

---

## 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

The *TUGboat* web pages are at:

<https://tug.org/TUGboat>

They provide **an article template**, information for authors and reviewers, and the complete run of all published *TUGboat* issues, among other things.

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 is 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 and one/two-side 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 cropmarks.

**harvardcite** Specify Harvard-style citation (not recommended in general; see section 12); 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 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).

**onecolumn** Typeset article in one column.

**preprint** Set up for a preprint.

**rawcite** Explicitly specify the standard L<sup>A</sup>T<sub>E</sub>X citation method, which is the default; see section 12.

**runningfull** Information in both header and footer (default).

**runningminimal** Information in header only.

**runningoff** Information in neither header nor footer.

Again, normally there is no need to use any document options. They are listed here for completeness.

### 5 Command syntax

In general, we have sought simply to keep to the spirit of L<sup>A</sup>T<sub>E</sub>X in the commands provided by the *TUGboat* class (`ltugboat`).

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 original author here) the gay abandon of the syntax of the ‘plain’ *tugboat* styles [6], we have done so. Nevertheless, on the whole, the new *ltugboat* macros 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 14 lists equivalences between macros defined by the ‘plain’ package and those defined by the new package.

### 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 may 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.

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 [3]) 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 `ltugboat` class provides two environments for abstracts, `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; this is rarely used.

## 6.2 Appendices

A paper may have appendices, which can be expressed in exactly the same way as they would be in the `LATEX` article class (confusing as that may be):

```
\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 using an `appendix` environment:

```
\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 `LATEX` users; the `\title` command is exactly that used in the standard `LATEX` 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 mention a 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{https://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{...}
...
```

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 written as a set of contributed parts under the general editorship of the author(s) of the article. The author of the article is presented (using `\author`, etc.) 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
```

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

## 8 Verbatim text

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

For displayed verbatim text, 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 common:

- 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. (This is not the recommended style in general, but it's available for when it helps.)
- The command `\makevmeta`, also available only in `verbatim`'s optional argument, and makes

the construct `!<...>` inside verbatim execute `\meta{...}`. For example,

```
\begin{verbatim}[\small\makevmeta]
The !<duration> is long ...
```

produces:

The *<duration>* is long ...

The `!` character is made a general escape character by `\makevmeta`, but `<>` are not made grouping characters.

- More generally, one of the `\make*` commands,<sup>1</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.

Two caveats about these optional arguments:

- The search for the optional argument can be confused by the appearance of a `[` character as the first character of the displayed verbatim. An author who wishes to start verbatim text with a `[` character should provide an empty optional argument (i.e., `['']`) to the `verbatim` environment.
- The *TUGboat* facility is lost when any package is loaded that also defines the `verbatim` environment, as discussed next.

Authors may wish to use a more featureful verbatim package, such as `verbatim` [4], `listings` [2], or `fancyvrb` [7]. This is ok; it just means the *TUGboat* optional-argument feature is not available.

If you use the `listings` package, please specify:

```
\lstset{columns=flexible}
```

The other values for the `columns` option don't work well in *TUGboat*. Also, if you use `\small` for displayed verbatim, please reset inline verbatim to the normal text size:

```
\lstset{columns=flexible,
        basicstyle=\ttfamily\small}
\lstdefinestyle{inline}
{basicstyle=\ttfamily\normalsize}
```

## 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

<sup>1</sup> `\makeescape`, `\makebgroup`, `\makecomment`; used, for example, as `\makeescape\|`.

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 [tug.org/TUGboat/location.html](http://tug.org/TUGboat/location.html) have a more complete list, and of course the class sources are the ultimate reference.

<i>Macro</i>	<i>Output</i>
<code>\ConTeXt</code>	ConTeXt
<code>\Cplusplus</code>	C++
<code>\CTAN</code>	CTAN
<code>\eTeX</code>	$\varepsilon$ -TeX
<code>\FAQ</code>	FAQ
<code>\HTML</code>	HTML
<code>\ISBN</code>	ISBN
<code>\KOMAScript</code>	KOMA-Script
<code>\LaTeXe</code>	L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub>
<code>\MacOSX</code>	Mac OS X
<code>\MathML</code>	MathML
<code>\MF</code>	METAFont
<code>\MP</code>	METAPOST
<code>\OMEGA</code>	Ω
<code>\PDF</code>	PDF
<code>\SGML</code>	SGML
<code>\TUB</code>	TUGboat
<code>\TUG</code>	T <sub>E</sub> X Users Group
<code>\tug</code>	TUG
<code>\XML</code>	XML

**Figure 1:** A few of the provided acronyms and logos

Authors are especially urged to note the `\acro` command, which is defined in the classes. The visual appearance of all-caps sequences among normal text is rather unpleasing in Computer Modern, unfortunately. Therefore, the `\acro` command typesets its argument one point size smaller than the surrounding text: compare ‘DANTE’ (`\acro{DANTE}`) with ‘DANTE’. Many of the provided macros merely generate calls to `\acro`; two examples, `\CTAN` and `\tug`

of the list in figure 1 have already been used in the present paper.

### 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.

`\Dash` Typeset an em-dash, ignoring preceding and following space, surrounded by thin spaces, only breakable *after* the dash; this is the preferred method of specifying a dash in running text.

`\cs{cmd}` Typeset a control sequence name: `\cs{fred}` produces `\fred`.

`\env{environ}` Typeset the command to begin an environment: `\env{fred}` produces `\begin{fred}`.

`\meta{var}` Typeset meta-syntactic text: `\meta{fred}` produces `<fred>`.

`\tubbraced{text}` Typeset typewriter text in typewriter braces: `\tubbraced{fred}` produces `{fred}`.

`\nth{n}` Typeset an ordinal number; `\nth{1}` is set as 1<sup>st</sup>, `\nth{27}` is set as 27<sup>th</sup>, and so on.

## 11 Use of packages

In general, authors can use non-standard packages in their papers. However, the paper does need to be processable on the TUGboat production computers (running current T<sub>E</sub>X Live). A sensible mechanism for submitting out-of-the-ordinary packages (as for paper-specific bibliographies) is by use of the `filecontents` environment.

In general, packages currently on CTAN, and known to work with *current* L<sup>A</sup>T<sub>E</sub>X are unlikely to give problems. In particular, the team is happy to accept papers using packages that are supported by members of the L<sup>A</sup>T<sub>E</sub>X3 team,<sup>2</sup> subject to the warning already mentioned (section 8): that use of the standard `verbatim` package eliminates the special `verbatim` facilities provided by the TUGboat classes.

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.

At present, the list of canonically recommended packages consists of only two, `mflogo.sty` [5] and `url.sty` [1].

<sup>2</sup> Those in the L<sup>A</sup>T<sub>E</sub>X base distribution, or one of those in the `macros/latex/required` sub-tree on CTAN.

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). Speaking of `urls`, nowadays we generally omit a leading `http://` or `https://`.

Although not necessarily recommended in all cases, many additional packages are commonly used. To mention a few: `microtype` can help reduce overfull boxes; sometimes it may be preferable to use the Latin Modern fonts via `lmodern`; and `hyperref` allows for many internal links and other features.

## 12 Bibliography

In short: our basic recommendation for handling bibliographies is to use `BIBTEX` and the `tugboat` bibliography style. No document options are needed or recommended. All that is required in the article source (as shown in the template available from [tug.org/TUGboat](http://tug.org/TUGboat)) is the following:

```
\bibliographystyle{tugboat}
\bibliography{yourbibfile}
```

If you don't have `tugboat.bst`, which was released in 2018, it's fine to use `plain` or `abbrv`. If you do have it, though, you may enjoy the following small but useful features:

- It is based on `abbrvurl.bst` (see [ctan.org/pkg/urlbst](http://ctan.org/pkg/urlbst)),
- thus supports `url` and `doi` fields, among others.
- The `url` field is ignored if either the `doi` or `howpublished` field is present, since in practice we observe that people put the same information in all those fields, and we don't want to use up space typesetting redundant information.
- Has even more abbreviations than `abbrv`, including printing only two author names (plus "et al.") if there are more than four authors (thanks to Mico Loretan).
- Defines an `@online` entry type as an alias for `@misc`.
- For completeness only: `tugboat.bst` provides "secret" fields `newpage` to force a page break after the current item, and `urlnewline` to force a line break before the `url` value. We've found that these presentation tweaks are often necessary for the final typeset output. Authors need not worry about them.

If you use `BIBTEX`, feel free to take advantage of the accumulated bibliography of *TUGboat* itself (`info/tugboat.bib` on CTAN), and the various compilations by Nelson Beebe in that same directory.

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

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

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

A last note for bibliographies: for references to other issues of *TUGboat*, please use the format *volno:issno*, e.g., "*TUGboat* 32:1" for volume 32, number 1.

## 13 Non-recommended bibliography facilities

The preceding gives the bibliography recommendations for current *TUGboat* article. If, for whatever reason, you cannot follow the recommendation, this section is about some of the myriad historical and other possibilities.

Notwithstanding the general recommendation for the `tugboat` (falling back to `plain`) `BIBTEX` style, *TUGboat*'s Harvard citation support may be selected by specifying `harvardcite` as an option of the `\documentclass` command.<sup>3</sup> If your article demands Harvard-style citations, you may prefer to load `natbib` or similar instead of using *TUGboat*'s facilities; that's ok.

The basic citation format is 'author(s), 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.

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

<sup>3</sup> The macros used derive rather directly from the 'harvard' styles written by Glenn Pauley and now maintained by Peter Williams; the `BIBTEX` style derives from one developed by Patrick Daly.

<i>Macro</i>	<i>Output</i>
<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) [also has A and NP variants]
<code>\citeyear{key}</code>	(1990) [also has an NP variant]

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

reference: whether this indeed happens is controlled by the `extralabel` and `noextralabel` class options. The default state (option `extralabel`) attaches the extra characters.

As for  $\text{\LaTeX}$ , we don't recommend it for *TUGboat*. If you feel you must use it, that is ok, but we may still change it to using the default  $\text{\LaTeX}$  and  $\text{\BibTeX}$  facilities in processing for publication if the output from  $\text{\BibLaTeX}$  is problematic, as we have often seen it to be.

#### 14 Equivalences between the ‘plain’ and $\text{\LaTeX}$ *TUGboat* 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>	itemize, enumerate, etc., environments
<code>\verbatim</code>	verbatim or <code>\verb</code>
<code>\figure</code>	figure or figure* environments

**Figure 3:** Equivalences between plain and  $\text{\LaTeX}$  *TUGboat* 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`, `listings`, and `fancyvrb` 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

- [1] D. Arseneau. The `url` package. [ctan.org/pkg/url](http://ctan.org/pkg/url), 2013.
- [2] C. Heinz et al. The `listings` package. [ctan.org/pkg/listings](http://ctan.org/pkg/listings), 2015.
- [3] S. Rahtz, H. Oberdiek, et al. The `hyperref` package. [ctan.org/pkg/hyperref](http://ctan.org/pkg/hyperref), 2017.
- [4] R. Schöpf. The `verbatim` package. [ctan.org/pkg/mflogo](http://ctan.org/pkg/mflogo), June 1996.
- [5] U. Vieth. The `mflogo` package. [ctan.org/pkg/mflogo](http://ctan.org/pkg/mflogo), 1999.
- [6] R. Whitney and B. Beeton. *TUGboat* authors’ guide. *TUGboat* 10(3):378–385, November 1989. [ctan.org/pkg/tugboat-plain](http://ctan.org/pkg/tugboat-plain).
- [7] T. V. Zandt et al. The `fancyvrb` package. [ctan.org/pkg/fancyvrb](http://ctan.org/pkg/fancyvrb), 2015.

◇ Robin Fairbairns & TUGboat  
editors  
[tugboat@tug.org](mailto:tugboat@tug.org)  
[tug.org/TUGboat](http://tug.org/TUGboat)