The tugboat package*

The *TUGboat* team

2026-06-17

Contents

*This file has version number v2.38, last revised 2026-06-17

1 Document preambles

```
⟨ltugboatcls | ltugproccls | ltugcomn⟩\NeedsTeXFormat{LaTeX2e}[1994/12/01]
⟨*dtx⟩
\ProvidesFile                  {tugboat.dtx}
⟨/dtx⟩
⟨ltugboatcls⟩\ProvidesClass  {ltugboat}
⟨ltugproccls⟩\ProvidesClass  {ltugproc}
⟨ltugboatsty⟩\ProvidesPackage{ltugboat}
⟨ltugprocsty⟩\ProvidesPackage{ltugproc}
⟨ltugcomn⟩    \ProvidesPackage{ltugcomn}
                    [2026-06-17 v2.38
⟨ltugboatcls⟩                        TUGboat journal class%
⟨ltugproccls⟩                        TUG conference proceedings class%
⟨ltugboatsty | ltugprocsty⟩          TUG compatibility package%
⟨ltugcomn⟩                             TUGboat 'common macros' package%
⟨*dtx⟩
                                         TUG macros source file%
⟨/dtx⟩
                    ]
```

2 Introduction

This file contains all the macros for typesetting *TUGboat* with both plain TeX and LaTeX 2ε.

2.1 Summary of control sequences

A few of the abbreviations we define, with indications of expansion where that may not be obvious. For full definitions, see real code below (Section 3.4).

`\AllTeX`	(LA)TeX
`\AMS`	American Mathematical Society
`\AmSTeX`	
`\aw`	A-W (abbreviation for Addison-Wesley)
`\API`	
`\AW`	Addison-Wesley
`\BibTeX`	
`\CandT`	Computers & Typesetting
`\ConTeXt`	ConTeXt

`\Cplusplus`	C++
`\DTD`	
`\DVD`	
`\DVI`	
`\DVIPDFMx`	DVIPDFMx
`\DVItoVDU`	DVItoVDU
`\ECMA`	
`\EPS`	
`\eTeX`	ε-TEX
`\ExTeX`	$\varepsilon\chi$TEX
`\Ghostscript`	
`\Hawaii`	Hawai‘i
`\HTML`	
`\ISBN`	ISBN
`\ISO`	
`\ISSN`	ISSN
`\JTeX`	
`\JoT`	The Joy of TEX
`\LaTeX`	
`\LyX`	
`\macOS`	mac OS
`\MacOSX`	Mac OS X
`\MathML`	
`\Mc`	M with raised c
`\MF`	METAFONT
`\mf`	METAFONT
`\MFB`	The Metafontbook
`\MP`	METAPOST
`\mp`	MetaPost (in text only: still ‘$\mp$’ in math)
`\OMEGA`	Omega ‘logo’ (Ω)
`\OCP`	Omega compiled process
`\OOXML`	
`\OTP`	Omega translation process
`\mtex`	multilingual TEX
`\NTS`	New Typesetting System
`\pcMF`	pcMF
`\PCTeX`	
`\pcTeX`	
`\Pas`	Pascal
`\PiCTeX`	
`\plain`	plain (in typewriter font)
`\POBox`	P. O. Box
`\PS`	PostScript (with hyphenation)
`\SC`	Steering Committee
`\SGML`	SGML
`\SliTeX`	
`\slMF`	Metafont, slanted: deprecated: use `\textsl` instead
`\stTeX`	TEX for the Atari ST
`\SVG`	

\TANGLE	
\TB	The TEXbook
\TeX	(Although nearly every package defines this, most, including plain, are missing the spacefactor adjustment)
\TeXhax	
\TeXMaG	(defunct)
\TeXtures	
\TeXXeT	
\Thanh	
\TFM	TFM
\TUB	*TUGboat*
\TUG	TEX Users Group
\UNIX	
\VAX	
\VnTeX	
\VorTeX	
\XeT	
\XeTeX	reflected and lowered first 'E'
\XeLaTeX	with extra space before 'L'
\XML	
\WEB	
\WEAVE	
\WYSIWYG	

Macros for things that are slightly more significant.

\NoBlackBoxes	turns off marginal rules marking overfull boxes
\BlackBoxes	turns them back on
\newline	horizontal glue plus a break
\ifundefined#1	checks argument with \csname against \relax
\topsmash	smashes above baseline (from AMSTeX)
\botsmash	smashes below baseline (from AMSTeX)
\smash	smashes both (from plain)
\ulap	lap upwards
\dlap	lap downwards
\xlap	reference point at center horizontally; 0 width
\ylap	reference point at center vertically; 0 height, depth
\zlap	combination \xlap and \ylap
\basezero	to avoid insertion of baselineskip and lineskip glue
\nullhrule	empty \hrule
\nullvrule	empty \vrule
\makestrut[#1;#2]	ad hoc struts; #1=height, #2=depth
\today	today's date
\SetTime	converts \time to hours, minutes
\now	displays time in hours and minutes
\Now	shows current date and time
\ifPrelimDraft	flag to indicate status as preliminary draft

`\rtitlex`	*TUGboat* volume and number info for running head
`\midrtitle`	information for center of running head
`\rtitlenexttopage`	next to page number in running head
`\HorzR@gisterRule`	pieces of registration marks ('trimmarks')
`\DownShortR@gisterRule`	
`\UpShortR@gisterRule`	
`\ttopregister`	top registration line with 'T' in center
`\tbotregister`	bottom registration line with inverted 'T' in center
`\topregister`	register actually used
`\botregister`	
`\raggedskip`	parameters used for ragged settings
`\raggedstretch`	
`\raggedparfill`	
`\raggedspaces`	
`\raggedright`	
`\raggedleft`	
`\raggedcenter`	
`\normalspaces`	
`\raggedbottom`	
`\bull`	square bullet
`\cents`	'cents' sign
`\Dag`	superscripted dagger
`\careof`	c/o
`\sfrac`	slashed fraction (arguments optionally separated by a slash)
`\cs`	control sequence name `\cs{name}`→`\name`
`\meta`	meta-argument name `\meta{name}`→⟨*name*⟩
`\dash`	en-dash surrounded by thinspaces; only breakable AFTER
`\Dash`	em-dash, as above
`\hyph`	permit automatic hyphenation after an actual hyphen
`\slash`	'breakable' slash
`\nth`	for obtaining '1st', '2nd', 3rd, etc.
`\tubissue`	gets `\TUB` followed by volume and issue numbers
`\xEdNote`	Editor's Note:
`\Review:`	Review: (for title of book review article)
`\reviewitem`	begin data for item being reviewed
`\revauth`	with one argument, author(s) of item being reviewed
`\revtitle`	with one argument, title of ...
`\revpubinfo`	with one argument, other info pertaining to ...
`\endreviewitem`	end data for item being reviewed
`\titleref`	one argument, format title as straight text (slanted, frenchspacing)

`\Input`	`\input` with some other bookkeeping for case where multiple articles are put together
`\TBremark`	reminder to *TUGboat* editorial staff
`\TBEnableRemarks`	enable `\TBremark`s (normally suppressed)
`\pagexref`	used to write out page numbers to screen and
`\pagexrefON`	external files
`\pagexrefOFF`	
`\xrefto`	used for symbolic cross-reference to other pages
`\xreftoON`	in *TUGboat*
`\xreftoOFF`	
`\TBdriver`	marks code which only takes effect when articles are run together in a driver file
`\signaturemark`	items for signatures
`\signaturewidth`	

3 LaTeX 2ε *TUGboat* class file

3.1 Setup and options

Occasionally we need to do different things when running under traditional (pdf)latex or a native Unicode engine. Since we don't need any fancier distinctions, instead of reading the `iftex` or another package, do the test directly.

```
⟨*common⟩
\newif\ifTBunicodeengine
\ifx\Umathchardef\@thisisundefined % not (xetex|luatex)
  \TBunicodeenginefalse
\else
  \TBunicodeenginetrue
\fi
⟨/common⟩
```

Check for reloading. Hmmm... Does this happen with LaTeX 2ε classes? Probably, in fact, as well that it doesn't, since the `\tugstyinit` referenced here doesn't exist; however, it's possible that we might need a similar mechanism in the future, so we retain its skeleton, without fleshing out the `\tugstyinit` bones.

```
⟨*ltugboatcls⟩
\csname tugstyloaded@\endcsname
\def\tugstyloaded@{\tugstyinit\endinput}
```

Acquire a name for this class if we don't already have one (by virtue of having been loaded by `tugproc.cls`). This name will be used in error messages and the like.

```
\providecommand{\@tugclass}{ltugboat}
```

Errors/warnings/information messages — if we're using LaTeX 2ε we can use the `\Class*` commands. `\tbdebug` is different from all the others, intended for temporary debugging messages (hence the all-lowercase name); they're started with `***` at the beginning of a line to make them stand out, and be parsable, e.g. by `texfot`(1).

```
\def\TBError{\ClassError{\@tugclass}}
\def\TBWarning{\ClassWarning{\@tugclass}}
```

```
\def\TBWarningNL{\ClassWarningNoLine{\@tugclass}}
\def\TBInfo{\ClassInfo{\@tugclass}}
\def\tbdebug#1{\message{^^J*** #1}}
```

Unfortunately, LaTeX's `\loggingall` does not turn off tracingonline. And `microtype` outputs useless verbose expansions to the terminal after `\loggingall`. So make our own:

```
\def\tbloggingall{\loggingall \tracingonline=0 }
```

Class options: `draft` vs. `preprint` vs. `final`.

```
\DeclareOption{draft}{% [draft], the default
  % If the user loads hyperref, avoid passing on the global draft option
  % (which would remove all links in the pdf).
  \PassOptionsToPackage{final}{hyperref}
  %
  \AtEndOfClass{%
    \setcounter{page}{901}%
    \BlackBoxes
    \def\MakeRegistrationMarks{}%
    \PrelimDrafttrue
  }%
}

\newif\ifpreprint
\def\preprint{\preprinttrue} % [preprint], hardly used
\DeclareOption{preprint}{%
   \preprinttrue
}

\newif\iftubfinaloption % [final], manually inserted by us for processing
\DeclareOption{final}{%
  \tubfinaloptiontrue
  \AtEndOfClass{%
    % Insert draft date into the header even with [final], if we are not
    % doing a production run. (|tugboat.dates| sets up page numbers
    % above 900 in such pseudo-draft mode.) We use [final] in the first
    % place for this case because draft vs. final can change page
    % layout, wrt registration marks, etc. (Not good, but too painful to
    % change at this late date.)
    \ifnum\value{page}>900 \PrelimDrafttrue \else \PrelimDraftfalse \fi
    \@tubrunningfull
  }%
}
```

We want to use `hyperref`'s `\texorpdfstring`, e.g., in the `draft` option above. If `hyperref` is not loaded, define our own trivial fallback to expand to the TeX (first) argument.

Similarly, disable and more if we have `hyperref`, so section titles using them don't cause useless warnings.

```
\AtBeginDocument{%
  \ifx\undefined\texorpdfstring
    \DeclareRobustCommand{\texorpdfstring}[2]{#1}%
  \fi
  %
```

```
  \ifx\undefined\pdfstringdefDisableCommands\else
    \pdfstringdefDisableCommands{%
      \let\acro\relax
      \let\origDash=\Dash \def\Dash{\texorpdfstring{\origDash}{--}}%
      % lots more could/should be added.
    }%
  \fi
}
```

TUGboat uses only 10pt for the main text.

```
\DeclareOption{11pt}{%
  \TBWarning{The \@tugclass\space class only supports 10pt fonts:
    \MessageBreak option \CurrentOption\space ignored}%
}
\DeclareOption{12pt}{\csname ds@11pt\endcsname}
```

Similarly, ignore one/two-side options.

```
\DeclareOption{oneside}{\TBWarning{Option \CurrentOption\space ignored}}
\DeclareOption{twoside}{\ds@oneside}
```

There are these people who seem to think `tugproc` is an option rather than a class... (Note that it's already been filtered out if we were calling from `ltugproc`.)

```
\DeclareOption{tugproc}{%
  \TBWarning{Option \CurrentOption\space ignored: use class ltugproc
    instead of \@tugclass}%
}
```

Option `rawcite` (the default) specifies the default citation mechanism (as built-in to LaTeX); option `harvardcite` specifies the author-date citation mechanism defined in section 3.24 below.

```
\DeclareOption{rawcite}{\let\if@Harvardcite\iffalse}
\DeclareOption{harvardcite}{\let\if@Harvardcite\iftrue}
```

Option `extralabel` (the default) specifies that the publication years of two successive references with otherwise identical labels will be tagged with distinguishing letters; option `noextralabel` causes those letters to be suppressed. Note that (a) no two references will in any case have the same labels in the default (plain) `rawcite` setup, and that (b) the distinguishing letters appear in the labels themselves; the reader can work out the correspondence one with the other...

```
\DeclareOption{extralabel}{\let\UseExtraLabel\@firstofone}
\DeclareOption{noextralabel}{\let\UseExtraLabel\@gobble}
```

The section-numbering style, so that we can allow the same heading layout as in the plain macros.

```
\DeclareOption{numbersec}{\let\if@numbersec\iftrue}
\DeclareOption{nonumber}{\let\if@numbersec\iffalse}
```

Minimal running headers/footers contain just the TUGboat volume/issue identification and page numbers. 'runningfull' is the default, and includes title and author. 'runningoff' makes both headers and footers empty.

```
\DeclareOption{runningoff}{\AtEndOfClass{\@tubrunningoff}}
\DeclareOption{runningminimal}{\AtEndOfClass{\@tubrunningminimal}}
\DeclareOption{runningfull}{\AtEndOfClass{\@tubrunningfull}}
```

Usually we want to print the doi if [final], else not. But sometimes we want to omit it even if [final], namely when we're posting a review or other item early.

```
\newif\iftubomitdoioption
\DeclareOption{omitdoi}{%
  \tubomitdoioptiontrue
}
```

\if@tubtwocolumn Occasionally (`tb107jackowski`, and past conference preprints), we need the option `onecolumn`. For alternative approaches to one-column articles, see `tb92hagen-euler` and `tb78milo`.

```
\newif\if@tubtwocolumn  \@tubtwocolumntrue
\DeclareOption{onecolumn}{\@tubtwocolumnfalse}
```

\ifsecondcolstart Occasionally, we need to start an article in the second column of a page, due to splicing with a previous article. Let's try declaring that. Then, before `\maketitle`, we'll force the move to the second column.

And sometimes we need to add space at the top of that second column (e.g., `tb136lettre`); there's no way to intervene in the article source, so define a hook `\tubsecondcolstartextra`.

```
\newif\iftubsecondcolstart
\DeclareOption{secondcolstart}{\tubsecondcolstarttrue}
\let\tubsecondcolstartextra\relax
```

Any other options, we pass on to `article.cls` before we load it:

```
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
```

Request default options (draft mode, standard citation, numbered sections, etc.), process all options, and then get the base document class on top of which we reside, namely `article`. Always call `article` with the `twoside` option, since we want the ability to have odd/even headers/footers.

```
\ExecuteOptions{draft,extralabel,numbersec,rawcite,runningminimal}
\ProcessOptions
\LoadClass[twoside]{article}
```

Various fonts used throughout. Some effort has been made to suppress these things with explicit sizes in the macro name (`\tensl` is an example below), but keeping in step with the documentation is one thing that restricts such a move.

```
\def\sectitlefont{\fontfamily\sfdefault\fontseries{bx}\fontshape{n}%
         \fontsize\@xviipt\stbaselineskip\selectfont}
\def\tensl{\fontseries{m}\fontshape{sl}\fontsize\@xpt\@xiipt
          \selectfont}
```

This font selection command is used *only* for the 'Editor's Note' introduction to notes; sadly it makes explicit reference to CMR, and Barbara Beeton has agreed that the reference may be constructed to use the current family such that, if no upright italic is defined, ordinary italics are used. A project for later...

```
\ifTBunicodeengine
  % there is no "LM unslanted" in OpenType, so use the standard cmu
  % scaled for the current text size. Not worth more effort.
  \def\EdNoteFont{\font\ednotefont = cmu10 at 1em }
\else % traditional engine:
```

```
  \def\EdNoteFont{\fontfamily{cmr}\fontseries{m}\fontshape{ui}\selectfont}
\fi
⟨/ltugboatcls⟩
```

If Ulrik Vieth's `mflogo.sty` is around, we'll use it. Otherwise (pro tem, at least) we'll warn the user and define the absolute minimum of machinery that *TUGboat* requires (that which was used prior to the invention of LaTeX 2ε).

```
⟨*common⟩
\IfFileExists{mflogo.sty}%
  {\RequirePackage{mflogo}}%
⟨!ltugcomn⟩ {\TBWarning
⟨ltugcomn⟩  {\PackageWarning{ltugcomn}
      {Package mflogo.sty not available --\MessageBreak
        Proceeding to emulate mflogo.sty}
   \DeclareRobustCommand{\logofamily}{%
     \not@math@alphabet\logofamily\relax
     \fontencoding{U}\fontfamily{logo}\selectfont}
   \DeclareTextFontCommand{\textlogo}{\logofamily}
   \def\MF{\textlogo{META}\tubdisc\textlogo{FONT}\@}
   \def\MP{\textlogo{META}\tubdisc\textlogo{POST}\@}
   \DeclareFontFamily{U}{logo}{}
   \DeclareFontShape{U}{logo}{m}{n}{%
     <8><9>gen*logo%
     <10><10.95><12><14.4><17.28><20.74><24.88>logo10%
   }{}
   \DeclareFontShape{U}{logo}{m}{sl}{%
     <8><9>gen*logosl%
     <10><10.95><12><14.4><17.28><20.74><24.88>logosl10%
   }{}
   \DeclareFontShape{U}{logo}{m}{it}{%
     <->ssub*logo/m/sl%
   }{}%
  }
```

3.2 Resetting at start of paper

`\ResetCommands` `\AddToResetCommands` `\StartNewPaper`

We store a set of commands that should be executed at the start of each paper, before any paper-specific customisation. These commands (stored in the token register `\ResetCommands`) include things such as resetting section and footnote numbers, re-establishing default settings of typesetting parameters, and so on. The user (or more typically, editor) may execute the commands by using the command `\StartNewPaper`. Things I've not yet thought of may be added to the list of commands, by

```
\newtoks\ResetCommands
\ResetCommands{%
  \setcounter{part}{0}%
  \setcounter{section}{0}%
  \setcounter{footnote}{0}%
  \authornumber\z@
}
\newcommand{\AddToResetCommands}[1]{%
  \AddToResetCommands\expandafter{\AddToResetCommands#1}%
}
```

3.3 Helpful shorthands (common code with Plain styles)

`\makeescape`, ..., `\makecomment` allow users to change the category code of a single character a little more easily. These require that the character be addressed as a control sequence: e.g., `\makeescape\/` will make '/' an escape character.

```
⟨*!latex⟩
\def\makeescape#1{\catcode`#1=0 }
\def\makebgroup#1{\catcode`#1=1 }
\def\makeegroup#1{\catcode`#1=2 }
\def\makemath  #1{\catcode`#1=3 }
⟨/!latex⟩
⟨*latex⟩
\def\makeescape#1{\catcode`#1=\z@}
\def\makebgroup#1{\catcode`#1=\@ne}
\def\makeegroup#1{\catcode`#1=\tw@}
\def\makemath  #1{\catcode`#1=\thr@@}
⟨/latex⟩
\def\makealign #1{\catcode`#1=4 }
\def\makeeol   #1{\catcode`#1=5 }
\def\makeparm  #1{\catcode`#1=6 }
\def\makesup   #1{\catcode`#1=7 }
\def\makesub   #1{\catcode`#1=8 }
\def\makeignore#1{\catcode`#1=9 }
\def\makespace #1{\catcode`#1=10 }
\def\makeletter#1{\catcode`#1=11 }
\chardef\other=12
\let\makeother\@makeother
\def\makeactive#1{\catcode`#1=13 }
\def\makecomment#1{\catcode`#1=14 }
```

`\savecat#1` and `\restorecat#1` will save and restore the category of a given character. These are useful in cases where one doesn't wish to localize the settings and therefore be required to globally define or set things.

```
\def\savecat#1{%
  \expandafter\xdef\csname\string#1savedcat\endcsname{\the\catcode`#1}}
\def\restorecat#1{\catcode`#1=\csname\string#1savedcat\endcsname}
⟨!latex⟩\savecat\@
⟨!latex⟩\makeletter\@
```

`\SaveCS#1` and `\RestoreCS#1` save and restore 'meanings' of control sequences. Again this is useful in cases where one doesn't want to localize or where global definitions clobber a control sequence which is needed later with its 'old' definition.

```
\def\SaveCS#1{\expandafter\let\csname saved@@#1\expandafter\endcsname
  \csname#1\endcsname}
\def\RestoreCS#1{\expandafter\let\csname#1\expandafter\endcsname
  \csname saved@@#1\endcsname}
```

To distinguish between macro files loaded

```
\def\plaintubstyle{plain}
\def\latextubstyle{latex}
```

Control sequences that were first defined in LaTeX 2_ε of 1995/06/01 (or later), but which we merrily use. Only define if necessary:

```
\providecommand\hb@xt@{\hbox to}
\providecommand\textsuperscript[1]{\ensuremath{\m@th
                                    ^{\mbox{\fontsize\sf@size\z@
                                        \selectfont #1}}}}
```

(Note that that definition of `\textsuperscript` isn't robust, but probably doesn't need to be... What's more, it doesn't appear in the mythical 2.09 version of the package.)

We end up wanting this fairly often, and LaTeX removed `\line`.

```
\def\tubline{\hbox to \hsize}
```

3.4 Abbreviations and logos

```
\def\tubdisc{\texorpdfstring{\-}{}} % avoid warning about \- in section titles
\DeclareRobustCommand{\AllTeX}{%
  \texorpdfstring{(\La\kern-.075em)\kern-.05em\TeX}{(La)TeX}}
\def\AMS{American Mathematical Society}
\def\AmS{$\mathcal{A}$\kern-.1667em\lower.5ex\hbox
    {$\mathcal{M}$}\kern-.125em$\mathcal{S}$}
\def\AmSLaTeX{\AmS-\LaTeX}
\def\AmSTeX{\AmS-\TeX}
\def\ANSI{\acro{ANSI}}
\def\API{\acro{API}}
\def\ASCII{\acro{ASCII}}
\def\aw{\acro{A\kern.04em\raise.115ex\hbox{-}W}}
\def\AW{Addison\kern.1em-\penalty\z@\hskip\z@skip Wesley}
%
% make \BibTeX work in slanted contexts too; it's common in titles, and
% especially burdensome to hack in .bib files.
\def\Bib{%
  \ifdim \fontdimen1\font>0pt
     B{\SMC\SMC IB}%
  \else
     B\textsc{ib}% LaTeX has more kerns, but they are too much to our eyes
  \fi
}
\def\BibLaTeX{\texorpdfstring{\Bib\tubdisc\kern.02em \LaTeX}{BibLaTeX}}
\def\BibTeX{\texorpdfstring{\Bib\tubdisc\kern-.04em \TeX}{BibTeX}}
% no good way to determine bold font, and we want to lose the kern, too:
% (we \let BibTeX to this in maketitle)
\def\bfBib{B{\SMC\SMC IB}}
\def\bfBibTeX{\texorpdfstring{\bfBib\TeX}{BibTeX}}
\def\bfBibLaTeX{\texorpdfstring{\bfBib\LaTeX}{BibLaTeX}}
%
\def\BSD{\acro{BSD}}
\def\CandT{\textsl{Computers~\& Typesetting}}
% must not define \CJK, because the CJK package does.
```

We place our `\kern` after `\-` so that it disappears if the hyphenation is taken:

```
\def\ConTeXt{\texorpdfstring{C\kern-.0333emon\tubdisc\kern-.0667em\TeX\kern-.0333emt}
  {ConTeXt}}
\def\CMkIV{\ConTeXt\ \MkIV}
\def\Cplusplus{C\plusplus}
%
```

```
% Turns out this original TUB definition has different output under
% pdftex and lua/xetex, because, ultimately, the axis in the OTF math
% cm fonts (newcm, latinmodern-math) fonts is at 57 units instead of 70
% units. Why Jacko did this is unknown, but can't be changed now.
%original TUGboat: \def\plusplus{\raisebox{.7ex}{$_{++}$}}
%
% We can't avoid OTF math in general. So we change the definition not to
% use math. The results are the same within a couple of decimal places
% (didn't seem to matter to make it exact), and it's simpler besides.
\def\plusplus{\texorpdfstring{\raise .351ex \hbox{\scriptsize ++}}{++}}
%
% consider rm vs. bold + tb139may-automata.ltx
\def\CPU{\acro{CPU}}
\def\CSczabbr{\ensuremath{\cal C}\kern-.1667em\lower.5ex\hbox{$\cal S$}}
\def\CSS{\acro{CSS}}
\def\CSTUG{\CSczabbr\kern.05em\acro{TUG}}
\def\CSV{\acro{CSV}}
\def\CTAN{\acro{CTAN}}
\def\DTD{\acro{DTD}}
\def\DTK{\acro{DTK}}
\def\DVD{\acro{DVD}}
\def\DVI{\acro{DVI}}
\def\DVIPDFMx{\acro{DVIPDFM}$x$}
\def\DVItoVDU{DVIto\kern-.12em VDU}
\def\ECMA{\acro{ECMA}}
\def\EPS{\acro{EPS}}
% no line break at this hyphen please, and try to get a bold \varepsilon.
\def\TUBdefaulteTeX{\ensuremath{\varepsilon}\mbox{-}\kern-.125em\TeX}%
\DeclareRobustCommand{\eTeX}{%
  \ifx\f@series\bfseries@rm
    \ifx\boldsymbol\undefined % \boldsymbol is from amsmath; also support bm?
      \TBWarning{bold varepsilon for \string\eTeX\space not available; load amsmath}%
      \TUBdefaulteTeX
    \else
      \ensuremath{\boldsymbol{\varepsilon}}\mbox{-}\kern-.125em\TeX
    \fi
  \else
    \TUBdefaulteTeX
  \fi
}
\DeclareRobustCommand{\ExTeX}{%
  \ensuremath{\textstyle\varepsilon_{\kern-0.15em\cal{X}}}\kern-.2em\TeX}
\def\FAQ{\acro{FAQ}}
\def\FTP{\acro{FTP}}
\def\Ghostscript{Ghost\tubdisc script}
\def\GNU{\acro{GNU}}
\def\GUI{\acro{GUI}}
\DeclareRobustCommand{\HarfBuzz}{Harf\discretionary{-}{}{\kern.02em}Buzz}
\def\Hawaii{Hawai‘i}
\def\HTML{\acro{HTML}}
\def\HTTP{\acro{HTTP}}
\def\HTTPS{\acro{HTTPS}}
\def\iOS{i\acro{OS}}
\def\IDE{\acro{IDE}}
```

```
\def\IEEE{\acro{IEEE}}
\def\ISBN{\acro{ISBN}}
\def\ISO{\acro{ISO}}
\def\ISSN{\acro{ISSN}}
\def\JPEG{\acro{JPEG}}
\def\JTeX{\leavevmode\hbox{\lower.5ex\hbox{J}\kern-.18em\TeX}}
\def\JoT{\textsl{The Joy of \TeX}}
\DeclareRobustCommand{\KOMAScript}{\textsf{K\kern.05em O\kern.05em%
     M\kern.05em A\kern.1em\hyph\kern.1em Script}}
\def\LAMSTeX{L\raise.42ex\hbox{\kern-.3em
                  $\m@th$\fontsize\sf@size\z@\selectfont
                  $\m@th\mathcal{A}$}%
    \kern-.2em\lower.376ex\hbox{$\m@th\mathcal{M}$}\kern-.125em
    {$\m@th\mathcal{S}$}-\TeX}
% This code is hacked from its definition of \cs{LaTeX}; it allows
% slants (for example) to propagate into the raised (small) 'A':
%    \begin{macrocode}
\DeclareRobustCommand{\La}%
   {L\kern-.36em
        {\setbox0\hbox{T}%
         \vbox to\ht0{\hbox{$\m@th$%
                           \csname S@\f@size\endcsname
                           \fontsize\sf@size\z@
                           \math@fontsfalse\selectfont
                           A}%
                     \vss}%
        }}
```

We started with the intention that we wouldn't redefine `\LaTeX` when we're running under it, so as not to trample on an existing definition. However, this proves less than satisfactory; a single logo may be OK for the run of documents, but for *TUGboat*, we find that something noticeably better is necessary; see section 3.12.

```
⟨!latex⟩\def\LaTeX{\La\kern-.15em\TeX}
\def\LMTX{\acro{LMTX}}
\def\LuaHBTeX{Lua\acro{HB}\tubdisc\TeX}%
\def\LuaHBLaTeX{Lua\acro{HB}\tubdisc\LaTeX}%
\def\LuaLaTeX{Lua\tubdisc\LaTeX}% dtk-logos defines it
\def\LuaTeX{Lua\tubdisc\TeX}% ditto
\def\luatex{\LuaTeX}% ditto
\def\LyX{L\kern-.1667em\lower.25em\hbox{Y}\kern-.125emX}
\def\macOS{mac\acro{OS}}
\def\MacOSX{Mac\,\acro{OS}\,X}}
\def\MathML{Math\acro{ML}}
\def\Mc{\setbox\TestBox=\hbox{M}M\vbox
   to\ht\TestBox{\hbox{c}\vfil}} %  for Robert McGaffey
```

If we're running under LaTeX 2ε, we use Ulrik Vieth's `mflogo.sty` if it's present. Otherwise, we're using a short extract of Vieth's stuff. Either way, we don't need to specify `\MF` or `\MP`.

```
\def\mf{\textsc{Metafont}}
\def\MFB{\textsl{The \MF\kern.1em\tubdisc book}}
\DeclareRobustCommand{\MiKTeX}{MiK\TeX}
\def\MkIV{Mk\acro{IV}}
\let\TB@@mp\mp
```

```
\DeclareRobustCommand{\mp}{\ifmmode\TB@@mp\else MetaPost\fi}
\def\mtex{T\kern-.1667em\lower.424ex\hbox{\^E}\kern-.125emX\@}
%
% In order that the \cs{OMEGA} command will switch to using the TS1
% variant of the capital Omega character if \texttt{textcomp.sty} is
% loaded, we define it in terms of the \cs{textohm} command.  Note
% that this requires us to interpose a level of indirection, rather
% than to use \cs{let}\dots
% Revised definition of \cs{NTS} based on that used by Phil Taylor.
%
%    \begin{macrocode}
\DeclareRobustCommand{\NTG}{\acro{NTG}}
\DeclareRobustCommand{\NTS}{\ensuremath{\mathcal{N}\mkern-4mu
  \raisebox{-0.5ex}{$\mathcal{T}$}\mkern-2mu \mathcal{S}}}
\DeclareTextSymbol{\textohm}{OT1}{'012}
\DeclareTextSymbolDefault{\textohm}{OT1}
\newcommand{\OMEGA}{\textohm}
\DeclareRobustCommand{\OCP}{\OMEGA\acro{CP}}
\DeclareRobustCommand{\OOXML}{\acro{OOXML}}
\DeclareRobustCommand{\OTF}{\acro{OTF}}
\DeclareRobustCommand{\OTP}{\OMEGA\acro{TP}}
\DeclareRobustCommand{\OpTeX}{\texorpdfstring{Op\kern-.05em\TeX}{OpTeX}}
\def\Pas{Pascal}
\def\pcMF{\leavevmode\raise.5ex\hbox{p\kern-.3\p@ c}MF\@}
\def\PCTeX{PC\thinspace\TeX}
\def\pcTeX{\leavevmode\raise.5ex\hbox{p\kern-.3\p@ c}\TeX}
\def\pdfLaTeX{pdf\/\tubdisc\LaTeX}% dtk-logos
\def\pdflatex{\pdfLaTeX}
\def\pdfTeX{pdf\/\tubdisc\TeX}% dtk-logos
\def\pdftex{\pdfTeX}
\def\PDF{\acro{PDF}}
\def\PDFUA{\acro{PDF/UA}}
\def\PGF{\acro{PGF}}
\def\PHP{\acro{PHP}}
\def\PiC{P\kern-.12em\lower.5ex\hbox{I}\kern-.075emC\@}
\def\PiCTeX{\PiC\kern-.11em\TeX}
\def\plain{\texttt{plain}}
\def\PNG{\acro{PNG}}
\def\POBox{P.\thinspace O.~Box }
\def\PS{{Post\tubdisc Script}}
\def\PSTricks{\acro{PST}ricks}
\def\RIT{\acro{RIT}}
\def\RTF{\acro{RTF}}
\def\SC{Steering Committee}
\def\SGML{\acro{SGML}}
\def\SliTeX{\textrm{S\kern-.06em\textsc{l\kern-.035emi}%
                      \kern-.06em\TeX}}
\def\slMF{\textsl{\MF}} % should never be used
\def\SQL{\acro{SQL}}
\def\stTeX{\textsc{st}\kern-0.13em\TeX}
\def\STIX{\acro{STIX}}
\def\SVG{\acro{SVG}}
\def\TANGLE{\texttt{TANGLE}\@}
\def\TB{\textsl{The \TeX\tubdisc book}}
```

```
\def\TIFF{\acro{TIFF}}
\def\TP{\textsl{\TeX:\ The Program}}
\DeclareRobustCommand{\TeX}{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX\@}
\def\TeXhax{\TeX hax}
\DeclareRobustCommand{\TeXLive}{\TeX\ Live}
\def\TeXMaG{\TeX M\kern-.1667em\lower.5ex\hbox{A}\kern-.2267emG\@}
\def\TeXtures{\textit{Textures}}
\let\Textures=\TeXtures
\def\TeXworks{\TeX\kern-.07em works}
\def\TeXXeT{\TeX-{}-\XeT}
\def\TFM{\acro{TFM}}
% \Thanh is defined below.
\def\TikZ{Ti\/{\em k}Z}
\def\ttn{\textsl{TTN}\@}
\def\TTN{\textsl{\TeX{} and TUG News}}
\def\TUB{\texttub{TUGboat}}\def\texttub{\textsl} % redefined in some situations
\def\TUG{\TeX\ \UG}
\def\tug{\acro{TUG}}
\def\UG{Users Group}
\def\UNIX{\acro{UNIX}}
% Don't define \UTF, since other packages use it for Unicode character access.
% On the other hand, we want a macro for UTF-8 that doesn't break at a
% following -, as in \tbUTF-8.
\def\tbUTF{\acro{UTF}\futurelet\@nextchar\@tbUTFcheck}
  \def\@tbUTFcheck{\ifx\@nextchar-%
                     \mbox{-}\let\next=\tbgobbledash
                   \else
                     \let\next=\empty
                   \fi\next}
  \def\tbgobbledash-{}
\def\VAX{V\kern-.12em A\kern-.1em X\@}
\def\VnTeX{V\kern-.03em n\kern-.02em \TeX}
\def\VorTeX{V\kern-2.7\p@\lower.5ex\hbox{O\kern-1.4\p@ R}\kern-2.6\p@\TeX}
\def\XeT{\texorpdfstring{X\kern-.125em\lower.424ex\hbox{E}\kern-.1667emT\@}{XeT}}
\def\XML{\acro{XML}}
\def\XMP{\acro{XMP}}
\def\WEB{\texorpdfstring{\texttt{WEB}\@}{WEB}}
\def\WEAVE{\texttt{WEAVE}\@}
\def\WYSIWYG{\acro{WYSIWYG}}
```

XeTEX requires reflecting the first E, hence we complain if the graphics package is not present. (For plain documents, this can be loaded via `miniltx` or Eplain.) Also, at Barbara's suggestion, if the current font is slanted, we rotate by 180 instead of reflecting so there is a better chance to look ok. (The magic values here seem more or less ok for `cmsl` and `cmti`.)

```
\def\tubreflect#1{%
  \@ifundefined{reflectbox}{%
    \TBError{A graphics package must be loaded to use \string\XeTeX}
            {Load graphicx or graphics.}%
  }{%
    \ifdim \fontdimen1\font>0pt
      \raise 1.75ex \hbox{\kern.1em\rotatebox{180}{#1}}\kern-.1em
    \else
      \reflectbox{#1}%
```

```
    \fi
  }%
}
\def\tubhideheight#1{\setbox0=\hbox{#1}\ht0=0pt \dp0=0pt \box0 }
\def\XekernbeforeE{-.125em}
\def\XekernafterE{-.1667em}
% From Max, mail of 13sep24:
% hyperref is trying to expand \Xe to get a string for
% the embedded PDF table of contents, but \Xe is unsafe in an
% expansion-only context [even when defined with \DeclareRobustCommand,
% for reasons unknown].
% An easy way to fix this is to replace \DeclareRobustCommand with
% \NewDocumentCommand, which defines the macro as ``\protected'' instead
% as ``\protect''ed.
\NewDocumentCommand\tub@Xe{}{\leavevmode
  \tubhideheight{\hbox{X%
    \setbox0=\hbox{\TeX}\setbox1=\hbox{E}%
    \ifdim \fontdimen1\font>0pt
      % XeTeX logo needs tinkering when slanted/italic font,
      % so make kerns changeable
      \def\XekernbeforeE{-.11em}%
      \def\XekernafterE{-.16em}%
      \dp1=-.17ex
    \fi
    \lower\dp0\hbox{\raise\dp1\hbox{\kern\XekernbeforeE\tubreflect{E}}}%
    \kern\XekernafterE}}}
% [But then,] For hyperref to be able to see the \texorpdfstring, it
% needs to be inside of a non-protected macro, but we still want the
% graphics commands to be protected, so we need to make a wrapper command:
\newcommand\Xe{\texorpdfstring{\tub@Xe}{Xe}}
\def\XeTeX{\texorpdfstring{\Xe\TeX}{XeTeX}}
\def\XeLaTeX{\texorpdfstring{\Xe{\kern.11em \LaTeX}}{XeLaTeX}}
%
\def\XHTML{\acro{XHTML}}
\def\XSL{\acro{XSL}}
\def\XSLFO{\acro{XSL}\raise.08ex\hbox{-}\acro{FO}}
\def\XSLT{\acro{XSLT}}
\def\YAML{\acro{YAML}}
```

3.5 General typesetting rules

```
\newlinechar=`\^^J
\normallineskiplimit=\p@
\clubpenalty=10000
\widowpenalty=10000
\def\NoParIndent{\parindent=\z@}
\newdimen\normalparindent
\normalparindent=20\p@
\def\NormalParIndent{\global\parindent=\normalparindent}
\NormalParIndent
\def\BlackBoxes{\overfullrule=5\p@}
\def\NoBlackBoxes{\overfullrule=\z@}
\def\newline{\hskip\z@\@plus\pagewd\break}
```

\Thanh Han The Thanh's name is complicated because it has a double-accented character over the e, U+1EBF (it also has grave accents over the a's, but they aren't a problem): Hàn Thế Thành. Many fonts do not have this character; hardly any, in the old days, and definitely not Computer Modern.

So, when the e-circumflex-acute character in Thanh's name is not in the font, we'll use the following definition. Until fall 2025, we used a definition that put the acute vertically over the e. But then Barbara said, and Thanh agreed, that it is preferable nowadays to put the acute to the right of the circumflex. So that's what this definition does:

```
\def\TBecircacute{\^e\llap{\raise 0.3ex\hbox{\'{}\kern-0.4ex}}}%
```

For posterity, this was our previous definition, used for many years, which put the acute directly over the circumflex:

```
%\def\Thanh{H\`an~Th\^e\llap{\raise 0.5ex\hbox{\'{}}}~Th\`anh}
```

Here's a way to get at our new definition in case we overwrite it below. This is because the lm fonts, which are the LaTeX default with Unicode engines, put the acute vertically over the circumflex (perhaps since *TUGboat* did), and we'll want to change that.

```
\let\TBecircacutebuilt=\TBecircacute
```

If running under `pdflatex`, allow the actual UTF-8 character to be used in text. `\DeclareUnicodeCharacter` is not defined under the Unicode engines. A possible definition is given here: https://tex.stackexchange.com/questions/195458, but let's wait to see if we need it.

```
\ifx\DeclareUnicodeCharacter\undefined\else
  \DeclareUnicodeCharacter{1EBF}{\TBecircacute} % for the literal character
\fi
```

Under Unicode engines, by default, we'll use the character from the font, if it's there.

```
\ifTBunicodeengine
  \AtBeginDocument{% in case a different font gets loaded
    % \iffontchar is from e-TeX; safe to use under Unicode engines.
    \iffontchar\font"1EBF
      % Use the font's version by default; perhaps we're using newcm or
      % another font that has the acute offset to the right.
      \def\TBecircacute{\char"1EBF }%
    \fi
  }%
\fi % (end TBunicodeengine)

% Finally, we can define \Thanh itself, using \TBecircacute.
\def\Thanh{H\`an~Th\TBecircacute~Th\`anh}%
```

One more complication: add tagging for Thanh using the UTF-8 characters. This comes from Frank Mittelbach, written for https://tug.org/TUGboat/tb47-1/tb145sojka-doctor-mittelbach.pdf.

```
\ExplSyntaxOn
\IfPDFManagementActiveT{
\AddToHook{cmd/Thanh/before}
 {
   \mode_leave_vertical:
   \tag_mc_end_push:
```

```
  % \char_generate would work for the agrave (E0) characters,
  % but it does not work for 1EBF under pdflatex ("out of engine range").
  % We can't use ^^^^ since it's not supported by pdftex. So it seems
  % best, and perhaps necessary, to use the literal UTF-8 characters.
  % What we'd prefer is to just use ASCII to specify the characters, like:
  %   actualtext=H^^e0n~ Th^^^^1ebf~ Th^^e0nh
  % but oh well.
  \tag_struct_begin:n {tag=Span,actualtext=Hàn~ Thế~ Thành}
  \tag_mc_begin:n{}
 }
\AddToHook{cmd/Thanh/after}
 {
  \tag_mc_end:
  \tag_struct_end:
  \tag_mc_begin_pop:n{}
 }
} % (end PDFManagementactiveT)
\ExplSyntaxOff
```

See the TUGboat test file tubibthanh.tex (and the Makefile) to exercise all this.

`\tubsentencespace` Occasionally, notably after citations that need to come after a sentence-ending period, we want to tell TEX that it's still at the end of a sentence. As in: `... whatever. \cite{foo}\tubsentencespace` This happens when, e.g., the reference applies to more than the final sentence. Also can be needed when `\@` cannot be used because the sentence-ending punctuation itself occurs inside a control sequence that prevents it.

```
\def\tubsentencespace{\spacefactor=3000{}\space\ignorespaces}
```

`\tubdots` Latin Modern and many other fonts irritatingly make the Unicode ellipsis character (U+2026) a single character's width, typically more squashed together than three period characters. This just looks wrong. It is too painful to try to redefine in general, but provide the normal definition to reset in individual papers with, e.g.: `\ifx\tubdots\undefined \else \let\dots\tubdots \let\ldots\tubdots \fi`

The `plain.tex` definition does not have the small space before the first dot, but that space makes the result look better in cases like `[\tubdots]` where something other than a space comes before the ellipsis.

```
\def\tubdots{\ifmmode\mathellipsis\else
    \kern\fontdimen3\font % space before first dot
   .\kern\fontdimen3\font
   .\kern\fontdimen3\font
   .\kern\fontdimen3\font\fi}
```

`\allowhyphens` Hyphen control: first, we save (via `\edef`) the hyphenpenalties in `\allowhyphens`. This allows us to permit hyphens temporarily in things like `\netaddresses`, which typically occur when `\raggedright` is set, but which need to be allowed to break at their artificial discretionaries.

```
\edef\allowhyphens{\noexpand\hyphenpenalty\the\hyphenpenalty\relax
  \noexpand\exhyphenpenalty\the\exhyphenpenalty\relax}
\def\nohyphens{\hyphenpenalty\@M\exhyphenpenalty\@M}
```

3.6 Utility registers and definitions

We define a few scratch registers (and the like) for transient use; they're all paired: an internal one (`\T@st*`) and an external one (`\Test*`).

Comment: Exercise for an idle day: find whether all these are necessary, or whether we can use the LaTeX temporaries for some (or all) of the `\T@st*` ones.

Comment: (bb) All these registers are used in the plain version, `tugboat.sty`.

```
\newbox\T@stBox                 \newbox\TestBox
\newcount\T@stCount             \newcount\TestCount
\newdimen\T@stDimen             \newdimen\TestDimen
\newif\ifT@stIf                 \newif\ifTestIf
```

Control sequence existence test, stolen from TeXbook exercise 7.7 (note that this provides functionality that in some sense duplicates something within LaTeX).

```
\def\ifundefined#1{\expandafter\ifx\csname#1\endcsname\relax }
```

LaTeX conventions which are also useful here.

```
⟨*!latex⟩
  \let\@@input\input
  \def\iinput#1{\@@input#1 }
  \def\@inputcheck{\if\@nextchar\bgroup
    \expandafter\iinput\else\expandafter\@@input\fi}
  \def\input{\futurelet\@nextchar\@inputcheck}
⟨/!latex⟩
```

Smashes repeated from AMS-TeX; plain TeX implements only full `\smash`.

```
\newif\iftop@           \newif\ifbot@
\def\topsmash{\top@true\bot@false\smash@}
\def\botsmash{\top@false\bot@true\smash@}
\def\smash{\top@true\bot@true\smash@}
\def\smash@{\relax\ifmmode\def\next{\mathpalette\mathsm@sh}%
        \else\let\next\makesm@sh\fi \next }
\def\finsm@sh{\iftop@\ht\z@\z@\fi\ifbot@\dp\z@\z@\fi\box\z@}
```

Vertical 'laps'; cf. `\llap` and `\rlap`

```
\long\def\ulap#1{\vbox to \z@{\vss#1}}
\long\def\dlap#1{\vbox to \z@{#1\vss}}
```

And centered horizontal and vertical 'laps'

```
\def\xlap#1{\hb@xt@\z@{\hss#1\hss}}
\long\def\ylap#1{\vbox to \z@{\vss#1\vss}}
\long\def\zlap#1{\ylap{\xlap{#1}}}
```

Avoid unwanted vertical glue when making up pages.

```
\def\basezero{\baselineskip\z@skip \lineskip\z@skip}
```

Empty rules for special occasions

```
\def\nullhrule{\hrule \@height\z@ \@depth\z@ \@width\z@ }
\def\nullvrule{\vrule \@height\z@ \@depth\z@ \@width\z@ }
```

Support ad-hoc strut construction.

```
\def\makestrut[#1;#2]{\vrule \@height#1 \@depth#2 \@width\z@ }
```

Construct box for figure pasteup, etc.; height = #1, width = #2, rule thickness = #3

```
\def\drawoutlinebox[#1;#2;#3]{\T@stDimen=#3
        \vbox to#1{\hrule \@height\T@stDimen \@depth\z@
            \vss\hb@xt@#2{\vrule \@width\T@stDimen
                \hfil\makestrut[#1;\z@]%
                \vrule \@width\T@stDimen}\vss
            \hrule \@height\T@stDimen \@depth\z@}}
```

Today's date, to be printed on drafts. Based on TEXbook, p.406.

```
⟨*!latex⟩
\def\today{\number\day\space \ifcase\month\or
        Jan \or Feb \or Mar \or Apr \or May \or Jun \or
        Jul \or Aug \or Sep \or Oct \or Nov \or Dec \fi
        \number\year}
⟨/!latex⟩
```

Current time; this may be system dependent!

```
\newcount\hours
\newcount\minutes
\def\SetTime{\hours=\time
        \global\divide\hours by 60
        \minutes=\hours
        \multiply\minutes by 60
        \advance\minutes by-\time
        \global\multiply\minutes by-1 }
\SetTime
\def\now{\ifnum\hours<10 0\fi\number\hours:%
         \ifnum\minutes<10 0\fi\number\minutes}
\def\Now{\today\ \now}
\newif\ifPrelimDraft    % true if ([draft] or [preprint] or pageno>900)
\def\midrtitle{} % center of running heads
\def\rtitlenexttopage{\ifPrelimDraft \textsl{\small draft: \Now}\fi}
%\def\rtitlenexttopage{\ifnum\value{page}>900 \textsl{\small draft: \Now}\fi}
```

Sometimes we want to refer to the pages of another article in the same issue. `tugboat.dates` makes the real definition; here we define a placeholder so it won't be undefined when we send the source back to the author.

```
\let\thisissuepageref\empty
```

3.7 Ragged right and friends

`\raggedskip` `\raggedstretch` `\raggedparfill` `\raggedspaces`

Plain TEX's definition of `\raggedright` doesn't permit any stretch, and results in too many overfull boxes. We also turn off hyphenation. This code lies somewhere between that of Plain TEX and of LATEX.

```
\newdimen\raggedskip    \raggedskip=\z@
\newdimen\raggedstretch \raggedstretch=5em    % ems of font set now (10pt)
\newskip\raggedparfill  \raggedparfill=\z@\@plus 1fil
\def\raggedspaces{\spaceskip=.3333em \relax \xspaceskip=.5em \relax }
```

`\raggedright` `\raggedleft` `\raggedcenter` `\normalspaces`

Some applications may have to add stretch, in order to avoid all overfull boxes. We define the following uses of the above skips, etc.

```
\def\raggedright{%
  \nohyphens \raggedspaces
```

```
  \rightskip=\raggedskip\@plus\raggedstretch
  \parfillskip=\raggedparfill
}
\def\raggedleft{%
  \nohyphens \raggedspaces
  \leftskip=\raggedskip\@plus\raggedstretch
  \parfillskip=\z@skip
  \let\\ \@centercr % else tabulararray fails,
                    % https://github.com/lvjr/tabularray/issues/348
}
\def\raggedcenter{%
  \nohyphens \raggedspaces
  \leftskip=\raggedskip\@plus\raggedstretch
  \rightskip=\leftskip
  \parindent=\z@
  \parfillskip=\z@skip
}
%
% Undo |\raggedspaces|.
\def\normalspaces{\spaceskip\z@skip \xspaceskip\z@skip}
```

`\tubjustifiedpar` Undo the `\raggedright` (or other such) settings, restoring normality.

```
\def\tubjustifiedpar{\rightskip=0pt \parfillskip=0pt plus1fil
                     \allowhyphens \normalspaces}
```

3.8 Assorted user-level markup

We provide a new definition of `~` by redefining `\ ` (`\DeclareRobustCommand` doesn't mind redefinition, fortunately). This is based on the version in AMS-TEX—the LATEX 2_ε version (`ltspace.dtx`) has `\leavevmode` and does not do anything with the surrounding space(s). Our version messes up with the `\pfill` used in `doc`-generated indexes (github.com/latex3/latex2e/issues/75), but later (2018++) versions of `doc` should protect against our redefinition.

```
\let\latexnobreakspace=\nobreakspace
\DeclareRobustCommand{\nobreakspace}{\unskip\nobreak\ \ignorespaces}
```

Plain TEX defines `\newbox` as `\outer`. We solemnly preserve the following, which removes the `\outer`ness; of course, we carefully exclude it from what we generate... (`\outer`ness is a spawn of the devil, is it not? Barbara Beeton responded to the previous sentence "`\outer`ness has its place: it avoids register buildup, hence running out of memory". In another context, David Carlisle remarked that an error control mechanism that causes more confusing errors than it prevents is rather a poor one. This is perhaps not the place to conduct a serious debate...)

```
\def\boxcs#1{\box\csname#1\endcsname}
\def\setboxcs#1{\setbox\csname#1\endcsname}
\def\newboxcs#1{\expandafter\newbox\csname#1\endcsname}
\let\gobble\@gobble
\def\vellipsis{%
  \leavevmode\kern0.5em
  \raise\p@\vbox{\baselineskip6\p@\vskip7\p@\hbox{.}\hbox{.}\hbox{.}}
  }
```

```
% \bull doesn't work with tagging; requires ActualText using, e.g.,
% accsup, but the ActualText is ignored since it's just a rule.
% (Lots of our other commands also are not properly tagged.)
% https://github.com/latex3/tagging-project/pull/535
\def\bull{\vrule \@height 1ex \@width .8ex \@depth -.2ex }
\DeclareRobustCommand{\cents}{\textcent}
\def\tubcentsold{{\rm\raise.2ex\rlap{\kern.05em$\scriptstyle/$}c}}
\def\careof{\leavevmode\hbox{\raise.75ex\hbox{c}\kern-.15em
                /\kern-.125em\smash{\lower.3ex\hbox{o}}} \ignorespaces}
\def\Dag{\raise .6ex\hbox{$\scriptstyle\dagger$}}
%
\DeclareRobustCommand{\sfrac}[1]{\@ifnextchar/{\@sfrac{#1}}%
                                                {\@sfrac{#1}/}}
\def\@sfrac#1/#2{\leavevmode\kern.1em\raise.5ex
         \hbox{$\m@th\mbox{\fontsize\sf@size\z@
                           \selectfont#1}$}\kern-.1em
         /\kern-.15em\lower.25ex
          \hbox{$\m@th\mbox{\fontsize\sf@size\z@
                            \selectfont#2}$}}
%
% don't stay bold in description items, bold italic is too weird.
\DeclareRobustCommand\meta[1]{%
  \ensuremath{\langle}%
  \ifmmode \expandafter\mbox \fi % if in math
  {\it #1\/}% no typewriter italics, please
  \ensuremath{\rangle}%
}
%
% Use \tt rather than \texttt because italic typewriter is just too strange
% and upright works well enough in both italic and bold contexts.
% Would be nice to change catcode of _ for \LaTeX3, but we don't.
%
% By the way, it would be possible to substitute typewriter slanted for
% typewriter italic in general:
% \url{https://tex.stackexchange.com/questions/692277}.
% But it feels like that is too intrusive a change, even though in
% practice we always prefer tt slanted.
\DeclareRobustCommand{\cs}[1]{\texorpdfstring
  {{\tt \char`\\#1}\@}%
  {\textbackslash #1}%
}
%
% This command was defined much later than the others around here, so
% let's not conflict with any existing definitions that might be out there.
% Don't allow hyphenations or other line breaks.
\DeclareRobustCommand{\tubbraced}[1]{\texorpdfstring
  {\mbox{\texttt{\char`\{#1\char`\}}}}%
  {\textbraceleft #1\textbraceright}%
}
%
% Literal text, such as class names, package names, filenames, etc,
% Trying to define separate commands for each seems impossible and pointless.
% Usually we don't want hyphenation or any other kind of break.
\DeclareRobustCommand{\tbcode}[1]{\mbox{\texttt{#1}}}
```

```
%
% On the other hand, sometimes we need to break such code fragments.
% If |hyperref| is loaded, we want |\nolinkurl|, else just |\url|.
\AtBeginDocument{%
\ifx\nolinkurl\undefined
  \DeclareRobustCommand{\tbcodebreak}{\url}
\else
  \DeclareRobustCommand{\tbcodebreak}{\nolinkurl}
\fi
}
%
% Not sure why we ever want this instead of LaTeX's \, (using \kern),
% but fine, just keeping it.
\DeclareRobustCommand{\thinskip}{\hskip 0.16667em\relax}
%
% Ah, urls. Nowadays, we like the visible url to not have any protocol,
% if it is \texttt{http://} or \texttt{https://}. But we need to include
% the protocol if we are making live links, since a string like
% \texttt{tug.org/whatever} will be taken as a local filename by
% browsers and PDF readers. Since we need to check for
% \texttt{hyperref}, make the definition \cs{AtBeginDocument}. In the
% end, \cs{tbsurl}\tubbraced{foo} produces \texttt{https://foo} and
% \cs{tbhurl}\tubbraced{foo}  produces \texttt{http://foo}.
\AtBeginDocument{%
\ifx\hyper@normalise\undefined
  \ifx\url\undefined      % define our own simplistic non-hyperref \url
    \def\url{\begingroup  % might as well catch common special chars
      \catcode`\#=12 \catcode`\$=12 \catcode`\%=12 \catcode`\^=12
      \catcode`\&=12 \catcode`\_=12 \catcode`\~=12
      \finish@tub@url}
  \def\finish@tub@url#1{\tt #1\endgroup}
  \fi
  \let\tburl\url  % no hyperref, so just \url is fine;
  \let\tbsurl\url % \let instead of \def so we can still
  \let\tbhurl\url % use \def\url{\tbsurl} without infloop.
\else
  % This hyperref hook-in is due to Ulrike Fischer.
  % \url{https://github.com/latex3/hyperref/issues/125}.
  % \tb[sh]url@ are defined next.
  \DeclareRobustCommand*{\tburl}{\tbsurl}%
  \DeclareRobustCommand*{\tbsurl}{\hyper@normalise\tbsurl@}%
  \DeclareRobustCommand*{\tbhurl}{\hyper@normalise\tbhurl@}%
\fi
}
%
% Outside \AtBeginDocument, back at the top level of the dtx, we
% turn on expl syntax for the main definitions of \tb[sh]url.  We want
% to auto-remove an explicit protocol in case it
% was given.
%
% Giving \verb|https://| to \cs{tbhurl} generates an invalid link; in
% practice there's no use for that so we don't bother to check for it.
%
\ExplSyntaxOn
```

```
% Helper function to test whether #1 matches the leading characters of #2.
\cs_new:Npn \str_if_starts_with:nnTF #1#2
  {
    % #1 = prefix to test
    % #2 = full string
    %
    % compare the first N characters (tokens) of the full string #2,
    %   where N is the length of the prefix #1,
    % to the prefix.
    \str_if_eq:eeTF
      { \str_range:nnn {#2} {1} { \str_count:n {#1} } }
      {#1}
  }

% the main function to handle url #1.
\def\tbsurl@#1  % https
 {
  \group_begin:
   % URL encoding for when pdf management (\DocumentMetadata) is active,
   % else a % in the input url ends up as a %25 in the active link.
   % From Ulrike, 4apr26.
   \IfPDFManagementActiveT{
      \bool_if:NTF \l__hyp_href_url_encode_bool
     {
       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/URI}
     }
     {
       \tl_set:Nn \l__hyp_text_enc_uri_print_tl {utf8/string}
     }
   }% end pdf management
   %
   \str_set:Nn \l_tmpa_str {#1}
   % this checks if http:// is contained anywhere within the argument url,
   % but that was not good enough, since http:// might occur inside the url,
   % e.g., https://web.archive.org/web/20090809184749/http://www.eco-log.de/
   %   \str_if_in:NnTF \l_tmpa_str {http://}
   %
   % Instead we check for "starts with http://".
   % Another fix would have been to insert a constant string "foo" before
   % the argument and the prefix, but since the clean solution is at hand,
   % might as well use it.
   \str_if_starts_with:nnTF {http://} {#1}
    {
      \tbhurl@{#1} % if http, redirect to remove protocol
      % this version prints the http, as we originally thought was better.
      % \expandafter\hyper@linkurl
      % \expandafter{\expandafter\Hurl\expandafter{\l_tmpa_str}}{\l_tmpa_str}
    }
    {
      \str_remove_once:Nn \l_tmpa_str {https://}
      \expandafter\hyper@linkurl
        \expandafter{\expandafter\Hurl\expandafter{\l_tmpa_str}}
                                                          {https://\l_tmpa_str}
    }
```

```
  \group_end:
 } % end \tbsurl@
%
% explicit http, handle similarly but don't bother checking for https
\def\tbhurl@#1
 {
   \str_set:Nn\l_tmpa_str{#1}
   \str_remove_once:Nn \l_tmpa_str {http://}
   \expandafter\hyper@linkurl\expandafter{\expandafter\Hurl\expandafter
                                          {\l_tmpa_str}}{http://\l_tmpa_str}
 }
\ExplSyntaxOff
%
% Now let's use those macros for putting a url into a simple
% ragged-right footnote.
\def\tburlfootnote{\tbsurlfootnote}
\def\tbsurlfootnote#1{\unskip\footnote{\raggedright\tbsurl{#1}}}
\def\tbhurlfootnote#1{\unskip\footnote{\raggedright\tbhurl{#1}}}
%
% Close up space between footnote mark and punctuation (``pre-punctuation'').
\DeclareRobustCommand{\tbppkernfoot}{\tubthinnerspace}

% Make \! work in text mode, for older LaTeX.
\DeclareRobustCommand{\!}{\ifmmode\mskip-\thinmuskip \else\kern-0.16667em \fi}
%
% Half a thinspace, positive and negative. Should have named these
% \cs{tb} instead of \cs{tub}, but not worth changing now.
\DeclareRobustCommand{\tubthinnerspace}
  {\ifmmode\mskip.5\thinmuskip \else\kern0.08333em \fi}
\DeclareRobustCommand{\tubthinnerspaceneg}
  {\ifmmode\mskip-.5\thinmuskip \else\kern-0.08333em \fi}
%
% Half a smallskip.
\DeclareRobustCommand{\tubsmallerskip}
  {\vskip 1.5pt plus .75pt minus .75pt\relax}
%
```

We play a merry game with dashes, providing all conceivable options of breakability before and after.

```
\def\endash{--}
\def\emdash{\endash-}
\def\d@sh#1#2{\unskip#1\thinskip#2\thinskip\ignorespaces}
\def\dash{\d@sh\nobreak\endash}
\def\Dash{\d@sh\nobreak\emdash}
\def\ldash{\d@sh\empty{\hbox{\endash}\nobreak}}
\def\rdash{\d@sh\nobreak\endash}
\def\Ldash{\d@sh\empty{\hbox{\emdash}\nobreak}}
\def\Rdash{\d@sh\nobreak\emdash}
```

Hacks to permit automatic hyphenation after an actual hyphen, or after a slash.

```
\def\hyph{-\penalty\z@\hskip\z@skip }
\def\slash{/\penalty\z@\hskip\z@skip }
```

Adapted from `comp.text.tex` posting by Donald Arseneau, 26 May 93. LaTeX 2$_\varepsilon$-isation added by Robin Fairbairns. Destroys both the `TestCount`s.

```
\def\nth#1{%
    \def\reserved@a##1##2\@nil{\ifcat##1n%
           0%
         \let\reserved@b\ensuremath
      \else##1##2%
         \let\reserved@b\relax
      \fi}%
    \TestCount=\reserved@a#1\@nil\relax
    \ifnum\TestCount <0 \multiply\TestCount by\m@ne \fi % subdue negatives
    \T@stCount=\TestCount
    \divide\T@stCount by 100 \multiply\T@stCount by 100
    \advance\TestCount by-\T@stCount     % n mod 100
    \ifnum\TestCount >20 \T@stCount=\TestCount
      \divide\T@stCount by 10 \multiply\T@stCount by 10
      \advance\TestCount by-\T@stCount   % n mod 10
    \fi
     \reserved@b{#1}%
       \textsuperscript{\ifcase\TestCount th%    0th
                        \or   st%                1st
                        \or   nd%                2nd
                        \or   rd%                3rd
                        \else th%                nth
                        \fi}%
}
```

3.9 Reviews

Format information on reviewed items for book review articles. For the LaTeX 2$_\varepsilon$ version, we follow Fairbairns' maxim, and define something that can even look like a LaTeX macro...

```
\def\Review{\@ifnextchar:{\@Review}{\@Review:}}
\def\@Review:{\@ifnextchar[%]
  {\@Rev}%
  {\@Rev[Book review]}}
\def\@Rev[#1]#2{{\ignorespaces#1\unskip:\enspace\ignorespaces
                                          \slshape\mdseries#2}}
\def\reviewitem{\addvspace{\BelowTitleSkip}%
  \def\revauth##1{\def\therevauth{##1, }\ignorespaces}%
  \def\revtitle##1{\def\therevtitle{{\slshape##1}. }\ignorespaces}%
  \def\revpubinfo##1{\def\therevpubinfo{##1.}\ignorespaces}%
}
\def\endreviewitem{{\noindent\interlinepenalty=10000
  \therevauth\therevtitle\therevpubinfo\endgraf}%
  \vskip\medskipamount
}
\def\titleref#1{{\slshape\frenchspacing#1\nocorr}}
\let\booktitle=\titleref % older name
```

3.10 Dates, volume and issue numbers, etc.

Dates and other items which identify the volume and issue. `\issueseqno` is a sequential issue number starting from the first issue published; volume 15,4 has `\issueseqno=45`.

```
\vol 19, 1.
\issdate March 1998.
\issueseqno=58
```

To use:

Starting with volume 23 (nominal 2002), we have `\issyear` instead of `\issdate`, because issues don't have months any more.

For production, these are set in a separate file, `tugboat.dates`, which is issue-specific.

```
\newcount\issueseqno \issueseqno=-1
\def\v@lx{\gdef\volx{Volume~\volno~(\volyr), No.~\issno}}
\def\volyr{}
\def\volno{}
\def\vol#1, #2.{%
        \gdef\volno{#1}%
        \gdef\issno{#2}%
        \setbox\TestBox=\hbox{\volyr}%
        \ifdim \wd\TestBox > .2em \v@lx \fi }
\def\issyear#1.{%
        \gdef\issdt{#1}\gdef\volyr{#1}%
        \gdef\bigissdt{#1}%
        \setbox\TestBox=\hbox{\volno}%
        \ifdim \wd\TestBox > .2em \v@lx \fi }
\def\issdate#1#2 #3.{%
        \gdef\issdt{#1#2 #3}\gdef\volyr{#3}%
        \gdef\bigissdt{#1{\smc\uppercase{#2}} #3}%
        \setbox\TestBox=\hbox{\volno}%
        \ifdim \wd\TestBox > .2em \v@lx \fi }
% The \vol command must be invoked precisely like this, including spaces.
% Since we are the only ones who write it, we can be strict.
\vol 0, 0.
\issdate Thermidor, 9999.
```

(The curious may like to know that *Thermidor* was one of the French revolutionary month names.)

For LaTeX use, define a version of the issue declaration that can take or leave the old plain syntax

```
⟨!latex⟩\def\tubissue#1(#2)%
⟨*latex⟩
\def\tubissue#1{\@ifnextchar(%)
  {\@tubissue@b{#1}}
  {\@tubissue@a{#1}}}
\def\@tubissue@b#1(#2){\@tubissue@a{#1}{#2}}
\def\@tubissue@a#1#2%
⟨/latex⟩
  {\TUB~#1, no.~#2}
```

TUGboat conventions include the sequential issue number in the file name. Permit this to be incorporated into file names automatically. If issue number = 11, `\Input filnam` will read `tb11filnam.tex`

```
\def\infil@{\jobname}
\def\Input #1 {\ifnum\issueseqno<0
    \def\infil@{#1}%
  \else
    \def\infil@{tb\number\issueseqno#1}
  \fi
  \edef\jobname{\infil@}\@readFLN
  \@@input \infil@\relax
  \if@RMKopen
    \immediate\closeout\@TBremarkfile\@RMKopenfalse
  \fi
}
```

\TBremarks are things that need to be drawn to the attention of the editors; the conscientious author will include such things in the article file. By default, remarks are suppressed, but their appearance may be enabled by the \TBEnableRemarks command, which can be included in the configuration file ltugboat.cfg (or ltugproc.cfg, if that's what we're at).

```
\newif\if@RMKopen        \@RMKopenfalse
\newwrite\@TBremarkfile
\def\@TBremark#1{%
  \if@RMKopen
  \else
    \@RMKopentrue\immediate\openout\@TBremarkfile=\infil@.rmk
  \fi
  \toks@={#1}%
  \immediate\write\@TBremarkfile{^^J\the\toks@}%
  \immediate\write16{^^JTBremark:: \the\toks@^^J}%
}
```

We initialise \TBremark to ignore its argument (this used to involve a \TBremarkOFF which was cunningly defined exactly the same as \gobble)

```
\let\TBremark=\gobble
```

\TBEnableRemarks simply involves setting \TBremark to use the functional \@TBremark defined above.

```
\def\TBEnableRemarks{\let\TBremark\@TBremark}
```

For marking locations in articles that pertain to remarks in another file of editorial comments

```
\def\TUBedit#1{}
```

For using different filenames in the production process than those supplied by authors

```
\def\TUBfilename#1#2{\expandafter\def\csname file@@#1\endcsname{#2}}
\newread\@altfilenames
\def\@readFLN{\immediate\openin\@altfilenames=\jobname.fln
  \ifeof\@altfilenames\let\@result\relax\else
  \def\@result{\@@input\jobname.fln }\fi
  \immediate\closein\@altfilenames
  \@result}
\@readFLN
\everyjob=\expandafter{\the\everyjob\@readFLN}
\InputIfFileExists{\jobname.fln}%
```

```
  {\TBInfo{Reading alternative file \jobname.fln}}
  {}
```

The following needs to work entirely in TEX's mouth

```
\def\@tubfilename#1{\expandafter\ifx\csname file@@#1\endcsname\relax
  #1\else\csname file@@#1\endcsname\fi}
\def\fileinput#1{\@@input\@tubfilename{#1} }
```

Write out (both to a file and to the log) the starting page number of an article, to be used for cross references and in contents. `\pagexref` is used for articles fully processed in the *TUGboat* run. `\PageXref` is used for 'extra' pages, where an item is submitted as camera copy, and only running heads (at most) are run.

```
⟨*!latex⟩
\def\pagexrefON#1{%
        \write-1{\def\expandafter\noexpand\csname#1\endcsname{\number\pageno}}%
        \write\ppoutfile{%
                \def\expandafter\noexpand\csname#1\endcsname{\number\pageno}}%
        }
\def\PageXrefON#1{%
        \immediate\write-1{\def\expandafter
                        \noexpand\csname#1\endcsname{\number\pageno}}%
        \immediate\write\ppoutfile{\def\expandafter
                        \noexpand\csname#1\endcsname{\number\pageno}}}
⟨/!latex⟩
⟨*latex⟩
\def\pagexrefON#1{%
        \write-1{\def\expandafter\noexpand\csname#1\endcsname{\number\c@page}}%
        \write\ppoutfile{%
                \def\expandafter\noexpand\csname#1\endcsname{\number\c@page}}%
        }
\def\PageXrefON#1{%
        \immediate\write-1{\def\expandafter
                        \noexpand\csname#1\endcsname{\number\c@page}}%
        \immediate\write\ppoutfile{\def\expandafter
                        \noexpand\csname#1\endcsname{\number\c@page}}}
⟨/latex⟩
\def\pagexrefOFF#1{}
\let\pagexref=\pagexrefOFF
\def\PageXrefOFF#1{}
\let\PageXref=\PageXrefOFF
\def\xreftoON#1{%
  \ifundefined{#1}%
    ???\TBremark{Need cross reference for #1.}%
  \else\csname#1\endcsname\fi}
\def\xreftoOFF#1{???}
\let\xrefto=\xreftoOFF
```

`\TBdriver` 'marks code for use when articles are run together in a driver file'. Since we don't yet have a definition of that arrangement, we don't have a definition of `\TBdriver`. Its argument (which one presumes was intended as the code for this unusual state) is just gobbled.

```
\let\TBdriver\gobble
```

Hyphenation exceptions. We read our own full `ushyphex.tex` (generated from `tb0hyf.tex`) if it's available. The additional exceptions are nearly all included in the file, but keep defining them anyway, since we have for many years.

But do not define any exceptions if `\tubomithyphenations` is defined. This is needed for the `hyf` articles themselves.

```
\ifx\tubomithyphenations\@thisisundefined
\InputIfFileExists{ushyphex.tex}{}{} % ok if it's missing
\hyphenation{Del-a-ware Dijk-stra Duane Eijk-hout
  Flor-i-da Free-BSD Ghost-script
  Hara-lam-bous Jac-kow-ski Ja-pa-nese Karls-ruhe Lua-Meta
  Mac-OS Math-Sci-Net
  Net-BSD Open-BSD Open-Office
  Pak-i-stan Post-Script Rich-ard Skoup South-all
  Vieth VM-ware Win-Edt
  acro-nym acro-nyms analy-sis ap-pen-di-ces ap-pen-dix asyn-chro-nous
  bib-lio-graph-i-cal bit-map bit-mapped bit-maps buf-fer buf-fers bool-ean
  col-umns com-put-able com-put-abil-ity
  data-base data-bases
   de-allo-cate de-allo-cates de-allo-cated de-allo-ca-tion
   de-riv-a-tive de-riv-a-tives de-riv-a-ble der-i-va-tion dis-trib-ut-able
  es-sence
  fall-ing
  half-way
  in-fra-struc-ture
  key-note
  long-est
  ma-gyar man-u-script man-u-scripts meta-table meta-tables
   mne-mon-ic mne-mon-ics mono-space mono-spaced
  name-space name-spaces
  off-line over-view
  pal-ettes par-a-digm par-a-dig-matic par-a-digms
   pipe-line pipe-lines
   plug-in plug-ins pres-ent-ly pro-gram-mable
  re-allo-cate re-allo-cates re-allo-cated re-printed
  set-ups se-vere-ly spell-ing spell-ings stand-alone strong-est
   sub-ex-pres-sion sub-tables sur-gery syn-chro-ni-city syn-chro-nous
  text-height text-length text-width
   time-stamp time-stamped time-stamps
  vis-ual vis-ual-ly
  which-ever white-space white-spaces wide-spread wrap-around
}
\fi
⟨!latex⟩\restorecat\@
⟨/common⟩
⟨*classtail⟩
\PrelimDrafttrue
```

3.11 Page dimensions, glue, penalties, etc.

```
\textheight 54pc     % 648pt = 645.58bp = 8.97in
\textwidth 39pc      % 468pt = 466.25bp = 6.48in
\columnsep 1.5pc     %  18pt =  17.93bp =  .249in
\columnwidth 18.75pc % 225pt = 224.16bp = 3.11in
```

```
\hfuzz 1pt
\parindent \normalparindent % 20pt
\parskip \z@ % \@plus\p@
\leftmargini 2em
\leftmarginv .5em
\leftmarginvi .5em
\oddsidemargin \z@
\evensidemargin \z@
\topmargin -2.5pc   % 30pt = 29.89bp = .415in
\headheight 12\p@
\headsep 20\p@
\marginparwidth 48\p@
\marginparsep 10\p@
\partopsep=\z@
\topsep=3\p@\@plus\p@\@minus\p@
\parsep=3\p@\@plus\p@\@minus\p@
\itemsep=\parsep
%
% The width of one column plus gutter (=243pt =242.09bp) is useful sometimes.
\newdimen\tubcolwidthandgutter
  \tubcolwidthandgutter=\columnwidth
  \advance\tubcolwidthandgutter by \columnsep
%
% Ordinarily we typeset in two columns, but the onecolumn option
% goes to one.  In which case we want to center the text block on an
% 8.5in width, given the default 72.27pt offset with margins of zero.
% We are always in LaTeX's twoside mode because of how we load article,
% and this is a good thing, since we want different headings.
\if@tubtwocolumn \twocolumn \else
  \onecolumn
  \textwidth=34pc
  \oddsidemargin=30.8775pt
  \evensidemargin=\oddsidemargin
\fi
%
\newdimen\pagewd        \pagewd=\textwidth
\newdimen\trimwd        \trimwd=\pagewd
\newdimen\trimlgt       \trimlgt=11in
\newdimen\headmargin    \headmargin=3.5pc
```

Don't go to a float page so soon. Not all of these are relevant to all articles, but we may as well set them all.

```
\renewcommand{\topfraction}{.9}  % don't go to a float page so soon
\renewcommand{\dbltopfraction}{.9}
\renewcommand{\bottomfraction}{.7}
\renewcommand{\textfraction}{.1}
\renewcommand{\floatpagefraction}{.8}
\renewcommand{\dblfloatpagefraction}{.8} % the most common one used
```

3.12 Messing about with the LaTeX logo

Barbara Beeton's pleas for LaTeX logos that look right in any font shape provoked me to generate the following stuff that is configurable.

Here's the command for the user to define a new version. The arguments are font family, series and shape, and then the two kern values used in placing the raised 'A' of LaTeX.

```
\newcommand{\DeclareLaTeXLogo}[5]{\expandafter\def
  \csname @LaTeX@#1/#2/#3\endcsname{{#4}{#5}}}
```

The default values are as used in the source of LaTeX itself:

```
\def\@LaTeX@default{{.36}{.15}}
```

More are defined in the initial version, for bold CM sans (which is used as `\SecTitleFont`), and CM italic medium and bold, and Bitstream Charter (which Nelson Beebe likes to use). Duplicate for Latin Modern.

```
\DeclareLaTeXLogo{cmss}{bx}{n}{.3}{.15}
\DeclareLaTeXLogo{lmss}{bx}{n}{.3}{.15}
%
\DeclareLaTeXLogo{cmr}{m}{it}{.29}{.2}
\DeclareLaTeXLogo{lmr}{m}{it}{.29}{.2}
%
\DeclareLaTeXLogo{cmr}{m}{sl}{.29}{.15}
\DeclareLaTeXLogo{lmr}{m}{sl}{.29}{.15}
%
\DeclareLaTeXLogo{cmr}{bx}{it}{.29}{.2}
\DeclareLaTeXLogo{lmr}{bx}{it}{.29}{.2}
%
\DeclareLaTeXLogo{cmr}{bx}{sl}{.29}{.2}
\DeclareLaTeXLogo{lmr}{bx}{sl}{.29}{.2}
%
\DeclareLaTeXLogo{bch}{m}{n}{.2}{.08}
\DeclareLaTeXLogo{bch}{m}{it}{.2}{.08}
```

Redefine `\LaTeX` to choose the parameters for the current font, or to use the default value otherwise:

```
\DeclareRobustCommand{\LaTeX}{\expandafter\let\expandafter\reserved@a
  \csname @LaTeX@\f@family/\f@series/\f@shape\endcsname
  \ifx\reserved@a\relax\let\reserved@a\@LaTeX@default\fi
  \expandafter\@LaTeX\reserved@a}
```

Here's the body of what was originally `\LaTeX`, pulled out with its roots dripping onto the smoking ruin of original LaTeX, and then bits stuck in on the side.

`\@LaTeX@default` provides parameters as one finds in the original; other versions are added as needed.

```
\newcommand{\@LaTeX}[2]{%
  %\wlog{latex logo family=\f@family/\f@series/\f@shape -> #1, #2.}%
  L\kern-#1em
  {\sbox\z@ T%
   \vbox to\ht0{\hbox{$\m@th$%
                       \csname S@\f@size\endcsname
                       \fontsize\sf@size\z@
                       \math@fontsfalse\selectfont
                       A}%
                \vss}%
  }%
```

```
  \kern-#2em%
  \TeX}
```

3.13 Authors, contributors, addresses, signatures

An article may have several authors (of course), so we permit an `\author` command for each of them. The names are then stored in a set of `\csname`s called `\author1`, `\author2`, ... Similarly, there are several `\address<n>` and `\netaddress<n>` and `\PersonalURL<n>` and `\ORCID<n>` commands set up for each article.

Comment: [RF] I would like to make provision for several authors at the same address, but (short of preempting the `*` marker, which it would be nice to retain so as to preserve compatibility with the `plain` style) I'm not sure how one would signal it. [KB] Current kludges for multiple author affiliations in tb143rishi-xml-first, tb140rishi-elsarticle, tb128ruckert-hint, etc.

```
\def\theauthor#1{\csname theauthor#1\endcsname}
\def\theaddress#1{\csname theaddress#1\endcsname}
\def\thenetaddress#1{\csname thenetaddress#1\endcsname}
\def\thePersonalURL#1{\csname thePersonalURL#1\endcsname}
\def\theORCID#1{\csname theORCID#1\endcsname}
```

The standard way of listing authors is to iterate from 1 to `\count@` and to pick the author names as we go.

```
⟨!latex⟩\newcount\@tempcnta
\def\@defaultauthorlist{%
  \@getauthorlist\@firstofone
}
```

`\@getauthorlist` processes the author list, passing every bit of stuff that needs to be typeset to the macro specified as its argument.

```
\def\@getauthorlist#1{%
  \count@\authornumber
  \advance\count@ by -2
  \@tempcnta0
```

Loop to output the first $n-2$ of the n authors (the loop does nothing if there are two or fewer authors)

```
  \loop
    \ifnum\count@>0
      \advance\@tempcnta by \@ne
      #1{\ignorespaces\theauthor{\number\@tempcnta}\unskip, }%
      \advance\count@ by \m@ne
  \repeat
  \count@\authornumber
  \advance\count@ by -\@tempcnta
  \ifnum\authornumber>0
```

If there are two or more authors, we output the penultimate author's name here, followed by 'and'

```
    \ifnum\count@>1
      \count@\authornumber
      \advance\count@ by \m@ne
      #1{\ignorespaces\theauthor{\number\count@}\unskip\@tubauthorlastsep}%
    \fi
```

Finally (if there were any authors at all) output the last author's name:

```
    #1{\ignorespaces\theauthor{\number\authornumber}\unskip}
  \fi
}
%
\def\@tubauthorlastsep{, }% until 2018, was: "\ and "
```

Signature blocks. The author can (in principle) define a different sort of signature block using `\signature`, though this could well cause the editorial group to have collective kittens (unless it had been discussed in advance...)

```
\def\signature#1{\def\@signature{#1}}
\def\@signature{\@defaultsignature}
```

`\@defaultsignature` loops through all the authors, outputting the details we have about that author, or (if we're in a sub-article) outputs the contributor's name and closes the group opened by `\contributor`. It is (as its name implies) the default body for `\makesignature`

```
\def\@defaultsignature{{%
    \let\thanks\@gobble
    \frenchspacing
    %
    \ifnum\authornumber<0
```

if `\authornumber`< 0, we are in a contributor's section

```
      \medskip
      \signaturemark
      \theauthor{\number\authornumber}\\
      \theaddress{\number\authornumber}\\
      \allowhyphens
      \thenetaddress{\number\authornumber}\\
      \thePersonalURL{\number\authornumber}\\
      \theORCID{\number\authornumber}\\
```

`\authornumber`≥ 0, so we are in the body of an ordinary article:

```
    \else
      \count@=0
      \loop
        \ifnum\count@<\authornumber
          \medskip
          \advance\count@ by \@ne
          \signaturemark
          \theauthor{\number\count@}\\
          \theaddress{\number\count@}\\
          {%
            \allowhyphens
            \thenetaddress{\number\count@}\\
            \thePersonalURL{\number\count@}\\
            \theORCID{\number\count@}\\
          }%
      \repeat
    \fi
  }%
}
\newdimen\signaturewidth   \signaturewidth=12pc
```

The optional argument to `\makesignature` is useful in some circumstances (e.g., multi-contributor articles)

```
\newcommand{\makesignature}[1][\medskipamount]{%
```

check the value the user has put in `\signaturewidth`: it may be at most `1.5pc` short of `\columnwidth`

```
  \@tempdima\signaturewidth
  \advance\@tempdima 1.5pc
  \ifdim \@tempdima>\columnwidth
    \signaturewidth \columnwidth
    \advance\signaturewidth -1.5pc
  \fi
  \par
  \penalty9000
  \vspace{#1}%
  \rightline{%
    \vbox{\hsize\signaturewidth \ninepoint \raggedright
      \parindent \z@ \everypar={\hangindent 1pc }%
      \parskip \z@skip
      \def\|{\unskip\hfil\break}%
      \def\\{\endgraf}%
      \def\phone{\rm Phone: }%
      \def\tubmultipleaffilauthor{\unskip,\\\hspace*{1em}}%
      \rm\@signature}%
  }%
  \ifnum\authornumber<0 \endgroup\fi
}
\def\signaturemark{\leavevmode\llap{$\diamond$\enspace}}
```

The idea here is that if multiple authors share affiliation information, we need only typeset the affiliation once. We separate by commas for the `\maketitle`, and put on separate lines, also with commas, in the `\makesignature`.

Similarly, within `\netaddress`, `!tubmultipleaffilnet` separates with a space before and after the comma. (All this per bb.) See `tb122childs-trotter.ltx`, `tb131sojka-czech.ltx` for examples.

```
\def\tubmultipleaffilauthor{\unskip,\ \ignorespaces}%
\def\tubmultipleaffilnet{\unskip\textrm{\,,\ \ignorespaces}}
```

Now all the awful machinery of author definitions. `\authornumber` records the number of authors we have recorded to date.

```
\newcount\authornumber
\authornumber=0
```

`\author` 'allocates' another author name (by bumping `\authornumber`) and also sets up the address and netaddress for this author to produce a warning and to prevent oddities if they're invoked. This last assumes that invocation will be in the context of `\signature` (`ltugboat.cls`) or `\maketitle` (`ltugproc.cls`); in both cases, invocation is followed by a line break (tabular line break `\\` in `ltugproc`, `\endgraf` in `\makesignature` in `ltugboat`).

```
\def\author{%
  \global\advance\authornumber\@ne
  \TB@author
}
```

`\contributor` is for a small part of a multiple-part article; it begins a group that will be ended in `\makesignature`.

```
\def\contributor{%
  \begingroup
  \authornumber\m@ne
  \TB@author
}
```

Both 'types' of author fall through here to set up the author name and to initialise author-related things. `\EDITORno*` commands allow the editor to record that there's good reason for an *address* or *netaddress* not to be there, but nowadays, we consider all address information optional.

```
\def\TB@author#1{%
  \expandafter\def\csname theauthor\number\authornumber\endcsname
      {\ignorespaces#1\unskip}%
%  \expandafter\def\csname theaddress\number\authornumber\endcsname
%    {\TBWarningNL{Address for #1\space missing}\@gobble}%
%  \expandafter\def\csname thenetaddress\number\authornumber\endcsname
%    {\TBWarningNL{Net address for #1\space missing}\@gobble}%
  \expandafter\let\csname thePersonalURL\number\authornumber\endcsname
    \@gobble
  \expandafter\let\csname theORCID\number\authornumber\endcsname
    \@gobble
  }
\def\EDITORnoaddress{%
  \expandafter\let\csname theaddress\number\authornumber\endcsname
    \@gobble
}
\def\EDITORnonetaddress{%
  \expandafter\let\csname thenetaddress\number\authornumber\endcsname
    \@gobble
}
```

`\address` copies its argument into the `\theaddress<n>` for this author.

```
\def\address#1{%
  \expandafter\def\csname theaddress\number\authornumber\endcsname
      {\leavevmode\ignorespaces#1\unskip}}
```

`\network` is for use within the optional argument of `\netaddress`; it defines the *name* of the network the user is on.

Comment: I think this is a fantasy, since everyone (in practice, nowadays) quotes an internet address. In principle, there are people who will quote `X.400` addresses (but they're few and far between) and I have (during 1995!) seen an address with an `UUCP` bang-path component on `comp.text.tex`, but *really*!

```
\def\network#1{\def\@network{#1: }}
```

`\netaddress` begins a group, executes an optional argument (which should not, presumably, contain global commands) and then relays to `\@relay@netaddress` with both `@` and `%` made active (so that they can be discretionary points in the address). If we're using LaTeX 2_ε, we use the default-argument form of `\newcommand`; otherwise we write it out in all its horribleness.

```
\newcommand{\netaddress}[1][\relax]{%
```

```
  \begingroup
  \def\@network{}%
```

Unfortunately, because of the catcode hackery, we have still to do one stage of relaying within our own code, even if we're using LaTeX 2_ε.

```
  #1\@sanitize\makespace\ \makeactive\@%
  \makeescape! \makebgroup[ \makeegroup]% seems more useful than literals
  \makeactive\.\makeactive\%\@relay@netaddress}%
```

`\@relay@netaddress` finishes the job. It sets `\thenetaddress` for this author to contain the network name followed by the address. As a result of our kerfuffle above, `@` and `%` are active at the point we're entered. We ensure they're active when `\thenetaddress` gets expanded, too. (***WOT?!***)

```
\def\@relay@netaddress#1{%
  \ProtectNetChars
  \expandafter\protected@xdef
      \csname thenetaddress\number\authornumber\endcsname
    {\protect\leavevmode\textrm{\@network}%
     {\protect\NetAddrChars\net
      \ignorespaces#1\unskip}}%
  \endgroup
}
```

We `\personalURL` quite differently from `\netaddress`: it is set up to simply call `\tburl`, which makes a live link if possible, and also removes a leading protocol. Thus the argument has to be a true url, not just a random string, but that restriction seems ok to get the benefits. Since `\tburl` handles all the catcoding, no need to do any of that here.

```
\def\personalURL#1{%
  % define \cs{thePersonalURL}\meta{n} for author \meta{n}'s \personalURL.
  \expandafter\protected@xdef
    \csname thePersonalURL\number\authornumber\endcsname{%
      \protect\leavevmode
      \ignorespaces
      \protect\tburl{#1}%
      \unskip
  }%
}
```

Previously: `\personalURL` was similar to `\netaddress`, apart from (1) the lack of the eccentric optional argument, (2) the activation of '`/`'. This is the old definition, no longer used (left here just for posterity); new definition is just above.

```
%\def\personalURL{\begingroup
%  \@sanitize\makespace\ \makeactive\@%
%  \makeactive\.\makeactive\%\makeactive\/%
%  \@personalURL}%
%\def\@personalURL#1{%
%  \ProtectNetChars
%  % define \cs{thePersonalURL}\meta{n} for author \meta{n}.
%  \expandafter\protected@xdef
%    \csname thePersonalURL\number\authornumber\endcsname{%
%      \protect\leavevmode
%      {%
```

```
%         \protect\URLchars
%         \net
%         \ignorespaces\protect\tburl{#1}\unskip
%       }%
%     }%
%   \endgroup
%}
```

Define the activation mechanism for '`@`', '`%`', '`.`' and '`/`', for use in the above. Note that, since the code has '`%`' active, we have '`*`' as a comment character, which has a tendency to make things look peculiar...

```
{%
  \makecomment\*
  \makeactive\@
  \gdef\netaddrat{\makeactive\@*
    \def@{\discretionary{\char"40}{}{\char"40}}}
  \makeactive\%
  \gdef\netaddrpercent{\makeactive\%*
    \def%{\discretionary{\char"25}{}{\char"25}}}
  \makeactive\.
  \gdef\netaddrdot{\makeactive\.*
    \def.{\discretionary{\char"2E}{}{\char"2E}}}
```

`\NetAddrChars` is what *we* use (we're constrained to retain the old interface to this stuff, but it *is* clunky...). Since URLs are a new idea, we are at liberty not to define a separate `\netaddrslash` command, and we only have `\URLchars`.

```
  \gdef\NetAddrChars{\netaddrat \netaddrpercent \netaddrdot}
  \makeactive\/
  \gdef\URLchars{*
    \NetAddrChars
    \makeactive\/*
    \def/{\discretionary{\char"2F}{}{\char"2F}}}
```

`\ProtectNetChars` includes protecting '`/`', since this does no harm in the case of net addresses (where it's not going to be active) and we thereby gain by not having yet another csname.

```
  \gdef\ProtectNetChars{*
    \def@{\protect@}*
    \def%{\protect%}*
    \def.{\protect.}*
    \def/{\protect/}*
    }
}
```

LaTeX 2ε (in its wisdom) suppresses `\DeclareOldFontCommand` when in compatibility mode, so that in that circumstance we need to use a declaration copied from `latex209.def` rather than the way we would normally do the thing (using the command LaTeX 2ε defines for the job).

```
\if@compatibility
  \DeclareRobustCommand{\net}{\normalfont\ttfamily\mathgroup\symtypewriter}
\else
  \DeclareOldFontCommand{\net}{\ttfamily\upshape\mdseries}{\mathtt}
\fi
\def\authorlist#1{\def\@author{#1}}
```

```
\def\@author{\@defaultauthorlist}
```

`\ORCID` inserts 'ORCID' and then argument into the `\theORCID<n>` for this author. Also, we want `\small` for this.

```
\def\ORCID#1{%
  \expandafter\def\csname theORCID\number\authornumber\endcsname
      {\leavevmode \ignorespaces {\SMC ORCID} #1\unskip}}
```

For the online re-publication (as of 2009) by Mathematical Sciences Publishers `http://mathscipub.org`, lots and lots of metadata is needed, much of it redundant with things we already do. They are flexible enough to allow us to specify it in any reasonable way, so let's make one command `\mspmetavar` which takes two arguments. Example: `\mspmetavar{volumenumber}{30}`. For our purposes, it is just a no-op. And this initiative never came to anything, so it is not used at all.

`\mspmetavar`

```
\def\mspmetavar#1#2{}
```

3.14 Article title

`\if@articletitle` `\maketitle` takes an optional "*"; if present, the operation is not defining the `\maketitle` title of a paper, merely that of a "business" section (such as the participants at `\@r@maketitle` a meeting) that has no credited author or other title. In this case, the command flushes out the latest `\sectitle` (or whatever) but does nothing else.

Provide machinery (`\PreTitleDrop` to skip extra space, even one or more full columns, above the top of an article to leave space to paste up a previous article that has finished on the same page. This is a fall back to accommodate the fact that multiple articles cannot be run together easily.

In addition, if the `secondcolstart` option was specified, do `\null\newpage` to move over. This is separate from `\PreTitleDrop`, for no particular reason.

```
\newif\if@articletitle
\def\maketitle{\@ifstar
  {\@articletitlefalse\@r@maketitle}%
  {\@articletitletrue\@r@maketitle}%
}
\def\@r@maketitle{\par
 \iftubsecondcolstart \null\newpage\tubsecondcolstartextra \fi
 \ifdim\PreTitleDrop > \z@
   \loop
   \ifdim \PreTitleDrop > \textheight
     \vbox{}\vfil\eject
     \advance\PreTitleDrop by -\textheight
   \repeat
   \vbox to \PreTitleDrop{\vfil}%
   \global\PreTitleDrop=\z@
 \fi
 \begingroup
 \setcounter{footnote}{0}
 \global\@topnum\z@ % disallow floats above the title
 \def\thefootnote{\fnsymbol{footnote}}
 \@maketitle
 \@thanks
 \endgroup
```

```
\setcounter{footnote}{0}
\gdef\@thanks{}
}
```

\title
\TB@title

We redefine the \title command, so as to set the \rhTitle command at the same time. While we're at it, we redefine it to have optional arguments for use as 'short' versions, thus obviating the need for users to use the \shortTitle command.

```
\def\rhTitle{}% avoid error if no author or title
\renewcommand{\title}{\@dblarg\TB@title}
\def\TB@title[#1]#2{\gdef\@title{#2}%
  \bgroup
    \let\thanks\@gobble
    \def\\{\unskip\space\ignorespaces}%
    \protected@xdef\rhTitle{#1}%
  \egroup
}
```

\shortTitle
\ifshortAuthor
\shortAuthor

The \rh* commands are versions to be used in the running head of the article. Normally, they are the same things as the author and title of the article, but in the case that there are confusions therein, the text should provide substitutes, using the \short* commands.

```
\def\shortTitle #1{\def\rhTitle{#1}}
\newif\ifshortAuthor
\def\shortAuthor #1{\def\rhAuthor{#1}\shortAuthortrue}
```

3.15 Section titles

The following macros are used to set the large *TUGboat* section heads (e.g. "General Delivery", "Fonts", etc.)

Define the distance between articles which are run together:

```
\def\secsep{\vskip 5\baselineskip}
```

Note that \stbaselineskip is used in the definition of \sectitlefont, in LaTeX 2ε, so that it has (at least) to be defined before \sectitlefont is used (we do the whole job).

```
\newdimen\stbaselineskip        \stbaselineskip=18\p@
\newdimen\stfontheight
\settoheight{\stfontheight}{\sectitlefont O}
```

Declaring section titles; the conditional \ifSecTitle records the occurence of a \sectitle command. If (when) a subsequent \maketitle occurs, the section title box will get flushed out; as a result of this, one could in principle have a set of \sectitle commands in a semi-fixed steering file, and inclusions of files inserted only as and when papers have appeared. Only the last \sectitle will actually be executed.

```
\newif\ifWideSecTitle
\newif\iftubtitlerulefullwidth
\newif\ifSecTitle \SecTitlefalse
\newcommand{\sectitle}{%
  \SecTitletrue
  \@ifstar
    {\WideSecTitletrue\def\s@ctitle}%
```

```
    {\WideSecTitlefalse\def\s@ctitle}%
}
```

`\PreTitleDrop` records the amount of column-space we need to eject before we start any given paper. It gets zeroed after that ejection has happened.

```
\newdimen\PreTitleDrop   \PreTitleDrop=\z@
```

The other parameters used in `\@sectitle`; I don't think there's the slightest requirement for them to be registers (since they're constant values, AFAIK), but converting them to macros would remove the essentially useless functionality of being able to change them using assignment, which I'm not about to struggle with just now...

`\AboveTitleSkip` is glue above the article title; `\BelowTitleSkip` is glue below the authors in the title block. `\strulethickness` is the value to use for `\fboxrule` when setting the title, and for the rule above titles when there is no box.

For `\BelowTitleSkip`, add some stretch and shrink since the first column of an article often needs it; otherwise, a first column of all text will come out underfull. Use `plus2pt` since that is the same as the glue above sections, but `minus1pt` since we'd usually prefer to shrink somewhere else if possible.

```
\newskip\AboveTitleSkip   \AboveTitleSkip=12pt
\newskip\BelowTitleSkip   \BelowTitleSkip=8pt plus2pt minus1pt
\newdimen\strulethickness \strulethickness=.6pt
```

`\@sectitle` actually generates the section title (in a rather generous box). It gets called from `\maketitle` under conditional `\ifSecTitle`; by the time `\@sectitle` takes control, we already have `\SecTitlefalse`. This implementation uses LaTeX's `\framebox` command, on the grounds that one doesn't keep a dog and bark for oneself...

```
\def\@sectitle #1{%
  \par
  \penalty-1000
```

If we're setting a wide title, the stuff will be at the top of a page (let alone a column) but inside a box, so that the separator won't be discardable: so don't create the separator in this case.

```
  \ifWideSecTitle\else\secsep\fi
  {%
    \fboxrule\strulethickness
    \fboxsep\z@
    \noindent\framebox[\hsize]{%
      \vbox{%
        \raggedcenter
        \let\\\@sectitle@newline
        \sectitlefont
        \makestrut[2\stfontheight;\z@]%
        #1%
        \makestrut[\z@;\stfontheight]\endgraf
      }%
    }%
  }%
  \nobreak
  \vskip\baselineskip
```

```
}
```

`\@sectitle@newline` For use inside `\sectitle` as `\\`. Works similarly to `\\` in the "real world"—uses an optional argument

```
\newcommand{\@sectitle@newline}[1][\z@]{%
  \ifdim#1>\z@
    \makestrut[\z@;#1]%
  \fi
  \unskip\break
}
```

We need to trigger the making of a section title in some cases where we don't have a section title proper (for example, in material taken over from TTN).

```
\def\@makesectitle{\ifSecTitle
    \global\SecTitlefalse
    \ifWideSecTitle
      \twocolumn[\@sectitle{\s@ctitle}]%
      \global\WideSecTitlefalse
    \else
      \@sectitle{\s@ctitle}%
    \fi
  \else
    \vskip\AboveTitleSkip
    \kern\topskip
    \hrule \@height\z@ \@depth\z@ \@width 10\p@
    \kern-\topskip
    \kern-\strulethickness
    \iftubtitlerulefullwidth
      \hrule \@height\strulethickness \@depth\z@ width\textwidth
    \else
      \hrule \@height\strulethickness \@depth\z@
    \fi
    \kern\medskipamount
    \nobreak
  \fi
}
```

`\@maketitle` Finally, the body of `\maketitle` itself.

```
\def\@maketitle{%
  \@makesectitle
  \if@articletitle{%
    \nohyphens \interlinepenalty\@M
    \setbox0=\hbox{%
      \let\thanks\@gobble
      \let\\=\quad
      \let\and=\quad
      \ignorespaces\@author}%
    {%
      \noindent\bf\raggedright\ignorespaces\frenchspacing
      \let\BibTeX=\bfBibTeX % else LaTeX Font Warning:
                            %      Font shape `OT1/cmr/bx/sc' undefined
      \@title\endgraf
    }%
    \ifdim \wd0 < 5\p@              % omit if author is null
```

```
    \else
```

Since we have `\BelowTitleSkip` + `4pt` = `\baselineskip`, we skip by `4pt` here. However, an all-text first column still comes out underfull, maybe because of the top rule? Thus `\BelowTitleSkip` is given a little stretch and shrink.

```
      \nobreak \vskip 4\p@
      {%
        \leftskip=\normalparindent
        \raggedright
        \def\and{\unskip\\}%
        \noindent\@author\endgraf
      }%
    \fi
    \nobreak
    \vskip\BelowTitleSkip
  }\fi%
  \global\@afterindentfalse
  \aftergroup\@afterheading
}
```

Dedications are ragged right, in italics.

```
\newenvironment{dedication}%
  {\raggedright\noindent\itshape\ignorespaces}%
  {\endgraf\medskip}
```

The `abstract` and `longabstract` environments both use `\section*`. For one-column articles (or in `ltugproc` class), indent the abstract. This is done in the usual bizarre LaTeX way, by treating it as a one-item list with an empty item marker.

```
\def\@tubonecolumnabstractstart{%
      \list{}{\listparindent\normalparindent
         \itemindent\z@ \leftmargin\@tubfullpageindent
         \rightmargin\leftmargin \parsep \z@}\item[]\ignorespaces
}
\def\@tubonecolumnabstractfinish{%
      \endlist
}
\renewenvironment{abstract}%
  {\begin{SafeSection}%
    \section*{%
        \if@tubtwocolumn\else \hspace*{\@tubfullpageindent}\fi
        Abstract}%
    \if@tubtwocolumn\else \@tubonecolumnabstractstart \fi
  }%
  {\if@tubtwocolumn\else \@tubonecolumnabstractfinish \fi
   \end{SafeSection}}
\newenvironment{longabstract}%
  {\begin{SafeSection}%
    \section*{Abstract}%
    \bgroup\small
  }%
  {\endgraf\egroup
    \end{SafeSection}%
  \vspace{.25\baselineskip}
```

```
\begin{center}
  {$--*--$}
\end{center}
\vspace{.5\baselineskip}}
```

3.16 Section headings

Redefine style of section headings to match plain *TUGboat*. Negative beforeskip suppresses following parindent. (So negate the stretch and shrink too).

These macros are called `\*head` in the plain styles.

Relaying via `\TB@startsection` detects inappropriate use of `\section*`. Of course, if (when) *we* use it, we need to avoid that relaying; this can be done by `\let`ting `\TB@startsection` to `\TB@safe@startsection`, within a group.

First the version for use in the default case, when class option NUMBERSEC is in effect.

The `\tubsecfmt` macro defines our standard formatting for section titles: ragged right, french spacing, no hyphenation. The `\tubruninsecfmt` macro is the simpler form for run-in section headings (when the afterskip is negative), with the afterskip glue given by `\tubruninglue`. The `\tubsechook` macro allows overriding the defaults.

```
\def\tubsechook{}
\def\tubsecfmt{\normalsize\bf\raggedright\frenchspacing\nohyphens\tubsechook}
\def\tubruninglue{-1em plus-2\fontdimen3\font minus-\fontdimen4\font}
\def\tubruninsecfmt{\normalsize\bf\tubsechook}
%
\if@numbersec
  \def\section{\TB@startsection{%
    {section}                 % name of counter
    {1}                       % level
    {0pt}                     % indent
    {-8pt plus-2pt minus-2pt} % beforeskip; negative -> \noindent after
    {4pt}                     % afterskip;  negative -> hspace for run-in
    {\tubsecfmt}}}            % style
  %
  \def\subsection{\TB@startsection{%
    {subsection}%
    2%
    \z@
    {-8\p@ \@plus-2\p@ \@minus-2\p@}%
    {4\p@}%
    {\tubsecfmt}}}
  %
  \def\subsubsection{\TB@startsection{%
    {subsubsection}%
    3%
    \z@
    {-8\p@ \@plus-2\p@ \@minus-2\p@}%
    {4\p@}%
    {\tubsecfmt}}}
  %
  \def\paragraph{\TB@startsection{%
    {paragraph}%
```

```
    4%
    \z@
    {4\p@ \@plus1\p@ \@minus1\p@}%
    {\tubruninglue}
    {\tubruninsecfmt}}}
```

Now the version if class option `nonumber` is in effect, i.e., if `\if@numbersec` is false.

```
\else
  \setcounter{secnumdepth}{0}
  \def\section{\TB@nolimelabel\TB@startsection{%
    {section}% same as numbered
    1%
    \z@
    {-8\p@ \@plus-2\p@ \@minus-2\p@}%
    {4\p@}%
    {\tubsecfmt}}}
  %
  \def\subsection{\TB@nolimelabel\TB@startsection{%
    {subsection}%
    2%
    \z@
    {-8\p@ \@plus-2\p@ \@minus-2\p@}%
    {\tubruninglue}
    {\tubruninsecfmt}}}
  %
  \def\subsubsection{\TB@nolimelabel\TB@startsection{%
    {subsubsection}%
    3%
    \parindent
    {-8\p@ \@plus-2\p@ \@minus-2\p@}%
    {\tubruninglue}
    {\tubruninsecfmt}}}
\fi
```

`\TB@startsection` used to warn about `*` versions of sectioning commands when numbering wasn't in effect. But that eventually seemed a useless complaint, since it can be useful to switch back and forth between numbered and unnumbered can be useful during article development. So now `\TB@startsection` is just a synonym for `\@startsection`.

```
\def\TB@startsection#1{\@startsection#1}%
```

`\TB@safe@startsection` is to be used where `\section*` (etc.) appear in places where the request is OK (because it's built in to some macro we don't fiddle with).

```
\def\TB@safe@startsection#1{\@startsection#1}
```

The `SafeSection` environment allows use of `*`-forms of sectioning environments. It's not documented for the general public: it's intended as an editor's facility.

```
\newenvironment{SafeSection}%
  {\let\TB@startsection\TB@safe@startsection}%
  {}
```

And now for the exciting sectioning commands that LaTeX defines but we don't have a definition for (whatever else, we don't want Lamport's originals, which come out 'like the blare of a bugle in a lullaby'[1]).

The three inappropriate ones are subparagraph (indistinguishable from paragraph), and chapter and part. The last seemed almost to be defined in an early version of these macros, since there was a definition of `\l@part`. I've not got down to where that came from (or why). If class option NONUMBER is in effect, we also suppress `\paragraph`, since it has no parallel in the plain style.

```
\if@numbersec
  \def\subparagraph{\TB@nosection\subparagraph\paragraph}
\else
  \def\paragraph{\TB@nosection\paragraph\subsubsection}
  \def\subparagraph{\TB@nosection\subparagraph\subsubsection}
\fi
\def\chapter{\TB@nosection\chapter\section}
\def\part{\TB@nosection\part\section}
\def\TB@nosection#1#2{\TBWarning{class does not support \string#1,
    \string#2\space used instead}#2}
```

`\l@<sectioninglevel>` is for table of contents (of an article). We define new macros to allow easily changing the font used for toc entries (for *TUGboat*, we usually want roman, not bold), and the space between entries. Nelson Beebe and Frank Mittelbach's articles often have toc's (and few others). Also turn off microtype protrusion after

Contents

, or leaders get messed up.

```
\def\TBtocsectionfont{\normalfont}
\newskip\TBtocsectionspace  \TBtocsectionspace=1.0ex\@plus\p@

% |#1| is both the section number and title, as in
%   |{\numberline {1}Introduction}|.
% |#2| is the page number.
%
% Per Ulrike, the hook calls are for tagging, introduced with the
% June 2023 \LaTeX.
% qqq need to also do subsections like tb137carlisle to avoid hyphenation
\def\l@section#1#2{%
  \addpenalty{\@secpenalty}%
  \addvspace{\TBtocsectionspace}%
  \@tempdima 1.5em
  \begingroup
    \parindent\z@
    \rightskip=0pt plus2em
    \parfillskip\z@
    \hyphenpenalty=10000
    \TBtocsectionfont
    \leavevmode
    \advance\leftskip by \@tempdima % space between section number and text
    \hskip-\leftskip
```

[1] Thurber, *The Wonderful O*

```
    %
    \ifx\UseHookWithArguments\undefined\else % hook before number and text
      \UseHookWithArguments{contentsline/text/before}{4}
        {\toclevel@part}{#1}{#2}{\@contentsline@destination}%
    \fi
    %
    % don't worry if this cs is not defined, hence the \csname.
    % If it doesn't exist, we just typeset #1 as text.
    \csname contentsline@text@1@format\endcsname
    {#1% number and title
     \unskip % avoid extra space just in case
     \csname pdffakespace\endcsname % fake space if pdftex
     ~% ensure at least a word space between text and page number
    }
    %
    \ifx\UseHookWithArguments\undefined\else % hook after number and text
      \UseHookWithArguments{contentsline/text/after}{4}
        {\toclevel@part}{#1}{#2}{\@contentsline@destination}%
    \fi
    \nobreak\hfil
    \nobreak
    % page number
    \hb@xt@\@pnumwidth{\hfil
      \ifx\UseHookWithArguments\undefined\else
        \UseHookWithArguments{contentsline/page/before}{4}
          {\toclevel@part}{#1}{#2}{\@contentsline@destination}%
      \fi
      \tubtypesetpageno{#2}%
      \ifx\UseHookWithArguments\undefined\else
        \UseHookWithArguments{contentsline/page/after}{4}
          {\toclevel@part}{#1}{#2}{\@contentsline@destination}%
      \fi
    }\par
  \endgroup}
```

3.17 Appendices

Appendices (which are really just another sort of section heading) raise a problem: if the sections are unnumbered, we plainly need to restore the section numbering, which in turn allows labelling of section numbers again (`\TBnolimelabel` happens before the `\refstepcounter`, so its effects get lost ... what a clever piece of design that was). So here we go:

```
\renewcommand{\appendix}{\par
  \renewcommand{\thesection}{\@Alph\c@section}%
  \setcounter{section}{0}%
  \if@numbersec
  \else
    \setcounter{secnumdepth}{1}%
  \fi
```

Now: is this the start of an appendix environment? This can be detected by looking at `\@currenvir`; if we are, we need to relay to `\@appendix@env` to pick up the optional argument.

```
  \def\@tempa{appendix}
  \ifx\@tempa\@currenvir
    \expandafter\@appendix@env
  \fi
}
```

Here we deal with `\begin{appendix}[`⟨*app-name*⟩`]`

```
\newcommand{\app@prefix@section}{}
\newcommand{\@appendix@env}[1][Appendix]{%
  \renewcommand{\@seccntformat}[1]{\csname app@prefix@##1\endcsname
    \csname the##1\endcsname\quad}%
  \renewcommand{\app@prefix@section}{#1 }%
}
```

Ending an appendix environment is pretty trivial...

```
\let\endappendix\relax
```

3.18 References

If the sections aren't numbered, the natural tendency of the author to cross-reference (which, after all, is one of the things LaTeX is for ever being advertised as being good at) can cause headaches.

The following command is used by each of the sectioning commands to make a following `\ref` command bloop at the author. Even if the author then ignores the complaint, the poor old editor may find the offending `\label` rather more easily.

(Note that macro name is to be read as "*noli me* label" (I don't know the mediæval Latin for 'label').

Comment To come (perhaps): detection of the act of labelling, and an analogue of `\ifG@refundefined` for this sort of label

```
\def\TB@nolimelabel{%
  \def\@currentlabel{%
    \protect\TBWarning{%
      Invalid reference to numbered label on page \thepage
      \MessageBreak made%
    }%
    \textbf{?!?}%
  }%
}
```

3.19 Title references

This is a first cut at a mechanism for referencing by the title of a section; it employs the delightfully simple idea Sebastian Rahtz has in the `nameref` package (which is part of `hyperref`). As it stands, it lacks some of the bells and whistles of the original, but they could be added; this is merely proof-of-concept.

The name label comes from the moveable bit of the section argument; we subvert the `\@sect` and `\@ssect` commands (the latter deals with starred section commands) to grab the relevant argument.

As of the June 2023 LaTeX (or somewhat earlier, but this is good enough), there are hooks that allow us to avoid redefinig `\@sect` and `\@ssect`.

```
\@ifl@t@r\fmtversion{2023-06-01}{}{%
\let\TB@@sect\@sect
\let\TB@@ssect\@ssect
\def\@sect#1#2#3#4#5#6[#7]#8{%
  \def\@currentlabelname{#7}%
  \TB@@sect{#1}{#2}{#3}{#4}{#5}{#6}[{#7}]{#8}%
}
\def\@ssect#1#2#3#4#5{%
 \def\@currentlabelname{#5}%
 \TB@@ssect{#1}{#2}{#3}{#4}{#5}%
}
} % LaTeX earlier than June 2023
```

We output the name label as a second `\newlabel` command in the `.aux` file. That way, packages such as `varioref` which also read the `.aux` information can still work. So we redefine `\label` to first call the standard LaTeX `\label` and then write our named label as `nr<label>`.

Similarly, we only need this with pre-June 2023 LaTeX. With more recent LaTeX, define currentlabelname via hooks.

```
\@ifl@t@r\fmtversion{2023-06-01}{%
  \RequirePackage{gettitlestring}
  \AddToHookWithArguments{cmd/@sect/before}{%
    \GetTitleString{#7}%
    \let\@currentlabelname\GetTitleStringResult}%
  \AddToHookWithArguments{cmd/@ssect/before}{%
    \GetTitleString{#5}%
    \let\@currentlabelname\GetTitleStringResult}%
}{% else older latex:
  \let\@savelatexlabel=\label  % so save original LaTeX command
  %
  \def\label#1{%
    \@savelatexlabel{#1}%
    \@bsphack
    \if@filesw
      \protected@write\@auxout{}%
        {\string\newlabel{nr@#1}{{\@currentlabel}{\@currentlabelname}}}%
    \fi
    \@esphack}
  % in case there are no sectioning commands:
  \let\@currentlabelname\@empty
}
```

Getting named references is then just like getting page references in the LaTeX kernel (see `ltxref.dtx`).

The above was written by RobinF decades ago; the macros in *TUGboat* were never changed. Meanwhile, the `\nameref` in `hyperref` has changed many times, and we want to use its version if available. So we provide our `\nameref` `\AtBeginDocument`, so as not to overwrite any previous version. Until May 2022, `hyperref` silently overwrote an existing definition, that is, *TUGboat*'s. But now it is no longer silent.

It seems that all the internal definitions above do not cause problems, so just let them alone.

```
\AtBeginDocument{%
  \@ifl@t@r\fmtversion{2023-06-01}%
  { % after June 2023, LaTeX stores the label name; use that.
    \long\def\@thirdoffive#1#2#3#4#5{#3}
    \providecommand\nameref[1]{%
      \expandafter\@setref
      \csname r@#1\endcsname\@thirdoffive{#1}}%
  }
  { % for earlier versions, still avoid overwriting \nameref per above.
    % but if not otherwise defined, use the "nr" label defined by our \label.
   \providecommand\nameref[1]{%
     \expandafter\@setref
     \csname r@nr@#1\endcsname\@secondoftwo{#1}}%
  }%
}
```

3.20 Float captions

By analogy with what we've just done to section titles and the like, we now do our best to discourage hyphenation within captions. We also typeset them in `\small` (actually `\tubcaptionfonts`).

First, let's define a dimension by which we will indent full-page captions. We'll also use this to indent abstracts in proceedings style.

`\@tubfullpageindent`

```
\newdimen\@tubfullpageindent
\@tubfullpageindent = \if@tubtwocolumn 4.875pc \else 3.875pc \fi
```

One-line captions are normally centered, but sometimes we want to set them flush left for consistency with other nearby figures.

`\tubcaptionleftglue`

```
\let\tubcaptionleftglue=\hfil
```

For *TUGboat*, we like 9pt captions to help differentiate from the main text. Make a macro so we can easily override.

```
\def\tubcaptionfonts{\small}%
```

Ok, here is `\@makecaption`.

```
\long\def\@makecaption#1#2{%
  \vskip\abovecaptionskip
  % try in an hbox:
  \sbox\@tempboxa{\tubcaptionfonts \frenchspacing \tubmakecaptionbox{#1}{#2}}%
  \ifdim \wd\@tempboxa > \hsize
    {% caption doesn't fit on one line; set as a paragraph.
     \tubcaptionfonts \raggedright \hyphenpenalty=\@M \parindent=1em
     % indent full-width captions {figure*}, but not single-column {figure}.
     \ifdim\hsize = \textwidth
       \leftskip=\@tubfullpageindent \rightskip=\leftskip
       \advance\rightskip by 0pt plus2em  % increase acceptable raggedness
     \fi
     \noindent \tubmakecaptionbox{#1}{#2}\par}%
```

```
  \else
    % fits on one line; use the hbox, usually centered.  Do not reset its glue.
    \global\@minipagefalse
    \hb@xt@\hsize{\tubcaptionleftglue\box\@tempboxa\hfil}%
  \fi
  \vskip\belowcaptionskip}
```

Also use `\tubcaptionfonts` for the caption labels, and put the label (e.g., "Figure 1") in bold. Use a macro so we can override.

```
\def\tubmakecaptionbox#1#2{{\tubcaptionfonts\textbf{#1}:\ #2}%
```

We used to switch the labels into bold this way, but it's better to do it as part of `\@makecaption` since then it will apply to other floating types, such as those created by the `newfloat` package. (E.g., `tb142duck-pylatex`.)

```
\def\fnum@figure{{\tubcaptionfonts \bf \figurename\nobreakspace\thefigure}}
\def\fnum@table{{\tubcaptionfonts \bf \tablename\nobreakspace\thetable}}
```

If the `listings` package is being used, bold for its label too; this `\def` is too early, but maybe listings will play nice later.

```
\def\lstlistingnamestyle{\bfseries}
```

Let's reduce the default space above captions a bit, and give it some flexibility. The default is `10pt`, which seems too much.

```
\setlength\abovecaptionskip{3pt plus1pt minus1pt}
```

Let's also reduce the space between floats, and between floats and text. Strangely, it seems to be these that count, rather than `\dbl...`, at least sometimes.

```
\setlength\floatsep    { 9pt plus3pt minus2pt} % default 12pt plus2pt minus2pt
\setlength\textfloatsep{12pt plus4pt minus3pt} % default 20pt plus2pt minus4pt
```

We want to allow more floats at the top/bottom/everywhere on a page; all depends on their content.

```
\setcounter{bottomnumber}{2} % default 1
\setcounter{topnumber}{4}    % default 2
\setcounter{totalnumber}{6}  % default 3
```

3.21 Size changing commands

In addition to their 'normal' effects, these commands change the glue around displays to match that of being around lists. Otherwise, in a column that contains both displays and "lists", including verbatim blocks, etc., the different spacing can become noticeable. For example, tb145beisert-eqnlines, page 4, column 1. LaTeX's default for `\abovedisplayskip` and `\belowdisplayskip` are much larger (and not equal).

`\normalsize` `\small` `\footnotesize` `\abovedisplayskip` `\belowdisplayskip`

```
\renewcommand{\normalsize}{%
   \@setfontsize\normalsize\@xpt\@xiipt
   \abovedisplayskip=\topsep % same as space above lists/verbatims/etc.
   \belowdisplayskip=\abovedisplayskip
   \abovedisplayshortskip=\z@\@plus 3\p@
   \belowdisplayshortskip=\p@\@plus 3\p@\@minus\p@
}
```

```
\renewcommand{\small}{%
   \@setfontsize\small\@ixpt{11}%
   \abovedisplayskip=2pt plus.66pt minus.66pt % 2/3 \topsep
   \belowdisplayskip=\abovedisplayskip
   \abovedisplayshortskip=\z@\@plus 2\p@
   \belowdisplayshortskip=\p@\@plus 2\p@\@minus\p@
}

\renewcommand{\footnotesize}{%
    \@setfontsize\footnotesize\@viiipt{9.5}%
   \abovedisplayskip=1pt plus.33pt minus.33pt % 1/3 \topsep
    \belowdisplayskip=\abovedisplayskip
    \abovedisplayshortskip=\z@\@plus 3\p@
    \belowdisplayshortskip=\p@\@plus 3\p@\@minus\p@
}
```

3.22 Lists and other text inclusions

```
\def\@listi{%
  \leftmargin\leftmargini\parsep=\p@\@plus\p@\@minus\p@
  \itemsep=\parsep
  \listparindent=1em
  }

\def\@listii{%
  \leftmargin\leftmarginii
  \labelwidth=\leftmarginii \advance\labelwidth-\labelsep
  \topsep=2\p@\@plus\p@\@minus\p@ % space between first item and preceding
  \parsep=\p@\@plus\p@\@minus\p@
  \itemsep=\parsep                % space between successive items
  \listparindent=1em              % indentation of subsequent paragraphs
  }

\def\@listiii{%
  \leftmargin=\leftmarginiii
  \labelwidth=\leftmarginiii \advance\labelwidth-\labelsep
  \topsep=\p@\@plus\p@\@minus\p@
  \parsep=\z@
  \itemsep=\topsep
  \listparindent=1em
  }
\def\quote{\list{}{\rightmargin.5\leftmargin}\item[]}
```

From Dominik Wujastyk's font article. First paragraph of a quotation will not be indented, and right margin is decreased for narrow columns.

```
\renewcommand{\quotation}{\list{}{\listparindent 1.5em
     \rightmargin.5\leftmargin\parsep \z@\@plus\p@}\item[]}
```

The `compactitemize`, `compactenumerate`, and `compactdescription` environments, without space between the items.

```
\newenvironment{compactitemize}%
   {\begin{itemize}%
     \setlength{\itemsep}{0pt}%
     \setlength{\parskip}{0pt}%
     \setlength{\parsep} {0pt}%
```

```
    }%
    {\end{itemize}}
%
\newenvironment{compactenumerate}%
    {\begin{enumerate}%
      \setlength{\itemsep}{0pt}%
      \setlength{\parskip}{0pt}%
      \setlength{\parsep} {0pt}%
    }%
    {\end{enumerate}}
%
\newenvironment{compactdescription}%
    {\begin{description}%
      \setlength{\itemsep}{0pt}%
      \setlength{\parskip}{0pt}%
      \setlength{\parsep} {0pt}%
    }%
    {\end{description}}
%
```

3.23 Some fun with `verbatim`

The `plain` *TUGboat* style allows `[optional]` arguments to its `\verbatim` command. This will allow the author (or editor) to specify a range of exciting features; we would definitely like the numbered verbatim style for code (that facility is reserved for a future version of this package), and the present little bit of code imposes the `\ruled` option on the built-in verbatim environment. (Note that we don't yet deal with verbatim∗, which is in itself an option to the `plain` original.)

We start by saving various bits and bobs whose operation we're going to subvert.

```
%\let\@TB@verbatim\@verbatim
\let\@TBverbatim\verbatim
\let\@TBendverbatim\endverbatim
```

Impose an optional argument on the environment.

We start the macro with `\par` to avoid a common error: if the optional argument is `\small`, and the document has no blank line before the verbatim block, we don't want that preceding paragraph to be set with `\small`'s line spacing.

(`\obeylines` added to prevent the `\futurelet` from propagating into the body of the verbatim, thus causing lines that start with odd characters (like `#` or even `\`) to behave peculiarly.)

```
\def\verbatim{\par\obeylines
  \futurelet\reserved@a\@switch@sqbverbatim}
%
\def\@switch@sqbverbatim{\ifx\reserved@a[%]
  \expandafter\@sqbverbatim\else
  \def\reserved@b{\@sqbverbatim[]}\expandafter\reserved@b\fi}
%
\def\@sqbverbatim[#1]{%
```

The optional argument consists entirely of functions that modify the appearance of the environment. Following the `plain` style, we define the functions we can execute in the optional argument here.

The command `\ruled` tells us that there should be rules above and below the verbatim block.

```
\def\ruled{\let\if@ruled\iftrue}%
```

The command `\makevmeta` says to make !¡...¿ do ⟨...⟩.

```
\def\makevmeta{\makeescape\! \let\<\tubverb@meta \tubverb@clearliglist}
\def\tubverb@meta##1>{\meta{##1}}
```

The default verbatim defines ‘’¡¿,- as active characters to stop ligatures; remove ¡¿ from the list so we get normal characters. Just hope that the CM ¡¿ ligatures aren't used.

```
\def\tubverb@clearliglist{%
  \def\verbatim@nolig@list{\do\‘\do\,\do\’\do\-}%
}
```

Then we execute the arguments we've got, and relay to a (hacked) copy of the LaTeX verbatim environment.

```
#1\@TBverbatim}
```

The built-in environment itself relays to `\@verbatim`, which we've subverted to impose our views on appearance.

```
\def\@verbatim{%
```

First, we deal with `\ruled`:

```
  \if@ruled\trivlist\item\hrule\kern5\p@\nobreak\fi
```

Now, the code out of the original verbatim environment:

```
  \trivlist \item\relax
  \if@minipage\else\vskip\parskip\fi
  \leftskip\@totalleftmargin\rightskip\z@skip
  \parindent\z@\parfillskip\@flushglue\parskip\z@skip
  \@@par
  \@tempswafalse
  \def\par{%
    \if@tempswa
      \leavevmode \null \@@par\penalty\interlinepenalty
    \else
      \@tempswatrue
      \ifhmode\@@par\penalty\interlinepenalty\fi
    \fi}%
  \obeylines \verbatim@font \@noligs
  \let\do\@makeother \dospecials
  \everypar \expandafter{\the\everypar \unpenalty}%
}% end |\@sqbverbatim|
```

To end the environment, we do everything in reverse order: relay via the copy we made of `\endverbatim`, and then finish off the option changes (again `\ruled` only, so far).

```
\def\endverbatim{\@TBendverbatim
  \if@ruled\kern5\p@\hrule\endtrivlist\fi}
```

Define the `\if` used by the `\ruled` option:

```
\let\if@ruled\iffalse
```

Finally, if `microtype` is loaded, we want it to be deactivated in verbatim blocks. It often manipulates a leading \ rather too much, thus messing with the visible fixed-width alignment.

```
\AtBeginDocument{%
   \@ifpackageloaded{microtype}
     {\g@addto@macro\@verbatim{\microtypesetup{activate=false}}}{}
}
```

3.24 Bibliography

This is more or less copied verbatim from Glenn Paulley's *chicago.sty* (`gnpaulle@bluebox.uwaterloo.ca`). It produces an author-year citation style bibliography, using output from the BibTeX style file based on that by Patrick Daly. It needs extra macros beyond those in standard LaTeX to function properly. The form of the bibitem entries is:

```
\bibitem[\protect\citeauthoryear{Jones, Baker, and Smith}
     {Jones et al.}{1990}{key}...
```

The available citation commands are:

`\cite{key}`	→ (Jones, Baker, and Smith 1990)
`\citeA{key}`	→ (Jones, Baker, and Smith)
`\citeNP{key}`	→ Jones, Baker, and Smith 1990
`\citeANP{key}`	→ Jones, Baker, and Smith
`\citeN{key}`	→ Jones, Baker, and Smith (1990)
`\shortcite`	→ (Jones et al. 1990)
`\citeyear`	→ (1990)
`\citeyearNP`	→ 1990

First of all (after checking that we're to use Harvard citation at all), make a copy of LaTeX's default citation mechanism.

```
\if@Harvardcite
\let\@internalcite\cite
```

Normal forms.

```
\def\cite{\def\@citeseppen{-1000}%
    \def\@cite##1##2{(##1\if@tempswa , ##2\fi)}%
    \def\citeauthoryear##1##2##3{##1, ##3}\@internalcite}
\def\citeNP{\def\@citeseppen{-1000}%
    \def\@cite##1##2{##1\if@tempswa , ##2\fi}%
    \def\citeauthoryear##1##2##3{##1, ##3}\@internalcite}
\def\citeN{\def\@citeseppen{-1000}%
    \def\@cite##1##2{##1\if@tempswa , ##2)\else{)}\fi}%
    \def\citeauthoryear##1##2##3{##1 (##3}\@citedata}
\def\citeA{\def\@citeseppen{-1000}%
    \def\@cite##1##2{(##1\if@tempswa , ##2\fi)}%
    \def\citeauthoryear##1##2##3{##1}\@internalcite}
\def\citeANP{\def\@citeseppen{-1000}%
    \def\@cite##1##2{##1\if@tempswa , ##2\fi}%
    \def\citeauthoryear##1##2##3{##1}\@internalcite}
```

Abbreviated forms (using *et al.*)

```
\def\shortcite{\def\@citeseppen{-1000}%
```

```
    \def\@cite##1##2{(##1\if@tempswa , ##2\fi)}%
    \def\citeauthoryear##1##2##3{##2, ##3}\@internalcite}
\def\shortciteNP{\def\@citeseppen{-1000}%
    \def\@cite##1##2{##1\if@tempswa , ##2\fi}%
    \def\citeauthoryear##1##2##3{##2, ##3}\@internalcite}
\def\shortciteN{\def\@citeseppen{-1000}%
    \def\@cite##1##2{##1\if@tempswa , ##2)\else{)}\fi}%
    \def\citeauthoryear##1##2##3{##2 (##3}\@citedata}
\def\shortciteA{\def\@citeseppen{-1000}%
    \def\@cite##1##2{(##1\if@tempswa , ##2\fi)}%
    \def\citeauthoryear##1##2##3{##2}\@internalcite}
\def\shortciteANP{\def\@citeseppen{-1000}%
    \def\@cite##1##2{##1\if@tempswa , ##2\fi}%
    \def\citeauthoryear##1##2##3{##2}\@internalcite}
```

When just the year is needed:

```
\def\citeyear{\def\@citeseppen{-1000}%
    \def\@cite##1##2{(##1\if@tempswa , ##2\fi)}%
    \def\citeauthoryear##1##2##3{##3}\@citedata}
\def\citeyearNP{\def\@citeseppen{-1000}%
    \def\@cite##1##2{##1\if@tempswa , ##2\fi}%
    \def\citeauthoryear##1##2##3{##3}\@citedata}
```

Place commas in-between citations in the same `\citeyear`, `\citeyearNP`, `\citeN`, or `\shortciteN` command. Use something like `\citeN{ref1,ref2,ref3}` and `\citeN{ref4}` for a list.

```
\def\@citedata{%
        \@ifnextchar [{\@tempswatrue\@citedatax}%
                                    {\@tempswafalse\@citedatax[]}%
}

\def\@citedatax[#1]#2{%
\if@filesw\immediate\write\@auxout{\string\citation{#2}}\fi%
  \def\@citea{}\@cite{\@for\@citeb:=#2\do%
    {\@citea\def\@citea{, }\@ifundefined% by Young
       {b@\@citeb}{{\bf ?}%
       \@warning{Citation `\@citeb' on page \thepage \space undefined}}%
{\csname b@\@citeb\endcsname}}}{#1}}%
```

Don't box citations, separate with ; and a space; Make the penalty between citations negative: a good place to break.

```
\def\@citex[#1]#2{%
\if@filesw\immediate\write\@auxout{\string\citation{#2}}\fi%
  \def\@citea{}\@cite{\@for\@citeb:=#2\do%
    {\@citea\def\@citea{; }\@ifundefined% by Young
       {b@\@citeb}{{\bf ?}%
       \@warning{Citation `\@citeb' on page \thepage \space undefined}}%
{\csname b@\@citeb\endcsname}}}{#1}}%
```

No labels in the bibliography.

```
\def\@biblabel#1{}
```

Set length of hanging indentation for bibliography entries.

```
\newlength{\bibhang}
\setlength{\bibhang}{2em}
```

Indent second and subsequent lines of bibliographic entries. Stolen from openbib.sty: `\newblock` is set to `{}`.

```
\newdimen\bibindent
\bibindent=1.5em
\@ifundefined{refname}%
   {\newcommand{\refname}{References}}%
   {}%
```

For safety's sake, suppress the `\TB@startsection` warnings here...

```
\def\thebibliography#1{% for harvardcite
  \let\TB@startsection\TB@safe@startsection
  \section*{\refname
    \@mkboth{\uppercase{\refname}}{\uppercase{\refname}}}%
  \list{[\arabic{enumi}]}{%
    \labelwidth\z@ \labelsep\z@
    \leftmargin\bibindent
    \itemindent -\bibindent
    \listparindent \itemindent
    \parsep \z@
    \usecounter{enumi}}%
  \def\newblock{}%
  \BibJustification
  \frenchspacing % more than just period, see comments below
}
```

`etal` Other bibliography odds and ends.

`\bibentry`

```
\def\etal{et\,al.\@}
\def\bibentry{%
  \smallskip
  \hangindent=\parindent
  \hangafter=1
  \noindent
  \sloppy
  \clubpenalty500 \widowpenalty500
  \frenchspacing
}
```

`\bibliography` Changes made to accommodate TUB file naming conventions

`\bibliographystyle`

```
\def\bibliography#1{%
  \if@filesw
    \immediate\write\@auxout{\string\bibdata{\@tubfilename{#1}}}%
  \fi
  \@input{\jobname.bbl}%
}
\def\bibliographystyle#1{%
  \if@filesw
    \immediate\write\@auxout{\string\bibstyle{\@tubfilename{#1}}}%
  \fi
}
```

`\thebibliography` If the user has asked to use LaTeX's default citation mechanism (using the `rawcite` `\TB@@thebibliography` option), we still need to patch `\sloppy` to support justification of the body of the bibliography. We kludge in a call to `\frenchspacing` too, since there is no

reason to change only period's `\sfcode`, as LaTeX's original `thebibliography` (in `classes.dtx`) does.

By the way, `amsgen.sty` changes `\frenchspacing` to set the `\sfcode` of punctuation character to successively decreasing integers ending at 1001 for comma. Thus its 1006 for period is overwritten to 1000 for `thebibliography`, making `amsgen`'s `\@addpunct` ineffective. Don't know what that means in practice, if anything.

Back here, we also play with *The TEXbook*@startsection since we always have, though that is no longer needed.

```
\else % not harvardcite
\let\TB@origthebibliography\thebibliography
\def\thebibliography{%
  \let\TB@startsection\TB@safe@startsection
  \def\sloppy{\frenchspacing\BibJustification}%
  \TB@origthebibliography} % latex's thebibliography now reads args.
\fi  % not harvardcite
```

`\BibJustification` `\SetBibJustification` `\TB@@sloppy`

`\BibJustification` defines how the bibliography is to be justified. The Lamport default is "`\sloppy`", but we find some sort of ragged right setting is almost always preferred, so (as of 2025) make that the default. (`\BibJustification` is nevertheless reset to its default value at the start of a paper.)

```
\let\TB@@sloppy\sloppy
\let\BibJustification\raggedright
\newcommand{\SetBibJustification}[1]{%
  \renewcommand{\BibJustification}{#1}%
}
\ResetCommands\expandafter{\the\ResetCommands
  \let\BibJustification\TB@@sloppy
}
```

3.25 Registration marks

We no longer use these since Cadmus does not want them.

```
\def\HorzR@gisterRule{\vrule \@height 0.2\p@ \@depth\z@ \@width 0.5in }
\def\DownShortR@gisterRule{\vrule \@height 0.2\p@ \@depth 1pc \@width 0.2\p@ }
\def\UpShortR@gisterRule{\vrule \@height 1pc \@depth\z@ \@width 0.2\p@ }
```

"T" marks centered on top and bottom edges of paper

```
\def\ttopregister{\dlap{%
        \hb@xt@\trimwd{\HorzR@gisterRule \hfil \HorzR@gisterRule
                        \HorzR@gisterRule \hfil \HorzR@gisterRule}%
        \hb@xt@\trimwd{\hfil \DownShortR@gisterRule \hfil}}}
\def\tbotregister{\ulap{%
        \hb@xt@\trimwd{\hfil \UpShortR@gisterRule \hfil}%
        \hb@xt@\trimwd{\HorzR@gisterRule \hfil \HorzR@gisterRule
                        \HorzR@gisterRule \hfil \HorzR@gisterRule}}}
\def\topregister{\ttopregister}
\def\botregister{\tbotregister}
```

3.26 Running headers and footers

```
\def\rtitlex{\def\texttub##1{{\normalsize\textrm{##1}}}\TUB, \volx}
```

registration marks; these are temporarily inserted in the running head

```
\def\MakeRegistrationMarks{}
\def\UseTrimMarks{%
  \def\MakeRegistrationMarks{%
    \ulap{\rlap{%
       \vbox{\dlap{\vbox to\trimlgt{\vfil\botregister}}%
             \topregister\vskip \headmargin \vskip 10\p@}}}}%
  }
% put issue identification and page number in header.
\def\@oddhead{\MakeRegistrationMarks
  \frenchspacing
  \normalsize\csname normalshape\endcsname\rm \tubheadhook
  \rtitlex\qquad \midrtitle\hfil
  \rtitlenexttopage\quad\tubtypesetpageno{\thepage}}
\def\@evenhead{\MakeRegistrationMarks
  \frenchspacing
  \normalsize\csname normalshape\endcsname\rm \tubheadhook
  \tubtypesetpageno{\thepage}\quad\rtitlenexttopage
  \hfil\midrtitle \qquad\rtitlex}

% Put a ? into the page number in the headers in all but a final run, so
% people aren't tempted to cite it.
%
\newcommand{\tubtypesetpageno}[1]{%
  \ifnum #1>900
    % in CM, numerals are exactly .5em.
    %
    % The \texorpdfstring avoids the usual hyperref warning:
    %   Token not allowed in a PDF string ... removing `\@ifnextchar'
    \texorpdfstring{\makebox[.5em][l]{\small ?}}{?}%
    %
    \textsl{\@arabic{\numexpr#1-900\relax}}% assuming e-tex
  \else
    \@arabic{#1}%
  \fi
}
%
% The above changes the page number in the headers and tocs. It does not
% change the page number in cross-references, which will still show up
% as `901' instead of `?1'. In order to do that, we'd have to redefine
% |\thepage| (https://tex.stackexchange.com/questions/687258).
%
% The problem is that |\thepage| is not expected to contain typesetting
% commands like |\makebox| and |\textsl|, but to expand to the simple
% page number (in whatever form). For example, when redefining
% |\thepage| to the above, terminal warnings then look like:
% |LaTeX Warning: Citation `foo' on page \makebox [.5em][l]{...|
% losing the actual page number.
%
% So apparently there is no way to add the ? correctly in all contexts.
%
% BTW, such a custom page number format would also break makeindex,
% etc., but for that we could provide the information. Per Ulrike:
%\usepackage{index}
```

```
%\newcommand\specialthepage{\inteval{\value{page}-900}}
%\newindex[specialthepage]*{default}{idx}{ind}{Index}

% can be used to reset the font, e.g., tb98kuester.
\def\tubheadhook{}

% in case the official \author is too verbose for the footline.
% (the \shortauthor / \rhAuthor stuff is only enabled for proceedings, fix!)
\def\tubrunningauthor{\@author}

% put title and author in footer.
\def\@tubrunningfull{%
  \def\@oddfoot{% make line break commands produce a normal space
    \def\\{\unskip\ \ignorespaces}%
    \let\newline=\\%
    \tubtypesetdoi
    \frenchspacing\hfil\rhTitle}
  \def\@evenfoot{%
    \let\thanks\@gobble
    \tubtypesetdoi
    \frenchspacing\tubrunningauthor\hfil}
}

% empty footer.
\def\@tubrunningminimal{%
  \def\@oddfoot{\tubtypesetdoi\hfil}%
  \def\@evenfoot{\tubtypesetdoi\hfil}%
}

% empty footer and header.
\def\@tubrunningoff{%
  \@tubrunningminimal
  \def\@oddhead{\hfil}%
  \def\@evenhead{\hfil}%
}

\def\ps@headings{}
\pagestyle{headings}
```

Typeset the doi. The format we decided on looks like: `https://doi.org/10.47397/tb/41-3/tb129n` where the last element is the `\jobname`.

We put this below the footline. The footer definitions above specify that it is always called, even if the regular footer is empty.

If the article started in the second column (option [secondcolstart]), we manually move the doi over.

We do not check for validity of `\volno`, `\issno`, `\jobname`. For testing, etc., seems simpler to just typeset what we've got. Other scripts will verify consistency.

```
%
\def\tubdoiprefix{10.47397/tb} % the number crossref assigned us
\def\tubabovedoi{}             % fudge spacing or whatever.
%
\def\tubtypesetdoi{%
  \iftubomitdoioption\else % do if not explicit omission ...
    \ifnum\volno>0         % and if being run for production ...
```

```
      \iftubfinaloption    % and if [final], even if pageno>900
       \vbox to 0pt{% don't impact normal layout
         \edef\thedoi{% but make url invalid if >900
                      \ifnum\count0>900 example.org%
                                  \else doi.org\fi
                       /\tubdoiprefix/\volno-\issno/\jobname}%
         \scriptsize
         \vskip\baselineskip
         \tubabovedoi
         \iftubsecondcolstart \moveright \tubcolwidthandgutter \fi
         \rlap{\expandafter\tbsurl\expandafter{\thedoi}}%
         \vss
       }%
     \fi  % tubfinaloption
   \fi    % volno>0
  \fi     % !tubomitdoioption
  \global\let\tubtypesetdoi\@empty % only do it once, no matter what.
}
%
%
```

3.27 Output routine

Modified to alter `\brokenpenalty` across columns

Comment We're playing with fire here: for example, `\@outputdblcol` has changed in LaTeX 2$_\varepsilon$ for 1995/06/01 (with the use of `\hb@xt@`). *This* time there's no semantic change, but...

```
\def\@outputdblcol{\if@firstcolumn \global\@firstcolumnfalse
    \global\setbox\@leftcolumn\box\@outputbox
    \global\brokenpenalty10000
  \else \global\@firstcolumntrue
    \global\brokenpenalty100
    \setbox\@outputbox\vbox{\hb@xt@\textwidth{\hb@xt@\columnwidth
      {\box\@leftcolumn \hfil}\hfil \vrule \@width\columnseprule\hfil
       \hb@xt@\columnwidth{\box\@outputbox \hfil}}}\@combinedblfloats
       \@outputpage \begingroup \@dblfloatplacement \@startdblcolumn
       \@whilesw\if@fcolmade \fi{\@outputpage\@startdblcolumn}\endgroup
    \fi}
```

3.28 Font-related definitions and machinery

These are mostly for compatibility with plain `tugboat.sty`

```
\newif\ifFirstPar       \FirstParfalse
\def\smc{\sc}
\def\ninepoint{\small}
⟨/classtail⟩
```

`\SMC` *isn't* small caps—Barbara Beeton says she thinks of it as "big small caps". She says (modulo capitalisation of things...):

> For the things it's used for, regular small caps are not appropriate—they're too small. Real small caps are appropriate for author names

(and are so used in continental bibliographies), section headings, running heads, and, on occasion, words to which some emphasis is to be given. \SMC was designed to be used for acronyms and all-caps abbreviations, which look terrible in small caps, but nearly as bad in all caps in the regular text size. The principle of using "one size smaller" than the text size is similar to the design of caps in German—where they are smaller relative to lowercase than are caps in fonts intended for English, to improve the appearance of regular text in which caps are used at the heads of all nouns, not just at the beginnings of sentences.

We define this in terms of the memory of the size currently selected that's maintained in \@currsize: if the user does something silly re. selecting fonts, we'll get the wrong results. The following code is adapted from an old version of relsize.sty by Donald Arseneau and Matt Swift. The order of examination of \@currsize is to do the most common cases first.

This is (I think) not a robust command since it's a conditional. https://tex.stackexchange.com/questi

```
⟨*common⟩
\newcommand{\SMC}{%
  \ifx\@currsize\normalsize\small\else
   \ifx\@currsize\small\footnotesize\else
    \ifx\@currsize\footnotesize\scriptsize\else
     \ifx\@currsize\large\normalsize\else
      \ifx\@currsize\Large\large\else
       \ifx\@currsize\LARGE\Large\else
        \ifx\@currsize\scriptsize\tiny\else
         \ifx\@currsize\tiny\tiny\else
          \ifx\@currsize\huge\LARGE\else
           \ifx\@currsize\Huge\huge\else
            \small\SMC@unknown@warning
 \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
}
\newcommand{\SMC@unknown@warning}{\TBWarning{\string\SMC: nonstandard
    text font size command -- using \string\small}}
\newcommand{\textSMC}[1]{{\SMC #1}}
```

The \acro command uses \SMC as it was originally intended. Since the argument never contains lowercase letters (except in error), it resets the spacefactor to 1000 after inserting the text. We define it using \NewDocumentCommand as an easy way to make the command robust, and thus be usable in titles when tagging is active. Suggestion from Ulrike Fischer: https://github.com/latex3/tagging-project/issues/1274#issuecomment-4078534466.

```
\NewDocumentCommand\acro{m}{\textSMC{#1}\@}
⟨/common⟩
```

3.29 Editor's notes and other footnotes

\EdNote allows the editor to enter notes in the text of a paper. If the command is given something that appears like an optional argument, the entire text of the note is placed in square brackets. (Yes, it really is!)

```
⟨*classtail⟩
\def\xEdNote{{\EdNoteFont Editor's note:\enspace }}
\def\EdNote{\@ifnextchar[%
```

```
  {%
    \ifvmode
      \smallskip\noindent\let\@EdNote@\@EdNote@v
    \else
      \unskip\quad\def\@EdNote@{\unskip\quad}%
    \fi
    \@EdNote
  }%
  \xEdNote
}
\long\def\@EdNote[#1]{%
  [\thinspace\xEdNote\ignorespaces
   #1%
   \unskip\thinspace]%
  \@EdNote@
}
\def\@EdNote@v{\par\smallskip}
```

Macros for Mittelbach's self-documenting style

```
\def\SelfDocumenting{%
  \setlength\textwidth{31pc}
  \onecolumn
  \parindent \z@
  \parskip 2\p@\@plus\p@\@minus\p@
  \oddsidemargin 8pc
  \evensidemargin 8pc
  \marginparwidth 8pc
  \toks@\expandafter{\@oddhead}%
  \xdef\@oddhead{\hss\hb@xt@\pagewd{\the\toks@}}%
  \toks@\expandafter{\@evenhead}%
  \xdef\@evenhead{\hss\hb@xt@\pagewd{\the\toks@}}%
  \def\ps@titlepage{}%
}
\def\ps@titlepage{}

% let's have a little space after the footnote marker.
% also a little more space for the marker itself.
\long\def\@makefntext#1{%
  \parindent 1em
  \noindent
  \hb@xt@2em{}%
  \llap{\@makefnmark}\null
  $\mskip5mu$% space after marker
  #1% footnote text
}
%
% For comparison, here is the original definition from classes.dtx:
% \long\def\@makefntext#1{%
%   \parindent 1em
%   \noindent
%   \hb@xt@2em{\hss\@makefnmark}#1}

```

`\tubraggedfoot` To get a ragged-right footnote.

```
\newcommand{\tubraggedfoot}{\rightskip=\raggedskip plus\raggedstretch\relax}
```

`\creditfootnote` `\supportfootnote` Sometimes we want the label "Editor's Note:", sometimes not.

```
\def\creditfootnote{\nomarkfootnote\xEdNote}
\def\supportfootnote{\nomarkfootnote\relax}
```

General macro `\nomarkfootnote` to make a footnote without a reference mark, etc. `#1` is an extra command to insert, `#2` the user's text.

```
\gdef\nomarkfootnote#1#2{\begingroup
  \def\thefootnote{}%
  % no period, please, also no fnmark. Also no hyperref warning.
  \def\@makefntext##1{##1}%
  \def\Hy@Warning##1{}%
  \footnotetext{\noindent #1#2}%
\endgroup}
```

3.30 Initialization

If we're going to use Harvard-style bibliographies, we set up the bibliography style: the user doesn't get any choice. (Not recommended.)

```
\if@Harvardcite
  \AtBeginDocument{%
    \bibliographystyle{ltugbib}%
  }
\fi
\authornumber\z@
\let\@signature\@defaultsignature
\InputIfFileExists{ltugboat.cfg}
  {\TBInfo{Loading ltugboat.cfg configuration information}}
  {}
```

⟨/classtail⟩

4 LaTeX 2ε proceedings class (no longer used)

`\@tugclass` Make the code of `ltugboat.cls` (when we load it) say it's really us:

```
⟨*ltugproccls⟩
\def\@tugclass{ltugproc}
```

`\if@proc@sober` `\if@proc@numerable` TUG'96 proceedings switched to more sober headings still; so the `tug95` option establishes the original state. In the absence of any other guidance, we use the '96 for TUG'97 proceedings, but also allow numbering of sections.

```
\newif\if@proc@sober
\newif\if@proc@numerable
\DeclareOption{tug95}{%
  \@proc@soberfalse
  \@proc@numerablefalse
}
\DeclareOption{tug96}{%
  \@proc@sobertrue
  \@proc@numerablefalse
}
```

```
\DeclareOption{tug97}{%
  \@proc@sobertrue
  \@proc@numerabletrue
}
\DeclareOption{tug2002}{%
  \@proc@sobertrue
  \@proc@numerabletrue
  \let\if@proc@numbersec\iftrue
  \PassOptionsToClass{numbersec}{ltugboat}%
}
```

`\if@proc@numbersec` If we're in a class that allows section numbering (the actual check occurs after `\ProcessOptions`, we can have the following:

```
\DeclareOption{numbersec}{\let\if@proc@numbersec\iftrue
  \PassOptionsToClass{numbersec}{ltugboat}%
}
\DeclareOption{nonumber}{\let\if@proc@numbersec\iffalse
  \PassOptionsToClass{nonumber}{ltugboat}%
}
```

`\ifTB@title` If we have a paper for which we want to create a detached title, with an editor's note, and then set the paper separately, we use option notitle.

```
\newif\ifTB@title
\DeclareOption{title}{\TB@titletrue}
\DeclareOption{notitle}{\TB@titlefalse
  \AtBeginDocument{\stepcounter{page}}}
```

There are these people who seem to think `tugproc` is an option as well as a class...

```
\DeclareOption{tugproc}{%
  \ClassWarning{\@tugclass}{Option \CurrentOption\space ignored}%
}
```

All other options are simply passed to `ltugboat`...

```
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{ltugboat}}
```

If there's a tugproc defaults file, input it now: it may tell us which year we're to perform for... (Note: this code *is* millenium-proof. It's not terribly classy for years beyond 2069, but then I'm not going to be around then—this will be an interesting task for a future TEXie...)

```
\InputIfFileExists{\@tugclass.cfg}{\ClassInfo{ltugproc}%
            {Loading ltugproc.cfg configuration information}}{}
\@ifundefined{TUGprocExtraOptions}%
   {\let\TUGprocExtraOptions\@empty}%
   {\edef\TUGprocExtraOptions{,\TUGprocExtraOptions}}
```

`\tugProcYear` Now work out what year it is

```
\@tempcnta\year
\ifnum\@tempcnta<2000
  \divide\@tempcnta by100
  \multiply\@tempcnta by100
  \advance\@tempcnta-\year
  \@tempcnta-\@tempcnta
\fi
```

And use that for calculating a year for us to use.

```
\edef\@tempa{\noexpand\providecommand\noexpand\tugProcYear
                {\ifnum10>\@tempcnta0\fi\the\@tempcnta}}
\@tempa
\ClassInfo{ltugproc}{Class believes year is
  \expandafter\ifnum\tugProcYear<2000 19\fi\tugProcYear
    \@gobble}
```

Check that this is a "sensible year" (one for which we have a class option defined). If not, make it a 'suitable' year, in particular, one that allows numbering sections.

```
\expandafter\ifx\csname ds@tug\tugProcYear\endcsname\relax
  \def\tugProcYear{2002}\fi
```

Now execute the default 'year' option and get on with processing. Note that this command gets ignored if the configuration file specifies a silly year.

```
\ExecuteOptions{tug\tugProcYear,title\TUGprocExtraOptions}
\ProcessOptions
\if@proc@numbersec
  \if@proc@numerable
  \else
    \ClassWarning{\@tugclass}{This year's proceedings may not have
      numbered sections}%
  \fi
\fi
```

Call ltugboat, adding whichever section numbering option is appropriate

```
\LoadClass[\if@proc@numbersec numbersec\else nonumber\fi]{ltugboat}
```

4.1 Proceedings titles

`\maketitle` `\ifTB@madetitle`

There's no provision for 'section titles' in proceedings issues, as there are in *TUGboat* proper. Note the tedious LaTeX bug-avoidance in the `\@TB@test@document` macro.

```
\def\maketitle{%
  \begingroup
```

first, a bit of flim-flam to generate an initial value for `\rhAuthor` (unless the user's already given one with a `\shortAuthor` comand).

```
    \ifshortAuthor\else
      \global\let\rhAuthor\@empty
      \def\g@addto@rhAuthor##1{%
        \begingroup
          \toks@\expandafter{\rhAuthor}%
          \let\thanks\@gobble
          \protected@xdef\rhAuthor{\the\toks@##1}%
        \endgroup
      }%
      \@getauthorlist\g@addto@rhAuthor
    \fi
```

now, the real business of setting the title

```
    \ifTB@title
```

```
      \setcounter{footnote}{0}%
      \renewcommand{\thefootnote}{\@fnsymbol\c@footnote}%
      \if@tubtwocolumn
        \twocolumn[\@maketitle]%
      \else
        \onecolumn
        \global\@topnum\z@
        \@maketitle
      \fi
      \@thanks
      \thispagestyle{TBproctitle}
    \fi
  \endgroup
  \TB@madetitletrue
}
\newif\ifTB@madetitle  \TB@madetitlefalse
```

`\@TB@test@document` `\@TB@test@document` checks to see, at entry to `\maketitle`, if we've had `\begin{document}`. See LaTeX bug report latex/2212, submitted by Robin Fairbairns, for details.

```
\def\@TB@test@document{%
  \edef\@tempa{\the\everypar}
  \def \@tempb{\@nodocument}
  \ifx \@tempa\@tempb
    \@nodocument
  \fi
}
```

`\AUTHORfont` `\TITLEfont` `\addressfont` `\netaddrfont` Define the fonts for titles and things

```
\def\AUTHORfont {\large\rmfamily\mdseries\upshape}
\def\TITLEfont  {\Large\rmfamily\mdseries\upshape}
\def\addressfont{\small\rmfamily\mdseries\upshape}
\def\netaddrfont{\small\ttfamily\mdseries\upshape}
```

`\aboveauthorskip` `\belowauthorskip` `\belowabstractskip` Some changeable skips to permit variability in page layout depending on the particular paper's page breaks.

```
\newskip\aboveauthorskip   \aboveauthorskip=18\p@ \@plus4\p@
\newskip\belowauthorskip   \belowauthorskip=\aboveauthorskip
\newskip\belowabstractskip \belowabstractskip=14\p@ \@plus3\p@ \@minus2\p@
```

`\@maketitle` The body of `\maketitle`

```
\def\@maketitle{%
  {\parskip\z@
   \frenchspacing
   \TITLEfont\raggedright\noindent\@title\par
     \count@=0
     \loop
     \ifnum\count@<\authornumber
       \vskip\aboveauthorskip
       \advance\count@\@ne
       {\AUTHORfont\theauthor{\number\count@}\endgraf}%
       \addressfont\theaddress{\number\count@}\endgraf
       {%
```

```
          \allowhyphens
          \hangindent1.5pc
          \netaddrfont\thenetaddress{\number\count@}\endgraf
          \hangindent1.5pc
          \thePersonalURL{\number\count@}\endgraf
        }%
      \repeat
  \vskip\belowauthorskip}%
  \if@abstract
     \centerline{\bfseries Abstract}%
     \vskip.5\baselineskip\rmfamily
     \@tubonecolumnabstractstart
           \the\abstract@toks
     \@tubonecolumnabstractfinish
     \global\@ignoretrue
  \fi
  \vskip\belowabstractskip
  \global\@afterindentfalse\aftergroup\@afterheading
}
```

`abstract` (*env.*) `\if@abstract` `\abstract@toks` Save the contents of the abstract environment in the token register `\abstract@toks`. We need to do this, as otherwise it may get 'typeset' (previously, it got put in a box) before `\begin{document}`, and experiments prove that this means our shiny new `\SMC` doesn't work in this situation.

If you need to understand the ins and outs of this code, look at the place I lifted it from: `tabularx.dtx` (in the tools bundle). The whole thing pivots on having stored the name of the 'abstract' environment in `\@abstract@`

```
\newtoks\abstract@toks  \abstract@toks{}
\let\if@abstract\iffalse
\def\abstract{%
```

we now warn unsuspecting users who provide an `abstract` environment *after* the `\maketitle` that would typeset it...

```
  \ifTB@madetitle
    \TBWarning{abstract environment after \string\maketitle}
  \fi
  \def\@abstract@{abstract}%
  \ifx\@currenvir\@abstract@
  \else
    \TBError{\string\abstract\space is illegal:%
      \MessageBreak
      use \string\begin{\@abstract@} instead}%
      {\@abstract@\space may only be used as an environment}
  \fi
  \global\let\if@abstract\iftrue
  {\ifnum0=`}\fi
  \@abstract@getbody}
\let\endabstract\relax
```

`\@abstract@getbody` gets chunks of the body (up to the next occurrence of `\end`) and appends them to `\abstract@toks`. It then uses `\@abstract@findend` to detect whether this `\end` is followed by `{abstract}`

```
\long\def\@abstract@getbody#1\end{%
  \global\abstract@toks\expandafter{\the\abstract@toks#1}%
```

```
  \@abstract@findend}
```

Here we've got to `\end` in the body of the abstract. `\@abstract@findend` takes the 'argument' of the `\end` do its argument.

```
\def\@abstract@findend#1{%
  \def\@tempa{#1}%
```

If we've found an 'end' to match the 'begin' that we started with, we're done with gathering the abstract up; otherwise we stuff the end itself into the token register and carry on.

```
  \ifx\@tempa\@abstract@
    \expandafter\@abstract@end
  \else
```

It's not `\end{abstract}`—check that it's not `\end{document}` either (which signifies that the author's forgotten about ending the abstract)

```
    \def\@tempb{document}%
    \ifx\@tempa\@tempb
      \TBError{\string\begin{\@abstract@}
          ended by \string\end{\@tempb}}%
        {You've forgotten \string\end{\@abstract@}}
    \else
       \global\abstract@toks\expandafter{\the\abstract@toks\end{#1}}%
       \expandafter\expandafter\expandafter\@abstract@getbody
    \fi
  \fi}
```

In our case, the action at the 'proper' `\end` is a lot simpler than what appears in `tabularx.dtx` ... don't be surprised!

```
\def\@abstract@end{\ifnum0=`{\fi}%
  \expandafter\end\expandafter{\@abstract@}}
```

`\makesignature` `\makesignature` is improper in proceedings, so we replace it with a warning (and a no-op otherwise)

```
\renewcommand{\makesignature}{\TBWarning
         {\string\makesignature\space is invalid in proceedings issues}}
```

`\ps@TBproctitle` `\ps@TBproc` `\dopagecommands` `\setpagecommands` `\TB@definefeet` `\pfoottext` `\rfoottext` Now we define the running heads in terms of the `\rh*` commands.

```
\def\ps@TBproctitle{\let\@oddhead\MakeRegistrationMarks
  \let\@evenhead\MakeRegistrationMarks
  \TB@definefeet
}
\def\ps@TBproc{%
  \def\@oddhead{\MakeRegistrationMarks
    {%
      \hfil
      \def\\{\unskip\ \ignorespaces}%
      \rmfamily\rhTitle
    }%
  }%
  \def\@evenhead{\MakeRegistrationMarks
    {%
      \def\\{\unskip\ \ignorespaces}%
      \rmfamily\rhAuthor
```

```
      \hfil
    }%
  }%
  \TB@definefeet
}

\advance\footskip8\p@    % for deeper running feet

\def\dopagecommands{\csname @@pagecommands\number\c@page\endcsname}
\def\setpagecommands#1#2{\expandafter\def\csname @@pagecommands#1\endcsname
  {#2}}
\def\TB@definefeet{%
  \def\@oddfoot{\ifpreprint\pfoottext\hfil\Now\hfil\thepage
    \else\rfoottext\hfil\thepage\fi\dopagecommands}%
  \def\@evenfoot{\ifpreprint\thepage\hfil\Now\hfil\pfoottext
    \else\thepage\hfil\rfoottext\fi\dopagecommands}%
}

\def\pfoottext{{\smc Preprint}:
   Proceedings of the \volyr{} Annual Meeting}
\def\rfoottext{\normalfont\TUB, \volx\Dash
   {Proceedings of the \volyr{} Annual Meeting}}

\pagestyle{TBproc}
```

4.2 Section divisions

Neither sections nor subsections are numbered by default in the proceedings style: note that this puts a degree of stress on authors' natural tendency to reference sections, which is a matter that needs attention. The class option NUMBERSEC once again numbers the sections (and noticeably changes the layout).

```
\if@proc@numbersec
\else
  \setcounter{secnumdepth}{0}
\fi
```

Otherwise, the `\section` command is pretty straightforward. However, the `\subsection` and `\subsubsection` are run-in, and we have to remember to have negative stretch (and shrink if we should in future choose to have one) on the ⟨*afterskip*⟩ parameter of `\@startsection`, since the whole skip is going to end up getting negated. We use `\TB@startsection` to detect inappropriate forms.

```
\if@proc@numbersec
\else
  \if@proc@sober
    \def\section
          {\TB@nolimelabel
           \TB@startsection{{section}%
                            1%
                            \z@%
                            {-8\p@\@plus-2\p@\@minus-2\p@}%
                            {6\p@}%
                            {\normalsize\bfseries\raggedright}}}
  \else
```

```
    \def\section
           {\TB@nolimelabel
            \TB@startsection{{section}%
                             1%
                             \z@%
                             {-8\p@\@plus-2\p@\@minus-2\p@}%
                             {6\p@}%
                             {\large\bfseries\raggedright}}}
  \fi
  \def\subsection
           {\TB@nolimelabel
            \TB@startsection{{subsection}%
                             2%
                             \z@%
                             {6\p@\@plus 2\p@\@minus2\p@}%
                             {-5\p@\@plus -\fontdimen3\the\font}%
                             {\normalsize\bfseries}}}
  \def\subsubsection
           {\TB@nolimelabel
            \TB@startsection{{subsubsection}%
                             3%
                             \parindent%
                             \z@%
                             {-5\p@\@plus -\fontdimen3\the\font}%
                             {\normalsize\bfseries}}}
\fi
⟨/ltugproccls⟩
```

5 Plain TEX styles

```
⟨*tugboatsty⟩
% err...
⟨/tugboatsty⟩
⟨*tugprocsty⟩
% err...
⟨/tugprocsty⟩
```

6 The LATEX 2ε compatibility-mode style files

```
⟨*ltugboatsty⟩
\@obsoletefile{ltugboat.cls}{ltugboat.sty}
\LoadClass{ltugboat}
⟨/ltugboatsty⟩
⟨*ltugprocsty⟩
\@obsoletefile{ltugproc.cls}{ltugproc.sty}
\LoadClass{ltugproc}
⟨/ltugprocsty⟩
```