Documenting a TEX Archive

Adrian F. Clark
University of Essex
email: alien@essex.ac.uk

Abstract

There are a number of TEX archives in existence: the master sources at Stanford, the LATEX (and other TEXware) collection at Clarkson, Don Hosek's archive at Claremont, and so on. The author is one of a group of volunteers who maintain the TEX archive located at Aston University in the UK. The Aston archive is probably the biggest of the TEX archives, holding complete distributions for Unix, VMS, MS-DOS and the Macintosh, as well as LATEX style files, support utilities, fonts, and so on — a few hundred megabytes in total. Documenting this volume of information is no mean task; indeed, the archive maintainers frequently have trouble remembering where things are!

There is growing interest in coordinating these various archives. While this can only be good for the user, an archive is only as good as its documentation: there's no point having a piece of software archived if no one can find it! The primary purpose of this paper is to raise awareness to the various needs of archive users and maintainers in the hope that it will provoke both discussion and involvement.

Accessing an Archive

The first thing one must know about an archive is *where* it is! Although this may seem a trivial point, a fair proportion of the messages flying around electronic mail and news networks are of the "Where do I find this?" variety.

The second, and by far the most significant, point is *access* to the archive. The majority of TEX (and other) archives are only accessible via anonymous file transfers; a few, such as Clarkson and Aston, run mail-servers, which interpret requests in incoming electronic mail messages and respond with file transfers, directory listings, and so on. But what of the community without electronic access at all? The author is aware of only four publicized sources of TEX on physical media: the "official" Maria Code distributions for several systems from Stanford, the University of Washington's UNIX TEX tapes, Jon Radel's PC distribution on floppies, and the Aston archive, which will send out VMS, UNIX, PC and Mac kits, as well as a complete tape of the entire archive (one fairly full VMS "backup" tape at 6250 bpi).

However, since it is by far the most common means, let us concentrate on electronic access for a few moments. In the Internet world, the standard means of accessing an archive is by anonymous FTP. This is an interactive program which allows one to "log in" to a remote file system (via the username ANONYMOUS), list directories and copy files back to the local machine. Although FTP implementations do not typically provide facilities for typing files to the screen, it is fairly trivial to do this in practice on most operating systems. Hence, FTP permits reasonably interactive access to an archive.

The UK academic network, JANET, is based on a rather different set of protocols and supports a rather different type of remote file access. When a request is made to copy a file from a remote machine, it is handled *asynchronously:* the file simply "appears" at some time after making the request. JANET style file transfer (NIFTP) means that a user cannot interactively scan through the filestore (directory) of the remote machine, thereby improving security; hence, directory listings have to be left lying around on the archive machine, stored in files with standardized names. These directory listings have to be updated frequently (this is a good habit for an archive, in any case); in the case of the Aston archive, listings of the entire archive and of the recently created files are generated daily.

For cross-network traffic (*e.g.,* between Bitnet and JANET), neither FTP or NIFTP is possible, and one must resort to other approaches. The most common solution is a *mail server,* a program which receives messages containing requests for directory listings, requests to send files to the requestor, and so on. (The author must accept the stigma of having written the mail server program which fronts the Aston archive.)

Finding the Right File

With only these types of access available, finding the file one wants can be a non-trivial exercise. Perhaps the simplest approach is via the "`whereis`" feature supported by several mail servers, one simply sends off a mail message containing, for example, `whereis recipe.sty`. The mail server program then scans through the archive for any files which have this name and returns their locations in a return mail message to the requestor. Of course, this only works if the name of the required file is known, and this is frequently not the case, and this brings us back to the problem of documentation.

If we consider an archive to be analogous to a book, we can think of the listing of the directory hierarchy as the electronic analogue of the table of contents, since it tells us where each file (topic) is located. However, there is no true analogue of a book's *index.* As anyone who has written a book will confirm, compiling the index is a task which cannot satisfactorily be automated. The closest approach to an index is in the hierarchical help systems offered by several operating systems (this most definitely *excludes* the UNIX on-line manual, which is a total abomination).

To be able to support all types of potential archive user, two forms of documentation are needed: hard copy documentation which can be sent out to people without electronic access, and on-line documentation which facilitates locating files by functionality rather than by name. To save the archivist's valuable time, the most sensible approach is to generate both of these from a common source file. Of course, such systems already exist, such as Digital's VAX Document and the Free Software Foundation's "TEXinfo" (a "LATEXinfo" has recently been devised too). Both these products use TEX as the typesetting engine for printed manuals. Document uses an enhanced version of `runoff` for generating VMS Help files, while TEXinfo uses GNU Emacs (a programmable editor) to generate "info" files, which can be scanned with an Emacs subsystem. The problem with info files is that they can only be read from within Emacs, and not all operating systems run it; indeed, many people prefer other editors, even on systems on which Emacs does run. The author has prototyped a similar system and investigated its use for documenting small sections of the Aston archive.

A Prototype Archive Documentation System

In order to evaluate the above approach to generating an archive "index," the author prototyped a documentation system in the `AWK` language. This system had a (deliberately) very restricted set of commands, which covered operations such as titles, lists, font changes (emphasis and teletype) and keyword specification. General descriptions of the contents of several directories (`README` files) were marked up in this way and processed through GNU `AWK` to generate the following types of documentation:

- LATEX input (each `README` file being a separate section of a larger document)
- `nroff` input, to generate UNIX manual pages
- `runoff` input, to generate VMS Help
- `info` format (from the VMS Help, for use with a stand-alone `info` program written by Nelson Beebe)
- plain text, to be inserted into the individual directories as `README` files

The keywords specified in the marked up text were used to generate index entries for the LATEX document and a specific sub-topic (`keywords`) for the Help. A public-domain Help program was then modified to support searches through only the titles (subject headings), through the keywords, or through the bodies of the help modules. The software was tested on a few volunteers who do not know their way around the Aston archive, and it was found that locating specific items in the archive was then much faster, particularly if the choice of keywords was made carefully. Of course, on such a small-scale experiment, it is dangerous to make sweeping conclusions; nevertheless, user response was favourable.

As a further experiment, a sub-section of each entry also contained a list of the relevant files in the archive, and this was used to automatically generate a set of (NIFTP) file-transfer commands to retrieve from the archive all the files needed to get the particular piece of software working. Alternatively, the same information could be used to prepare a printed request to the archive maintainers to specify the files on magnetic media.

A Real Archive Documentation System

While the prototype outlined above could not be used in the real world, it does indicate a practicable way of documenting an archive. However, one can take the idea a step further. There is no reason why the Help system which one uses to scan through the archive's contents should reside only on the archive machine. Given a machine-portable Help program, one could arrange that its databases (i.e., the descriptions of various archives' contents) could be kept up-to-date automatically. This could be achieved either by updating the local databases on (say) a weekly basis by automatic jobs requesting modifications, or by "registering" the local system with the archive and having it send out modifications as and when required. (The latter approach is similar to the one used in the UK for distributing the database of network names and addresses for certain types of computer, while the latter is similar in concept to the distribution mechanism for network news.)

There is no reason why the above scheme should be limited to an archive of TEX-related material. What it would require would be the willingness of archive maintainers *and* people making submissions to archives to settle on a common but painless mark-up format for `README` files, and the standardisation of a Help program across several platforms.