

# Developing a Pop-Up Help Facility for T<sub>E</sub>X on PCs

Peter Flynn†

University College of Cork, Cork, Republic of Ireland

BITnet: cbts8001@iruccvax

## Abstract

The manuals, tutorials, and descriptive books about T<sub>E</sub>X and its allied products are fairly comprehensive, but their very technical completeness means that the novice user is often unable to locate quickly the information needed from among the quantity provided. Many new users also ask, “where is the HELP file?” and too often receive an unsatisfactory response. This paper describes the construction of a memory-resident pop-up help facility for T<sub>E</sub>X on PCs, developed using publicly-available software. The problems discussed include determining the level of complexity that a help system needs; the balance between fine and coarse detail; cross-referencing and circular references; context-sensitivity; visual presentation; the selection of topics; the regard for later conversion to other systems with structured help facilities (e.g., VMS); and the development of tools for assisting the help file development process. The help file itself, when completed, will be made available with the distributable software in the public archives.

## Introduction

At the 5th European T<sub>E</sub>X Conference in Cork, a paper was presented which summarised the approaches taken by the various authors of a selection of documentation on T<sub>E</sub>X.[1] The author’s conclusions were that the quality was uneven: some documents contained errors of logic, some were difficult for the novice to understand or find a path through (and these were documents *intended* for novices!), and some manifested aesthetic problems in the design and layout which made using them less easy.

The problem facing the present author, in his user-support function, was to provide a faster startup path for novice users of T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X.<sup>1</sup> Training courses were out, as the numbers involved were too great for the resources of the Computer Centre, so the use of machine-based assistance was investigated.

---

† The author would like to express his thanks to all those who generously gave their time and expertise by electronic discussions during the development of this system.

<sup>1</sup> From here on, references to T<sub>E</sub>X imply also L<sup>A</sup>T<sub>E</sub>X, for brevity, unless explicitly designated ‘Plain’ or by mention of some other macro package.

Several Computer-Based Training (CBT) packages were examined, but all were too expensive or required too great an investment in time to get running, even at a low level. A simple help facility was therefore identified as a mechanism for providing the user with the information necessary to solve the most common low-level queries. This has now been expanded to handle some more complex material as well, and contains additional matter on more general topics such as design, layout, and typography.

## Existing HELP Facilities

Identifying existing HELP files was not difficult as there are relatively few of them. E-mail messages were sent to many contacts asking for copies, where available, of help texts.

The most numerous responses were for the VMS `latex.hlp` file, summarised below. In addition, the help texts supplied with various versions were retrieved and examined (not a long task!).

There is, of course, comprehensive assistance available on T<sub>E</sub>X through the electronic mail networks for those fortunate enough to be connected; but this is of an asynchronous nature, and something more immediate was required.

Table 1: Content of latex.hlp

Level	Sections	Content
1	1	Description of T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X
2	3	Commands, Parameters and Qualifiers
3	22	Things you can do in L <sup>A</sup> T <sub>E</sub> X
4	106	Command and environment names
5	50	Math and more specialist commands

**VMS (L<sup>A</sup>T<sub>E</sub>X).** A DEC VAX/VMS help file is a structured multi-level hierarchy. Sections labelled for level 1 appear as headings when you type `help latex`, and selecting one of these first gives you the associated help text and then displays the relevant level 2 headings associated with it. Selecting one of these repeats the mechanism and takes you to level 3, and so on. Pressing the Return key takes you back one level and eventually quits from `HELP`. The source text of help files is compiled into a help database (`.h1b` file) for speed of access.[3]

The `latex.hlp` file is fairly comprehensive and some 2,100 lines long: its structure is described in Table 1. No information was available about the decision-making process which was used as to what to include and what to omit.

**VMS (LSE).** The author has reports of a on-line help system for use with the VAX/VMS Language-Sensitive Editor (LSE), but has been unable to locate the text. There is documentation on the use of editor command keys for using the templating facilities of the LSE, but no evidence of help on T<sub>E</sub>X commands for doing the formatting.

**UNIX man pages.** The man pages consist of a brief description of the T<sub>E</sub>X system with some notes on directory storage and environment variables.

**PC and Macintosh versions.** No existing help files were located on how to use T<sub>E</sub>X commands. All the known PC and Mac versions of T<sub>E</sub>X have some form of help documentation, but this is restricted almost entirely to technical details (usually very good) of how to get the program running and how to organise logical names and subdirectories.

Some commercial versions supply a manual (e.g., PC-T<sub>E</sub>X, T<sub>E</sub>Xtures); others rely on the T<sub>E</sub>Xbook. None supply any of the public-domain workbooks which are available.[2]

**Other environments.** In the time available no other environments were examined, although there are some help facilities available on Atari and IBM mainframes. It is hoped to include some of this material (if it can be located) at the  $\beta$  test stage.

## Selection of Topics

The overriding need to provide some structure within which to develop a help system meant that a categorisation of topics was required. The structure used would need to apply equally to Plain T<sub>E</sub>X and to L<sup>A</sup>T<sub>E</sub>X as well as to other macro packages for which help might be needed at a later date.

The commands and facilities of T<sub>E</sub>X fall into several discrete types:

- Operating system commands needed to run the programs, and the switches, setups and arguments;
- Commands which affect the positioning of text;
- Commands which affect the appearance of text;
- Values such as dimensions and counters;
- Commands to operate logical structures like sectioning, index, contents, tables and mathematics modes;

but as the objective was to create a system which would answer the “how do I...” questions, it was necessary to marry these abstract categories with the requirements of the users. There is of course a considerable degree of overlap between the coarse categories described above.

**User requirements.** Working out which commands are most useful to a user was the most difficult of all the tasks. Commands used daily by some users are never used at all by others. The only guideline available was experience, and this is naturally site-dependent.

One of the prime expectations of this paper is that other user-support workers will be able to contribute their expertise in order to improve the composition of the system.

From logs kept of user queries, some common requirements were identified:

- How to process a file through T<sub>E</sub>X and display or print it;
- How to change type size and style, and find out what is available;
- How to change spacing and layout;
- How to insert graphics;
- How to handle logical structures;
- How to do mathematics; and
- How to interpret error messages.

These categories match fairly closely those described in the previous section pertaining to T<sub>E</sub>X commands themselves, with the addition of graphics and error messages. It was felt that the interests of the users would be best served by a system which attempted to model these needs, rather than mirror the traditional sequential presentation (*à la* T<sub>E</sub>Xbook), even though some of this would mean extensive cross-references to items which the user may not be interested in.

**Levels of foreknowledge.** The system as it currently stands assumes no foreknowledge whatever about T<sub>E</sub>X itself. It does however assume that the user knows how to switch on the machine, type a command and use a text editor, and follow simple single-key instructions to run the help system itself.

Although this may be considered a valid approach for novice users, it also means that a substantial amount of explanatory text is required at a low level. Some basic concepts about typesetting therefore need to be covered, which are wholly external to T<sub>E</sub>X or any other form of desktop publishing. This forms an additional category to those described above.

**Depth of detail.** No attempt is made to explain *why* things are done as they are in T<sub>E</sub>X unless this knowledge is required to understand *how* a command or facility works as it does. Thus, for example, no details of the internals of T<sub>E</sub>X or L<sup>A</sup>T<sub>E</sub>X are included.

There is perhaps scope for a more advanced version of this help system, to act as an *aide-mémoire* to experienced users.

## Structure

Although the structure of a help text is to some extent conditioned by the software used, it is possible to approach the problem in different ways. Before making a final decision on the software, several options were considered:

**Flat file.** This is the simplest format. Topics appear as headings, and the file can be searched by system utilities to locate the text required. However, unless more sophisticated software is used, display of search 'hits' is usually restricted to a line or so either side of the target keywords.

**Multi-level.** Multi-level (hierarchical) structures (like VMS HELP) require much more sophisticated software but provide highly tailorable context-sensitive displays, although locating the help required is usually restricted by the need to navigate

down, up, or across the hierarchy, rather than by searching the whole text for matches.

**Hypertext.** This very powerful information concept is available on a variety of platforms other than the Apple Macintosh, where it is popularised in the Hypercard<sup>TM</sup> program. The software is highly complex, and developing 'stacks' (help texts) can be fairly lengthy business if the extensive cross-referencing and multimedia management facilities are used. If such a system could be provided as a memory-resident option under MS-DOS, complete with graphics, serious consideration would be given to using this method.

**Hybrid structures.** These offer some combination of the facilities of those structures mentioned. Specifically, they use flat files, but with entries tagged in such a way as to facilitate compiling (like VMS HELP); and there is usually some element of hypertext, in that the user can wander *ad libitum* from one cross-reference to another without the restrictions of a strictly hierarchical system.

**Compatibility with other systems.** In developing the help text, it has been borne in mind that it should be possible to port the system to another file structure without too much rewriting. Although the file-marking itself is specific to the software used, the one selected nevertheless uses a plain text file, which can be edited or stripped apart by data management software for reworking into other systems.

## Implementation

Six PC-based systems were evaluated: HELP, PAINTHLP, HELP\_SYS, QUIKHELP, QHELP and HELPSB. They provided a mix of file structures and access methods. They were evaluated by checklist and examination against the following criteria:

- Ability to run as TSR (memory-resident pop-up) over T<sub>E</sub>Xware and editors;
- Size of memory taken;
- Ease of use by novice;
- Ease of writing/formatting files for help input;
- Need or otherwise for special configuration of PC; and
- Support and documentation.

**Choice of delivery system.** The software chosen was QHELP, written by Mark van Kekerix. This product was chosen because it is the simplest and fastest to work with, requires no special facilities, takes a plain input file, and sits as a TSR over the T<sub>E</sub>X executables tested without interference.

**Disk storage.** The help file in its present state is around 50kb of text, and a similar size when compiled. The QHELP executable is only 21Kb, and no further disk space is required by users. It is envisaged that the help file should probably contain additional explanatory material, and there will doubtless be additions to the commands covered as a result of  $\beta$ -test users' comments. It is suggested that user-support people develop and keep their own site-specific help file, and append this to the main file for recompilation and distribution locally. Thus the resultant size of the completed text in compiled form is difficult to predict.

**Memory constraints.** QHELP requires approximately 35kb memory. Future versions may require more or less, depending on added facilities or reprogramming to minimise usage.

**Availability and Cost.** The software is freely available in the public domain. The author specifically states in his introduction:

[...] you are free to use QHELP and any help files you have created yourself or obtained from bulletin boards, etc., but if you want to package QHELP with another program you are selling for profit, contact me for written permission to do so (I will ask for a small fee for such a license).

The limitation on "bundling" QHELP with other programs applies only if the program is being sold for profit. If you have developed a public domain program and would like to use QHELP and a help file you have written as the program's help system, please feel free to do so.[4]

**Physical file structure.** A QHELP source (text) file is plain ASCII, using the colon in column 1 to flag keywords, and the tilde to mark references to them in the text. During use, references in the text are highlighted, and selecting one with the cursor and pressing the Enter key causes the help text for that keyword to be displayed.

Each section (description of a keyword) is formatted to fit the selected size of the help window, and can extend to multiple pages. From user experience it is more effective to have more subdivisions of keywords, with one screen each, than to use multiple screens for a single keyword. Within each description, text can be freely formatted as desired.

**Referencing.** Because of the indiscriminating nature of the reference-to-keyword mechanism of QHELP, the marking of occurrences of keywords in the text as references must be done with care. Too many, and the user may be confused by which one to choose next; too few, and the user may not be able to find the help required. Marking a word (*i.e.*, a command) as a reference means that that word must

occur as a keyword (QHELP will refuse to compile otherwise), but otherwise there are no logical restrictions on what can or cannot be thus marked. The tendency was noted early on to mark too many words; but it is not certain until the system is used more extensively by its target audience whether the level chosen is adequate. The average number of references marked in a section is seven. Care must also be taken to avoid purely circular references which will confuse the user, although an Alt-key function enables the user to jump back to the opening screen if needed.

**Development tools.** QHELP normally recognises a space character (ASCII 32) or any non-alphabetic character as the end of a keyword or reference, but has the ability to allow the user to specify additional non-alphabetic characters which are to be taken as valid characters in a keyword string. The present file uses backslash, underscore, and curly braces as additional characters, for obvious reasons. This specification is only possible interactively at compile time, and cannot be stored as a compiler configuration. The author has used the public-domain STACKEY utility to assist compilation, as this enables the passing of keystrokes to an application, in this case to drive file and parameter selection and initiate compilation from an MS-DOS batch file. The STACKEY software is included in the present distribution.

The QHELP compiler reports unmatched references and aborts on the first such occurrence. There is no facility for listing or otherwise reporting on the file structure, so a routine was developed in the P-Stat data management language to read the help file and do its own parsing for keywords and references, and report in listed form on matched, unmatched and unapplied references. This routine is supplied in the distribution, but it should be possible to rewrite it for other data manipulation languages where P-Stat is not available.

**Presentation.** The colours of the help display can be changed by the user or distributor, as can the hot-key for activating the system. Upon activation, the system will search the keyword list for the word under the cursor (delimited by spaces and punctuation), so the system is to a small extent context-sensitive. If this fails, the default (opening) screen is displayed, and the user can search through the highlighted topics using the arrow keys.

**Compatibility with editors, tex.exe, etc.** QHELP has been tested with PC-Write, PC-TEX (both 8086 and 80386 versions), emTEX (both 8086 and 80286 versions) and the associated .DVI drivers on a variety of IBM and compatible PCs. No errors

have been reported, and there was no noticeable impact on system performance.

### Text Generation

As discussed above, the source of the present text has been generated by the author, in the case of Plain T<sub>E</sub>X commands, and taken from the VMS help file, in the case of L<sup>A</sup>T<sub>E</sub>X, with authors' additions and substantial modifications to the language.

**Complexity of wording.** The level used in writing is that normally associated with instructional material for novice usage. A conscious effort was made to avoid computing jargon and the use of typographical terminology beyond the experience of the normal wordprocessing user. It was assumed that the user has reasonable fluency in the use of English, but the author would be delighted to see versions of the text made available in other languages.

### Conclusions

The system provides a first attempt at making T<sub>E</sub>X formatting information available to the user without the need for tuition or a manual. Although this may be viewed as 'wrong' by educationalists and educators alike, in the situation where demand for help far exceeds supply, it is virtually the only solution.

**Work outstanding.** As noted above, more work is needed to add some of the less frequently used commands which may come to the users' notice and require explanation.

In particular, the many add-ons for T<sub>E</sub>X beyond L<sup>A</sup>T<sub>E</sub>X, such as PICT<sub>E</sub>X, musT<sub>E</sub>X, CIR<sub>E</sub>T<sub>E</sub>X and so on, should be documented in outline at least, so that the user is aware of their existence, and is not left trying to reinvent some of the existing wheels unnecessarily.

**Availability.** The software is available from the servers at Aston, YMIR and Heidelberg as file `texhelp.zip` (MS-DOS format). Commercial suppliers would need to adhere to the requirements of the author of QHELP if they wish to distribute the system, but no charge is made for the use of the help text file itself, provided it is distributed unmodified (it may be appended to but not changed without arrangement with the present author).

### References

- [1] Barden, Angela. *Purchasing Pain with all that Joy*. TUGboat 12:1, 1990 Conference Proceedings.

- [2] Doob, Michael. *A Gentle Introduction to T<sub>E</sub>X*. In: `y mir.claremont.edu::[tex.documentation]gentle.tex`, 1991.
- [3] [Author unknown] L<sup>A</sup>T<sub>E</sub>X.HLP (Help file for the DEC VAX/VMS version of L<sup>A</sup>T<sub>E</sub>X). In: `tex.ac.uk::[tex-archive.utils.editors.lse.help]latex.hlp`, 1990.
- [4] Van Kekerix, Mark. *QHELP Pop-Up Help Program Users Guide*. In: `simtel20.army.mil::pd1:/msdos/info/qhelp.arc`, 1990.