THE HP TEX MACROS

HEWLETT-PACKARD CO.

Susan Daniels

1 Introduction

This set of macros was written as part of HP TEX, Hewlett-Packard's software package designed to facilitate the use of TEX on the HP Series 200 Desktop Computers. The HP TEX program can be run on either an SRM based, or a stand-alone (LIF) system, and requires 1.25 Mbytes main memory and a minimum of 15 Mbytes mass storage. It uses an HP 2688A Laser Printer.

The HP2688A is a page printer which uses scanning electrophotograph technology to print computer output on single sheet paper. The 300 dots per inch resolution, high contrast and speed, plus sophisticated character printing, allow a high degree of application flexibility. Unique features include 90° rotation of page print, multiple pages of data output per printed sheet, and graphics capability. The print rate is approximately 12 pages per minute.

The HP TEX macro package is designed to supplement and enhance Plain TEX by providing a higher level of functionality and an example of a macro package oriented toward a simple set of user formatting tasks. It also includes macros that simplify the use of the special features of the HP2688A Laser Printer.

The functions selected for HP TEX macros are general formatting functions. They are not oriented toward a particular application, however the needs of manual production were considered in order to obtain a useful set of commands.

The HP TEX macros are intended to be a superset of the Plain TEX macros, and fully compatible with them. Whenever possible, Plain TEX functions are unchanged in HP TEX. In the few instances it was necessary to redefine a Plain function, the new definition remains consistent with the intent of the Plain definition. Hence, the user of HP TEX has access to the Plain TEX macros and can use the *TEXbook* and its exercises and examples.

The HP TEX Macros will soon be available on the Stanford Distribution Tape.

On syntax lines, when curly braces "{...}" are shown, they are required in the source file, text between angle brackets "⟨...⟩" identifies a parameter variable, and the character "␣" represents a required space.

2 Document Formatting Macros

The following HP TEX commands control the basic style and formatting of a document:

2.1 Page Layout

The HP TEX defaults for page layout are the same as they are in the TEX program. That is, all margins are set to approximately one inch from the edges. (Plain TEX sets `\vsize=8.9in` and `\hsize=6.5in`.) Headings and footings are printed **outside** those margins.

The following macros affect the running head and the footing of each page. The heading and footing lists are expanded during the output routine so the TEX commands `\firstmark`, `\topmark` and `\botmark` are compatible here.

`\centerheading`{⟨horizontal list⟩}
`\leftheading`{⟨horizontal list⟩}
`\rightheading`{⟨horizontal list⟩}
`\outsideheading`{⟨horizontal list⟩}
`\insideheading`{⟨horizontal list⟩}
These macros accept text which is to be placed at the top of every page. The first three macros place the argument in the center, left, or right of the page respectively. `\outsideheading` and `\insideheading`, if specified, override the left and right headings. On odd pages, the inside heading will appear on the left, outside on the right. Even pages are the opposite. (This feature is helpful when the output is to be used as two-sided copy.)

`\centerfooting`{⟨horizontal list⟩}
`\leftfooting`{⟨horizontal list⟩}
`\rightfooting`{⟨horizontal list⟩}
`\outsidefooting`{⟨horizontal list⟩}
`\insidefooting`{⟨horizontal list⟩}
These are like the above commands, only they are for footings at the bottom of each page. For example, if page numbers are desired in the left side of the bottom of each page, type: `\leftfooting{\folio}`

`\outsidefooting` and `\insidefooting` will produce similar results as `\outsideheading` and `\insideheading` (see above). The default in HP TEX is `\centerfooting{\folio}`. This produces page numbers at the center of the bottom of each page.

`\noheading`
`\nofooting`
These macros turn off headings or footings for the current and all successive pages until the `\resumefooting` or `\resumeheading` command is used.

`\suspendheading`⟨integer⟩
`\suspendfooting`⟨integer⟩
These macros suspend headings or footings for the specified number of pages or until the use of a `\resumeheading` or `\resumefooting`.

`\resumeheading`
`\resumefooting`
These macros undo the effects of the `\suspendheading`, `\noheading`, `\suspendfooting` and `\nofooting` macros above.

2.2 Paging

\newpage
This macro forces a page eject if not on a new page.[1]

\oddpage
This macro causes a page eject, and if the current page is an odd-numbered page, leaves an extra blank page so that the following text is guaranteed to begin on an odd-numbered page.

\evenpage
This macro is similar to oddpage, but the following text will appear on an even rather than an odd numbered page.

The Plain TEX command, **\pageno** will set the current page number to the specified page number. (See "Some Useful TEX Commands" in Chapter 2 of this manual.)

2.3 Paragraph Style

As TEX accepts text from the input file, the text is formatted into paragraphs. The following commands can be used to control the shape of the paragraphs.

\inset=⟨dimension⟩
Specifies a general amount of indentation to be used with itemized lists, notes, warnings and indent blocks. Default is 0.5 inch.

\start{indent}
\finish{indent}
These commands are used to indent the left margin by the inset dimension. The right margin is unaffected by this block.

\indentspace=⟨dimension⟩
This command assigns the indentation value for the first line of all succeeding paragraphs when **\indentstyle** is in effect. The default is 20 points.

\indentstyle
This causes the first line of each paragraph to be indented (the amount of the indentation is assigned by the **\indentspace** command). This is the default paragraph style.

\noindentstyle
This causes the paragraphs to be formatted with no indentation. The nominal spacing between paragraphs is 5 points greater without indentation than when **\indentstyle** is being used.

– NOTE –

Spacing between paragraphs can be user specified by using the **\parskip** command.

[1] If **\newpage**, **\oddpage**, or **\evenpage** occur at the bottom of a full page, they will cause an extra page eject. If this happens, insert an **\eject** just before the page command.

2.4 Itemized and Bulleted Lists

The following macros are aids in producing lists. The various list commands cause indentation at various levels and use various tokens (numbers, letters, dots, dashes, *etc.*) to the left of the first line of the listed item to set off the list entry.

\numbereditems
\letthereditems
\Letthereditems
\romanitems
\Romanitems
\squareditems
\dotteditems
\dasheditems
These macros initialize the tag allocation macro **\itemtag**. Default is **\numbereditems**.

\numberedsubitems
\letteredsubitems
\Letteredsubitems
\romansubitems
\Romansubitems
\squaredsubitems
\dottedsubitems
\dashedsubitems
These macros initialize the tag allocation macro **\subitemtag**. Default is **\letteredsubitems**.

\itemtag
This macro causes a number, letter, roman numeral, square, dot or dash to be printed as the item specifier depending upon what initialization has taken place (see above).

\subitemtag
Similar to **\itemtag** but pertains to subitems.

\square
This macro prints a .4em-by-.4em square (■). This macro can be used at any time a horizontal list is being built (such as the middle of a paragraph), but it is particularly useful as the argument for the **\itemlist** macro. (This command produces the same symbol as TEX's **\bull** command.)

\dott
This macro prints a solid round dot (•). (Same as Plain TEX's **$\bullet$**.)

\emdash
Sets a horizontal rule one "em" long (—).

\itemlist{⟨horizontal list⟩}
This macro introduces an item of a list by indenting both left and right margins by the ⟨dimension⟩ specified in the last **\inset** command. The ⟨horizontal list⟩ is inserted within braces to specify the token which is to set off the text. If the token has already been specified (using **\dotteditems**, **\Romanitems**, *etc.*), or the default **\numbereditems** is acceptable, then the command **\itemtag** can be inserted here. The text of the item follows. (The text does **not** need to be enclosed within braces.) Indentation continues only for the duration of one paragraph (if more than one paragraph is desired in an item, **\itempar** should be used). The use of **\itemlist** resets the subitem tag.

\subitem{⟨horizontal list⟩}
This macro is similar to \itemlist, but the indentations are twice as large as for \itemlist. This can be used to indicate a second level of list. (See \itemlist for contents of the ⟨horizontal list⟩.

\itempar
\subitempar
These macros are used to start new paragraphs within an item or subitem respectively.

\enditems
This commands properly ends an itemized list by resetting the item counter and appending \bigskip glue.

\itm
Similar to \itemlist{\itemtag} except that a period is appended to the tag if it is a letter, number or roman numeral. With this command, the item tag must be specified ahead of time using the command \dotteditems, \letteredits, *etc.*, unless the default (\numbereditems) is acceptable. Again, \itm will **not** accept any ⟨horizontal list⟩ as an item tag specifier.

\sitm
Similar to \itm but pertains to subitems. Like \itm, \sitm will **not** accept any ⟨horizontal list⟩ as a subitem tag specifier. The subitem tag must be specified beforehand, unless the default (\letteredsubitems) is acceptable.

– NOTE –

If you enter a local block structure prior to setting an itemized list, and want to exit that structure immediately after your last item, then you must first be sure that you are in **vertical mode**. Otherwise, the last item may not be indented properly.

Three ways to accomplish this are by using \enditems, \vskip, or \par.

2.5 Line Specifier Macros

The content of individual lines can be controlled, and they can be centered or justified using the following commands.

\centerline{⟨horizontal list⟩}
This takes the text within the brackets and centers it between the margins. The line does not count as a paragraph, so the normal interparagraph space is not inserted above the line. If this command follows a paragraph, it is usually appropriate to precede it with a \vskip. This command has been redefined from Plain TEX in that it uses the HP macro \lline (described below), so \leftskip and \rightskip values are taken into account.

\lline{⟨horizontal list⟩}
This is very similar to the plain TEX \line command in that it creates a horizontal box that is the width of the current \hsize, except that \lline takes into account the values of \leftskip and \rightskip if they have been specified. In other words \lline stays within the current margins. The ⟨horizontal list⟩ is the contents of the horizontal box.

\leftline{⟨horizontal list⟩}
This left justifies the contents of the ⟨horizontal list⟩. The comments about vertical spacing above apply here, also. This command has also been redefined to use \lline (see \centerline, above).

\rightline{⟨horizontal list⟩}
This right justifies the contents of the ⟨horizontal list⟩. The comments about vertical spacing above apply here, also. This command has also been redefined to use \lline (see \centerline, above).

\raggedright
The \raggedright macro causes paragraphs to be formatted in such a way that they are not necessarily justified on the right margin. This command has also been redefined from Plain TEX in that \rightskip values are preserved.

\justify
This macro has the opposite effect of the \raggedright macro and causes the text to be right justified (\justify is the default in HP TEX).

2.6 Boxes

\boxline=⟨dimension⟩
This macro assigns the width of the lines used for boxes. The default boxline dimension is 0.01332 inch.

\boxspace=⟨dimension⟩
This macro assigns the width of the space between a boxed item and the line of the box. The default width in HP TEX is 5 points.

\boxit{⟨horizontal list⟩}
This macro encloses the argument in a box. If encountered while making a line of text, the box's bottom line will be along the baseline of the text. The ⟨horizontal list⟩ may contain multiple lines separated by \cr to be centered within the box. Note that each line is then treated as a **group**, so font changes, *etc.*, on one line will not affect the next line. When using this command, the ⟨horizontal list⟩ will be raised somewhat above the line, like this. If you desire that the text not be raised above the baseline, use the \textbox command (described below).

\textbox{⟨horizontal list⟩}
This command causes a box to be placed around the ⟨horizontal list⟩, **without** altering the text, like this. A textbox cannot be broken from one line of text to another.

\centerbox{⟨horizontal list⟩}
This macro centers a box horizontally on the page and inserts space above and below. Multiple lines can be specified using \cr (see \boxit description). A \centerbox within a \centerbox will not work, but \boxit inside \centerbox will.

2.7 Notes and Warnings

Two other text structures available with the HP TEX macro package are notes and warnings.

\start{note}
\finish{note}
A note is inset twice the \inset dimension on both margins and set apart from the rest of the text by extra vertical space. If a note would otherwise start less than half an inch from the bottom of a page, a page eject is performed prior to the note. \finish{note} signifies the end of the note.

\start{warning}
\finish{warning}
A warning is similar to a note, except that is also set apart by horizontal rules above and below the text. **\finish{warning}** signifies the end of the warning text.

2.8 Verbatim Mode

Verbatim mode will cause text to be printed "as is," without any justification. Special characters in this mode are the backslash (\) and the curly brackets ({), (}). Most other characters can be used and will be printed verbatim.

Verbatim mode is intended for use with simple text which does **not** contain a large number of control sequences. The reason for this restriction is that there are many characters (such as a space, a tilde, *etc.*) which take on different meanings in verbatim mode. Some of these may be imbedded within a control sequence and can cause problems when they are expanded.

You will usually want to select a "non-proportionally spaced" font (like "tt") for use in verbatim mode. The reason for this is that "proportional fonts" (like "rm") will cause the printed output to be aligned differently than what appears on the CRT during source file typing. Other non-proportional fonts available with the HP 2688A Laser Printer are Courier, Gothic, Pica, Script, Prestige and Line Printer.

– NOTE –

> Although verbatim mode using non-proportional fonts will **usually** produce output that exactly matches what you see on the CRT, there is at least one exception to this. Long sequences of characters may be spaced slightly differently than a string of blank spaces of identical length . This is a result of the rounding anomalies that occur when TEX's **ideal** character sizes are converted to the printer's **actual** sizes.

\start{verbatim}
This macro causes the following text to be printed "verbatim" without any justification.

\finish{verbatim}
This macro ends the verbatim mode described above.

2.9 Paragraph Levels

These macros can be used to create paragraph headings of four different levels. The ⟨horizontal list⟩, and page number can be written to the file ⟨jobname⟩* to be used for a table of contents (the asterisk "*" signifies the **contents** file).

The command **\ctswrite**⟨horizontal list⟩ will automatically open a file and write the ⟨horizontal list⟩ and the page number to the file. The **\contents** command then uses the ⟨jobname⟩* file to create the table of contents (refer to the **\contents** command description in this section).

\level⟨integer between 1 and 4⟩{⟨horizontal list⟩}
This is the command used to print a paragraph heading. Whatever format is specified using the commands listed below (**\firstlevelhead**, **\secondlevelhead**, *etc.*) will be used. For example, the command used to set the paragraph heading of this section looked like this:

\level3{Paragraph Levels}

The **\level** command will automatically update the **\levelno** to print the paragraph head.

\firstlevelhead{⟨horizontal list⟩}
\secondlevelhead{⟨horizontal list⟩}
\thirdlevelhead{⟨horizontal list⟩}
\fourthlevelhead{⟨horizontal list⟩}
These commands specify the tokens that are inserted to format each of the various head levels. They should be used when the default conditions of HP TEX, as described below, are not satisfactory. A control sequence \title is defined to be the ⟨horizontal list⟩ from the corresponding \level command. For example, the command:

```
\secondlevelhead{\need.75in\bigskip\leftline{\fourteenbf%
    \levelno\enspace\title}\medskip%
    \ctswrite{\hskip15pt\tenrm\title}}
```

would cause all second level paragraph heads to be printed on a new page if less than .75 inch remains on the current page. The title will be left justified in fourteen point boldface type. The level number is followed by an "enspace." "Bigskip" glue is inserted before the title with a "medskip" after. The \ctswrite command specifies how the level heading will be written in the table of contents, as described below.

Whenever you define a \firstlevelhead, \secondlevelhead, *etc.*, command to specify the formatting of paragraph headings, you must also specify how it will be written in the table of contents (if you plan to create one). The command for doing this is \ctswrite. For example, the fourth level head was redefined from the HP TEX defaults in this document, and the contents writing command was used as follows:

```
\ctswrite{\hskip45pt\ninerm\title}
```

This causes all fourth level headings in the table of contents to be inset 45 points and be written in nine-point roman type.

The default HP TEX paragraph level headings are formatted as follows: \level1 begins on a new page, prints the title in 14 point bold face type and down 1.5 inches from the top of the page. \level2 requires that at least .75 inch remain on the page. If so, the title is printed with some vertical space above it, otherwise, it is set on the top of the next page. In any case, it is printed in 12 point boldface type. \level3 and \level4 are similar, but require less space to remain on the page and print the title in 10 point bold face and 10 point roman respectively.

The "levelhead" macros are **local** to their block structure. This provides a simple means to return to the HP TEX default levelhead style. All you would need to do is enclose the command and any text you want affected within braces. After the closing brace, TEX will return to the default format. Note that this should only be attempted for **shorter** blocks. Creating large blocks of several pages in length can cause TEX to run out of memory.

– NOTE –

If you need to change the font style or size within a level heading, you must specify the **entire** font name, (for example: \tentt... not \tt).

\levelno
This macro can be used whenever the author desires the number of the heading to appear in the document. The number of the various levels of headings appear, separated by periods. For example, typing **Section** \levelno might result in: Section 1.2.7.1 if this was the first fourth-level heading of the seventh third-level section of the second... *etc.* Levels lower than the last modified level do not appear in \levelno.

\setlevelno{⟨integer⟩.⟨integer⟩...⟨integer⟩}
This macro can be used for presetting the one through four heading numbers. If you use this command before a \level command, then the counter will increment to the next number when the \level command is encountered. Therefore, you should preset to one number below the desired level. You cannot preset the level 2 counter without specifying the level 1, or the level 3 without level 1 and 2 and so on. (For example, you cannot specify \setlevelno{1. . .4}, you must specify {1.⟨integer⟩.⟨integer⟩.4}.)

\contents
This macro forces a page eject and produces a table of contents on the following pages as dictated by the previous heading macros. No vertical glue is inserted, so you may want to use a \vfil command immediately before \contents.

\topofcontents{⟨vertical list⟩}
\botofcontents{⟨vertical list⟩}
These macros specify a list to be placed above and below the table of contents. The ⟨vertical list⟩ may include logos, rules, *etc.* The table of contents is quite rigid, so a \vfill is appropriate in at least one ⟨vertical list⟩. Defaults in HP TEX will format the table of contents as they appear in the beginning of this manual (with the exception of the headings).

- NOTE -

\topofcontents and \botofcontents must **both** be specified **before** the \contents command is used. (That is, you cannot specify \botofcontents after \contents, even though this seems logical.)

2.10 Multiple Columns

The following macros control the number of columns on the page. Multiple column and single column text can be mixed on a page by using the following commands.

\columnspace=⟨dimension⟩
This command specifies the amount of space between columns on a page. This dimension should be assigned before entering multicolumn mode and should remain unchanged for the duration thereof. The default column space in HP TEX is 0.5 inch.

\start{twocolumns}
\start{threecolumns}
\finish{twocolumns}
\finish{threecolumns}
These four commands cause the text to be formatted into multiple columns. If a \balance, \newpage, \evenpage or \oddpage macro is encountered while in multicolumn mode, the columns are balanced on the page prior to the page eject. Exit from multicolumn mode causes the columns to be balanced as well. Unbalanced columns may be obtained by using \vfill \eject while in multicolumn mode. Balancing forces the top lines of each column to be lined up. The bottom lines are lined up as well unless \raggedbottom has been specified. Discardable items (such as glue, penalties, *etc.*) immediately following these commands will be ignored. (To prevent this, you may use \null immediately after \finish{*n*columns}.)

\balance
This command causes the columns to become balanced. Discardable items (such as glue, penalties, *etc.*) immediately following these commands will be ignored. (To prevent this, you may use \null immediately after \finish{*n*columns}.)

- NOTE -

When using multiple columns, you may find it helpful to use the plain TEX commands \tolerance and \hbadness (to increase the stretch and shrinkability of interword glue and decrease complaints regarding "underfull hboxes"). See *The TEXbook* for more information.

2.11 Tables

The following macros are useful for setting tables. These macros are different from the plain TEX \halign and \valign commands in that they format one row at a time. The table can be justified as a whole either left, right or centered. As long as the number of columns in each entry remains constant, the entire table appears justified. (Interesting figures, such as pyramids, hopscotch boards, *etc.*, can be produced by changing the format and the number of columns in each entry.) Each column of the table has its own user specified width. Between each pair of columns is a vertical line (which can be easily made invisible). This vertical rule takes no space from any column so the rule width may be varied without altering the column dimensions. An entry may also be designed to span two or three columns. Horizontal bars are treated exactly the same as data entries. The macro \tbar can be used to produce such a bar.

\tableline=⟨dimension⟩
This parameter specifies the dimension of bars and rules in all subsequent entries. This parameter should remain unchanged throughout the entire table. Default value is 0.01332 inch.

- NOTE -

There may be some occasion when printing a table that you get the error message,
"Printer error:
**Could not process all the data, data lost."
The printer uses a variety of different length line segments to "build" a line. If the tableline specification you have chosen requires too many individual segments to make up the exact specification, it may cause this error. There are a few remedies available, as follows:

—If you are using the default magnification (1.0), then try using the default tableline dimension (just leave out the \tableline command).

—If you are using a file magnification value other than 1.0, then alter your \tableline specification so that its value, when converted to dots (multiply length in inches × 300), is a power of 2 (*i.e.* 1, 2, 4, 8, 16, 32 or a multiple of any of these except 1).

\tablespace=⟨dimension⟩
This parameter specifies the minimum amount of space to be placed between the vertical rules and data entries in the table. Default in HP TEX is 5 points.

\tableformat{⟨table spec.⟩|⟨column spec.⟩⟨dimension⟩|...|⟨column spec.⟩⟨dimension⟩}
This command specifies the table format. Note that a vertical bar (|) is required as a separator between specifications, but not at the beginning or end of the list (different than \tablerow or \tablebar).

⟨table spec.⟩ can be any of \leftline, \rightline, or \centerline. ⟨column spec.⟩ can be any of \leftline, \rightline, \centerline or \paragraph. Other options are allowed if you make them yourself; for example, if you type:

```
\def\mything#1{\line{\hskip 1in #1\hss}}
```

then \mything would be a valid ⟨table spec.⟩ option that would print the table 1 inch from the left margin. The ⟨column spec.⟩ is always a single token (\leftline, \rightline, \centerline or \paragraph), and is followed by a dimension. The token specifies the standard justification of the column (\paragraph must be used if the column is to contain paragraphs), and the dimension specifies the absolute width of the column. Again, if you are not satisfied with the selection you can make your own—the rules are that the macro must consume one token (containing the text) and produce a box of width \hsize.

– NOTE –

If the \paragraph column specification is used, \parskip must be 0 points (this is the default). Otherwise, the paragraph entry will be raised or lowered by the current \parskip amount and the table will be out of alignment. If you are using \noindentstyle and have not reset \parskip, then inserting an \indentstyle command just prior to the table will return to the proper (0 points) parskip.

\tablerow{|⟨column list⟩|⟨column list⟩...|⟨column list⟩|}
\tablebar{|⟨column list⟩|⟨column list⟩...|⟨column list⟩|}
These commands are used to build tables. \tablerow should be used if the ⟨column list⟩ contains text, while \tablebar should be used for bars or other non-text entries. Horizontal bars are obtained by using either \vrule height ⟨dimension⟩ width ⟨dimension⟩ or \tbar as a ⟨column list⟩ (\tbar uses \tableline as the rule height). The following syntax holds true for both macros:

- A vertical bar (|) or a tilde (~) must be present to indicate the beginning and end of each column, including the first and last entry (differs from \tableformat syntax).
- The vertical bars (|) may be replaced by tildes (~) if a visible vertical bar in that position is not desired.
- If the first thing in a ⟨column list⟩ happens to be the control sequence \span{⟨integer⟩}\tbar, then the number of columns specified as ⟨integer⟩ are spanned by the entry (in this case a horizontal bar). The natural justification for the spanned entry will be that of the leftmost column spanned. (Note that span{⟨integer⟩} is **not** followed by a blank space as this can cause difficulties.)

– NOTE –

While building tables, if a \tablebar or \tablerow runs more than one line of text (on the CRT), it is good practice to use the percent sign (%) at the end of each line. This will tell TEX to ignore anything else on that line and prevent possible problems with extra spaces being misinterpreted by TEX.

2.12 Fonts

Only a few fonts are preloaded in HP TEX. These macros allow you to access a variety of font families, styles, and sizes.

\fontdef\⟨command⟩={⟨library⟩,⟨font name⟩}
This command equates a command of your choice with a font (as used in the library). You can use ⟨library⟩(optional) to specify other than the default font library. This command differs from TEX's **\font** command in that the font won't actually be loaded until the first request (if any).

\fivepoint
\sixpoint
\sevenpoint
\eightpoint
\ninepoint
\tenpoint
\twelvepoint
\fourteenpoint
\eighteenpoint
\twentyfourpoint
These commands select the font point size from five to twentyfour. (In **Math Mode**, these commands only apply to the default font within a font family. If you are using a current family other than **\fam0**, you must make the appropriate font assignments (see Appendix A, "Changing Fonts in HP TEX").

\rm
\it
\bf
\sl
\sa
\tt
These commands select roman style, italic, bold face, slanted, sans serif, and typewriter style.

If the selected font is undefined, the font style is changed to roman. If that new font is still undefined, the size is changed to ten points. Computer Modern ten point roman will always be defined in HP TEX.

2.13 Miscellaneous Macros

\note{⟨horizontal list⟩}
This macro puts a footnote at the bottom of the current page and inserts a superscript footnote number at the location of the command and the footnote. Footnote numbers are allocated starting with 1 and can be reset by the use of **\resetnotes**. The ⟨horizontal list⟩ is the footnote text.

\need⟨dimension⟩
The result of this macro is that if the page breaking algorithm of TEX determines that the current position would optimally fall within ⟨dimension⟩ of the bottom of the page, the page is broken leaving some empty space at the bottom.

– NOTE –

Since the **\need** command uses a negative penalty to **encourage** (not **force**) TEX to break the page, it will not always have the effect you might anticipate. This is especially true if the command is encountered near the top of the current page and the remaining text will not adequately fill the current page.

\super{⟨horizontal list⟩}
The argument is printed as a superscript.

\sub{⟨horizontal list⟩}
The argument is printed as a subscript.

\lbreak
This macro forces a line break within a paragraph. It inserts "hfil" glue before the break, so the line will **not** be right justified. If you want right justification, use plain TEX's \break command.

\uline{⟨horizontal list⟩}
This macro causes the argument to be underlined.

\mon
This command prints the current month name. For example, if the current month were August, \mon would print the letters "August" in the current font.

\date
This command prints the current date in the format Month Date, Year. For example, if today's date were 12 December 1984, \date would print the characters "December 12, 1984" in the current font.

\hour
This command prints the current time of day (for example – 4:07 PM)

2.14 Block Structure

Certain local document styles are considered to belong to a "block." (Block structure, as used here, simply refers to a portion of a file that has some common formatting instructions applied to it.)[2]

The current block is defined to be the most recently opened block. Pending blocks are blocks that have been opened but not closed. The HP TEX macros \start and \finish are used to open and close blocks.

Valid ⟨blockname⟩s pre-loaded in HP TEX include **indent**, **note**, **warning**, **verbatim**, **twocolumns** and **threecolumns**, all of which have been described in this chapter. User defined blocks may be implemented by defining a control sequence, \BEGIN⟨blockname⟩ and, optionally, another control sequence, \END⟨blockname⟩.

\start{⟨blockname⟩}
This command determines if a control sequence \BEGIN⟨blockname⟩ has been defined. If one exists, a new block is opened and the control sequence is invoked. Otherwise, an error message is issued and the command is ignored.

\finish{⟨blockname⟩}
This command is used to close the current block and invoke the control sequence \END⟨blockname⟩ if one has been defined. If ⟨blockname⟩ matches the current block name, the current block is closed. If not, an appropriate error message is issued and corresponding corrective action is taken. If ⟨blockname⟩ is not valid or there are no pending blocks, the command is ignored. If ⟨blockname⟩ is valid but does not match the current block name, \done commands are inserted until either ⟨blockname⟩ matches the current block name or until all pending blocks are closed.

[2] The concept of **grouping**, explained in this manual and in *The TEXbook*, is basically the same as the "block structure" referred to here.

\done
This command closes the current block without any error checking and invokes the control sequence \END⟨blockname⟩ if one has been defined. A TEX error message will be issued if there are no pending blocks.

Following is an sample usage of the \BEGIN and \END commands in creating user-defined blocks:

- EXAMPLE -

This example was created by defining the following block:

```
\def\BEGINexample{\bigskip\centerline{- EXAMPLE -}
    \medskip\leftskip.75in\rightskip.75in}
```

The block was opened using the command: **\start{example}**. The optional control sequence \END⟨blockname⟩ was also used, as follows:

```
\def\ENDexample{\medskip}
```

The **\medskip** vertical glue will be inserted after the block is closed using the command: **\finish{example}**.

3 HP 2688A Control Macros

These macros are for controlling certain features of the 2688A Laser Printer:

3.1 Page Copy Control

\copies⟨integer⟩
This macro assigns the number of copies per page starting with the current page. The copies are uncollated; in other words, if **\copies5** occurred on page 3, the output will have five copies of page 3, then five copies of page 4, *etc.* The control of the number of copies is accomplished through one of TEX's counters. Default is counter number 1. Possible integers and their effect are shown below:

- 1 to 32,767 will produce the specified number of copies.
- Greater than 32,767 will produce 32,767 copies.
- 0 will produce 1 copy.
- Less than 0 (negative number) no copies (can be used to suppress certain sections of a document, *etc.*)

The counter can be changed using the following macro.

\selectcopycounter⟨integer from 0 to 9⟩
This is used for changing which counter is used for controlling the number of copies per page (default is 1). This command generally should not be used except by people who are writing their own macros.

`\copieson`
`\copiesoff`
If the multiple copies feature is being used in a document, these macros can be used to alternate between one copy per page and the number of copies assigned by the `\copies` macro. The difference between `\copiesoff` and `\copies1` is that when the former is followed with `\copieson`, the original number of copies is restored, whereas when the latter is followed with `\copieson`, there is, of course, no effect.

3.2 Logical Page Control

The Print Server features a concept called "logical paging" which is controlled using the following set of macros. A logical page is a rectangular addressing space on the sheet of paper (referred to as the "physical page"). Logical pages have an orientation (portrait, landscape, reverse portrait or reverse lanscape) and a position on the physical page. Through the operator interface of PS2688A, you may define a list of logical pages that may be written to on any sheet of paper. These logical page specifications may also be stored in a PS2688A input file to avoid typing in the specifications for each job. Using these macros, you can address the logical pages with either of two methods: either by explicitly specifying a logical page for each page of TEX output, or by specifying an ordered list of logical pages and letting the system cycle through the list.

`\lpdef{`⟨logical page definition⟩`}`
This command defines a logical page size and orientation. The ⟨logical page definition⟩consists of the logical page number, followed by the **left**, then the **top** dimensions (distance from the edges of the physical page), and the logical page orientation (portrait, landscape, rev–portrait or rev–landscape). Orientation may be abbreviated `P`, `L`, `RP`, and `RL`, respectively, and either upper or lower case characters will work. The edges of the page that would normally be the top and left as viewed from that logical page's orientation are always considered top and left in this context. An example usage follows:

```
\lpdef {1, 1.5in, 3in, L}
```

This would cause logical page number 1 to be printed in **landscape** orientation, with the left edge of the print (as viewed from landscape orientation) beginning 1.5 inches from the edge of the physical page. The "top" (left-hand edge along the paper path) will begin 3 inches from the edge.

The default values for logical pages in HP TEX are: LEFT=1in, TOP=1in.

- NOTE -

> The dimension specified as distance from the top edge does not take into account the space needed for headings. If running headings are to be used, the logical page definition should allow extra room for them when specifying "top" dimension. (Footings are also printed "in the margin," so when specifying `vsize` be sure to leave room for them.)

`\lplist{`⟨logical page list⟩`}`
This command specifies the logical page list to be used whenever logical pages are being used implicitly. Only one such specification is allowed within a document; if more than one exist, all but the last are ignored. This command initializes implied logical paging.

The ⟨logical page list⟩ is a list of integer numbers from 1 to 32 which have been defined using the `\lpdef` command. Positive number entries will cause a physical page eject prior to initiating the logical page entry. Negative numbers require no such physical page eject. Each entry number should be separated from the next by a comma.

The remaining commands control explicit logical paging. If any of these commands occur within a page, a page eject is issued and the next page to be printed will be dependent upon the particular command.

If any of these commands occur on the top of a page, the prescribed action will be delayed until the next page break (be it natural or forced by a \newpage, \eject, *etc.*). For example, suppose the first page of every chapter in a document uses logical page 5, then \ppageto5 \lpresume would eject the current page (assuming it is not empty), issue a physical page eject and begin printing on page 5. Later, when the page is full, TEX breaks the page and resumes printing according to the logical page list.

\lpageto⟨integer between 1 and 32⟩
This command causes the next page to be printed on the specified logical page. No physical page eject is implied.

\ppageto⟨integer between 1 and 32⟩
This command is similar to \lpageto but implies a physical page eject prior to printing on the specified logical page.

\lpresume
This command is used after a \ppageto or \lpageto command to resume printing on the current page of the logical page list.

\lpreset
This command resets the logical page list and prints on the first page in the list.

\lpexit
This command exits the current loop in the logical page list and prints on the next page in the list.

\selectlpcounter⟨integer⟩
This is used for changing which counter is used for specifying the logical page (default is 2). This macro generally should not be used except by people who are writing their own macros.

4 Formatting a Sample Document

In this section we will format a simple document utilizing many of the HP TEX macros. For this example, we will accept all the HP TEX default values for the various commands.

A Sample Document

Isn't that nice? If we wanted to, we could have easily set that off to the right like this:

A Sample Document

Now, how about a textbox? Those are always fun. Or, if we really want to set something off, we could ask TEX to:

put it inside
a nice little
centered box,

or, maybe just "boxit" – like this.

But say you're really serious about getting someone's attention.

LOOK!
You can use the "Warning" command, like we did here.

Now, let's use the extraordinary capabilities of TEX to generate a mathematical formula:

$$\hat{n}_2(s) = \frac{1}{\alpha_2}\left[\left(\frac{\partial C_2}{\partial x}\right)_{x=0} + \frac{k_1 \hat{n}_1}{D_1}\right]$$

Next, we'll create a table to show the dimensional units available in TEX:

TEX UNIT	DESCRIPTION	PER INCH	TEX UNIT	DESCRIPTION	PER INCH
in	inch	1	mm	millimeter	25.4
cm	centimeter	2.54	dd	Didot point	67.54
pt	printer's point	72.27	bp	big point	72
pc	pica	6.02			

Itemized lists are very useful:

1. For listing things.
2. For making a series of points.
   a. You can even use subitems.

Now, look at the next page to see exactly how this sample document was formatted.

Here's the source file for "Sample Document"

```
\noindentstyle
\centerfooting{\eightbf THIS DOCUMENT WAS CREATED USING \HPTEX}
\null\bigskip
\centerline{\twelvebf A Sample Document}

Isn't that nice? If we wanted to, we could have easily set that off to the
rightlike this:\lbreak
\rightline{\twelvebf A Sample Document}

Now, how about \textbox{a textbox?}  Those are always
fun. Or, if we really want to set something off, we could ask \TeX\ to:
\centerbox{put it inside\cr
           a nice little\cr
           centered box,}
or, maybe just \boxit{``boxit'' -- like this.}

But say you're really serious about getting someone's attention.

\start{warning}
\centerline{\bf LOOK!}
\centerline{You can use the ``Warning'' command, like we did here.}
\finish{warning}

Now, let's use the extraordinary capabilities of \TeX\ to
generate a mathematical formula:
$$\hat{n}_2(s)={1\over \alpha_2} \biggl[ \biggl({\partial C_2\over
     \partial x}  \biggr)_{x=0} +{k_1 \hat{n}_1\over D_1}  \biggr]$$
Next, we'll create a table to show the dimensional units available in \TeX :
\medskip
\tableformat{\centerline|\centerline 1in|\leftline 1.2in|%
     \centerline 1in|\centerline .05in|%
     \centerline 1in|\leftline 1.2in|\centerline 1in}
\tablebar{|\span7\tbar|}
\tablerow{|\TeX\ UNIT|DESCRIPTION|PER INCH||\TeX\ UNIT|DESCRIPTION|PER INCH|}
\tablebar{|\span7\tbar|}
\tablerow{|{\tentt in}|inch|1||{\tentt mm}|millimeter|25.4|}
\tablerow{|{\tentt cm}|centimeter|2.54||{\tentt dd}|Didot point|67.54|}
\tablerow{|{\tentt pt}|printer's point|72.27||{\tenttbp}|big point|72|}
\tablerow{|{\tentt pc}|pica|6.02|| | | |}
\tablebar{|\span7\tbar|}
\medskip
Itemized lists are very useful:
\itm For listing things.
\itm For making a series of points.
\sitm You can even use subitems.
\enditems
Now, look at the next page to see exactly how this sample document was
formatted.
\vfil\eject
```