      [STAMP char                 ]
      [LIT                        ]      [STOP                       ]
      [LOCPARM [=] {ON }          ]      [STREAM                     ]
      [            {OFF}          ]      [SYSERROR [=] {n}           ]
      [LONGERROR                  ]      [TAB                        ]
      [LOWC [=] {n}               ]      [TABCHAR [= char]           ]
      [MISS[OK]                   ]      [        [= tabkey]         ]
      [MPE                        ]      [TABSTOP [[=] n1,n2,...,n15]]
      [MUF [[=]ON ]               ]      [        [[=] SPREAD n     ]]
      [    [[=]OFF]               ]      [TERMINAL {string }         ]
      [MUST                       ]      [TERMTYPE {n}               ]
      [NO                         ]      [VBIGBLOCK [[=] ON ]        ]
      [NOBLIT                     ]      [          [[=] OFF]        ]
      [NOBREAK                    ]      [WINDOW [=] {n}             ]
      [NOCOMP                     ]      [YES                        ]
      [NOCONTROLY                 ]      [ZFILL [[=] ON ]            ]
      [NOMPE                      ]      [      [[=] OFF]            ]
      [NONLIT                     ]      [ {n}                       ]
```

SHOWSPOOL

```
SHOWSPOOL
```

SKIP

```
\SKIP {n}
```

SPACE

```
\SPACE [EVERY] n
```

SPELL

```
SPELL[Q] [rangelist] [,UNN     ]
                     [,OFFLINE ]
                     [,NEW     ]
                     [,LONG    ]
```

SQUEEZE

```
SQUEEZE[Q] [rangelist] [,NOTEXT]
```

STARTSPOOL

```
STARTSPOOL
```

STOPSPOOL

```
STOPSPOOL
```

STOREPARMS

```
STOREPARMS
```

TAB

```
TAB
```

TABLE FIGURE

```
\TABLE {FIGURE} {string} [,IN n              ]
                {BEGIN } [,NEW               ]
                {ON    } [,SPACE n [AFTER]]
                {OFF   } [,NEED n            ]
                         [,NO[DOTS]          ]
       {END}
```

TERMCAP

```
TERMCAP
```

TERMSTAT

```
TERMSTAT
```

TEXT

```
TEXT filename [filerange] [,CODED]
                          [,UNN  ]
```

TIME

```
TIME
```

TOP

```
\TOP n
```

TOTAL

```
TOTAL[Q] [rangelist] [,APPEND]
                     [,NOTEXT]
```

TRY

```
\TRY
```

UL

```
\UL [n  ]
    [ON ]
    [OFF]
```

UNCRUNCH

```
UNCRUNCH filename
```

UPSHIFT

```
UPSHIFT[Q] [rangelist] [,ADJUST] [,NOTEXT]
```

USE

```
USE[Q] filename [,NOTEXT]
```

UW

```
\UW [n  ]
    [ON ]
    [OFF]
```

VERIFY

```
\VERIFY [PAGES      ]
        [FONTS      ]
        [ENVIRONMENT]
        [ALL        ]
```

VERIFY

```
VERIFY

      [ACTIVE      ]        [HOLD       ]        [PROMPT      ]
      [ALL         ]        [LANGUAGE   ]        [QUIET       ]
      [AM          ]        [LEFT       ]        [RESTRICTIONS]
      [BATCHERROROK]        [LENGTH     ]        [RIGHT       ]
      [BIGBLOCK    ]        [LINE       ]        [SCREENMAX   ]
      [BLIT        ]        [LIT        ]        [SESSQUIET   ]
      [CLEAN       ]        [LOCPARM    ]        [SHORTERROR  ]
      [COLSTOP     ]        [LOWC       ]        [SIZE        ]
      [CONFIG      ]        [LOTS       ]        [SM          ]
      [CONTINUE    ]        [MISS       ]        [SPOOLER     ]
      [DELTA       ]        [MUF        ]        [STAMP       ]
      [DEPTH       ]        [MUST       ]        [STOP        ]
      [DICTIONARY  ]        [NO         ]        [TAB         ]
      [DISPLAY     ]        [NONLIT     ]        [TERMINAL    ]
      [ERROR       ]        [NOTAB      ]        [TERMTYPE    ]
      [ESCAPE      ]        [PERMYES    ]        [TIME        ]
      [FATHER      ]        [PMARK      ]        [TOTAL       ]
      [FILE        ]        [PNEWS      ]        [VBIGBLOCK   ]
      [FILETYPE    ]        [POINTER    ]        [WINDOW      ]
      [FORMAT      ]        [POSTADD    ]        [YES         ]
      [HIGHC       ]        [PRIORITY   ]        [ZFILL       ]
```

WIDOW

```
\WIDOW [ON ]
       [OFF]
```

WIDTH

```
\WIDTH n [INCHES]
```

WORDMOVE

```
WORDMOVE[Q] tokenrange,position [,rangelist]
```

Z: :

```
Z[Q]::=
ZP::= "prompt"
Z::%name
Z::="string"
```

@

```
@[n]
 [Z]
```

@D

```
@D[A]
  [B]
  [C]
```

@F

```
@F[A[M]]
  [B[M]]
  [C[M]]
```

@GO

```
@GO[TO] {record number}
        {label}
```

@IF

```
@IF [NOT](FILE "filename") THEN GO TO {label         }
         {string         }             {record number}
         {x/y/z  </=/>  n}
```

@L

```
@Ln
```

@M

```
@M
```

@S

```
@S[A [= command string]]
  [B [= command string]]
  [C [= command string]]
```

=

```
=[x[= {arithmetic expression}]]
=[y[= {arithmetic expression}]]
=[z]
```

=C

```
=C
```

=L

```
=L
```

=M

```
=M
```

=S

```
=S[=expression]
```

=TOTAL

```
=TOTAL[Q] rangelist [,APPEND]
                    [,NOTEXT]
```

*****

```
\* any text
```

^A	^A{text}^N	
^B	^B{text}^S	
^C	^C{char}	
^D	^D{0}	6/10/88
	{1}	June 10, 1988
	{2}	Friday, June 10, 1988
	{3}	Friday, June 10, 1988, 2:54 PM
	{4}	June 10
	{5}	1988
	{6}	2:54 PM
	{7}	88
	{8}	Friday
	{9}	10/6/88
	{A}	10 June, 1988
	{B}	Friday, 10 June, 1988
	{C}	Friday, 10 June, 1988, 2:54 PM
	{D}	10 June
	{E}	10.6.88
	{F}	10/6/1988
	{G}	6/10/1988
	{H}	6.10.1988
	{I}	06.10.1988
	{J}	Fri
	{K}	FRI
	{L}	10
	{M}	June
	{N}	Jun
	{O}	JUN
	{P}	6
	{Q}	2:54
	{R}	14:54
	{S}	PM
	{T}	161
^E	^E{char}	
^F	^F{id} text ^S	
^G	^G{text}^S	
^M	^M{id}	
^N	^N	
^R	^R{text}^S	

^S	`^S`
^U	`^U(text)^S`
^W	`^W(text)^S`
^_	`^_`
^+	`^+`
^-	`^-`
^>	`^>`
^<	`^<`
^#n	`^#n`
^#(n)	`^#(n)`
^#F	`^#F`
^#P	`^#P or ^#(P)`
^#S	`^#S`

#n	`\#n {string}`
	`{n      }`

#P	`\#P {string}`
	`{n     }`

Technical information

This appendix contains some technical details on the internal operation of TDP that might be useful to the system manager or other sophisticated user. Most TDP users will have no need for this information.

Workfile size

The maximum size of the workfile is either the number of text records allowed (NTR) or the value of SIZE, whichever is greater. The number of text records allowed (NTR) depends on the size of the file TEXTed (TFS), as follows:

TEXT File Size (TFS)	Number of Text Records Allowed
(# of records)	In the Work File (NTR)
0<=TFS<=1000	2000
1000<=TFS<=2000	TFS + 1000
2000< TFS<=4000	TFS + 50%
4000< TFS	TFS + 2000

Note that if no file was TEXTed in (that is, if the work file was created with ADD, MAKE, or NAME), then TFS=0; thus, NTR=2000. Since the default value of SIZE=2000, and the maximum size of the workfile is the greater of NTR and SIZE, the maximum workfile size in this case is 2000 records.

Note that when DELETE and MOVE are used, the space is not recovered. Thus, a series of MOVE commands, while apparently not altering the number of lines in the work file, will actually use available space.

All extents for the workfile are allocated immediately when the workfile is made.

Workfile record size

The workfile record size is either 96 bytes (default) or the "input record size" plus 2 bytes (for a character count) plus 22 bytes (for room to expand the record), whichever is greater. The "input record size" depends on how the workfile was created, as follows:

1 If the file TEXTed was a default numbered file, then the "input record size" is the TEXT file record size minus 8 bytes.

2 If the file TEXTed was a COBOL numbered file, then the "input record size" is the TEXT file record size minus six bytes.

3 If the file TEXTed was an unnumbered file, then the "input record size" is the same as the TEXT file record size.

4 If no file was TEXTed (that is, if the workfile was created with an ADD, MAKE, or NAME), then the "input record size" is the value of LENGTH. Note that the default value of LENGTH is 72 if DEFAULT or DIARY format is set and 74 if COBOL format is set.

Note that the LENGTH parameter is always set to the "input record size" (as determined above). Thus, increasing the line length by up to 22 characters can be done by resetting the value of LENGTH (see the SET command, Chapter 3).

If you wish to increase the record size of a line by more than 22, you can:

1 KEEP and re-TEXT, or

2 JOIN the file to an empty workfile with the length SET to the required size.

Workfile structure

The workfile used by TDP is maintained as a link list file. Eight records of text are preceded by one internal record with line numbers and pointers for the text records. This internal record contains a line number, a backward pointer, and a forward pointer for each of the eight records. The pointers are double integers showing the logical record number of the records preceding and following the current record. The line number is a double integer indicating the line number associated with the current record.

The first block of records is a special case. Record 0 of the workfile is the first internal record with pointers, and line numbers, but record one is not the first text record. Record one is used for parameters applicable to the workfile, which are read from TDPPARMS and PARMSET (if LOCPARM is SET). Examples of parameters which are held here are: the format of the file; the number of active lines in the workfile; the settings of RIGHT, LEFT, HIGHC, LOWC, etc. The end-of-file pointer and the parameters of record one are updated each time a command line is executed that alters a parameter or the end-of-file location. Also, when in add mode, the updating occurs according to the value of POSTADD (see SET command).

The text records begin with record 2. Each text record begins on a word boundary. Byte 0 of each record is not used; byte one contains the character count for the line; the rest of the record contains the text for that line.

There is a DISPLAY command that displays the link list structure of your workfile. This command is not included in earlier sections of this manual. An example and explanation follow.

```
               /DISPLAY - displays the internal structure of a
                          workfile as shown below.


                        +--------------------- Last Block Size
                        !
                        !     +--------------- Record number of
                        !     !                last line

LAST BLOCK ----> 9     5     14    9000  <--- Last Line Number
(logical record  LB    LBSZ  LLR   LLINE        x 1000
 number)         1             0       2
                 2       1     1       3
                 3       2     2       4
Line number ---- 4 --+   3     3       5
                 5   !   3.1   4       6
Logical record ->6   +-> 4     5 <---- 7  <-- Backward and Forward
number for line  7       4.1   6       8      Pointers for Record 6
number 4         8       5     7       10     (logical record
                 10      6     8       11      numbers)
                 11      7     10      12
                 12      7.2   11      13
                 13      8     12      14
                 14      9     13      999999
                 15  99999.999 99999999999999
                 16  99999.999 99999999999999
                 17  99999.999 9999999910
```

The top two rows list the following information: the number above LB is the logical record number of the internal record of the Last Block; the number above LBSZ is the Last Block Size in number of records; the number above LLR is the logical record of the last text line; the number above LLINE is the line number of the last line (a decimal point belongs three places to the left).

The listing below the first two lines consists of four columns, as follows: column 1 is the logical record number; column 2 is the corresponding line number; and columns 3 and 4 are the corresponding backward and forward logical record pointers.

Note that in the above example, logical record number 9 is not listed, since it is the internal record for the next 8 records.

Records 11 through 17 in the example are unused records of the second block. They are filled with 9's to indicate this. Note that in record 14, there is no valid forward pointer--it is also filled with 9's.

You may specify a starting record number for the display, but not a range. This is a text record number i.e. the pointer records are not counted. Thus the command DISPLAY 900 will display starting at text record number 900 and continuing until the end of the file is reached, or until CTRL-Y is typed.

Extra data segments

When a numbered file is TEXTed into the workfile, TDP builds an extra data segment containing one double word for each block of text. The first block contains seven lines of text; all others contain eight. This facilitates finding lines in the file.

The CLOSE command copies the extra data segment into your workfile. Space for this is not reserved in the workfile; if there is not enough room, the user is told and the information is not copied.

For very large files, the maximum extra data segment size (configurable by the system manager) may not be sufficient. A GET DSEG FAILURE error will result.

Several solutions are possible: you could TEXT the file unnumbered, since no extra data segment is built for unnumbered files; or you could request that the system manager reconfigure a larger data segment for your use.

Calling TDP programatically

It is possible to call TDP from a user program. The program must be created, prepped and compiled and able to call TDP to set desired parameters, and/or issue TDP editing commands.

There are two primary uses for this capability:

1 To restrict a specific user to a specific task (a word processing user, for example) so there is no access to other capabilities of the HP 3000, and;

2 To allow a programmer access to TDP's capabilities almost as if they were system intrinsics.

A sample program in SPL follows:

```
$CONTROL USLINIT, NOLIST, NOWARN
BEGIN
INTRINSIC CREATE, ACTIVATE, SENDMAIL;
ARRAY A (0:39);
BYTE ARRAY BA(*)=A;
INTEGER PIN;
MOVE BA := "TDP.PUB.SYS";
CREATE (BA,  ,  PIN,  ,1);
A := "  ";
MOVE A(1)  := A,(39);
MOVE BA :=
"(your command line goes here)"
;
SENDMAIL (PIN,40,A,0);
ACTIVATE (PIN,2);
END.
```

Line 12 of the program contains the command line to be passed to TDP. Quotation marks are required around the line, and no quotes can be included within it. More than one TDP editing command can be listed on the line as long as the total character count is less than 72. Multiple commands are separated by semicolons.

Three examples of line 12 are shown below:

```
12  "SET NORUN,NOCOMP,NOSTREAM,NOBREAK"
12  "TEXT SAM;F 'IQ', ALL,HOLD;L HOLD,OFFLINE;END"
12  "USE MYUSE;EXIT"
```

The first example sets the parameters shown in the command line. The second texts in the file named SAM, finds all occurrences of the "word" IQ, puts the found lines in the HOLD files, lists that file on the system printer; and then exits. The third example simply calls upon TDP to execute the commands in the usefile MYUSE and then exits.

The program is kept in a file named CALLTDP. The program is compiled, prepped and saved with these MPE commands called from TDP

```
    /SPLPREP CALLTDP ( PH); RENAME $OLDPASS,fn; SAVE fn
```

where fn is the name you have chosen for the program file to be run by the user.

The program may now be moved to other accounts or groups, or to PUBSYS.

Please consult the **MPE Intrinsics Manual** for details on the CREATE, ACTIVATE, and SENDMAIL intrinsics.

Filecodes used by TDP

The file codes used by TDP are shown below:

Filecode	Mnemonic	Explanation
1052	EDTCT	COBOL source file
1054	TDPDT	TDP DIARY file
1056	TDPP	TDP proof-marked file
1057	TDPCP	Proof-marked COBOL file
1058	TDPQ	TDP workfile (DEFAULT and DIARY formats)
1059	TDPXQ	TDP workfile (COBOL format)

Comparison with Edit/3000

TDP

This appendix is written for the experienced user of EDIT/3000 who wants to begin using TDP as quickly and easily as possible. For the most part, TDP can be considered a functional superset of EDIT/3000, i.e., nearly all the EDIT/3000 commands have a TDP counterpart; however, TDP offers a much greater range of capabilities.

TDP provides a more extensive set of editing commands than those found in the EDIT/3000 subsystem. In addition, TDP includes some 80 commands for formatting your text file for final output.

- Programs in COBOL, SPL, FORTRAN, PASCAL and RPG can be compiled and run without leaving the TDP environment
- The current line pointer is updated as a range is scanned (see Section 2 for details)
- Most MPE commands are recognized with or without the leading colon
- The text file can be "proofmarked" to identify lines that have been modified, and the date the modifications were made
- A built-in calculator allows you to add, subtract, multiply, divide, and compute square roots
- The FIND command finds all occurrences of a string in the file eliminating most of the need need for WHILE loops
- Conditional and unconditional branching are both allowed within USE files
- TDP does not handle variable length records; however the CRUNCH command allows files to be greatly compacted
- TDP can be called programatically from a user program
- Maximum record length for TDP is 168 characters compared to 255 for EDIT/3000
- The string search logic is different and can be controlled by the user (see Chapter 2)
- New options have been added to some of the commands to make them more useful
- A great many new editing commands have been added to provide more powerful editing capabilities

Command Comparison

All of the EDIT/3000 commands are listed below, with the equivalent TDP command noted beside it. Where new commands have been added that expand the TDP capability compared to EDIT/3000, they are noted, as are additional options. The few commands that do not have a TDP equivalent are either no longer necessary or have been replaced by simpler constructs.

EDIT/3000	TDP	Comments
ADD	ADD	
	ADDLINE	new command
	ADDSINGLE	new command
	MAKE	new command
BEGIN/END		used only with WHILE; @command constructs replace WHILE
CHANGE	CHANGE	has an ASK option
	COLINSERT	new command
	COLREPLACE	new command
COPY	COPY	
DELETE	DELETE	requires range or line specification
END/EXIT	END/EXIT	prevents exit without explicit keep or delete of a modified workfile
FIND	FINDNEXT	different name for same command
	FIND	finds all occurrences of the string within a range
GATHER	MOVE	different name for same command
GATHER ALL	RESEQ	new name for this form
HOLD	HOLD	can keep and list the hold file
INSERT	INSERT	
JOIN	JOIN	params are optional for range
	MERGE	new command
	OVERLAY	new command
KEEP	KEEP	HOLD, ASK, CODED new options
KEEPQ	CLOSE	new name for this form

LIST	LIST	no translate option; several new options
	PRINT	new command to list file other than the workfile
MODIFY	MODIFY	double slash does not terminate; S and DE subcommands are new
NOT/OR		used only with WHILE; @command constructs replace WHILE
PROCEDURE	PROCEDURE	abbreviation is PROC
Q	Q	
REPLACE	REPLACE	use of HOLD option assumes NOW option
SET	SET	many parameters added and several are deleted
TEXT	TEXT	handles CRUNCHED files, but not variable record lengths; CODED option added
USE	USE	allows conditional or unconditional branching
VERIFY	VERIFY	different options
WHILE		replaced by @command constructs
XPLAIN	HELP	
	PROMPT	puts user in "menu" mode
Z: :=	Z: :=	
	ZP: :=	new form
:		many MPE commands are recognized without the use of leading colon

EDIT/3000 USE files can be easily converted for use with TDP by altering calls to the FIND and GATHER commands and by using the @constructs to replace use of the WHILE command.

MPE commands and subsystems

MPE commands and subsystems allowed while running TDP

Many MPE commands are recognized by TDP with or without a leading colon. Where a leading colon is supplied the whole command line is passed to MPE for execution. In this way, TDP will accept any MPE command and try to execute it programatically. Where there is no leading colon the operation depends on which MPE command is to be executed.

The following list consists of those MPE commands that TDP will accept without a leading colon and all the subsystems currently recognized by TDP. **NOTE: The commands available to you depend on the version of MPE you have installed on your system.**

ALTACCT (1)
ALTGROUP (1)
ALTLOG (1)
ALTSEC (1)
ALTUSER (1)
BASIC (4)
BUILD (1)
COBOL (2)
COBOLGO (2)
COBOLPREP (2)
DSG (4)
EDITOR (4)
EZCHART (4)
FCOPY (4)
FILE (1)
FORMSPEC (4)
FORTGO(2)
FORTPREP (2)
FORTRAN (2)
HP(xxxx) (4)
IDSCHAR (4)
IDSFORM (4)
IFS (4)
LISTDIR (4)
LISTEQ (4)
LISTF
PASCAL (2)
PASCALGO (2)
PASCALPREP (2)
PURGE
RECALL
REFORMAT (4)
REFSPEC (4)
RELEASE
RELLOG
REMOTE
RENAME
REPORT
RESET
RESTORE (1)
RPG (2)
RPGGO (2)
RPGPREP (2)
RUN (3)
SAVE
SECURE
SEGMENTER (4)
SETJCW
SETMSG
SHOWALLOW
SHOWDEV
SHOWIN
SHOWJCW
SHOWJOB
SHOWME
SHOWOUT
SHOWTIME
SPL (2)
SPLGO (2)
SPLPREP(2)
SPOOK (4)
STORE (1)
STREAM
TDP (4)
TELL (1)
TELLOP (1)
VISICALC (4)

Notes:

1 The semicolon " ; " is used in TDP to separate commands. Some MPE commands (identified above as referring to this note) use the semicolon to separate parameters. In order to eliminate any ambiguity, when TDP encounters these commands it assumes that the rest of the line is part of the MPE command. Therefore, such commands must either be on a line by themselves, or appear as the last command on a command line.

2 See the section **Using compilers from TDP** below.

3 See the section **Running programs from TDP** below.

4 See the section **Running subsystems from TDP** below.

Using compilers from TDP

The following compilers are accepted by TDP:

SPL,FORTRAN,COBOL,COBOLII,PASCAL and RPG.

TDP compiler commands differ from the equivalent commands within the system. Primarily this is to allow a freer syntax and to allow the inclusion of information at any stage of the process.

N.B. In the description that follows "comp" stands for any of the above compilers.

The following compiler commands are accepted by TDP:

```
comp[Q]      [filename] [(parameters)]
compPREP[Q] [filename] [(parameters)]
compGO[Q]    [filename][,entryname] [(parameters)]
```

If the filename is not supplied then the current workfile is assumed. If the workfile is not 'clean' then a 'keep' will be done, and the user is prompted to purge any old copy. (If a 'keep' is not allowed, then there is no further processing of the command). If a filename is specified, it will be passed to the compiler as the source filename.

The entryname is separated from the source filename (if supplied) by a comma. The comma must immediately follow the source filename, but leading blanks on the entryname are skipped. If no source filename is supplied then the entryname follows a comma following the command, any blanks between the command and the comma are skipped, as are leading blanks on the entryname.

Parameters are delimited by the following:

$ = , ;

Other special characters are considered as delimiters for a string or a number.

The following parameters are accepted for the comp command:

USL [=] filename — Indicates the filename into which the object program is written. If not specified the object program is written to $NEWPASS.

list [=] filename — Indicates the filename into which the program listing is written. If not specified the program listing is written to $STDLIST, unless overridden by commands below.

MASTER [=] filename — Indicates the filename which is merged with the source file to produce composite source used as input by the compiler.

NOLIST — Indicates that no program listing is to be produced, unless overridden by the LIST parameter.

NOTEXT — Same as NOLIST

OFF or OFFLINE — Indicates that the program listing is to be sent to device class LP, unless overridden by the LIST parameter.

LP — Same as OFFLINE.

The following parameters are accepted for the compPREP command:

All those parameters accepted by comp plus:

PROG [=] filename — Indicates the filename into which the program file is written. If not specified the program file will be written to $OLDPASS. If no USL parameter was entered then the program file will overwrite the object program.

MAXDATA [=] n — Indicates, in words, the maximum stack area size permitted (Z-DL).

STACK [=] n — Indicates, in words, the initial local data area on the stack (Z-Q).

DL [=] n — Indicates the initial DL-DB area to be assigned on the stack.

ZERODB — Indicates that the area between DL and Q be initialized to zero.

CAP [=] — Included for completeness but not required.

- IA — The program can be run interactively.
- BA — The program can be run in batch mode.
- PH — The program requires process handling capabilities.
- DS — The program uses extra data segments.
- MR — The program uses multiple RINs.
- PM — The program requires privilege mode.

RL [=] filename — Indicates the filename to be searched to satisfy external references during preparation.

PMAP

Indicates that a descriptive listing of the program is to be output. If OFF, OFFLINE, or LP was specified then the listing is sent to class LP, otherwise it is sent to $STDLIST.

The following parameters are accepted for the compGO command:

All those parameters accepted by compPREP plus:

LIB [=] P/G/S

Indicates the order of searching segmented procedure libraries to satisfy remaining undefined externals.

NOPRIV

Place all segments in non-privilege mode.

NOCB

The file system is not to use stack segment (PCBX) for its control blocks.

PARM [=] n

Value to be passed to the program.

LMAP

Indicates that a descriptive listing of the allocated program is to be output. This is sent to the same destination as any PMAP information that may have been output.

INFO [=] string

String to be passed to the program.

STDLIST [=] string

Indicates the filename of the $STDLIST file used by this program.

STDIN [=] string

Indicates the filename of the $STDIN file used by this program.

string

String to be passed to the program as 'mail'.

Running programs from TDP

The format of the RUN command in TDP is:

```
RUN filename[,entryname] [(parameters)]
```

The entryname, if supplied, must follow a comma which immediately follows the name of the file to execute. If no group or account is supplied for the program filename and the file is not found locally, then TDP will search in PUB.SYS and then UTIL.SYS for the file.

The parameters that are accepted on the run command are:

MAXDATA [=] n	Indicates the maximum stack area (Z-DL) size permitted, in words.
STACK [=] n	Indicates the initial local data area (Z-Q) on the stack, in words.
DL [=] n	Indicates the initial DL-DB area to be assigned on the stack.
LIB [=] P/G/S	Indicates the order of searching segmented procedure libraries to satisfy remaining undefined externals.
NOPRIV	Place all segments in non-privilege mode.
NOCB	The file system is not to use stack segment (PCBX) for its control blocks.
PARM [=] n	Value to be passed to the program.
LMAP	Indicates that a descriptive listing of the allocated program is to be output. This is sent to $STDLIST.
INFO [=] string	String to be passed to the program.
STDLIST [=] string	Indicates the filename of the $STDLIST file used by this program.
STDIN [=] string	Indicates the filename of the $STDIN file used by this program.
string	String to be passed to the program as 'mail'.

Running subsystems from TDP

It is possible to run various programs and subsystems simply by using the program or subsystem name as a command. TDP will execute the relevant program in PUB.SYS. If any of the subsystems or programs require entrynames, then the entryname must follow a comma immediately after the command name. Any other parameters can be passed to the subsystem or program in the same manner as running a program (see the section on **Running programs from TDP** in this chapter).

Note that any program prefixed by "HP" and residing in PUB.SYS can be run in this way; e.g. to run HPSLATE.PUB.SYS simply enter the command HPSLATE.

Formatted output from TDP

Formatted output

Formatted output is obtained through the FINAL and DRAFT commands. The destination of the output, if not predefined through the SET TERMINAL command, can be identified through the TO option. The filename following the TO option must be preceded by an asterisk (*) to identify a particular printer or terminal. If an asterisk is not given, the output is formatted for a standard CRT and is placed in a disc file of the specified name.

The following Hewlett-Packard output devices are supported by TDP A.05.00:

Device	Identifier	Output filename	Device Class
HP 150 terminal	HP150	STERM	
HP 262x terminals	HP262x	STERM	
HP 263x printers	HP263x	STERM	
	LP263x	SLP	LP
HP 264x terminals	HP264x	STERM	
HP 2382 terminal	HP2382	STERM	
HP 2392A terminal	HP2392A	STERM	
HP 239x terminals	HP239x	STERM	
HP 2601 printer	HP2601/SF	STERM	DAISEY
HP 2687 printer	HP2687	STERM	LP2687
	LP2687	SLP	LP2687
HP 2680 printer	HP2680	SLP	PP
	LASER	SLP	LASER
	PP	SLP	PP
HP 2688 printer	HP2688	SLP	PP88
	PP88	SLP	PP88
HP 700/9x terminals	HP700/9x	STERM	
Standard line printer	LP	SLP	
Standard printer/terminal	TERM/SF	STERM	LP
Standard CRT	CRT	STERM	

The "identifier" is the output identifier recognized by TDP for the given device; e.g. to send output to the HP 2680A laser printer you can FINAL TO *HP2680 or *LASER or *PP. The "SF" appended to the identifier indicates that the terminal has a sheet feeder.

The "output filename" is the formal file designator used by TDP when sending output to the given device; e.g. when formatting output for a standard line printer, TDP uses the formal file designator SLP. The characteristics of these files may be altered by setting up a file equation for the output filename; e.g. to defer the output to the system line printer you can use the following :FILE command:

```
:FILE SLP;DEV=LP,1
```

When the output identifier is preceded by an asterisk, TDP checks for the existence of a TDP configuration file. If one is found, and the specified name is found in it, the data in the configuration file associated with the given name is used for output. If the name is not in the configuration file, TDP determines whether or not the file is defined through an MPE :FILE command. If not, and the identifier is not one of those recognized by TDP, an error is generated.

Spooler printer support

With the exception of the HP 2680 and HP 2688 laser printers, all of the printers listed above are supported by the TDP spooler. However, you cannot MPE spool and TDP spool to the same device.

Offline list file

When the OFFLINE parameter is used with the LIST or PRINT command, output goes to a file with formal file designator LP of device class LP. If your system doesn't have a device class LP, the message "CAN'T OPEN OUTPUT FILE" will be displayed. In this case, you must use a file equation to specify the device class to use; e.g. :FILE LP;DEV=PP.

Using laser printers

This appendix provides an introduction to using TDP with the HP 2680 and HP 2688 Laser Printing System (LPS). The appendix describes briefly the major features of LPS and provides basic guidance on how to use laser printers with TDP.

What is the Laser Printing System?

The Laser Printing System (LPS) consists of an HP 2680A or HP 2688A Page Printer and two interactive application programs, IDS/3000 and IFS/3000. The printers use a laser based electro-photographic process to print up to 45 pages per minute (in the case of the HP 2680) or 12 pages per minute (in the case of the HP 2688) and use 11 by 8.5 inch paper (or A4 paper). The Laser Printing System can make use of up to 32 software designed character sets and can merge standard pre-printed forms with the text to form the complete output. The two programs, IDS/3000 and IFS/3000, allow interactive development of the character sets and the pre-printed forms, and the ability to merge them, thus forming an operating environment.

The benefits and features of laser printing

The main advantage offered by laser printing is the high print quality at a high printing speed. This is enhanced by the many different character sizes, styles and orientations; the inclusion of pre-printed forms and logos when required; and the different sizes and numbers of pages, and print directions that can be included on one physical page. An added advantage offered by the Laser Printing System, is that the descriptions of the characters, forms and pages, are held together in one file called the Environment file. In addition, the Laser Printing System allows text and graphics to be merged in single pass runs. (Graphics may be produced using the Gallery Graphics, HP EZCHART, HP DRAW or DSG/3000 software packages.)

Used with the power and flexibility of TDP, the Laser Printing System forms the basis of a professional documentation system for in-house publishing departments. For example, this manual was written, edited and formatted using TDP, illustrations were prepared using HP DRAW and camera ready page masters were set on an HP 2688.

Print quality

The print quality of individual characters comes from the laser printing process, which prints at 180 dots per inch in both directions (in the case of the HP 2680) or 300 dots per inch (in the case of the HP 2688).This leads to clear, precise characters and excellent reproduction of graphics.

Multiple character sets

The characters themselves vary in size from 0.05 to 1.38 inches per character; can be oriented horizontally or vertically, relative to the page; and can be individually designed. Within the LPS up to 32 character sets, or fonts, may be defined. Each font is uniquely named and numbered. The name contains up to 16 characters, and the number is in the range 0 to 31. The name, number and orientation of each font, along with each character's size and dot map is held in the environment file.

Pre-printed forms

The inclusion of pre-printed forms or logos, along with the ability to turn them off, means that a first page and continuation pages can be printed as a single document at the maximum print speed.

Variable page layout

The ability to have different sizes and numbers of **logical pages** on one **physical page**, subject to character size, means that more than one page of normal output may be printed on one physical page. Thus the archive copy of a report can be printed at four pages per physical page, while the customer copy is printed on separate physical pages.

This is further enhanced by the ability to define the orientation of the print on the page. To distinguish the direction of printing the terms landscape and portrait are used. **Landscape** mode considers the page to be wider than long, while **portrait** mode considers the page to be longer than wide. Thus horizontally and vertically orientated pages can be used within the same document, without losing the ability to print more than one page per physical page. For example, a projected financial summary may be printed in the middle of a report: the report being printed in portrait mode and the summary in landscape mode.

The LPS holds up to 32 logical page definitions in the environment file. Each logical page definition is uniquely numbered, in the range 0 to 31, and contains its size and position on the physical page, together with the print orientation. It is possible for logical page definitions to overlay each other, either partially or completely. It is also possible for a sub-set of these page definitions to be in use at any time. The LPS will only use the definition of **active** logical pages. Pages can be **activated** or **deactivated** at any time.

At the beginning of a physical page, the lowest numbered active logical page is selected. When the end of that page is reached, the LPS will select the next highest numbered active logical page definition. If none other exists, a physical page eject occurs. This leaves a high degree of flexibility in the selection and formatting of the text.

The environment file

The environment file holds the description of the character sets, pre-printed forms and the logical page layouts. No extra description of the characters sets, forms or the pages is required. This means that those users who have a constant page layout and only make use of the normal and alternate character sets, need only identify the environment file. It is only if you wish to change the page layout or character set that you need to do anything. For printing to the HP 2680 or the HP 2688 an environment must be specified. (If you don't, TDP will use its default environment).

The command used to specify the environment file to be used for a document is the \ENVIRONMENT command.

An environment file is specific to one device, for example you cannot use an HP 2688 environment for printing to the HP 2680.

IFS

An environment file is created using IFS. Using this product you can:

1 Define the size and other characteristics of the physical page

2 Define up to 32 logical pages, including attaching forms

3 Define the fonts to be included

4 Compile the environment for use with other products, eg TDP, HP WORD, etc

Base Font

Like most text processors, TDP works in lines and characters. For example, commands like LFT, RHT, INLFT, INRHT, and COLUMN define the space between the text and the edge or margin in terms of characters. That is, the commands use a character as the unit of horizontal measurement. Similarly, the TOP, BOTTOM, PAGELENGTH, and SPACE commands use a linespace as the unit of measurement for the vertical.

On line printers and letter quality mechanical printers this is a simple and effective method of controlling the format of printed output. However, to deal effectively with the many different font sizes and finer resolution available with laser printers, TDP must convert its basic units of measure, the character width and linespace, to units of measure appropriate to each document. To do this it identifies a **base font**. Because the base font is used by TDP for calculating columns, etc, it is important that you know which base font is in use. TDP determines the base font in the following sequence:

1 The base character set defined in the Physical Page Dimensions or, if none:

2 The character set defined in the first active logical page (usually LP0) or, if none:

3 Character set 0

(Probably the best method is to use character set 0 as the base font. This way you can tell what your base font is by using the \VERIFY ENVIRONMENT command, see later, instead of having to use IFS to find out.) Once TDP has determined the base font it uses that font to answer the following basic questions. They are:

1 How wide is a character/column?

2 How long is a line and how many lines are there on a page?

3 What is the default font?

The size of the character/column is the width of the characters of the base font. In general, the width of a character is specified in dots. For non-proportional fonts, this will be the same for all characters in the character set.

However, since with proportional fonts, some characters are wider than others, the width of the font is defined as the width of a lower case letter m. This unit is called an "em" (and is a convention borrowed from typesetting).

Therefore, columns, etc, are calculated in em's of the proportional base font.

\VERIFY ENVIRONMENT

You can find out the base font for your document (if it is character set 0) and all the fonts used, by using the \VERIFY ENVIRONMENT command. The command is placed in the workfile immediately after the \ENVIRONMENT command. The output will be displayed on the screen when your document is formatted. A typical display is shown below:

```
TDP/V    (A.05.00) HP36578 Formatter (c) COPYRIGHT Hewlett-Packard Co. 1988
OUTPUT MODE = FINAL
INPUT = EDITOR WORKFILE, TEXT FROM NOTES
OUTPUT = *HP2688
  Environment file    TDPDOCEN
  Current EM-Width     25
  Current Line Height  50

PAGE  ACTIVE  HEIGHT WIDTH  PRINT  DIR
  0     YES      66    90    YES     0
 31      NO      46   105    YES     0

FONT   PROP?  HEIGHT WIDTH   EMS.  DIR     NAME
  0      NO      50    25     25     0 ELIT88
  1      NO      35    20     20     0 LINEPR8
  2     YES      33    37     28     0 ROM8
  3     YES      33    41     29     0 ROMB8
  4     YES      33    43     27     0 ROMI8
  5     YES      42    46     36     0 ROM10
  6     YES      42    55     34     0 ROMB10
  7     YES      42    55     34     0 ROMI10
  8     YES      33    37     27     0 GEN8
  9     YES      42    48     38     0 GEN10
 10     YES      42    48     34     0 GENB10
 11     YES      50    56     43     0 GENB12
 12     YES      67    75     58     0 GENB16
 13     YES      75    81     60     0 GENB18
 14     YES     100   113     84     0 GENB24
 15      NO      65   255    255     0 HPLOGO
```

Using different character sets

A TDP document can be printed using a variety of different fonts, or just a single font, as required. You can have up to 32 different fonts in any one document.

These fonts are declared in the environment file, which must be specified at the start of the TDP file (see later).

There are several different ways of changing fonts within TDP. The basic methods are discussed here; for further details, see Chapter 4.

The \FONT command

A document will be printed in its base font unless you put in a command to change fonts. The simplest of these commands is the \FONT command.

```
\FONT "elite12" 15
```

This will cause the next 15 output lines to be printed in the font "elite12", provided that "elite12" is one of the fonts defined in the environment file being used.

\FONTID and ^F....^S

If you want to change font without working out how many lines to print in the new font, you can use the \FONTID command to allocate a variable name (one letter or number) to represent the font, for example:

```
\FONTID I "romital"
```

Now you can use ^F followed by the variable name to access the font. To return to the base font use ^S, for example:

```
In this line ^FI these three words^S are in "romital"
```

The output will look like this:

In this line *these three words* are in "romital"

Note: the variable name assigned by the \FONTID command can also be used in the \FONT command, for example:

```
\FONT I 12
```

This will print the next 12 lines in the font indicated by "I", ie. in this case, "romital".

Bold and Ghost - The \FONTEQ command

The \BOLD and \GHOST commands (also ^B and ^G), can also be used to change fonts. To do this define the font to be used for bold/ghost text with the \FONTEQ command, for example:

```
\FONTEQ BOLD "helvbold14"
\FONTEQ GHOST "romital"
```

Now, whenever the \BOLD command is used, text will be printed in "helvbold14". Similarly, whenever \GHOST is used, "romital" will be printed.

The ^B. . .^S and ^G. . .^S commands also work, for example:

```
This is ^Bbold^S and this is ^Gghost^S.
```

This will produce:

This is **bold** and this is *ghost*.

Note: the font defined as bold doesn't have to be a bold character set. It can be any font in the environment file. In the example above, ^G is a convenient way of accessing an italic font.

Notes on choosing fonts

The fonts available for the laser printers are shown in **Appendix D of the IFS Reference Manual**. A font is either proportional or non-proportional.

Proportional fonts

In a proportional font, each character may be a different width, eg "m" is wider than "i". In a non-proportional font, all characters take up the same width.

For example, the first line below is in a non-proportional font, and the second is in a proportional font:

Here is some text
Here is some text

Problems arise when using a proportional font to print tables, or aligned text in \IMAGE mode. This can only be overcome by changing to a non-proportional font.

Base Font - \INFORMAT and COLUMN

Because font sizes may vary within a document, commands which calculate distances in terms of numbers of characters, such as \INFORMAT and \COLUMN, have to use a standard character size. As described earlier, this standard is derived from the base font, and will remain constant for the entire document, although it may be different for other documents.

The base font is defined in the environment file and cannot be changed using TDP commands. The result of calculating column sizes in terms of base font characters can be seen below:

Assume the base font is 10 point (ie approximately 12 chars/inch) and the current font is 14 point (ie approximately 8 chars/inch).

If you specify a column to be 24 characters wide, this will be calculated as about 2 inches (ie 24 base font characters).

But you are printing in a much larger font. The number of large characters (14 point) that can fit in 2 inches is about 16. Therefore the maximum number of characters that will fit in the column is about 16.

This is not necessarily a problem. It is, however, something of which you should be aware.

Linespacing considerations

If a font larger than the base font is being used, you may have to adjust line spacing to avoid overprinting. In order to do this, use the \LINESPACE command.

Creating your own font

It is possible to create your own character set or logo using IDSCHAR. These characters can then be included in the environment file and used for TDP output.

Accessing eight bit character sets

To access 8-bit characters, specify a pair of fonts instead of a single font, for example:

```
\FONT  ("pica","picax")  5
```

Where "pica" is a standard font and "picax" is its Roman extension. Whenever there is an 8-bit character in the document, it will be printed in the alternate, extended font, ie."picax".

ISO substitution characters

To access the ISO substitution characters for a particular language, use the appropriate ISO font. For example, use PICAF for French text.

Now if you have an "@" sign in your document, it will print as an "a" with a grave accent, a back slash "\" will print as a "c" circumflex, etc.

See the **IFS manual, Appendix D**, for more details. Note that ISO substitution character sets are not available on the HP 2688.

Physical and logical pages

Two types of page are defined in the environment file.

The **physical page** is the actual piece of paper to be printed on. The **logical page** is the area on the physical page which TDP prints to. There may be several (up to 32) logical pages on any physical page - TDP prints to each logical page in turn, and then does a physical page throw. (It is possible to miss out a logical page if needed).

A logical page may be either **landscape** or **portrait**.

Landscape is wider than long:

```
=============================
= ......................... =
= ......................... =
= ......................... =
= ......................... =
= ......................... =
=============================
```

Portrait is longer than wide:

```
==========
= ...... =
= ...... =
= ...... =
= ...... =
= ...... =
= ...... =
= ...... =
==========
```

Logical pages can be either **active** or **non active**.

A logical page will be printed on if it is active and ignored if it is not active.

The initial states of the logical pages are defined in the environment file. Normally you wouldn't use all 32 logical pages, as multiple active logical pages can cause overprinting on the same physical page. A "page" throw becomes an indication to move to the next active logical page. Movement from one logical page to another is in sequence from LP 0 to LP 31.

The text is written to the current active logical page and printed on the current physical page. On a page throw, the next active logical page is written to and the process is repeated until there are no more active logical pages. Only when there are no more active logical pages will a physical page throw occur. Printing then begins on the lowest numbered active logical page and moves up the logical pages up to LP 31 once again. For example, suppose an environment file had 3 active logical pages 0,1,7.

Logical page	Active	
0	YES	<-- LP POINTER
1	YES	
2	NO	
3	NO	
4	NO	
5	NO	
6	NO	
7	YES	
8	NO	
.		
.		

The text for the first page will be written on logical page 0 and on the current physical page. When a page throw occurs, then the Logical Page Table is searched for the next active logical page; in this case it is LP 1 and the following text is written to LP 1 and printed on the current physical page until the next page throw. Once again the Logical Page Table is searched for the next active logical page, which is LP 7. This logical page is used until the next page throw, when no more logical pages are active, hence a physical-page throw occurs and printing begins on the lowest active logical page, in this case LP 0.

Using logical pages in TDP

Most often you will only want 1 logical page printed on each physical page. Therefore, you need only 1 logical page active at a time.

To switch between the logical pages (with a physical page throw every time) use the \LAYOUT NEW command, for example:

```
\LAYOUT  NEW  1
```

This will cause a physical page eject, and start printing on logical page 1.

Forms

The most common reason for defining more than 1 logical page is if you are using forms.

You create your form using IDSFORM and attach it to a particular logical page when creating the environment file. For example, you might create 2 logical pages (LP0 and LP1) and attach a form to 1 of them (LP1). Whenever LP1 is printed on, the form will be printed too. Whenever LP0 is used, there will be no form. Thus:

```
\LAYOUT  NEW  1
```

would start a new page and printing would be on the form.

```
\LAYOUT  NEW  0
```

would start a new blank page.

Overlapping logical pages

If you want to print more than one logical page on a physical page, use the \ACTIVATE and \DEACTIVATE commands. For example:

```
\ACTIVATE  3 , 1 , 4
```

This will make logical pages 1, 3 and 4 active. Therefore, LP1 will be printed, then LP3, then LP4, and then a physical page eject will occur. Printing will resume on LP1, etc.

If you only want logical pages 1 and 4 active for the next page, then use:

```
\DEACTIVATE  3
```

Note: there must be at least 1 active logical page at all times, therefore always **ACTIVATE before DEACTIVATE.**

Text and graphics

Graphics to be incorporated into a TDP document can be produced using one of the following software packages: Gallery Graphics, HP EZCHART, HP DRAW or DSG/3000 II.

The \ILLUSTRATION Command

The general form is:

```
\ILLUSTRATION filename:figure n,options
```

The \ILLUSTRATION command will print a picture or graph as part of a document. (Unlike the \FIGURE or \BOX commands which merely leave space for pictures to be stuck on).

The illustration must be in either a figure file (eg created in HP DRAW using "Save Figure" option) or a raster file (eg produced using the LPS interpreter or the \NAME RASTER command in TDP, which will be discussed later). If the illustration is in a figure file, the filename and the figure name must be specified.

Scaling

An illustration from a raster file cannot have its size or orientation changed.

By default, an illustration from a figure file will be scaled to fit between the current margins, and it will be the same orientation as the current logical page. It is possible to override the defaults by supplying the number of lines to contain the figure, for example:

```
\ILLUSTRATION file:figure 10
```

The illustration will occupy 10 lines. You can also specify whether the illustration is to be rotated, and its position on the page, eg centered, at left margin, etc (see Chapter 4).

The illustration will be printed on the current page if there is room, and on the next page if there is not.

When outputting to a printer other than an HP 2680 or HP 2688 laser, the space which would have been occupied by the illustration will be occupied by a box of the same size.

Figure to raster conversion

Before a figure file can be printed it must be converted to a RASTER image (in dots) of the figure, of the final size and orientation. TDP will normally do this automatically, and put the raster in a temporary file. This process is time-consuming, and the more complicated the figure the longer it will take.

Saving the raster file

If you are going to print the same document several times, it would make sense to save this temporary raster file for use next time. TDP provides this facility with the \NAME RASTER command (see Chapter 4).

Commands relevant to laser printing

The following list shows the TDP commands that are especially relevant to laser printing. They are all described in Chapter 4.

ACTIVATE
DEACTIVATE
LAYOUT
ENVIRONMENT
FONT
FONTEQ
FONTID
^F
ILLUSTRATION
NAME
RASTER

Default environments

If no environment file is defined then TDP will attempt to select an appropriate environment file.

The defaults for the HP 2680 are as follows:

10 pitch, and the text width plus the left margin is less than 75 characters:
PICA.HPENV.SYS

12 pitch, and the text width plus the left margin is less than 90 characters:
ELITE.HPENV.SYS

In all other cases: LP.HPENV.SYS

The defaults for the HP 2688 are as follows:

10 pitch, and the text width plus the left margin is less than 75 characters:
COUR88A.HPENV.SYS

12 pitch, and the text width plus the left margin is less than 90 characters:
GOTH88A.HPENV.SYS

In all other cases: LP88.HPENV.SYS

Formatting hints

This appendix provides some brief notes on those features of TDP that may cause formatting problems for the new user.

Automatic paragraphs

TDP will automatically conform to the pattern of paragraph indentation you set in the first two lines of each paragraph. The first line is always indented the number of spaces you indicated when entering it. This is true of the second line as well. However, the indentation of the second line also indicates the indentation of the rest of the lines in the file. For example:

```
        A paragraph that looks like
this in the workfile will be
    formatted by TDP to automatically indent the
            first line the same distance as the first
paragraph line in the workfile; all
     subsequent lines will be in line with the second line
  of the paragraph.
```

        A paragraph that looks like this in the workfile will be formatted by TDP to automatically indent the first line the same distance as the first paragraph line in the workfile; all subsequent lines will be in line with the second line of the paragraph.

This is only valid for FORMAT mode.

Stepped paragraphs

In FORMAT mode, a paragraph of one input line with two spaces (though not necessary blanks) between the first and second tokens will be formatted as a stepped paragraph.

This means that if the line has to be wrapped, it will be indented level with the second token.

If you had entered:

```
1  Example   this is an example of stepped paragraph indentation.
```

With LFT=1 and RHT=40, you would get:

```
Example  this is an example of stepped
         paragraph indentation.
```

Labeled paragraphs

A line containing only special characters will be treated as being in IMAGE mode. A labeled paragraph takes this one step further.

A line followed by line of special characters constitutes a label for the following paragraph.

So, for example, if you enter:

```
       This is a Heading
       *****************
This is the main body of the
paragraph. It will be aligned with the
label.
```

It will format as:

This is a Heading
****
This is the main body of the paragraph. It will be aligned with the label.

Widow Watch

With "WIDOW ON" (the default) TDP will avoid putting the first line of a paragraph at the bottom of a page and the rest on the next page and, conversely, avoid putting most of a paragraph on one page and the last line on the next page.

This will sometimes have the effect of making multi-column work, for example, appear unbalanced. To overcome this, turn Widow Watch off using \WIDOW OFF.

Resetting formatting commands

A very common cause of formatting problems when first using TDP is caused by the failure to reset or turn off formatting commands. For example, failing to add an INFORMAT 0 after using the INFORMAT command; failure to reset FORMAT mode after processing a part of the workfile in IMAGE mode; or failing to reset temporary left and right margins with INLFT 0 or INRHT 0 are all common occurrences.

If formatting is producing unexpected results, look at the last place in the textfile you used a formatting command before the problem occurs and then check further on in the file to see if the command was reset.

LFT and RHT values

When FINALing to an HP 2680 or HP 2688 laser printer, you may get the message **No page big enough for output**.

This is probably because the value of RHT is greater than the number of characters TDP thinks it can get on a line.

Do a \VERIFY ENVIRONMENT at the start of your document (just after the \ENVIRONMENT command). Set LFT and RHT less than the width of the logical page (shown on the screen during FINALing).

Envlronment files for different printers

An environment file is specific to a particular device type, eg. an environment file for the HP 2688 will not work properly on an HP 2680.

If you use an HP 2688 environment file for FINALing to an HP 2680, your output will be twice the size expected and printed sideways. Text will probably be lost on the printed document.

If you use an HP 2680 environment file on a HP 2688, your output will be half the size expected and, again, printed sideways. This may be useful for printing draft documents.

\ENVIRONMENT command ignored

If you have any output lines before the \ENVIRONMENT command, even a blank line or column, the default environment file will be used. This is because TDP must open an environment file before output can start, and blank lines count as the start of output.

The effect of this is that any font commands you may have, such as \FONTID, \FONTEQ, etc. will produce error messages because these fonts are not defined in the default environment.

INLFT and INDENT - the difference

At first glance you may think that \INDENT 5 and \INLFT 5 will produce the same result. They may not, however, if you are using an HP 2680 or an HP 2688 laser printer.

The command \INDENT 5 will print text 5 spaces in from the left margin. This means 5 spaces of the right size for the font you are printing in at the time the command is obeyed. (Remember that if you are printing in a proportional font, then a space will probably be smaller than other characters typically this will occupy a 1/3 of "M".; so 5 spaces may occupy as much room as, say, 3 letters such as "a".)

The command \INLFT 5 will print text 5 base font character spaces in from the left margin; **Not necessarily the font being used at the time the command is obeyed.** (Refer to Appendix G for an explanation of base fonts.)

So if you have a small base font, but are currently printing in a large (non-proportional) font, \INLFT 5 will move the text in by 5 small character spaces, and \INDENT 5 will move the text in by 5 large character spaces.

Characters lost on page

If you get the message CHARACTERS LOST ON PAGE when using proportional fonts, it may be necessary to reduce the value of the right margin. You should use the \VERIFY ENVIRONMENT command to determine the width of the logical page, and set the right margin (\RHT command) to three less than this value.

Adding Blank Lines to Columns

If you need to add blank lines to the end of a column, you should include the required number of blank lines in your workfile, rather than use a \SPACE command.

Numbered and Unnumbered Files

If a file is created with numbers in record three, then TDP assumes the file is numbered. If you save a file with numbers in this record as UNnumbered, then TEXT it back in, it is treated as numbered and the last eight bytes of the first line are not displayed.

Printing Drawing Gallery Illustrations

You can include illustrations produced with Drawing Gallery on your personal computer in a TDP document. You will need to move your Drawing Gallery figure on to your MPE system. There are two ways you can do this:

1 If you have AdvanceMail (A.02.00) or later on your personal computer and HP Desk on your HP 3000, you can specify .GAL as the file type and send you picture to HP Desk in a message. Copy the picture from HP Desk to MPE, and then include it in your document with the \ILLUSTRATION command.

2 If you have AdvanceLink then you can use it to transfer Drawing Gallery file. For example, suppose you are transferring your Drawing Gallery figure into an MPE file called picture. Set up a file equation on MPE:

```
FILE picture;CODE=1422;REC=64
```

Run AdvanceLink and use File Transfer to copy your local Drawing Gallery file to *picture. You can use \ILLUSTRATION to include picture in a TDP document.

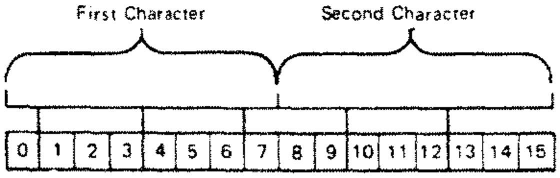
If you need any further information about using AdvanceLink or AdvanceMail, look at the manuals for these two products.

Table of ASCII character codes

The table of ASCII character codes is shown on the following page.

ASCII Character	First Character Octal Equivalent	Second Character Octal Equivalent
A	040400	000101
B	041000	000102
C	041400	000103
D	042000	000104
E	042400	000105
F	043000	000106
G	043400	000107
H	044000	000110
I	044400	000111
J	045000	000112
K	045400	000113
L	046000	000114
M	046400	000115
N	047000	000116
O	047400	000117
P	050000	000120
Q	050400	000121
R	051000	000122
S	051400	000123
T	052000	000124
U	052400	000125
V	053000	000126
W	053400	000127
X	054000	000130
Y	054400	000131
Z	055000	000132
a	060400	000141
b	061000	000142
c	061400	000143
d	062000	000144
e	062400	000145
f	063000	000146
g	063400	000147
h	064000	000150
i	064400	000151
j	065000	000152
k	065400	000153
l	066000	000154
m	066400	000155
n	067000	000156
o	067400	000157
p	070000	000160
q	070400	000161
r	071000	000162
s	071400	000163
t	072000	000164
u	072400	000165
v	073000	000166
w	073400	000167
x	074000	000170
y	074400	000171
z	075000	000172
0	030000	000060
1	030400	000061
2	031000	000062
3	031400	000063
4	032000	000064
5	032400	000065
6	033000	000066
7	033400	000067
8	034000	000070
9	034400	000071
NUL	000000	000000
SOH	000400	000001
STX	001000	000002
ETX	001400	000003
EOT	002000	000004
ENQ	002400	000005

ASCII Character	First Character Octal Equivalent	Second Character Octal Equivalent
ACK	003000	000006
BEL	003400	000007
BS	004000	000010
HT	004400	000011
LF	005000	000012
VT	005400	000013
FF	006000	000014
CR	006400	000015
SO	007000	000016
SI	007400	000017
DLE	010000	000020
DC1	010400	000021
DC2	011000	000022
DC3	011400	000023
DC4	012000	000024
NAK	012400	000025
SYN	013000	000026
ETB	013400	000027
CAN	014000	000030
EM	014400	000031
SUB	015000	000032
ESC	015400	000033
FS	016000	000034
GS	016400	000035
RS	017000	000036
US	017400	000037
SPACE	020000	000040
!	020400	000041
"	021000	000042
#	021400	000043
$	022000	000044
%	022400	000045
&	023000	000046
'	023400	000047
(	024000	000050
)	024400	000051
*	025000	000052
+	025400	000053
,	026000	000054
-	026400	000055
	027000	000056
/	027400	000057
	035000	000072
;	035400	000073
<	036000	000074
=	036400	000075
>	037000	000076
?	037400	000077
@	040000	000100
[	055400	000133
\	056000	000134
]	056400	000135
Δ	057000	000136
—	057400	000137
`	060000	000140
{	075400	000173
|	076000	000174
}	076400	000175
~	077000	000176
DEL	077400	000177



ASCII character codes

Index