

---

## The L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> TUGboat Macros

Robin Fairbairns & TUGboat editors

### 1 Introduction

This is the documentation for the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> macros to be used by *TUGboat* authors. The macros represent a development of the earlier *ltugboat* and *ltugproc* styles that were written for use with L<sup>A</sup>T<sub>E</sub>X 2.09; major contributors have been Robin Fairbairns, Sebastian Rahtz, Michel Goossens, Nico Poppelier and Johannes Braams. Many others have been involved, including Barbara Beeton, Karl Berry, Mimi Burbank, and the L<sup>A</sup>T<sub>E</sub>X3 team.

### 2 Availability

*TUGboat*'s web pages are at:

<http://tug.org/TUGboat>

They include an article template, information for authors and reviewers, all the back issues, and more.

The macros are released for general use, and are distributed via CTAN (directory `macros/latex/contrib/tugboat`) in the usual L<sup>A</sup>T<sub>E</sub>X way as files `tugboat.dtx` and `tugboat.ins`. When the `.ins` file is processed by L<sup>A</sup>T<sub>E</sub>X, the files `ltugboat.cls` and `ltugbib.bst` (for use with articles) and `ltugcomm.sty` (a cooking pot of perhaps-useful macros, for documentation, etc.) are produced. (`ltugproc.cls` is also produced for compatibility, but no longer used for proceedings or anything else.)

The `.dtx` file may itself be processed by L<sup>A</sup>T<sub>E</sub>X to produce a formatted (somewhat 'literate') source listing for those who would like more detailed descriptions of the *TUGboat* macros.

### 3 The general structure of a paper

The basic idea is to start your L<sup>A</sup>T<sub>E</sub>X document with `\documentclass{ltugboat}`, which defines the appearance of *TUGboat* articles. This uses the file `ltugboat.cls` as usual.

Each paper, therefore, is written as a document that may stand on its own. It starts with a `\documentclass` command, and its body is enclosed in a `document` environment. There are some options to the document class, described in the next section, but ordinarily the author needn't bother with them. The defaults are designed for creating proof copies of papers.

The proof output differs from the final production output with respect to page numbers and other material. The changes required for final production are the responsibility of the *TUGboat* editors, and the author need not be concerned with them.

### 4 Class options: The *ltugboat* class

The *ltugboat* class accepts many of the options of the *article* class (it suppresses the font-size selection, one/two-side and one/two-column options).

**draft** Set up for a draft copy of a paper (this is the default setting—the author need not explicitly set it): page numbering starts at a high number, black marks for overfull boxes.

**extralabel** Use the extra label-distinguishing mark in the body of the reference; see section 12.

**final** Set up for the final copy of a paper: page numbering to come from elsewhere, no black marks.

**harvardcite** Specify Harvard-style citation (not especially recommended in general, though they are used in the present document); see section 12.

**noextralabel** Don't use the extra label-distinguishing mark in the body of the reference; see section 12.

**nonumber** Sections are not to be numbered; section heading layout is to be as in the 'plain' *tugboat* styles.

**numbersec** Sections, subsections and subsubsections are to be numbered (this is the default setting—the author need not explicitly set it).

**preprint** Set up for a preprint.

**rawcite** Specify default (unnumbered) citation; see section 12.

Again, normally there is no need to use any document options.

### 5 Command syntax

We would have liked to offer perfectly uniform syntax for people to use when preparing their papers. Unfortunately, uniform syntax is not available with any widely-available set of macros (though see, for example, the discussions in Ogawa, 1994, Baxter, 1994, and Swift, 1995). In the circumstances, we have sought simply to keep to the spirit of Lamport's (1994), as modified<sup>1</sup> by the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> work (see, for example, Rowley, 1994).

In the few cases that it has proved possible to emulate (what seems to a staid old L<sup>A</sup>T<sub>E</sub>X programmer, such as the present author) the gay abandon of the syntax of the 'plain' *tugboat* styles (Whitney and Beeton, 1989), we have done. Nevertheless, on the whole, the new *ltugboat* macros simply define L<sup>A</sup>T<sub>E</sub>X commands and environments, or modify the definitions of L<sup>A</sup>T<sub>E</sub>X 'standard' commands. Section 13 lists equivalences between macros defined by the 'plain' package and those defined by the new package.

---

<sup>1</sup> The cited edition of Lamport's book documents the modified version of L<sup>A</sup>T<sub>E</sub>X, but it's worth emphasising that we use the modified version as the reference.

The ‘down’ side of this decision is, of course, the ‘welter of L<sup>A</sup>T<sub>E</sub>X braces’ that Barbara Beeton has been heard to complain of in production team discussions. One has to hope that the (near) uniformity of syntax offers those who think like Barbara some small recompense!

## 6 Divisions of the paper

Papers in *TUGboat* may be subdivided in the normal way of a L<sup>A</sup>T<sub>E</sub>X article (the classes are defined in terms of L<sup>A</sup>T<sub>E</sub>X’s article class). Thus the author may use `\section`, `\subsection`, ..., `\paragraph` commands (but `\part` and `\subparagraph` from article are suppressed, and `\chapter`, which doesn’t even appear in the parent class, receives the same treatment).

Authors should note that the style of ordinary issues of *TUGboat* makes no distinction between the titles of the divisions; the visual style relies on the section numbers to indicate where the divisions lie in the hierarchy. As a result, the un-numbered ‘\*’ forms of the `\section`, etc., commands, are inappropriate. The *ltugboat* class therefore warns the author who (possibly inadvertently) uses one of these forms. (The author who wishes to use un-numbered sections throughout her paper may use the `nonumber` class option (see 4).

Reference may, however, be made to the ‘title’ of divisions of the paper, whether they are numbered or not. The `\nameref` command (which uses the technique developed for the `hyperref` package (Rahtz and Oberdiek, 1997)) permits such references; for example, the present section was introduced by:

```
\section{Divisions of the paper}
\label{sec:divs-paper}
```

and the command `\nameref{sec:divs-paper}` produces ‘Divisions of the paper’.

### 6.1 Abstracts

The classes make provision for abstracts, but the provision is different for the two classes.

The *ltugboat* class provides two environments, `abstract` and `longabstract`. The `abstract` environment simply typesets its body as an un-numbered section whose title is ‘Abstract’. The `longabstract` environment typesets its body in small text, and separates the abstract from the rest of the paper with a decorative line.

### 6.2 Appendices

A paper may have appendices, which are expressed in exactly the same way as they would be in the L<sup>A</sup>T<sub>E</sub>X article class:

```
\appendix
```

```
\section{This is appendix A}
```

```
...
```

```
\section{This is appendix B}
```

Which will produce ‘section’ headings similar to:

## A This is appendix A

*TUGboat* articles may have a small extension to this format: this extension was originally developed for proceedings of past years, but is also available in normal issues:

```
\begin{appendix}
\section{This is the first one}
...
\end{appendix}
```

Which will produce ‘section’ headings similar to:

## Appendix A This is the first one

In both cases, the subsections are numbered as normal (i.e., as ‘A.n’ in normal *TUGboat* papers):

## 7 Titles, addresses and so on

The title and author(s) of a paper are quoted using commands that are familiar (in syntax, at least) to most L<sup>A</sup>T<sub>E</sub>X users; the `\title` command is exactly that used in the standard L<sup>A</sup>T<sub>E</sub>X classes. There is also `\shortTitle{<your-short-title>}` to define the form used in running heads or footers; similarly `\shortAuthor`.

The `\author` command is used once for each co-author of the paper, and for each `\author` there should be a `\address` command that gives a (postal) correspondence address. In addition (wherever possible), *TUGboat* likes to quote an email address for authors: for this, the `\netaddress` command is used. Finally, each author may advertise a ‘home’ Web page, using a `\personalURL` command.

For example, the present paper has at its start:

```
\title{The LATEX \TUB{} Macros}
\author{TUGboat editors}
\EDITORnoaddress
\netaddress{tugboat@tug.org}
\personalURL{http://tug.org/TUGboat}
\maketitle
```

Lines in the title information can get quite long. If the information being given is to be typeset as ordinary text (as in the case of the `\address` line above), it can be ‘wrapped’ perfectly happily, as in normal text. If one of the verbatim items (`\netaddress` or `\personalURL` commands) is going to be too wide for the column, what is the author to do? (Abbreviating the text, as in the `\personalURL` above, is *not* usually an acceptable option!) Unfortunately, the % sign is an entirely acceptable element of both email addresses and URLs, so that the normal ‘fall-back’ isn’t available. Therefore, the classes typeset these

electronic addresses in an environment where some of the characters (notably ‘.’ and ‘/’) are treated as word-divisions for the purposes of laying out the line.

If the paper is the result of more than one author’s labours, a sequence of `\author`, `\address`, `\netaddress` and `\personalURL` commands may be given, as in the following, which comes from a paper given at TUG’95 (abbreviated):

```
\author{Michel Goossens}
\address{CN Division, CERN\
...}
\netaddress{...}

\author{Sebastian Rahtz}
\address{Elsevier Science Ltd\
...}
\netaddress{...}

\author{Robin Fairbairns}
\address{University of Cambridge
Computer Laboratory\
...}
\netaddress{...}
\personalURL{...}
```

The class files will take care of arranging author names and addresses between the `\maketitle` and (possibly) `\makesignature` commands.

## 7.1 Compilation articles

Compilation articles are built from a set of contributed parts, and are under the general (sub-) editorship of the author<sup>2</sup> of the article. The author of the article is presented (using `\author` and suchlike commands) in the usual way, and writes the introductory text. Each contributors’ part then follows. The contributor’s name is quoted in the `\contributor` command, which is an analogue of the `\author` command; contributors’ `\address`, `\netaddress` or `\personalURL`. The `\contributor` command opens a group in which the contribution appears, and the contributor’s signature (produced with a `\makesignature` command) closes the group. The general scheme looks like:

```
\title{Example compilation article}
\author{Robin Fairbairns}
\address{University of Cambridge ...}
\netaddress{...}
...
introductory text
...
\makesignature
```

<sup>2</sup> Or authors: there’s no reason in particular that compilation articles should not be put together by more than one person.

```
\contributor{Betsy the Dog}
\address{Romsey Town, Cambridge}
...
Betsy’s contribution
...
\makesignature
...
```

## 8 Verbatim text

The classes do not at present provide the same wide range of facilities as the ‘plain’ *tugboat* style (Whitney and Beeton, 1989); the author had hoped to ‘borrow’ facilities from a package which is believed to be in development, but in the event that package has not materialised.

For in-line verbatim text, authors should ordinarily employ the facilities of L<sup>A</sup>T<sub>E</sub>X itself (the `\verb` macro). This macro, of course, is highly restricted as to its usage (primarily, that it may not appear in the argument of *any* other macro, even `\footnote`).

For ‘display verbatim’ (to employ the term used by Whitney and Beeton), the classes add a small increment to the functionality of L<sup>A</sup>T<sub>E</sub>X’s *verbatim* environment, by introducing an optional argument. The optional argument may contain commands to be executed before starting the verbatim text; the set of commands which have useful effect is strictly limited, but the following are commonly used:

- Font size selection commands: for example, all the display verbatim in the present paper starts with:  
`\begin{verbatim}[\small]`
- The command `\ruled`, which is available *only* in *verbatim*’s optional argument, and specifies that a column-wide rule should be drawn before and after the verbatim text
- One of the `\make*` commands,<sup>3</sup> which change the category code of characters within the verbatim text. This is (of course) a facility that should only be used with the utmost caution, but it can, for example, be employed to provide interesting effects by knowledgeable authors.

Two caveats about the use of this facility should be noted:

- The search for the optional argument can be confused by the appearance of a `[` character as the first of the displayed verbatim. An author who wishes to start verbatim text with a `[` character should provide an empty optional argument (i.e., simply `[]`) to the *verbatim* environment.

<sup>3</sup> `\makeescape`, `\makebgroup`, `...`, `\makecomment`; used, for example, as `\makeescape\l`

- The facility is lost when certain packages are loaded. An example is the `verbatim` package (Schöpf, 1996), which redefines the `verbatim` environment in its entirety. Of course, any package that loads `verbatim` (such as `moreverb`, Duggan et al., 1997) will necessarily have the same effect. (It should be noted that `verbatim` and `moreverb` provide some of the facilities that are available in the ‘plain’ `tugboat` styles, so that an author could reasonably be tempted to use them. There is no objection in principle to authors using these packages.)

## 9 Floating inserts

The classes do not make any change to L<sup>A</sup>T<sub>E</sub>X’s built-in provision for floating inserts, so that authors may generate figures and tables just as they would in any ‘normal’ L<sup>A</sup>T<sub>E</sub>X document. Figure and table captions, and labels referring to them, are also substantially untouched.

However, since both classes typeset in two columns, authors must distinguish between the `figure` and `table` environments (which produce floats that are the same width as the column) and the `figure*` and `table*` (which produce floats that are the same width as the page).

## 10 Special-purpose typesetting

The classes define a rather large set of commands for special-purpose typesetting. Some of them are available for historical reasons only, and many are only useful in somewhat restricted circumstances. For this reason, the present paper only outlines a representative, small set of the macros.

### 10.1 Acronyms and logos

The classes provide macros that produce ‘correct’ representations of a large number of acronyms and logos; a small representative selection is shown in figure 1. The sample documents at <http://tug.org/TUGboat/location.html> have a more complete list, and of course the class sources are the ultimate reference.

Authors are especially urged to note the `\acro` command, which is defined in the classes. The visual appearance of (mostly) lower-case English text, with interpolated acronyms in the same point size, is generally unpleasing. Therefore, the `\acro` command typesets its argument slightly smaller than it would otherwise appear: compare ‘URL’ (`\acro{URL}`, as used above) with ‘URL’. Many macros that simply generate calls to `\acro` are defined by the classes; two examples, `\CTAN` and `\tug` of the list in figure 1 have already been used in the present paper.

| <i>Macro</i>            | <i>Output</i>                  |
|-------------------------|--------------------------------|
| <code>\ConTeXt</code>   | ConT <sub>E</sub> Xt           |
| <code>\Cplusplus</code> | C++                            |
| <code>\CTAN</code>      | CTAN                           |
| <code>\eTeX</code>      | ε- <sub>T</sub> E <sub>X</sub> |
| <code>\FAQ</code>       | FAQ                            |
| <code>\HTML</code>      | HTML                           |
| <code>\ISBN</code>      | ISBN                           |
| <code>\ISSN</code>      | ISSN                           |
| <code>\MacOSX</code>    | Mac OS X                       |
| <code>\MathML</code>    | MathML                         |
| <code>\MF</code>        | METAFONT                       |
| <code>\MP</code>        | METAPOST                       |
| <code>\NTS</code>       | <i>N</i> 7 <i>S</i>            |
| <code>\OMEGA</code>     | Ω                              |
| <code>\OTP</code>       | ΩTP                            |
| <code>\PDF</code>       | PDF                            |
| <code>\SGML</code>      | SGML                           |
| <code>\TUB</code>       | <i>TUGboat</i>                 |
| <code>\TUG</code>       | T <sub>E</sub> X Users Group   |
| <code>\tug</code>       | TUG                            |
| <code>\XML</code>       | XML                            |

Figure 1: A few of the classes’ acronyms and logos

### 10.2 Other special typesetting

A small list of special typesetting commands follows: a large set of such commands is defined in the classes, but the list covers most of the ‘everyday’ ones.

- `\cs{cmd}` Typeset a control sequence name. (The command `\cs{fred}` is typeset as `\fred`.)
- `\env{environment}` Typeset an environment name as if at the start of the environment. (The command `\env{fred}` is typeset as `\begin{fred}`.)
- `\meta{arg}` Typeset a formal argument name. (The command `\meta{fred}` is typeset as *fred*.)
- `\Dash` Typeset an em-dash, surrounded by thin spaces, only breakable *after* the dash; this is the preferred method of specifying a dash in running text.
- `\dash` Typeset an en-dash, in the same way as `\Dash` does.
- `\nth{n}` Typeset an ordinal number. For example, `\nth{1}` is set as 1<sup>st</sup>, `\nth{27}` is set as 27<sup>th</sup>, and so on.
- `\sfrac{num}{denom}` Typeset a fraction to match running text; for example `\sfrac{3}{4}` is set as 3/4.

## 11 Use of packages

In general, the *TUGboat* team will be sympathetic to authors who wish to use non-standard packages in

their papers; indeed, in a journal devoted to the usage of  $\text{\TeX}$ , the editor would be churlish indeed to refuse such usage. However, the team does need to be able to process the paper on the *TUGboat* production computers, and the author who is slapdash about the way she submits her paper will cause confusion and delay. (Remember that the editorial team is a group of volunteers, each working in his or her spare time.)

In general, packages currently on CTAN, and known to work with *current*  $\text{\LaTeX}$  are unlikely to give problems.

In particular, the team is happy to accept papers using packages that are supported by members of the  $\text{\LaTeX}$ 3 team,<sup>4</sup> subject to the two provisos:

- Use of the `verbatim` package has implications for the `verbatim` facilities provided by the classes — see section 8.
- Use of `babel` almost inevitably implies use of hyphenation patterns that the team may not have installed in their  $\text{\LaTeX}$  format; it is therefore important that the author explains her `babel` configuration to the editorial team. The minimum documentation required is a copy of the author's `language.dat` file, and copies (or CTAN pointers to) any hyphenation files used.

Usage of other packages should always be subject to negotiation with the team. If the team does not have access to a copy of the package, life is going to be very difficult; authors are urged to be sensible in this regard. A sensible mechanism for submitting out-of-the-ordinary packages (as for paper-specific bibliographies) is by use of the `filecontents` environment.

TUG has a policy that macro packages described in *TUGboat* should be available for readers to use. Since typing macros from printed sources is such an error-prone undertaking, authors of publicly available packages are urged to submit their macros to the CTAN archives. If a package is only available under restricted terms, authors are urged to make this fact clear when first submitting an article to the editor.

Some facilities are considered inappropriate to delivery by the *TUGboat* classes, and as a result, the *TUGboat* team recommend certain packages to authors.

At present, the list of recommended packages consists of only two, `mflogo.sty` (Vieth, 1995) and `url.sty` (Arseneau, 1996).

Both classes will load the `mflogo` package if it is present on the author's system; if the package is not

present, the classes will emulate its more important features; the package defines `METAFONT` and `METAPOST` logos using recent versions of Knuth's `logo10` font family.

The `url` package is useful when one is typesetting significant numbers of file names, network addresses or URLs; it is being used in the present paper (not least in the bibliography).

## 12 Bibliography

Short form: our recommendation for handling bibliographies is to use the `BIBTEX` and `plain` bibliography style. No document options are needed or recommended. All that is required in the article source (as shown in the sample available from <http://tug.org/TUGboat>) is the following:

```
\bibliographystyle{plain}
\bibliography{yourbibfile}
```

The rest of this section is about cases where, for whatever reason, you don't want to do that.

Bibliographic citations give much grief to the editorial team. Good publishing practice requires that there be editorial control of the way citations in a journal are presented yet, all too often, authors submit articles whose bibliography is formatted according to their preference. The important rules for authors, then, are that they *shouldn't* supply a `.bbl` file (`BIBTEX` processed output), and that they *shouldn't* write out a `thebibliography` environment, but should rather submit a working `.bib` file with their paper.<sup>5</sup> As with uncommon packages, the `filecontents` environment is a convenient way to deliver the bibliography file.

A special case is the accumulated bibliography of *TUGboat* itself;<sup>6</sup> it is always available to the production team, so that authors may make reference to items from that `.bib` file without further ado.

Notwithstanding the general recommendation for the `plain` `BIBTEX` style, two citation styles are supported within *TUGboat* articles, '`raw`' and '`harvard`' (the present article is employing *harvard* citation). The `raw` citation style uses the 'standard' `BIBTEX` '`plain`' (numeric) citation style; its modification by use of Donald Arseneau's `cite` package<sup>7</sup> is acceptable.

<sup>5</sup> The program `bibextract`, available from CTAN in directory `biblio/bibtex/utils/bibextract`, provides a convenient mechanism for extracting just the relevant portions of your bibliography database for submitting your paper; many of the common `BIBTEX`-database management packages, such as `BIBDB` or `BIBTools` offer similar facilities.

<sup>6</sup> Available on CTAN as `digests/tugboat/biblio/tugboat.bib`

<sup>7</sup> Available on CTAN as `macros/latex/contrib/cite/cite.sty`

<sup>4</sup> Those in the  $\text{\LaTeX}$  base distribution, or one of those on the `macros/latex/required` sub-tree on CTAN.

Raw citation is selected by default (by execution of class option `rawcite`).

Harvard citation may be selected by specifying `harvardcite` as an option of the `\documentclass` command. The macros used derive pretty directly from the ‘harvard’ styles written by Glenn Paulley and now maintained by Peter Williams; the  $\text{\LaTeX}$  style derives from one developed by Patrick Daly.

The basic citation format is ‘author-year’, but the macros are capable of many variations: this in turn places somewhat of a load on the author to use the correct citation macro. The macros available are shown in figure 2; the figure assumes an entry in the bibliography with authors Tom, Dick, and Harry, and with a 1990 date.

| Macro                        | Output   |
|------------------------------|--|
| <code>\cite{key}</code>      | (Tom, Dick, and Harry, 1990)                       |
| <code>\citeA{key}</code>     | (Tom, Dick, and Harry)                             |
| <code>\citeNP{key}</code>    | Tom, Dick, and Harry, 1990                         |
| <code>\citeANP{key}</code>   | Tom, Dick, and Harry                               |
| <code>\citeN{key}</code>     | Tom, Dick, and Harry (1990)                        |
| <code>\shortcite{key}</code> | (Tom et al., 1990)<br>[also has A and NP variants] |
| <code>\citeyear{key}</code>  | (1990)<br>[also has an NP variant]                 |

**Figure 2:** The range of citations in `harvard` style

Note that, if Tom, Dick, and Harry are a prolific team, there can easily be more than one reference to their work in one year. In such a case, the citations will be (Tom, Dick, and Harry, 1990a), (Tom, Dick, and Harry, 1990b), and so on. These extra ‘a’, ‘b’, etc., tags may also appear in the references section of the paper, attached to the year recorded for the reference: whether this indeed happens is controlled by the `extralabel` and `noextralabel` class options. The default state (option `extralabel`) attaches the extra characters.

Bibliographies provide further problems because they’re notoriously difficult to typeset at the best of times.  $\text{\LaTeX}$  sets `\sloppy` when typesetting the bibliography, but this can often lead to pretty unpleasant output in the narrow columns typical of *TUGboat*. The author may control the typesetting using the command `\SetBibJustification`. The classes set `\sloppy`, by default (just like  $\text{\LaTeX}$ ), but the author may (for example) say:

```
\SetBibJustification{\raggedright}
```

as the present article does, to achieve somewhat better results.

### 13 Equivalences between the ‘plain’ and the $\text{\LaTeX}$ packages

A good proportion of the commands in the ‘plain’ packages also appear (with the same meaning) in the  $\text{\LaTeX}$  classes. Figure 3 gives a brief summary of where the macros differ significantly.

| Plain macro              | $\text{\LaTeX}$ macro   |
|--------------------------|---|
| <code>\head</code>       | <code>\section</code>   |
| <code>\subhead</code>    | <code>\subsection</code>  |
| <code>\subsubhead</code> | <code>\subsubsection</code>   |
| <code>\list</code>       | <code>itemize</code> , <code>enumerate</code> , etc.,<br>environments |
| <code>\verbatim</code>   | <code>verbatim</code> or <code>\verb</code>                           |
| <code>\figure</code>     | <code>figure</code> or <code>figure*</code> environments              |

**Figure 3:** Equivalences between plain and  $\text{\LaTeX}$  macros

$\text{\LaTeX}$  itself makes comprehensive provision for lists; the *TUGboat* classes make no attempt to emulate the list facilities of the ‘plain’ macros.

The ‘plain’ styles’ provision for verbatim text is also somewhat different from the  $\text{\LaTeX}$  approach; the *TUGboat* classes offer a small subset of the extra facilities that the ‘plain’ styles provide; for more elaborate facilities, the user is referred to the `verbatim` and `moreverb` packages (see section 8).

Of course, the syntax of commands given to the  $\text{\LaTeX}$  classes is different (as discussed in section 5); arguments are (almost always) enclosed in braces, and neither of the forms of argument provision promulgated by the ‘plain’ macros (`\macro{argument}` `\endmacro` and `\macro * {argument} *`) are provided by the  $\text{\LaTeX}$  classes.

### References

- Arseneau, Donald. “The `url` package”. Available from CTAN, `macros/latex/contrib/misc/url.sty`, 1996.
- Baxter, William Erik. “An object-oriented programming system in  $\text{\TeX}$ ”. *TUGboat* **15**(3), 331–338, 1994.
- Duggan, Angus, R. Schöpf, V. Eijkhout, and R. Fairbairns. “The `moreverb` package”. Available from CTAN, `macros/latex/contrib/moreverb`, 1997.
- Lamport, Leslie.  *$\text{\LaTeX}$ : A Document Preparation System*. Addison-Wesley, 2<sup>nd</sup> edition, 1994.
- Ogawa, Arthur. “Object-oriented programming, descriptive markup, and  $\text{\TeX}$ ”. *TUGboat* **15**(3), 325–330, 1994.

- Rahtz, Sebastian, and H. Oberdiek. “The hyperref system”. Available from CTAN, `macros/latex/contrib/hyperref`, 1997.
- Rowley, Chris. “ $\text{\LaTeX} 2_{\epsilon}$  update, dateline: 31 January 1994”. *TUGboat* **15**(1), 63, 1994.
- Schöpf, Rainer. “The verbatim package”. Part of the tools bundle, available from CTAN, `macros/latex/required/tools`, 1996.
- Swift, Matt. “Modularity in  $\text{\LaTeX}$ ”. *TUGboat* **16**(3), 269–275, 1995.
- Vieth, Ulrik. “The mflogo package”. Available from CTAN, `macros/latex/contrib/mflogo`, 1995.
- Whitney, Ron, and B. Beeton. “*TUGboat* authors’ guide”. *TUGboat* **10**(3), 378–385, 1989.

◇ Robin Fairbairns & TUGboat  
editors  
`tugboat@tug.org`  
<http://tug.org/TUGboat>