The LaTeX 2ε *TUGboat* macros

TUGboat editors

Contents

1   Introduction

This is the documentation for the LaTeX 2ε 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 LaTeX 2.09; the main original author was Robin Fairbairns, with Sebastian Rahtz, Michel Goossens, Nico Poppelier and Johannes Braams. Many others have been involved in the years since, including Barbara Beeton, Karl Berry, Mimi Burbank, and the LaTeX team.

1.1   Availability

The *TUGboat* web pages are at:

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 distributed via CTAN (`ctan.org/pkg/tugboat`) in the usual LaTeX way as files `tugboat.dtx` and `tugboat.ins`. When the `.ins` file is processed by LaTeX, the files `ltugboat.cls` and `ltugcomn.sty` (a melange of perhaps-useful macros, for documentation, etc.) are produced. (For compatibility, `ltugproc.cls` is also produced, but is no longer used for proceedings or anything else.)

The `.dtx` file may itself be processed by LaTeX to produce a formatted (somewhat 'literate') source listing for those interested in the implementation of the *TUGboat* macros.

2   The general structure of a paper

The basic idea is to start your LaTeX 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; authors will see the final version in proof.

In general, we have sought simply to keep to the spirit of LaTeX in the commands provided by the *TUGboat* class (`ltugboat`). On the whole, the new `ltugboat` macros define LaTeX commands and environments, or modify the definitions of LaTeX standard commands. For those interested, section 6 lists equivalences between macros defined by the 'plain' package [4] and those defined by the new package.

2.1   Class options for `ltugboat`

The `ltugboat` class accepts many of the options of the `article` class (it suppresses the font-size selection and one/two-side options). Normally there is no need to use any document option. They are listed here for completeness.

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 5.

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 5.

noextralabel Don't use the extra mark for distinguishing labels in the body of the reference; see section 5.

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 LATEX citation method, which is the default; see section 5.

runningfull Information in both header and footer (default).

runningminimal Information in header only.

runningoff Information in neither header nor footer.

secondcolstart Start the article in the second column.

2.2 Use of packages

Being a TEX 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 TEX 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* LATEX are likely to be fine. In particular, *TUGboat* is happy to accept papers using packages that are supported by members of the LATEX team.

TUG has a policy that macro packages described in *TUGboat* should be available for readers to use. Since typing macros from printed sources is tedious, 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` [3] for typesetting the METAFONT logo. If this package is not present by some mischance, `ltugboat` will emulate it.

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 (`\usepackage{microtype}`);

hyperref supports live and internal hyperlinks, outlines, and much more (`\usepackage[hidelinks,pdfa]{hyperref}`).

2.3 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 \LaTeXe\ \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.

2.3.1 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 ...
```

2.4 Divisions of the paper

Papers in *TUGboat* may be subdivided in the normal way of a LaTeX article (the classes are defined in terms of LaTeX's `article` class). Thus the author may use `\section`, `\subsection`, ..., `\paragraph` commands (but `\part` and `\subparagraph` from `article` are suppressed, and `\chapter`, which does not 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 [2]) 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.

2.4.1 Abstracts

The `ltugboat` class provides two environments for abstracts. The `abstract` environment simply typesets its body as an unnumbered 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.

2.4.2 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):

3 Floating inserts

The classes do not make any change to LaTeX's built-in provision for floating inserts, so that authors should write figures and tables as usual,

The default is for floats to be the width of the column. To make a float which is the width of the whole page, use `figure*` and `table*`.

Regarding caption placement, *TUGboat*'s convention is to put captions for figures below the figure, but captions for tables and listings above the table/listing. Please follow this convention unless there is a specific reason not to.

As a reminder, `\label` commands should always follow the `\caption`.

4   Special-purpose typesetting

The classes define a rather large set of commands for special-purpose typesetting.

4.1   Assorted simple markup

A short list of commonly-needed special typesetting commands follows; a larger set of such commands is defined in the classes. Feel free to peruse.

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

`\Dash` Typeset an em-dash, ignoring preceding and following space, surrounded by thin spaces, only breakable after the dash; this is our preferred method of specifying an em-dash in running text, over `---` or (especially) the Unicode character(s).

`\meta{`*var*`}` Typeset meta-syntactic text: `\meta{fred}` produces ⟨*fred*⟩.

`\tbcode{`*text*`}` Literal text, such as class names, package names, filenames, environment variables, etc. The text is typeset in typewriter and is not breakable.

`\tbcodebreak{`*text*`}` Like `\tbcode`, but breakable as with urls, e.g., at periods. Still not hyphenated.

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

For commands to typeset urls, see section 4.4.

4.2   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 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`

Macro	Output
`\ConTeXt`	ConTeXt
`\Cplusplus`	C++
`\CTAN`	CTAN
`\eTeX`	ε-TeX
`\FAQ`	FAQ
`\HTML`	HTML
`\ISBN`	ISBN
`\KOMAScript`	KOMA-Script
`\LaTeXe`	LaTeX 2_ε
`\macOS`	macOS
`\MathML`	MathML
`\MF`	METAFONT
`\PDF`	PDF
`\SGML`	SGML
`\TUB`	*TUGboat*
`\TUG`	TeX Users Group
`\tug`	TUG
`\XML`	XML

Figure 1: A few of the provided acronyms and logos

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

4.3   Verbatim text

For inline verbatim text, authors should ordinarily employ the facilities of LaTeX 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 LaTeX'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.

If the *TUGboat* verbatim environment suffices, please use it. But if you need to use a more featureful verbatim package, such as `listings` [1] or `fancyvrb` [5], this is ok (the *TUGboat* optional-argument feature will not be available). We will almost always wish to print code listings in straight black, not colored or even grayscaled, so please avoid colorized listings if at all reasonable.

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

```
\lstset{columns=fullflexible,
  keepspaces=true,
  commentstyle=\slshape,
  basicstyle=\ttfamily
           \lst@ifdisplaystyle\small\fi,}
```

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.
- `keepspaces=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 `keepspaces`.
- `commentstyle=\slshape`: We prefer slanted to Computer Modern typewriter italic. Using regular upright typewriter for comments is fine (preferred) too.
- `basicstyle=...`: We usually prefer `\small` for displayed verbatim, but not inline verbatim, hence the conditional.

---

[1] `\makeescape`, `\makebgroup`, ..., `\makecomment`; used, for example, as `\makeescape\|`.

- `listings` does not easily support UTF-8 for code points above 255. If you need such characters, a different package is probably easier. See: pdfLaTeX: `tex.stackexchange.com/q/24528`; LuaLaTeX, XƎLaTeX: `tex.stackexchange.com/q/25391`.

4.4 Typesetting urls

In short:

1. Please load either the `url` or (preferably) the `hyperref` package so that reasonable line breaking of urls can happen.
2. Add `\def\url{\tburl}` to your preamble and use `\url` in your document as usual. (It seems too intrusive for the *TUGboat* classes to redefine something as fragile and widespread as `\url`.)

The rest of this section is more details about the above.

The main reason for preferring the above is that for the printed (visible) *TUGboat* page, nowadays we typically prefer to omit a leading `https://`. But for the link to actually work in the output PDF or HTML, the protocol is required (else the PDF reader thinks it is a local filesystem path). Therefore the `ltugboat` class provides (as of 2023) the command `\tburl` for this.[2]

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 `tug.org`.

Since so few web sites do not support https, `\tburl{http://example.org}` typesets the full url as text '`http://example.org`' as a link to itself. If it's desirable to elide the http://, the variant `\tbhurl` can be used.

Thus, as shown, the correct protocol can be explicitly specified, and will be stripped for display. Just use `\tburl`.

In short, `\tburl` and related make live links and omit the protocol if `hyperref` is loaded. We highly recommend this.

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}`.

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

---

[2] This and related commands are wrappers around `\hyper@linkurl`. Thanks to Ulrike Fischer for doing the real work: `github.com/latex3/hyperref/issues/125`

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

Finally, sometimes it is best to allow line breaks at hyphens in urls. We've found that this is not confusing in practice. To enable this and `hyperref` at the same time, it's necessary to load `url` explicitly and enable the `hyphens` option:

```
\usepackage[hyphens]{url}
\usepackage[hidelinks]{hyperref}
```

4.4.1   Url footnotes

The shortcut macro `\tburlfootnote` makes a ragged-right footnote using the `\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 urls as either parentheticals in the main text or in bibliography entries, to ease page breaking and reading flow. It is ideal to minimize/avoid footnotes in general.

4.4.2   Url shortcuts

A different aspect of urls: the TUG web server supports a shortcut url mechanism, `tug.org/l/⟨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/l/` shortcut is by request, which we are happy to receive.

5   Bibliographies

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 `abbrvurl.bst` (see `ctan.org/pkg/urlbst`),

- 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 `doi` or `howpublished` 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 several fields intended to be used by the editors:

  journaltie to output a tie instead of space after the `journal` value,

  monthtie for the same after `month`,

  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),

  pagesnodashify to avoid autoconverting `-` to `--` in the `pages` field,

  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 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 this can be changed with (for example),

> `\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.

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.[3] If your article demands Harvard-style citations, you may prefer to load `natbib` or similar instead of using *TUGboat*'s facilities; that's fine.

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
`\cite{key}`	(Tom, Dick, and Harry, 1990)
`\citeA{key}`	(Tom, Dick, and Harry)
`\citeNP{key}`	Tom, Dick, and Harry, 1990
`\citeANP{key}`	Tom, Dick, and Harry
`\citeN{key}`	Tom, Dick, and Harry (1990)
`\shortcite{key}`	(Tom et al., 1990)
	[also has `A` and `NP` variants]
`\citeyear{key}`	(1990)
	[also has an `NP` variant]

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

---

[3] 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.

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.

6　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.

Plain macro	LATEX macro
`\head`	`\section`
`\subhead`	`\subsection`
`\subsubhead`	`\subsubsection`
`\list`	`itemize`, `enumerate`, etc., environments
`\verbatim`	`verbatim` or `\verb`
`\figure`	`figure` or `figure*` environments

Figure 3: Equivalences between `plain` and LATEX *TUGboat* macros

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 4.3).

The syntax of commands given to the LATEX classes is different; arguments are (almost always) enclosed in braces instead of the various forms provided by the 'plain' macros.

References

[1] C. Heinz, et al. The `listings` package. ctan.org/pkg/listings

[2] S. Rahtz, H. Oberdiek, et al. The `hyperref`
    package. `ctan.org/pkg/hyperref`

[3] U. Vieth. The `mflogo` package.
    `ctan.org/pkg/mflogo`

[4] R. Whitney, B. Beeton. *TUGboat* authors'
    guide. *TUGboat*, 10(3):378–385, November
    1989. `ctan.org/pkg/tugboat-plain`

[5] T.V. Zandt, et al. The `fancyvrb` package.
    `ctan.org/pkg/fancyvrb`

⋄ TUGboat editors
`tugboat@tug.org`
`tug.org/TUGboat`