
The L^AT_EX 2_ε *TUGboat* macros

Robin Fairbairns & TUGboat editors

1 Introduction

This is the documentation for the L^AT_EX 2_ε 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^AT_EX 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^AT_EX 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^AT_EX way as files `tugboat.dtx` and `tugboat.ins`. When the `.ins` file is processed by L^AT_EX, 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^AT_EX to produce a formatted (somewhat ‘literate’) source listing for those interested in the implementation of the *TUGboat* macros.

3 The general structure of a paper

The basic idea is to start your L^AT_EX 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 13.

`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 13).

`noextralabel` Don’t use the extra label-distinguishing mark in the body of the reference; see section 13.

`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^AT_EX citation method, which is the default; see section 13.

`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^AT_EX 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^AT_EX programmer, such as the original author here) the gay abandon of the syntax of the ‘plain’ `tugboat` styles [5], we have done so. Nevertheless, on the whole, the new `ltugboat` macros define L^AT_EX commands and environments, or modify the definitions of L^AT_EX ‘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^AT_EX article (the classes are defined in terms of L^AT_EX’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. If you use `\paragraph`, consider ending the paragraph label with a period; sometimes it is helpful, sometimes not.

For references to numbered sections, our style is to always use the word ‘Section’ in the text, e.g., `Section~\ref{sec:whatever}` without worrying about whether it is technically a sub(sub)section. It's also ok to use the section sign §, if that suits the material better.

Reference can also 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’. However, as you can see here, reusing the literal text of section titles often results in awkward results. We recommend numbered references in general.

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.

Please write an abstract, however short.

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. Each author may also mention a web page, using a `\personalURL` command, and an ORCID (from `orcid.org`), using `\ORCID`.

For example, the present paper has (approximately) this at its start:

```
\title{The  $\LaTeX$   $\TUB$  Macros}
\author{TUGboat editors}
\address{\TeX Users Group}
\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^AT_EX 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^AT_EX'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,¹ 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 `listings` [2] or `fancyvrb` [6]. This is ok; it just means the *TUGboat* optional-argument feature is not available. On the other hand, please do not use the `minted` package if possible; it is much harder to customize and correct at the T_EX level, and the shell escape requirement is troublesome. In any case, we will almost always wish to print code listings in simple black, not colored or even grayscaled.

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

¹ `\makeescape`, `\makebgroup`, `...`, `\makecomment`; used, for example, as `\makeescape\l`.

```
\lstset{columns=fullflexible, keepspaces=true,
        commentstyle=\slshape,
        basicstyle=\ttfamily\small}
\lstdefinestyle{inline}
  {basicstyle=\ttfamily\normalsize}
```

Explanations:

- `columns=fullflexible`: The other values for the `columns` option don't work well in *TUGboat*; we want the program text to be typeset normally, not forcibly aligned into large character cells.
- `keepspace=true`: However, having flexible columns makes spaces in the input not necessarily correspond to spaces in the output. That's usually desired, for alignment of the sources, hence `keepspace`.
- `commentstyle=\slshape`: We prefer slanted to Computer Modern typewriter italic. Using regular upright typewriter for comments is fine too.
- `basicstyle=...`: We usually prefer `\small` for displayed verbatim; when using `listings`, it is necessary to reset inline verbatim to the normal text size.

9 Floating inserts

The classes do not make any change to L^AT_EX's built-in provision for floating inserts, so that authors may generate figures and tables just as they would in any 'normal' L^AT_EX 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 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

<i>Macro</i>	<i>Output</i>
<code>\ConTeXt</code>	ConT _E Xt
<code>\Cplusplus</code>	C++
<code>\CTAN</code>	CTAN
<code>\eTeX</code>	ε-T _E X
<code>\FAQ</code>	FAQ
<code>\HTML</code>	HTML
<code>\ISBN</code>	ISBN
<code>\KOMAScript</code>	KOMA-Script
<code>\LaTeXe</code>	L ^A T _E X 2 _ε
<code>\MacOSX</code>	Mac OS X
<code>\MathML</code>	MathML
<code>\MF</code>	METAFONT
<code>\OMEGA</code>	Ω
<code>\PDF</code>	PDF
<code>\SGML</code>	SGML
<code>\TUG</code>	<i>TUGboat</i>
<code>\TUG</code>	T _E X Users Group
<code>\tug</code>	TUG
<code>\XML</code>	XML

Figure 1: A few of the provided acronyms and logos

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 Assorted other markup

A small list of special typesetting commands follows; a larger set of such commands is defined in the classes.

`\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 1st, `\nth{27}` is set as 27th, and so on.

For commands to typeset urls, see section 12.

11 Use of packages

Being a T_EX journal, authors may use both standard and non-standard external packages for their articles. The overriding criterion is that articles need to be processable on the *TUGboat* production computers (running current T_EX Live). A sensible mechanism for submitting personal packages is by use of the `filecontents` environment. It's also fine to submit manuscript source with additional packages in a zip or other archive.

In general, packages currently on CTAN, and known to work with *current* L^AT_EX are likely to be fine. In particular, the team is happy to accept papers using packages that are supported by members of the L^AT_EX team.²

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

The `ltugboat` class loads the package `mflogo.sty` [4] for typesetting the METAFONT logo. (If this package is not present by some mischance, `ltugboat` will emulate its important feature.)

An additional canonically-recommended package is `url.sty` [1], for typesetting filenames, email addresses, urls, etc.; it is being used throughout the present paper (not least in the bibliography), either on its own or via `hyperref`.

Although not necessarily recommended in all cases, many additional packages are commonly used. To mention a few:

microtype can help reduce overfull boxes and improve appearance;

hyperref supports live and internal hyperlinks, outlines, and many other features.

12 Typesetting urls

As mentioned above, please load either `url` or (preferably) `hyperref` so that reasonable line breaking of urls can happen.

Furthermore, for the printed (visible) *TUGboat* page, nowadays we typically prefer to omit a leading `http://` or `https://`. But for the link to actually

work in the output PDF or HTML, the protocol is required (or it appears to be a local filesystem path). Therefore the `ltugboat` class provides (as of version 2.28, released January 2023) the command `\tburl` for this. Related commands `\tbsurl` for `https` and `\tbhurl` for `http` were defined in version 2.23 (March 2020).³

For example, the commands `\tburl{tug.org}` and `\tburl{https://tug.org}` both typeset the text ‘tug.org’ (with the usual url line breaks possible) as a link to `https://tug.org`.

Similarly, `\tburl{http://example.org}` typesets the text ‘example.org’ as a link to `http://example.org`, for the (unusual nowadays) case when `http` is required.

As shown, the correct protocol can be explicitly specified, and will be stripped for display.

`\tburl` and related add the protocol and make live links if `hyperref` is loaded. Without `hyperref`, they are merely synonyms for `\url`. This is ok, and we still request that the protocol not be included; if live links are not being produced in the output, the printed url without the protocol suffices. (When a user copies/pastes url text into a browser, it will normally work.)

In order to keep using `\url` in a document body, we suggest `\def\url{\tburl}`.

The shortcut macro `\tburlfootnote` makes a ragged-right footnote using `\tburl` command. We recognize that when writing url references, sometimes the best option is to put urls in footnotes. However, when it's sensible, we prefer to have them as either parentheticals in the main text or in bibliography entries, to ease page breaking.

For `ftp`, `rsync`, and other protocols, it is best to always include them explicitly and use `\url`:

```
\url{ftp://tug.org},
\url{rsync://tug.org}, etc.
```

12.1 Url shortcuts

Another aspect of urls: the TUG web server supports a shortcut url mechanism, `tug.org/1/<ident>`, where `<ident>` can be any tag, similar to `tinyurl.com` and similar sites. The idea is that `tug.org` shortcuts can be used in *TUGboat* articles needing to link to excessively long and/or unstable web resources; then we update the shortcut if needed, and not worry that a commercial shortcut provider will disappear.

The only way to create a `tug.org/1/` shortcut is by request, which we are happy to receive.

² Those in the L^AT_EX base distribution, or one of those in the `macros/latex/required` sub-tree on CTAN.

³ All these commands are wrappers around `\hyper@linkurl`, and are due to Ulrike Fischer. <https://github.com/latex3/hyperref/issues/125>

13 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 in the template available from tug.org/TUGboat) is the following:

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

If you use `BIBTEX`, feel free to take advantage of the accumulated bibliography of *TUGboat* itself (mirror.ctan.org/info/biblio/tugboat.bib on CTAN, also in `TEX Live`, etc.), and the other compilations by Nelson Beebe in that same directory.

If you don't have `tugboat.bst` (our `BIBTEX` style file), 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 `abbrurl.bst` (see ctan.org/pkg/abbrurlbst),
- and thus supports `url` and `doi` fields, among others. Please use `url` instead of putting `urls` in the `note` field, where possible. Also, please don't bother to include "access date" information for *TUGboat*; we find that extraneous.
- The `url` field is ignored if either the `howpublished` or `doi` field is present. In practice we observe that people put the same information in all those fields, and we don't want to typeset redundant information.
- Does even more abbreviating than `abbrv`, such as printing only two author names (plus "et al.") if there are more than four authors (thanks to Mico Loretan and Oren Patashnik).
- New field `bookauthor` for the `@incollection` and `@inproceedings` allows for citing a part, written by author X, of a publication written by author Y, and not just edited by Y.
- For the `@misc` entry type, `editor` is accepted as well as `author`.
- Defines entry types `@online` and `@software` as aliases for `@misc`.
- Defines an `@ctan` entry type to reference packages on CTAN, following the fields output by the `ctanbib` script (in the package of the same name, ctan.org/pkg/ctanbib).
- For completeness only: `tugboat.bst` provides editor-intended fields `journaltie` to output a tie instead of space after the `journal` value, `newpage` to force a page break after the current item, `nowarning` to omit empty field warnings, `prebibitem` to output material before `\bibitem` (e.g.,

a section heading), and `urlnewline` to force a line break before the `url` value. As editors, we've found that these presentation tweaks can be desirable for the final typeset output. Authors need not worry about them.

By the way, we recommending using commas to terminate all fields in `.bib` files, including the last one in an entry. That makes for one less thing to worry about when changing fields around in the source.

Bibliographies can be difficult to typeset at the best of times. `LATEX` sets `\sloppy` when typesetting the bibliography, but this typically leads to unpleasant output with *TUGboat*'s narrow columns. The author can specify typesetting parameters using the command `\SetBibJustification`. The classes remain `\sloppy` by default, but the author may (for example) say:

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

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

Multiple citations: when citing more than one item, please include them in a single `\cite` command, as in `\cite{foo,bar}`. Also, please ensure that the result is in numerical order, so if `foo` ends up as reference '[2]' and `bar` as '[1]', use `\cite{bar,foo}`.

One more note on references: for *TUGboat* issues, please use the format *volno:issno*, e.g., "*TUGboat* 32:1" for volume 32, number 1.

By the way, if anyone would like to volunteer to implement a `BIBLATEX` style with the same features and output as our `BIBTEX` style, we would be grateful.

13.1 Non-recommended bibliography facilities

The preceding gives the bibliography recommendations for current *TUGboat* articles. If, for whatever reason, you do not wish to follow those recommendations, this section is about some of the myriad historical and other possibilities.

Notwithstanding that general recommendation for the `tugboat` (falling back to `plain`) `BIBTEX` style, *TUGboat*'s Harvard-style citation support may be selected by specifying `harvardcite` as an option of the `\documentclass` command.⁴ If your article demands Harvard-style citations, you may prefer to load `natbib` or similar instead of using *TUGboat*'s facilities; that's fine.

⁴ The macros used derive rather directly from the 'harvard' styles written by Glenn Paulley and later maintained by Peter Williams; the `BIBTEX` style derives from one developed by Patrick Daly.

This 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.

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) [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

Furthermore, 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.

As for `BIBLATEX`: 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 `LATEX` and `BIBTEX` facilities in processing for publication if the output from `BIBLATEX` is problematic, as we have often seen it to be.

14 Equivalences between the ‘plain’ and `LATEX` `TUGboat` packages

A good proportion of the commands in the ‘plain’ packages also appear (with the same meaning) in the `LATEX` classes. Figure 3 gives a brief summary of where the macros differ significantly.

`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 `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).

Plain macro	<code>L^AT_EX</code> 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., 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 `LATEX` `TUGboat` macros

Of course, the syntax of commands given to the `LATEX` classes is different (as discussed in section 5); arguments are (almost always) enclosed in braces instead of the various forms provided by the ‘plain’ macros.

References

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

◇ Robin Fairbairns & `TUGboat` editors
tugboat@tug.org
tug.org/TUGboat