Macros

Eplain 3: Expanded plain TEX

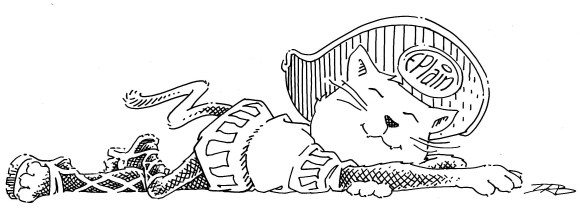
Karl Berry and Oleg Katsitadze

Abstract

Eplain is a macro package which extends the definitions in plain TEX. It provides many conveniences which most document writers need, but without forcing any specific typographic style on an author. Since its inception around 1990, Eplain has provided facilities for citations, tables of contents, symbolic cross-references, indexing, and multiple column output.

As of version 3, Eplain also provides support for live hyperlinks in PDF documents and for loading conveniently a few LATEX packages (notably `graphics.sty` for including images and rotating and scaling text, `color.sty` for colored text, and `url.sty` for typesetting paths and URLs).

This article gives a brief introduction to Eplain for newcomers to the package and then discusses some of the new features in more detail.



Introduction

The original plain TEX is more or less a low-level interface to TEX primitives (while providing a few user-oriented macros). It lacks many features which most document writers will reasonably expect, implementation of which requires some experience in TEX macro programming language. A good example is using labels for cross-references. Instead of manually inserting absolute numbers throughout the manuscript, authors like to assign labels to various parts of the document such as sections, figures, etc. When later TEX encounters a cross-reference with a label, it automatically changes the label to the appropriate number. This saves lots of manual work when parts of the document are reordered.

Such features are available in LATEX (originated by Leslie Lamport), but the approach taken by LATEX is much different from that of plain TEX. LATEX hides many details by providing users with high-level capabilities, making it easier to write documents with predefined styles and harder to produce bad typesetting. A vast number of additional packages has been developed to allow users to customize LATEX, but desired changes can still sometimes be hard to accomplish.

The Eplain macro package takes another approach. The philosophy of Eplain is to provide functionality which can be used (or not used) as desired, without forcing any typographic style on an author. Thus, the Eplain macro package expands on and extends the definitions in plain TEX, providing features such as symbolic cross-referencing, lists, citations, indexing and many other capabilities. Eplain provides both macros intended to be used directly in documents and macros to be used as tools in developing formats.

Eplain is extensively documented and has an active mailing list, so problems or questions related to using Eplain are usually quickly resolved. (One new user's experience with Eplain is reported in [6].) Also, the development sources are publicly available via `http://sarovar.org/projects/eplain`, so contributing is straightforward. For additional information about Eplain, please visit the Eplain home page, `http://tug.org/eplain`.

Eplain at a glance

The simplest way to use Eplain is to put the line

```
\input eplain
```

at or near the beginning of your TEX document. You then need to process your document with the plain-based `tex` (or `pdftex`, `etex`, etc.), *not* `latex`.

The `eplain.tex` file should already be included in your TEX installation. If not, you can get it from the Eplain home page or from CTAN [1].

Now let's take a quick tour of some of the features provided by Eplain. More detailed discussion can be found in the Eplain manual [2].

Multiple columns. Eplain provides the high-level commands `\doublecolumns`, `\triplecolumns`, and `\quadcolumns` to start multiple-column output. Single column output can then be restored with `\singlecolumn`. These macros do not start a new page, so you can have a multiple column passage in the middle of a page.

Displays. To obtain left-aligned math displays, Eplain provides the command `\leftdisplays`. The command `\centereddisplays` switches back to the default of centered displays, should you need to.

Lists. Eplain provides for arbitrarily nested lists that can be either numbered (`\numberedlist ... \endnumberedlist`) or bulleted (`\unorderedlist ... \endunorderedlist`). In both kinds of lists, you begin an item with `\li`, which accepts a cross-reference label as an optional argument. As with practically everything else, Eplain provides many parameters for customizing the lists.

Checking for PDF output. Eplain incorporates `ifpdf.sty`, written by Heiko Oberdiek, which provides an `\ifpdf` switch for detecting whether PDF or DVI is being emitted by TEX.

Verbatim. Eplain supports both in-line verbatim text (with the construct `\verbatim` ⟨*verbatim text*⟩ `|endverbatim`) and typesetting the contents of an entire file verbatim (`\listing{`⟨*filename*⟩`}`). The verbatim listing produced by `\listing` can have optional line numbers in a customizable format.

Footnotes. Eplain extends the definitions of plain TEX to support automatically numbered footnotes (`\numberedfootnote`) and to allow general customization of footnote spacing, rules, etc.

Arrow theoretic diagrams. Eplain incorporates macros (written by Steven Smith) for drawing commutative diagrams. This capability is similar to that found in LATEX's picture mode for drawing slanted lines and vectors of certain directions, and depends upon the LATEX font `line10`.

Cross-references. Referring readers to other parts of your document is a basic need for authors; but putting literal page, section, equation, or whatever numbers in the text is of course undesirable.

Eplain therefore supports symbolic cross-referencing, both generically (with `\definexref` to define labels, and `\ref` and its variants to refer to them) and specifically to page numbers (`\xrdef` and `\xref`) and (sub)equations (`\eqdef`, `\eqsubdef`, `\eqref` and variants).

Citations. The citation macros provided by Eplain are designed to work with the BIBTEX program, written by Oren Patashnik. The macros are defined in a separate source file, `btxmac.tex` (which can also be used on its own, without the rest of Eplain).

Citations are typeset with the `\cite{`⟨*label*⟩`}` command. The actual bibliography is produced with the `\bibliography` command (which reads the requested `.bib` files), and the bibliography style can be selected by `\bibliographystyle`. Many macros and parameters are provided for fine-tuning the formatting of citations and the bibliography.

Contents. Producing a table of contents that is both useful and aesthetic is one of the most difficult design problems in any work. Therefore Eplain does not attempt to solve the design problem; instead, it merely provides helper macros (`\writetocentry` and `\writenumberedtocentry`) for collecting the raw data for a table of contents, using an auxiliary file with extension `.toc` (and the same base name as your document). During the next run of TEX on the document, the collected information is read at the place(s) where you call the Eplain-provided `\readtocfile` to produce the table of contents.

To obtain a nicely typeset table of contents you will need to define one macro per contents entry type (chapters, sections, etc.), to specify the styles of the entry types.

This functionality can be reused for other kinds of contents lists, such as lists of figures and lists of tables. Invoking `\definecontentsfile{`⟨*abbrev*⟩`}` creates a set of the macros with which you can manage your own contents lists, with ⟨*abbrev*⟩ replacing `toc` in the macro names and file name extension of the intermediate file.

Indexing. Eplain provides macros to produce raw material for an index in the form accepted by the MakeIndex program, and to typeset the sorted index produced by MakeIndex.

Besides specifying the basic index entries, the indexing commands facilitate indexing of people's names, creating subentries, "see ..." and "see also ..." entries, page ranges, and entries with page numbers set in a different style (italicized, underlined, etc.). Eplain also supports specifying explicit sort strings; for example, in mathematics, it is usually desirable to typeset π with `$\pi$` and sort it as `pi`.

Setting `\indexproofingtrue` instructs Eplain to typeset index terms in the margin of each page, for help when proofreading.

Programming definitions. Numerous helper definitions turned out to be useful when implementing Eplain's user-level features. Many of these are documented and available to Eplain users on the chance that people writing other macros will also find them useful. Among them are: "inner" variants of `\newcount` and friends which can be called inside other macros; a `\for` macro for iteration over a comma-separated list of items; macros for defining general hooks and properties and named environments; macros for managing auxiliary files and manipulating character category codes; and many others.

Miscellaneous. Eplain also provides nifty typesetting conveniences such as rules with adjustable default width, height and depth; solid and unfilled boxes of specified dimensions, and boxed text; the time and date in various formats; fractions; long pathnames, electronic mail addresses and URLs (this support comes from `path.sty`, written by Nelson Beebe and Philip Taylor); and various TEX-related logos (from `texnames.sty`, compiled by Nelson Beebe). Finally, when things inevitably go wrong, there are macros for tuning the diagnostic output.

LATEX packages in (E)plain

In version 3.0, which was released in September 2005, Eplain acquired the capability to load *some* LATEX packages. You may find this feature of Eplain useful when working with plain TEX, even if you do not plan to use the rest of Eplain. Eplain uses David Carlisle's `miniltx.tex` [4] for this (described below), with extensions to support package options and a few other features.

Of course, most LATEX packages don't make sense under plain TEX; the overwhelming majority of LATEX packages that have been developed can't be used with Eplain, or plain TEX in general. However, some packages provide general capabilities which in principle are independent of the LATEX engine.

The `graphics` bundle is a notable example; it provides for graphics inclusion and also rotating, scaling and coloring of text. These features are not provided by TEX itself; instead the packages must rely on the capabilities of the output driver (typically for PostScript or PDF) to do the job.

These features are just as useful in plain TEX as they are in LATEX, but instead of rewriting all the packages in the `graphics` bundle for plain TEX, the LATEX3 team developed `miniltx.tex`, a "mini-LATEX environment", which provides stubs for or simplified parts of LATEX used by the packages in the `graphics` bundle, so that those packages can be loaded in plain TEX after loading `miniltx.tex`.

The definitions in `miniltx.tex` were designed with the `graphics` packages in mind; therefore these definitions are generally not sufficient for loading other packages. Furthermore, `miniltx.tex` provides no support for specifying package options. So Eplain builds on top of `miniltx.tex` to support package options and a few additional packages.

Loading LATEX packages. To load a LATEX package in Eplain, call `\usepackage` (the same name is used in LATEX), wrapped in a `\beginpackages ... \endpackages` block. This block serves as a substitute for LATEX's preamble, so it is best to specify only one such block per TEX job. For example:

```
\beginpackages
  \usepackage{graphicx,color}
  \usepackage{url}
\endpackages
```

will load the `graphicx`, `color` and `url` packages, with no options.

The following LATEX packages (all on CTAN) are known to work under Eplain:

- `autopict` (LATEX picture mode);
- `color` (color support);
- `graphics`, `graphicx` (graphics inclusion);
- `psfrag` (overlay LATEX onto EPS figures);
- `url` (smart line breaking for URLs).

We hope to support other packages in the future.

Implementation. For those who may be interested in the TEXnical details, here is an overview of the extensions made by Eplain to `miniltx.tex`.

- We redefine the `\DeclareOption` macro (a no-op in `miniltx.tex`) to save the code implementing the option. Next, we redefine the `\ProcessOptions` and `\ExecuteOptions` macros to execute the code which enables relevant options (they are defined as no-ops by `miniltx.tex`). Also, we implement the star ('`*`') option to declare a default option handler, used to process undeclared options.
- We define `\PassOptionsToPackage` (missing in `miniltx.tex`) to let packages pass options to other packages which they load.
- We define `\AtBeginDocument` to accumulate its arguments for execution at the end of the `\beginpackages ... \endpackages` block, instead of the immediate execution defined by `miniltx.tex`. Also, `\AtEndOfPackage` (missing from `miniltx.tex`) is defined to delay execution of the argument until after the current package is loaded.
- In `\ProvidesFile` and `\ProvidesPackage`, we define the macros `\ver@`⟨*package*⟩`.sty` and `\ver@`⟨*filename*⟩`.`⟨*ext*⟩; these are used by some packages to detect that a package or a file has been loaded. (When a package is requested which has already been loaded, Eplain currently displays an error message; it does no checking for the legitimate situation of a package having already been loaded with a superset of the options in the second request.)

- `\ProvidesPackage` also checks that the date of the package is not older than the date specified by the user in `\usepackage`.
- `\RequirePackage` is redefined to save all extant parameters before loading other packages, and restore them afterwards. That way, if a recursively loaded package loads other packages, or defines its own options or `\AtEndOfPackage` commands, they do not interfere with the state of the upper-level package.

Hyperlinks

Another feature of Eplain which is new in version 3 is the ability to create live hyperlinks in PDF documents through pdfTeX or dvipdfm(x).

Hyperlink drivers. Since, in addition to pdfTeX, there are several `.dvi` processors with the ability to generate PDF files with hyperlinks, the hyperlink support in Eplain has a two-layered structure: 1) hyperlink drivers, which provide low-level hyperlink commands (primitives in the case of pdfTeX and `\special` commands in case of `.dvi` processors); and 2) user commands, which are the same across the drivers (but supporting different subsets of functionality, depending on the driver's capabilities).

Currently, Eplain has two drivers: `pdftex` for pdfTeX, and `dvipdfm` for dvipdfm and dvipdfmx. We hope to add more drivers in the future.

One other pseudo-driver named `nolinks` is beneficial when one wishes to typeset a document both with and without hyperlinks.[1]

Eplain comes with the hyperlink support disabled by default (for various reasons). To enable hyperlinks you specify:

`\enablehyperlinks[`⟨*driver*⟩`]`

after you have loaded Eplain in your document. If the optional argument, `[`⟨*driver*⟩`]`, is omitted, Eplain tries to detect the appropriate driver.

Implicit hyperlinks. When hyperlinks are enabled, many Eplain macros automatically start to use hyperlinks in their output. For example, cross-reference macros then render the cross-reference as a hyperlink pointing to the location being referenced. Here is a list of features which create such implicit hyperlinks:

- `\url` (from `url.sty`, see previous section);
- cross-reference macros;
- BibTeX citations;
- numbered and unnumbered lists;
- indexing;
- footnotes.

All macros which create implicit hyperlinks are assigned to one of the so-called *hyperlink groups*, roughly corresponding to the above features, so that parameters such as link border width or colors can be set individually for each group. For example, all equation reference macros are assigned to the 'eq' hyperlink group; thus, you can customize parameters for equation hyperlinks without affecting other kinds.

Explicit hyperlinks. Sometimes you might need to create a hyperlink explicitly. This is done by first creating a hyperlink destination with the command

`\hldest{`⟨*type*⟩`}{`⟨*options*⟩`}{`⟨*label*⟩`}`

Supported destination types and options depend on the selected hyperlink driver, while ⟨*label*⟩ identifies the destination.

Now, to create a link to that destination, use:

```
\hlstart{⟨type⟩}{⟨options⟩}{⟨label⟩}
...
\hlend
```

Whatever text you write in the `...` becomes a hyperlink pointing to the destination identified by ⟨*label*⟩. Here again, available link types and options depend on the selected hyperlink driver.

Eplain provides a way to set default destination and link types and options, so that you don't need to specify the same parameters over and over in each call.

Implementation. Our first attempt at implementing hyperlinks was to write wrappers around relevant Eplain macros, extending them with hyperlinking capabilities. Although this was functionally implemented and even used to typeset an electronic book, coding of the wrappers turned out to be quite difficult, and adapting the wrappers for other

[1] Here is the TeXnical rationale for the `nolinks` driver (feel free to skip). When a hyperlink is inserted, TeX creates a *whatsit* (an internal TeX object). Whatsits may introduce legitimate breakpoints at places where none would exist otherwise. Now imagine that you want to generate a PDF document both with and without hyperlinks. Completely disabling the hyperlinks for the latter is not ideal, because then the whatsits will not be generated and the resulting PDF may end up with different page and line breaking than the former. Therefore, it is best to keep hyperlinks enabled, while selecting the `nolinks` driver. This defines all hyperlink commands to produce a whatsit that does nothing (writes to a log file), thus imitating the whatsits from the "real" hyperlink commands. (This trick was borrowed from `color.sty`.)

projects would not have been an easy task. Even worse, many wrappers were fragile, in the sense that they were greatly relying on the internals of Eplain macro definitions; thus, they would have been hard to maintain in future versions.

The logical solution was to add the hyperlink capability directly into Eplain macros. Let's look briefly at the hyperlink implementation in Eplain.

Hyperlink macros in Eplain are structured so that it is relatively easy to add support for new hyperlink engines by writing a new driver. A new driver can be modeled on the existing ones, and, in short, should define macros with certain names so that the driver-independent hyperlink macros can detect the driver, plus define link and destination handlers for each of the types it supports.

Each driver defines default destination and link types and default values for the supported options, to use in the absence of user-specified values in a call to `\hldest` or `\hlstart`. Of course, these defaults can be overridden by the user. In addition, the user can set default options and types for each of the link and destination groups, which, when set, will override the global defaults.

Maintaining default options for destination and link groups is a little tricky. We don't want to define a macro per group per option to hold the value, because a lot of TEX's memory would be wasted just storing the names of those macros. Instead, for each destination or link group, a list of default options given by the user is saved as a comma-separated list of assignments in the form ⟨*option*⟩=⟨*value*⟩. This list is consulted whenever a macro from the group needs to create a link or destination. If any option is missing from this list, a global default for this option (defined by the driver or specified by the user) is used.

An annotated example

Suppose we are using pdfTEX, have a figure we want to insert, scaled to fit our format, and we want to refer to this figure from the text via a cross-reference label. Also, we want to typeset a URL as a live hyperlink and a line of colored text.

Let's see how this can be done in Eplain. Consider the source file `eplndemo.tex` shown in Example 1, along with its output. Numbers at the beginning of the lines are not part of `eplndemo.tex`, they are merely to ease references in the comments to follow.

In order to compile `eplndemo.tex`, you will need the CTAN lion drawing by Duane Bibby, which you can download from [3]. Place the image

```
1. \input eplain
2.
3. \beginpackages
4.   \usepackage{url}[2005/06/27]
5.   \usepackage[dvipsnames]{color}
6.   \usepackage{graphicx}
7. \endpackages
8.
9. \enablehyperlinks
10. \hlopts{bwidth=0}
11. \hlopts[url]{colormodel=named,%
12.              color=OliveGreen}
13. \nopagenumbers
14. \def\figureword{fig.}
15.
16. \vbox{
17.   \definexref{CTANlion}{1}{figure}
18.   \noindent
19.   \includegraphics[width=200pt]
20.                   {ctan_lion_350x350}
21.
22.   \noindent Figure~1: lion in the archives.
23. }
24. \medskip
25.
26. See the lion in \ref{CTANlion}.
27.
28. Take me to \url{http://tug.org/eplain}.
29.
30. {\color[rgb]{0.3,0.3,0.3} Paint it gray.}
31.
32. \bye
```

Figure 1: lion in the archives.

See the lion in fig. 1.
Take me to http://tug.org/eplain.
Paint it gray.

Example 1: Eplain source file and output.

file in the same directory with `eplndemo.tex`, and change to that directory. Now, to produce a PDF, run pdfTEX (twice):

```
pdftex eplndemo.tex
pdftex eplndemo.tex
```

During the first run, Eplain will write the information about the cross-reference into `eplndemo.aux`, and during the second run this information will be read by Eplain to typeset the reference.

Now, let's see what all these commands mean.

- On line 1, we load Eplain.
- On lines 3–7, we load three LATEX packages.
  - `url.sty` provides the `\url` command to conveniently typeset a URL. We request that `url.sty` be the version from June 27, 2005, or later, because earlier versions had problems interacting with plain TEX.
  - `color.sty` provides support for colored text; all hyperlinks are automatically colored by Eplain when this package is loaded. We give the `dvipsnames` option because we want to use named colors from the `dvips` graphics driver.
  - Finally, we load `graphicx.sty`, for the macro `\includegraphics`.
- Recall that hyperlinks are off by default. Therefore, we enable them on line 9.
- On lines 10–12, we customize some hyperlink options.
  - First, we set the border width to 0 for all links, to omit the default boxes around links (we prefer colored links).
  - Next, we specify that all links in the `url` hyperlink group (meaning the `\url` command from `url.sty`) should be colored using the named color `OliveGreen`. The default is the dark red shown in the `\ref` output.
  - The `%` at the end of line 11 prevents TEX from converting the end-of-line character into a space token. (We have short source lines simply because of *TUGboat*'s narrow columns.)
- On line 13, we inhibit page numbering for this one-page document. (A plain TEX command.)
- On line 14, we define the output word for the cross-reference class `figure`. This word will be prepended by Eplain to references created via `\ref` (read on to see its use).
- Now comes the fun part! On lines 16–23, we create the figure with the CTAN lion image.
- We start by defining a symbolic label so that we can later refer to the figure with `\ref`. The command `\definexref` on line 17 takes the following arguments:
  - The cross-reference label (`CTANlion`).
  - The reference text (`1`, the figure number). If you have many figures, you may want to use a count register to keep track of the current figure number, and to use its value here as the reference text.
  - The class of the label (`figure`). The label class determines the word placed by `\ref` in front of the reference text; recall that we've defined `\figureword` on line 14. (There is nothing magic about the name `figure`; Eplain just uses whatever is defined.)

  In addition to the cross-reference label, `\definexref` creates a hyperlink destination with the same label.
- On lines 18–20, we use the `\includegraphics` command (from the package `graphicx.sty` [5]) to load the image, scaled to the width of 200 pt, and place it in a paragraph of its own, without indentation (`\noindent`).
- On line 22, we put the figure's caption underneath the image.
- Now let's typeset some hyperlinks. First, we use the `\ref` cross-reference command (line 26) to refer to the figure using our chosen cross-reference label (`CTANlion`). Eplain automatically inserts the label's class word (`fig.`, defined on line 14) as part of the link (to make sure the reader does not have to aim too hard).
- Let's now point somewhere outside our document; the Eplain home page seems a good target. On line 28, we use the `\url` command from `url.sty`. Remember that we have customized the color of `url` hyperlinks on lines 11 and 12, so it will be rendered with a color different from the default dark red.
- Finally, we produce a line of colored text on line 30. We use the `\color` command from `color.sty`, specifying an exact RGB color `0.3,0.3,0.3` (which results in gray). To limit the effect of the color command to this single line, we enclose the command and the text in a group.
- On line 32, we say good-bye to TEX.

Additional example documents are available from `http://tug.org/eplain/demo`.

Summary

Eplain tries to make plain TEX a more user-friendly environment while at the same time allowing the user to retain complete control. Thus, especially if you are familiar with plain TEX, Eplain may be able to save you time and effort in typesetting anything from whole books where you need low-level control to short documents where you need just one or two document-level features.

We appreciate any and all bug reports, suggestions for enhancement, and offers of help; please write via the Eplain mailing list, http://tug.org/mailman/listinfo/tex-eplain.

Acknowledgements

Thanks to Dave Walden, Steve Peter, Greg Black, and Barbara Beeton for reading drafts of this article and substantially improving it. Also thanks to Duane Bibby for the Eplain lion (named TEXimilian; more details and downloadable images are on the Eplain home page), and to Duane and CTAN for the CTAN lion drawing.

References

[1] Berry, Karl, Oleg Katsitadze, and Steven Smith. Eplain. http://www.ctan.org/tex-archive/macros/eplain

[2] Berry, Karl, Oleg Katsitadze, and Steven Smith. *Eplain: Expanded Plain TEX*, 1990–2005. http://tug.org/eplain/doc/eplain (online version) http://www.lulu.com/content/113810 (paper version)

[3] Bibby, Duane. CTAN lion drawing. ftp://tug.ctan.org/ftpmaint/CTAN_lion/ctan_lion_350x350.png

[4] Carlisle, David. Mini-LATEX. http://www.ctan.org/tex-archive/macros/plain/graphics/miniltx.tex

[5] Carlisle, David, et al. Graphics bundle. http://www.ctan.org/tex-archive/macros/latex/required/graphics

[6] Walden, David. Travels in TEX Land: [...] plain TEX with Eplain, [...]. *The PracTEX Journal*, 2005-4. http://tug.org/pracjourn/2005-4/walden

◇ Karl Berry
88609 Wickizer Lane
Bandon, OR 97411
USA

◇ Oleg Katsitadze
24 Pushkin St., apt. 10
Simferopol 95011
Ukraine
eplain (at) tug.org