The fixltx2e and fix-cm packages*

Frank Mittelbach, David Carlisle, Chris Rowley, Walter Schmidt†

2006/03/24

Abstract

These packages provides fixes to LaTeX 2_ε which are desirable but cannot be integrated into the LaTeX 2_ε kernel or the font definition files directly as they would produce a version incompatible to earlier releases (either in formatting or functionality).

By providing these fixes in the form of packages, users can benefit from them without the danger that their documents will fail or produce unexpected results at other sites since the documents contain a clear indication (the `\usepackage` line, preferably with a required date) that the fixes are needed.

Contents

---

*This file has version number v1.1n, last revised 2006/03/24.

†Walter wrote fix-cm

1 Introduction

In the newsletter `ltnews07.tex`, which accompanied the LaTeX 2_ε maintenance release of June 1997, we wrote:

> Many of the problem reports we receive concerning the standard classes are not concerned with bugs but are suggesting, more or less politely, that the design decisions embodied in them are 'not optimal' and asking us to modify them.
>
> There are several reasons why we have decided not to make such changes to these files.
>
> - However misguided, the current behaviour is clearly what was intended when these classes were designed.
> - It is not good practice to change such aspects of 'standard classes' because many people will be relying on them.
>
> We have therefore decided not to even consider making such modifications, nor to spend time justifying that decision. This does not mean that we do not agree that there are many deficiencies in the design of these classes, but we have many tasks with higher priority than continually explaining why the standard classes for LaTeX cannot be changed.

Back then we probably should have said that this decision also covers changes to the LaTeX kernel and font definitions, if the change results in noticeable differences in the formatting of documents or otherwise produces severe incompatibilities between releases. The important point to stress here is that "people rely on the fact that a document formatted at one site produces identical output at a different site". By fixing a certain problem in version $\langle date \rangle$, people making use of the fix will get incorrectly formatted documents if they send them to others who still run on a version prior to $\langle date \rangle$.

In theory one could get around this by adding a line like

    \NeedsTeXFormat{latex2e}[⟨date⟩]

on top of the document. However, this fails for two reasons. Firstly, most people will not be aware that they make use of a feature or fix that is only available in their version of LaTeX; and thus do not add such a line in their documents. Secondly, even if there is such a line the receiving site might not be able to upgrade their LaTeX in time to process the document properly (the latter is a sad fact of life).

By providing the fixltx2e and fix-cm packages we hope to help people in this respect since, when they are used, a document will contain a clear indication that special features/fixes are needed and if the receiving site does not have the packages available (or not available with the right version) it is far easier to download and install them from some archive than to upgrade LaTeX in a rush.

The packages are independent from each other and deal with different subjects: fixltx2e provides general changes to the LaTeX kernel, while fix-cm improves the definitions of the Computer Modern font families.

We will try to maintain the packages in such a way that they can be used with all maintenance releases of LaTeX 2_ε so that, if urgently needed, people can simply add them to the current directory in case they cannot upgrade their LaTeX for whatever reason.

The packages are **NOT** provided so that people can stop upgrading their LaTeX system. They will contain only fixes of a certain nature, others will still go into the kernel and extensions in form of packages, and support files will still be added to the base system at regular intervals.

1.1 Using **fixltx2e**

To use the fixltx2e package include the line

> \usepackage{fixltx2e}[⟨*date*⟩]

into the preamble of your document, where ⟨*date*⟩ is the date of the fixltx2e package that you are using. This way your document will produce a warning if processed at a site that only has an older version of of this package.

1.2 Using **fix-cm**

!!

To use the fix-cm package, load it *before* \documentclass, and use the command \RequirePackage to do so, rather than the normal \usepackage:

> \RequirePackage{fix-cm}
> \documentclass ...

Do not to load any other package before the document class, unless you have a thorough understanding of the LaTeX internals and know exactly what you are doing!

2 Fixes added

This section describes all the fixes/features that have been added to the initial release of the package. If applicable the bug report info (see `bugs.txt`) is given.

2.1 2-col: 1-col fig can come before earlier 2-col fig (pr/2346)

```
>Number:        2346
>Category:      latex
>Synopsis:      2-col: 1-col fig can come before earlier 2-col fig
>Arrival-Date:  Wed Dec 18 15:41:07 1996
>Originator:    w.l.kleb@larc.nasa.gov (bil kleb)
>Description:
as documented in lamport's book, p. 198, concerning figure
placement, "a figure will not be printed before an earlier
figure, and a table will not be printed before an earlier
table."  however, there is a footnote stating, "However,
in two-column page style, a single-column figure can come before
an earlier double-column figure, and vice versa."

this twocolumn behavior is undesireable---at least by me and
most professional organizations i publish in.  ed snyzter developed
a hack fix for 2.09 several years ago which links the two
counters, but i have not run across a similar "fix" for 2e...
```

Originally fixed in package fix2col which was merged into this package. Documentation and code from this package have been merged into this file.

2.1.1 Notes on the Implementation Strategy

The standard output routine maintains two lists of floats that have been 'deferred' for later consideration. One list for single column floats, and one for double column floats (which are always immediately put onto their deferred list). This mechanism means that LaTeX 'knows' which type of float is contained in each box by the list that it is processing, but having two lists means that there is no mechanism for preserving the order between the floats in each list.

The solution to this problem consists of two small changes to the output routine.

Firstly, abandon the 'double column float list' `\@dbldeferlist` and change every command where it is used so that instead the same `\@deferlist` is used as for single column floats. That one change ensures that double and single column floats stay in the same sequence, but as LaTeX no longer 'knows' whether a float is double or single column, it will happily insert a double float into a single column, overprinting the other column, or the margin.

The second change is to provide an alternative mechanism for recording the two column floats. LaTeX already has a compact mechanism for recording float information, an integer count register assigned to each float records information about the 'type' of float 'figure', 'table' and the position information 'htp' etc.

The type information is stored in the 'high' bits, one bit position (above '32') allocated to each float type. The 'low' bits store information about the allowed positions, one bit each allocated for `h t b p`. In the LaTeX2.09 system, the bit corresponding to '16' formed a 'boundary' between these two sets of information, and it was never actually used by the system. Ed Sznyter's fixfloats package not unreasonably used this position to store the double column information, setting the bit for double column floats. Then at each point in the output routine at which a float is committed to a certain region, an additional check must be made to check that the float is (or is not) double column. If it spans the wrong number of columns it is deferred rather than being added.

Unfortunately the bit '16' is not available in LaTeX 2_ε. It is used to encode the extra float position possibility '!' that was added in that system. It would be possible to use position '32' and to move the flags for 'table', 'figure',... up one position, to start at 64, but this would mean that in principle one less float type would be supported, and more importantly is likely to break any other packages that assume anything about the output routine internals. So here I instead use another mechanism for flagging double column floats: By default all floats have depth 0pt. This package arranges that double column ones have depth 1sp. This information may then be used in the same manner as in the fixfloats package, to defer any floats that are not of the correct column spanning type.

2.2   Wrong header for twocolumn (pr/2613)

```
>Number:         2613
>Category:       latex
>Synopsis:       wrong headline for twocolumn
>Arrival-Date:   Mon Sep 22 16:41:09 1997
>Originator:     daniel@cs.uni-bonn.de (Daniel Reischert)
>Description:
When setting the document in two columns
the headline shows the top mark of the second column,
but it should show the top mark of the first column.
```

Originally fixed in package fix2col which was merged into this package. Documentation and code from this package have been merged into this file.

2.2.1   Notes on the Implementation Strategy

The standard LaTeX twocolumn system works internally by making each column a separate 'page' that is passed independently to TeX's pagebreaker. (Unlike say the multicol package, where all columns are gathered together and then split into columns later, using `\vsplit`.) This means that the primitive TeX marks that are normally used for header information, are globally reset after the first column. By default LaTeX does nothing about this. A good solution is provided by Piet van Oostrum (building on earlier work of Joe Pallas) in his fixmarks package.

After the first column box has been collected the mark information for that box is saved, so that any `\firstmark` can be 'artificially' used to set the page-level marks after the second column has been collected. (The second column

`\firstmark` is not normally required.) Unfortunately TeX does not provide a direct way of knowing if any marks are in the page, `\firstmark` always has a value from previous pages, even if there is no mark in this page. The solution is to make a copy of the box and then `\vsplit` it so that any marks show up as `\splitfirstmark`.

The use of `\vsplit` does mean that the output routine will globally change the value of `\splitfirstmark` and `\splitbotmark`. The fixmarks package goes to some trouble to save and restore these values so that the output routine does *not* change the values. This part of fixmarks is not copied here as it is quite costly (having to be run on every page) and there is no reason why anyone writing code using `\vsplit` should allow the output routine to be triggered before the split marks have been accessed.

2.3 `\@` discards spaces when moving (pr/3039)

```
>Number:         3039
>Category:       latex
>Synopsis:       \@ discards spaces when moving
>Arrival-Date:   Sat May 22 09:01:06 1999
>Originator:     asnd@triumf.ca (Donald Arseneau)
>Description:
The \@ command expands to \spacefactor\@m in auxiliary files,
which then ignores following spaces when it is reprocessed.
```

2.4 `\setlength` produces error if used with registers like `\dimen0` (pr/3066)

```
>Number:         3066
>Category:       latex
>Synopsis:       \setlength{\dimen0}{10pt}
>Arrival-Date:   Tue Jul  6 15:01:06 1999
>Originator:     oberdiek@ruf.uni-freiburg.de (Heiko Oberdiek)
>Description:
The current implementation of \setlength causes an error,
because the length specification isn't terminated properly.
More safe:
\def\setlength#1#2{#1=#2\relax}
```

2.5 `\addpenalty` ruins flush-bottom (pr/3073)

```
>Number:         3073
>Category:       latex
>Synopsis:       \addpenalty ruins flush-bottom
>Arrival-Date:   Sat Jul 17 05:11:05 1999
>Originator:     asnd@triumf.ca (Donald Arseneau)
>Description:
Just to keep in mind for further development eh?
A page break at an \addpenalty after \vspace does *not*
give a flush-bottom page.  (The intent of \addpenalty is
apparently just to preserve the flush bottom by putting
the breakpoint 'above' the skip.)
```

3   Fixes added for 2003/06/01

3.1 `\fnsymbol` should use text symbols if possible (pr/3400)

```
>Number:         3400
>Category:       latex
>Synopsis:       \fnsymbol should use text symbols if possible
>Arrival-Date:   Fri Jan 04 20:41:00 CET 2002
```

```
>Originator:    was@VR-Web.de  ( Walter Schmidt )
```

```
The \fnsymbol command can be used in both text and math
mode.  The symbols produced are, however, always taken from
the math fonts.  As a result, they may not match the text
fonts, even if the symbols are actually available, for
instance from the TS1 encoding.  Since \fnsymbol is
primarily used for footnotes in text, this should be fixed,
IMO.
```

3.2 No hyphenation in first word after float environment (pr/3498)

```
>Number:        3498
>Category:      latex
>Synopsis:      No hyphenation in first word after float environment
>Arrival-Date:  Thu Jan 30 13:21:00 CET 2003
>Originator:    h.harders@tu-bs.de (Harald Harders)
```

```
If a float environment (figure, table) is written within a paragraph,
the first word after the environment is not hyphenated.
```

3.3 Allowing \emph to produce small caps, etc

By default \em or \emph switches to roman in an italic context but some designers prefer a switch to small caps in that case. This can be achieved by setting \eminnershape, e.g.,

```
\renewcommand\eminnershape{\scshape}
```

3.4 Using EC fonts (T1 encoding) makes my documents look bl**dy horrible (from c.t.t.)
I can't use arbitrary sizes with CM fonts (from c.t.t.)

No I'm not trying to collect any cites from the news group discussion on this topic. In a nutshell, if one adds

```
\usepackage[T1]{fontenc}
```

to a document that uses the Computer Modern typefaces, then not only the T1 encoding is used but the fonts used in the document look noticeably different. This is due to the fact that the EC fonts have more font series designs, e.g. a 14.4 pt bold etc and those get used in the standard `.fd` files, while with Computer Modern (in OT1 encoding) such sizes were scaled versions of smaller sizes—with a noticeable different look and feel.

So we provide a package fix-cm to ensure that comparable definitions are used. In addition to that, the package fix-cm also enables continuous scaling of the CM fonts. This package was written by Walter Schmidt.

3.4.1 What fix-cm does

Loading the package fix-cm changes the font definitions of the Computer Modern fonts, in order to achieve the following effects:

- The appearance of the T1 and TS1 encoded CM fonts (aka 'EC') is made as similar as possible to the traditional (OT1 encoded) ones. Particularly, a number of broken or ugly design sizes are no longer used, the look of the bold sans serif typeface at large sizes is considerably improved, and mismatches between the text fonts and the corresponding math fonts are avoided. As a side effect, PostScript and PDF documents may become smaller, because fewer fonts need to be embedded.

- The Computer Modern fonts are made available with arbitrary sizes.

- Only those design sizes of the fonts will be used, that are normally available in Type1 format, too. You need not load the extra package cmmib57 for this purpose.

The package acts on the following font families:

- The text font families cmr, cmss, cmtt and cmvtt with OT1, T1 and TS1 encoding.

- The default math fonts used by LATEX, i.e., the font families cmm with encoding OML and cms with encoding OMS.

- The symbols used by the package latexsym, i.e., the font family lasy.

Note that the package does *not* act on:

- Font families such as CM Fibonacci, CM Dunhill etc., which are provided for experimental purposes or for fun only.

- CM text fonts with character sets other than Latin, e.g., Cyrillic. Loading of the required font and encoding definitions while the fonts are not actually used, would not be a good idea. This should be addressed by particular packages or by changing the standard FDs of these fonts.

- Extra math fonts such as the AMS symbol fonts. While they match the style of Computer Modern, they are frequently used in conjunction with other font families, too. Thus, fix-cm is obviously not the right place to make sure that they can be scaled continuously. Ask the maintainers of these fonts to provide this feature, which is badly needed!

- The math extension font cmex. Whether or not this font should be scaled is a question of its own, and there are other packages (exscale, amsmath, amsfonts) to take care of it.

3.4.2 How to load the package

!! The package should be loaded *before* \documentclass, using the command \RequirePackage{fix-cm}, rather than the normal \usepackage. Rationale: If the package is loaded in the preamble, a preceding package or even the code of the document class may have used any of the CM fonts already. However, the definitions of those fonts, that are already in use, cannot be changed any more.

3.4.3 Usage notes

In contrast to what you may expect, fix-cm does *not* ensure that line and page breaks stay the same, when you switch an existing document from OT1 to T1 encoding. The package does not turn off all of the additional design sizes in the EC fonts collection: Those, that contribute considerably to the typographical quality and do not conflict with the math fonts, are—indeed—used.

!! Be careful when using arbitrary, non-standard font sizes with applications that need bitmap fonts: You may end up with lots of possibly huge .pk files. Also, Metafont chokes sometimes on extremely small or large sizes, because of arithmetic problems.

fix-cm supersedes the experimental packages cmsd and fix-ec, which are no longer distributed.

The packages type1cm and type1ec must not be loaded additionally; they enable only continuous scaling.

4 Fixes added for 2005/12/01

4.1 \textsubscript not defined in latex.ltx (pr/3492)

```
>Number:         3492
>Category:       latex
>Synopsis:       \textsubscript not defined in latex.ltx
>Arrival-Date:   Tue Jan 14 23:01:00 CET 2003
>Originator:     tgakic@chem.tue.nl  (Ionel Mugurel Ciobica)


I use \textsubscript much more often than \textsuperscript, and
\textsubscript it is not defined in latex.ltx. Could you please
consider including the definition of \textsubscript in the latex.ltx
for the next versions of LaTeX.    Thank you.
```

4.2 \DeclareMathSizes only take pts. (pr/3693)

```
>Number:         3693
>Category:       latex
>Synopsis:       \DeclareMathSizes only take pts.
>Arrival-Date:   Fri Jun 11 16:21:00 CEST 2004
>Originator:     moho01ab@student.cbs.dk  (Morten Hoegholm)


The last three arguments of \@DeclareMathSizes cannot take a dimension
as argument, making it inconsistent with the rest of the font changing
commands and itself, as the second argument can take a dimension
specification.
```

4.3 \addpenalty ruins flush-bottom (pr/3073)

```
>Number:         3073
>Category:       latex
>Synopsis:       \addpenalty ruins flush-bottom
>Arrival-Date:   20 Oct 2005 14:46:35 -0700
>Originator:     asnd@triumf.ca (Donald Arseneau)
>Description:
 The (revised) definition of \addpenalty has been
 incorporated into fixltx2e, but now Plamen Tanovski has found a
 problem:  since the \vskip is increased by the previous depth,
 consecutive \addpenalty and \addvspace commands keep enlarging
 the \vskip.
```

4.4 \footnotemark[x] crashes with fixltx2e.sty (pr/3752)

```
>Number:         3752
>Category:       tools
>Synopsis:       feature \footnotemark[x] crashes with fixltx2e.sty
>Arrival-Date:   Fri Dec 17 10:11:00 +0100 2004
>Originator:     stefan.pofahl@zsw-bw.de (Stefan Pofahl)


 If I use /fnsymbol together with fixltx2e.sty I can not use
 optinal parameter [num]
 \footnotemark[1] is not showing the mark number 1 but
 the mark \value{footnote}.
```

This bug was related to pr/3400, where **\@fnsymbol** was made robust.

4.4.1 Notes on the implementation strategy

Pr/3400 made **\@fnsymbol** decide between text-mode and math-mode, which requires a certain level of robustness somewhere as the decision between text and math must be made at typesetting time and not when inside **\protected@edef**

or similar commands. One way of dealing with this is to make sure the value seen by `\@fnsymbol` is a fully expanded number, which could be handled by code such as

```
\def\fnsymbol#1{\expandafter\@fnsymbol
  \expandafter{\the\csname c@#1\endcsname}}
```

This would be a good solution if everybody used the high level commands only by writing code like `\fnsymbol{footnote}`. Unfortunately many classes (including the standard classes) and packages use the internal forms directly as in `\@fnsymbol\c@footnote` so the easy solution of changing `\fnsymbol` would break code that had worked for the past 20 years.

`\TextOrMath`   Therefore the implementation here makes `\@fnsymbol` itself a non-robust command again and instead uses a new robust command `\TextOrMath`, which will take care of typesetting either the math or the text symbol. In order to do so, we face an age old problem and unsolvable problem in TeX: A reliable test for math mode that doesn't destroy kerning. Fortunately this problem can be solved when using eTeX so if you use this as engine for your LaTeX format, as recommended by the LaTeX3 Project, you will get a fully functioning `\TextOrMath` command with no side effects. If you use regular TeX as engine for your LaTeX format then we have to choose between the lesser of two evils: 1) breaking ligatures and preventing kerning or 2) face the risk of choosing text-mode at the beginning of an alignment cell, which was suppodes to be math-mode. We have decided upon 1) as is costumary for regular robust commands in LaTeX.

4.5   Fewer fragile commands

```
>Number:        3816
>Category:      latex
>Synopsis:      Argument of \@sect has an extra }.
>Arrival-Date:  Sat Oct 22 23:11:01 +0200 2005
>Originator:    susi@uriah.heep.sax.de (Susanne Wunsch)


Use of a \raisebox in \section{} produces the error message
mentioned in the subject.


PR latex/1738 described a similar problem, which has been solved
10 years ago. Protecting the \raisebox with \protect solved my
problem as well, but wouldn't it make sense to have a similar fix
as in the PR?


It is particulary confusing, that an unprotected \raisebox in a
\section*-environment works fine, while in a \section-environment
produces error.
```

While not technically a bug, in this day and age there are few reasons why commands taking optional arguments should not be robust.

4.5.1   Notes on the implementation strategy

Rather than changing the kernel macros to be robust, we have decided to add the
`\MakeRobust`   macro `\MakeRobust` in fixltx2e so that users can easily turn fragile macros into robust ones. A macro `\foo` is made robust by doing the simple `\MakeRobust{\foo}`. fixltx2e makes the following kernel macros robust: `\(`, `\)`, `\[`, `\]`, `\makebox`, `\savebox`, `\framebox`, `\parbox`, `\rule` and `\raisebox`.

5   Implementation

We require at least a somewhat sane version of LaTeX 2ε. Earlier ones where really quite different from one another.

```
1 ⟨∗fixltx2e⟩
2 \NeedsTeXFormat{LaTeX2e}[1996/06/01]
```

5.1 2-col: 1-col fig can come before earlier 2-col fig (pr/2346) Wrong headline for twocolumn (pr/2613)

Originally fixed in package fix2col which was merged into this package. Code and documentation are straight copies from that package.

5.1.1 Preserving Marks

This is just a change to the single command `\@outputdblcol` so that it saves mark information for the first column and restores it in the second column.

```
3 \def\@outputdblcol{%
4   \if@firstcolumn
5     \global\@firstcolumnfalse
```

Save the left column

```
6     \global\setbox\@leftcolumn\copy\@outputbox
```

Remember the marks from the first column

```
7     \splitmaxdepth\maxdimen
8     \vbadness\maxdimen
9     \setbox\@outputbox\vsplit\@outputbox to\maxdimen
```

One minor difference from the current fixmarks, pass the marks through a token register to stop any `#` tokens causing an error in a `\def`.

```
10    \toks@\expandafter{\topmark}%
11    \xdef\@firstcoltopmark{\the\toks@}%
12    \toks@\expandafter{\splitfirstmark}%
13    \xdef\@firstcolfirstmark{\the\toks@}%
```

This test does not work if truly empty marks have been inserted, but LaTeX marks should always have (at least) two brace groups. (Except before the first mark is used, when the marks are empty, but that is OK here.)

```
14    \ifx\@firstcolfirstmark\@empty
15      \global\let\@setmarks\relax
16    \else
17      \gdef\@setmarks{%
18        \let\firstmark\@firstcolfirstmark
19        \let\topmark\@firstcoltopmark}%
20    \fi
```

End of change

```
21  \else
22    \global\@firstcolumntrue
23    \setbox\@outputbox\vbox{%
24     \hb@xt@\textwidth{%
25        \hb@xt@\columnwidth{\box\@leftcolumn \hss}%
26        \hfil
27        \vrule \@width\columnseprule
28        \hfil
29        \hb@xt@\columnwidth{\box\@outputbox \hss}}}%
30  \@combinedblfloats
```

Override current first and top with those of first column if necessary

```
31    \@setmarks
```

End of change

```
32    \@outputpage
33    \begingroup
34      \@dblfloatplacement
35      \@startdblcolumn
36      \@whilesw\if@fcolmade \fi{\@outputpage\@startdblcolumn}%
```

```
37      \endgroup
38   \fi}
```

5.1.2   Preserving Float Order

Changes `\@dbldeferlist` to `\@deferlist` are not explicitly noted but are flagged by blank comment lines around the changed line.

```
39 \def\end@dblfloat{%
40 \if@twocolumn
41   \@endfloatbox
42   \ifnum\@floatpenalty <\z@
43     \@largefloatcheck
```

Force the depth of two column float boxes.

```
44     \global\dp\@currbox1sp %
```

Next line assumes that first token of `\end@float` is `\@endfloatbox` so we gobble that.

```
45 %     \@cons\@deferlist\@currbox
46       \expandafter\@gobble\end@float
```

`\@Esphack` is then added by `\@endfloat` above.

```
47   \fi
48 %   \ifnum \@floatpenalty =-\@Mii \@Esphack\fi
49 \else
50   \end@float
51 \fi
52 }
```

Test if the float box has the wrong width. (Actually as noted above the test is for a conventional depth setting rather than for the width of the float).

```
53 \def\@testwrongwidth #1{%
54   \ifdim\dp#1=\f@depth
55   \else
56     \global\@testtrue
57   \fi}
```

Normally looking for single column floats, which have zero depth.

```
58 \let\f@depth\z@
```

but when making two column float area, look for floats with 1sp depth.

```
59 \def\@dblfloatplacement{\global\@dbltopnum\c@dbltopnumber
60     \global\@dbltoproom \dbltopfraction\@colht
61     \@textmin \@colht
62     \advance \@textmin -\@dbltoproom
63     \@fpmin \dblfloatpagefraction\textheight
64     \@fptop \@dblfptop
65     \@fpsep \@dblfpsep
66     \@fpbot \@dblfpbot
67     \def\f@depth{1sp}}
```

All the remaining changes are replacing the double column defer list or insering the extra test `\@testwrongwidth{⟨box⟩}` at suitable places. That is at plces where a box is taken off the deferlist.

```
68 \def \@doclearpage {%
69     \ifvoid\footins
70       \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
71       \setbox\@tempboxa\box\@cclv
72       \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
73       \global \let \@toplist \@empty
74       \global \let \@botlist \@empty
75       \global \@colroom \@colht
76       \ifx \@currlist\@empty
77       \else
```

```
78          \@latexerr{Float(s) lost}\@ehb
79          \global \let \@currlist \@empty
80        \fi
81      \@makefcolumn\@deferlist
82      \@whilesw\if@fcolmade \fi{\@opcol\@makefcolumn\@deferlist}%
83      \if@twocolumn
84        \if@firstcolumn

85          \xdef\@deferlist{\@dbltoplist\@deferlist}%

86          \global \let \@dbltoplist \@empty
87          \global \@colht \textheight
88          \begingroup
89            \@dblfloatplacement

90            \@makefcolumn\@deferlist
91            \@whilesw\if@fcolmade \fi{\@outputpage
92                                      \@makefcolumn\@deferlist}%

93          \endgroup
94        \else
95          \vbox{}\clearpage
96        \fi
97      \fi
```

the next line is needed to avoid loosing floats in certain circumstances a single call
to the original \doclearpage will now no longer output all floats.

```
98        \ifx\@deferlist\@empty \else\clearpage \fi
99      \else
100       \setbox\@cclv\vbox{\box\@cclv\vfil}%
101       \@makecol\@opcol
102       \clearpage
103     \fi
104 }
105 \def \@startdblcolumn {%
106   \@tryfcolumn \@deferlist
107   \if@fcolmade
108   \else
109     \begingroup
110       \let \reserved@b \@deferlist
111       \global \let \@deferlist \@empty
112       \let \@elt \@sdblcolelt
113       \reserved@b
114     \endgroup
115   \fi
116 }
117 \def\@addtonextcol{%
118   \begingroup
119     \@insertfalse
120     \@setfloattypecounts
121     \ifnum \@fpstype=8
122     \else
123       \ifnum \@fpstype=24
124       \else
125         \@flsettextmin
126         \@reqcolroom \ht\@currbox
127         \advance \@reqcolroom \@textmin
128         \ifdim \@colroom>\@reqcolroom
129           \@flsetnum \@colnum
130           \ifnum\@colnum>\z@
131             \@bitor\@currtype\@deferlist
132             \@testwrongwidth\@currbox
133             \if@test
134             \else
```

```
135            \@addtotoporbot
136          \fi
137        \fi
138      \fi
139    \fi
140    \fi
141    \if@insert
142    \else
143      \@cons\@deferlist\@currbox
144    \fi
145  \endgroup
146 }
147 \def\@addtodblcol{%
148   \begingroup
149   \@insertfalse
150   \@setfloattypecounts
151   \@getfpsbit \tw@
152   \ifodd\@tempcnta
153     \@flsetnum \@dbltopnum
154     \ifnum \@dbltopnum>\z@
155       \@tempswafalse
156       \ifdim \@dbltoproom>\ht\@currbox
157         \@tempswatrue
158       \else
159         \ifnum \@fpstype<\sixt@@n
160           \advance \@dbltoproom \@textmin
161           \ifdim \@dbltoproom>\ht\@currbox
162             \@tempswatrue
163           \fi
164           \advance \@dbltoproom -\@textmin
165         \fi
166       \fi
167       \if@tempswa
168         \@bitor \@currtype \@deferlist
```

not in fixfloats?

```
169         \@testwrongwidth\@currbox
170         \if@test
171         \else
172           \@tempdima -\ht\@currbox
173           \advance\@tempdima
174             -\ifx \@dbltoplist\@empty \dbltextfloatsep \else
175                                     \dblfloatsep \fi
176           \global \advance \@dbltoproom \@tempdima
177           \global \advance \@colht \@tempdima
178           \global \advance \@dbltopnum \m@ne
179           \@cons \@dbltoplist \@currbox
180           \@inserttrue
181         \fi
182       \fi
183     \fi
184   \fi
185   \if@insert
186   \else
187     \@cons\@deferlist\@currbox
188   \fi
189  \endgroup
190 }
191 \def \@addtocurcol {%
192    \@insertfalse
193    \@setfloattypecounts
194    \ifnum \@fpstype=8
```

```
195    \else
196      \ifnum \@fpstype=24
197      \else
198        \@flsettextmin
199        \advance \@textmin \@textfloatsheight
200        \@reqcolroom \@pageht
201        \ifdim \@textmin>\@reqcolroom
202          \@reqcolroom \@textmin
203        \fi
204        \advance \@reqcolroom \ht\@currbox
205        \ifdim \@colroom>\@reqcolroom
206          \@flsetnum \@colnum
207          \ifnum \@colnum>\z@
208            \@bitor\@currtype\@deferlist
```

We need to defer the float also if its width doesn't fit.

```
209            \@testwrongwidth\@currbox
210            \if@test
211            \else
212              \@bitor\@currtype\@botlist
213              \if@test
214                \@addtobot
215              \else
216                \ifodd \count\@currbox
217                  \advance \@reqcolroom \intextsep
218                  \ifdim \@colroom>\@reqcolroom
219                    \global \advance \@colnum \m@ne
220                    \global \advance \@textfloatsheight \ht\@currbox
221                    \global \advance \@textfloatsheight 2\intextsep
222                    \@cons \@midlist \@currbox
223                    \if@nobreak
224                      \nobreak
225                      \@nobreakfalse
226                      \everypar{}%
227                    \else
228                      \addpenalty \interlinepenalty
229                    \fi
230                    \vskip \intextsep
231                    \box\@currbox
232                    \penalty\interlinepenalty
233                    \vskip\intextsep
234                    \ifnum\outputpenalty <-\@Mii \vskip -\parskip\fi
235                    \outputpenalty \z@
236                    \@inserttrue
237                  \fi
238                \fi
239                \if@insert
240                \else
241                  \@addtotoporbot
242                \fi
243              \fi
244            \fi
245          \fi
246        \fi
247      \fi
248    \fi
249    \if@insert
250    \else
251      \@resethfps
252      \@cons\@deferlist\@currbox
253    \fi
254 }
```

```
255 \def\@xtryfc #1{%
256   \@next\reserved@a\@trylist{}{}%
257   \@currtype \count #1%
258   \divide\@currtype\@xxxii
259   \multiply\@currtype\@xxxii
260   \@bitor \@currtype \@failedlist
261   \@testfp #1%

262   \@testwrongwidth #1%

263   \ifdim \ht #1>\@colht
264      \@testtrue
265   \fi
266   \if@test
267     \@cons\@failedlist #1%
268   \else
269     \@ytryfc #1%
270   \fi}
271 \def\@ztryfc #1{%
272   \@tempcnta\count #1%
273   \divide\@tempcnta\@xxxii
274   \multiply\@tempcnta\@xxxii
275   \@bitor \@tempcnta {\@failedlist \@flfail}%
276   \@testfp #1%
```

not in fixfloats?

```
277   \@testwrongwidth #1%

278   \@tempdimb\@tempdima
279   \advance\@tempdimb\ht #1%
280   \advance\@tempdimb\@fpsep
281   \ifdim \@tempdimb >\@colht
282      \@testtrue
283   \fi
284   \if@test
285     \@cons\@flfail #1%
286   \else
287     \@cons\@flsucceed #1%
288     \@tempdima\@tempdimb
289   \fi}
```

5.2 \@ discards spaces when moving (pr3039)

\@ Ensure that \@m can't eat spaces. Alternative would be to make \@ robust but that takes more space.

```
290 \def\@{\spacefactor\@m{}}
```

5.3 \setlength produces error if used with registers like \dimen0 (pr/3066)

\setlength Add space after register (#1) but only if this is still the original definition. When, for example, calc was already loaded this wouldn't be a good idea any more.

```
291 \def\@tempa#1#2{#1#2\relax}
292 \ifx\setlength\@tempa
293   \def\setlength#1#2{#1 #2\relax}
294 \fi
```

5.4 \addpenalty ruins flush-bottom (pr/3073)

\addpenalty Fix provided by Donald (though the original fix was not good enough). In 2005 Plamen Tanovski discovered that this fix wasn't good enough either as the \vskip

kept getting bigger if several `\addpenalty` commands followed each other. Donald kindly send a new fix.

```
295 \def\addpenalty#1{%
296   \ifvmode
297     \if@minipage
298     \else
299       \if@nobreak
300       \else
301         \ifdim\lastskip=\z@
302           \penalty#1\relax
303         \else
304           \@tempskipb\lastskip
```

We have to make sure the final `\vskip` seen by TeX is the correct one, namely `\@tempskipb`. However we may have to adjust for `\prevdepth` when placing the penalty but that should not affect the skip we pass on to TeX.

```
305           \begingroup
306             \advance \@tempskipb
307               \ifdim\prevdepth>\maxdepth\maxdepth\else
```

If `\prevdepth` is -1000pt due to `\nointerlineskip` we better not add it!

```
308                 \ifdim \prevdepth = -\@m\p@ \z@ \else \prevdepth \fi
309               \fi
310             \vskip -\@tempskipb
311             \penalty#1%
312             \vskip\@tempskipb
313           \endgroup
314           \vskip -\@tempskipb
315           \vskip \@tempskipb
316         \fi
317       \fi
318     \fi
319   \else
320     \@noitemerr
321   \fi}
```

5.5 `\fnsymbol` should use text symbols if possible (pr/3400)

`\@fnsymbol` This macro is another example of an ever recurring problem in TeX: Determining if something is text-mode or math-mode. It is imperative for the decision between text and math to be delayed until the actual typesetting is done as the code in question may go through an `\edef` or `\write` where an `\ifmmode` test would be executed prematurely. Hence in the implementation below, `\@fnsymbol` is not robust in itself but the parts doing the actual typesetting are.

In the case of `\@fnsymbol` we make use of the robust command `\TextOrMath` which takes two arguments and typesets the first if in text-mode and the second if in math-mode. Note that in order for this command to make the correct decision, it must insert a `\relax` token if run under regular TeX, which ruins any kerning between the preceding characters and whatever awaits typesetting. If you use eTeX as engine for LaTeX (as recommended) this unfortunate side effect is not present.

```
322 \def\@fnsymbol#1{%
323     \ifcase#1\or \TextOrMath\textasteriskcentered *\or
324     \TextOrMath \textdagger \dagger\or
325     \TextOrMath \textdaggerdbl \ddagger \or
326     \TextOrMath \textsection  \mathsection\or
327     \TextOrMath \textparagraph \mathparagraph\or
328     \TextOrMath \textbardbl \|\or
329     \TextOrMath {\textasteriskcentered\textasteriskcentered}{**}\or
330     \TextOrMath {\textdagger\textdagger}{\dagger\dagger}\or
331     \TextOrMath {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}\else
```

```
332    \@ctrerr \fi
333 }
```

`\TextOrMath` When using regular TEX, we make this command robust so that it always selects the correct branch in an `\ifmmode` switch with the usual disadvantage of ruining kerning. For the application we use it for here that shouldn't matter. The alternative would be to mimic `\IeC` from inputenc but then it wil have the disadvantage of choosing the wrong branch if appearing at the beginning of an alignment cell. However, users of eTEX will be pleasantly surprised to get the best of both worlds and no bad side effects.

First some code for checking if we are running eTEX but making sure not to permanently turn `\eTeXversion` into `\relax`.

```
334 \begingroup\expandafter\expandafter\expandafter\endgroup
335 \expandafter\ifx\csname eTeXversion\endcsname\relax
```

In case of ordinary TEX we define `\TextOrMath` as a robust command but make sure it always grabs its arguments. If we didn't do this it might very well gobble spaces in the input stream.

```
336 \DeclareRobustCommand\TextOrMath{%
337   \ifmmode  \expandafter\@secondoftwo
338   \else     \expandafter\@firstoftwo  \fi}
339 \protected@edef\TextOrMath#1#2{\TextOrMath{#1}{#2}}
340 \else
```

For eTEX the situation is similar. The robust macro is a hidden one so that we again avoid problems of gobbling spaces in the input.

```
341 \protected\expandafter\def\csname TextOrMath\space\endcsname{%
342   \ifmmode  \expandafter\@secondoftwo
343   \else     \expandafter\@firstoftwo  \fi}
344 \edef\TextOrMath#1#2{%
345   \expandafter\noexpand\csname TextOrMath\space\endcsname
346   {#1}{#2}}
347 \fi
```

5.6   No hyphenation in first word after float environment(pr/3498)

`\@esphack` Fix suggested by Donald Arseneau.
`\@Esphack`
```
348 \def\@esphack{%
349   \relax
350   \ifhmode
351     \spacefactor\@savsf
352     \ifdim\@savsk>\z@
353       \nobreak \hskip\z@skip  % <------
354       \ignorespaces
355     \fi
356   \fi}
```

```
357 \def\@Esphack{%
358   \relax
359   \ifhmode
360     \spacefactor\@savsf
361     \ifdim\@savsk>\z@
362       \nobreak \hskip\z@skip  % <------
363       \@ignoretrue
364       \ignorespaces
365     \fi
366   \fi}
```

5.7   Allowing \emph to produce small caps, etc

`\em`

`\eminnershape`

```
367 \DeclareRobustCommand\em
368         {\@nomath\em \ifdim \fontdimen\@ne\font >\z@
369                     \eminnershape \else \itshape \fi}
370 \def\eminnershape{\upshape}
```

5.8 \textsubscript not defined in latex.ltx (pr/3492)

\textsubscript This macro is almost identical to \textsuperscript from the kernel.

```
371 \DeclareRobustCommand*\textsubscript[1]{%
372   \@textsubscript{\selectfont#1}}
373 \def\@textsubscript#1{%
374   {\m@th\ensuremath{_{\mbox{\fontsize\sf@size\z@#1}}}}}
```

5.9 \DeclareMathSizes only take pts. (pr/3693)

\@DeclareMathSizes This fix given by Michael J. Downes on comp.text.tex on 2002/10/17 allows the user to have settings such as \DeclareMathSizes{9.5dd}{9.5dd}{7.4dd}{6.6dd}.

```
375 \def\@DeclareMathSizes #1#2#3#4#5{%
376   \@defaultunits\dimen@ #2pt\relax\@nnil
377   \if $#3$%
378     \expandafter\let\csname S@\strip@pt\dimen@\endcsname\math@fontsfalse
379   \else
380     \@defaultunits\dimen@ii #3pt\relax\@nnil
381     \@defaultunits\@tempdima #4pt\relax\@nnil
382     \@defaultunits\@tempdimb #5pt\relax\@nnil
383     \toks@{#1}%
384     \expandafter\xdef\csname S@\strip@pt\dimen@\endcsname{%
385       \gdef\noexpand\tf@size{\strip@pt\dimen@ii}%
386       \gdef\noexpand\sf@size{\strip@pt\@tempdima}%
387       \gdef\noexpand\ssf@size{\strip@pt\@tempdimb}%
388       \the\toks@
389     }%
390   \fi
391 }
```

5.10 Fewer fragile macros

\MakeRobust The macro firstly checks if the controls sequence in question exists at all.

```
392 \providecommand*\MakeRobust[1]{%
393   \@ifundefined{\expandafter\@gobble\string#1}{%
394     \@latex@error{The control sequence '\string#1' is undefined!%
395       \MessageBreak There is nothing here to make robust}%
396     \@eha
397   }%
```

Then we check if the macro is already robust. We do this by testing if the internal name for a robust macro is defined, namely \foo␣. If it is already defined do nothing, otherwise set \foo␣ equal to \foo and redefine \foo so that it acts like a macro defined with \DeclareRobustCommand.

```
398   {%
399     \@ifundefined{\expandafter\@gobble\string#1\space}%
400     {%
401       \expandafter\let\csname
402       \expandafter\@gobble\string#1\space\endcsname=#1%
403       \edef\reserved@a{\string#1}%
404       \def\reserved@b{#1}%
405       \edef\reserved@b{\expandafter\strip@prefix\meaning\reserved@b}%
406       \edef#1{%
407         \ifx\reserved@a\reserved@b
408           \noexpand\x@protect\noexpand#1%
409         \fi
```

```
410        \noexpand\protect\expandafter\noexpand
411        \csname\expandafter\@gobble\string#1\space\endcsname}%
412     }%
413     {\@latex@info{The control sequence '\string#1' is already robust}}%
414     }%
415 }
```

Here we make some kernel macros robust.

```
416 \MakeRobust\(
417 \MakeRobust\)
418 \MakeRobust\[
419 \MakeRobust\]
420 \MakeRobust\makebox
421 \MakeRobust\savebox
422 \MakeRobust\framebox
423 \MakeRobust\parbox
424 \MakeRobust\rule
425 \MakeRobust\raisebox
```

426 ⟨/fixltx2e⟩

5.11 Using EC fonts (T1 encoding) makes my documents look bl**dy horrible

5.11.1 Preliminaries

The LaTeX kernel does not declare the font encoding TS1. However, we are going to set up font definitions for this encoding, so we have to declare it now.

427 ⟨∗fix − cm⟩
428 \input{ts1enc.def}

In case the package is loaded in the preamble, any of the CM fonts may have been used already and cannot be redefined. Yet we try to intercept at least the problem that is most likely to occur, i.e., a hidden \normalfont. Most of the standard definitions are ok, but those for T1 encoding and 10.95 pt need to be removed:

```
429 \expandafter \let \csname T1/cmr/m/n/10.95\endcsname \relax
430 \expandafter \let \csname T1/cmss/m/n/10.95\endcsname \relax
431 \expandafter \let \csname T1/cmtt/m/n/10.95\endcsname \relax
432 \expandafter \let \csname T1/cmvtt/m/n/10.95\endcsname \relax
```

fix-cm may still fail, if the EC fonts are preloaded in the LaTeX format file. This situation is, however, very unlikely and could occur only with a customized format.

The remainder of the package is enclosed in a group, where the catcodes are guaranteed to be appropriate for the processing of font definitions.

433 \begingroup
434 \nfss@catcodes

5.11.2 T1 encoding

CM Roman

```
435 \DeclareFontFamily{T1}{cmr}{}
436 \DeclareFontShape{T1}{cmr}{m}{n}{
437        <-6>    ecrm0500
438        <6-7>   ecrm0600
439        <7-8>   ecrm0700
440        <8-9>   ecrm0800
441        <9-10>  ecrm0900
442        <10-12> ecrm1000
443        <12-17> ecrm1200
444        <17->   ecrm1728
```

```
445        }{}
446 \DeclareFontShape{T1}{cmr}{m}{sl}{
447        <-6>    ecsl0500
448        <6-7>   ecsl0600
449        <7-8>   ecsl0700
450        <8-9>   ecsl0800
451        <9-10>  ecsl0900
452        <10-12> ecsl1000
453        <12-17> ecsl1200
454        <17->   ecsl1728
455        }{}
456 \DeclareFontShape{T1}{cmr}{m}{it}{
457        <-8>    ecti0700
458        <8-9>   ecti0800
459        <9-10>  ecti0900
460        <10-12> ecti1000
461        <12-17> ecti1200
462        <17->   ecti1728
463        }{}
464 \DeclareFontShape{T1}{cmr}{m}{sc}{
465        <-6>    eccc0500
466        <6-7>   eccc0600
467        <7-8>   eccc0700
468        <8-9>   eccc0800
469        <9-10>  eccc0900
470        <10-12> eccc1000
471        <12-17> eccc1200
472        <17->   eccc1728
473                }{}
474 \DeclareFontShape{T1}{cmr}{m}{ui}{
475        <-8>    ecui0700
476        <8-9>   ecui0800
477        <9-10>  ecui0900
478        <10-12> ecui1000
479        <12-17> ecui1200
480        <17->   ecui1728
481        }{}
482 \DeclareFontShape{T1}{cmr}{b}{n}{
483        <-6>    ecrb0500
484        <6-7>   ecrb0600
485        <7-8>   ecrb0700
486        <8-9>   ecrb0800
487        <9-10>  ecrb0900
488        <10-12> ecrb1000
489        <12-17> ecrb1200
490        <17->   ecrb1728
491        }{}
492 \DeclareFontShape{T1}{cmr}{bx}{n}{
493        <-6>    ecbx0500
494        <6-7>   ecbx0600
495        <7-8>   ecbx0700
496        <8-9>   ecbx0800
497        <9-10>  ecbx0900
498        <10-12> ecbx1000
499        <12->   ecbx1200
500        }{}
501 \DeclareFontShape{T1}{cmr}{bx}{sl}{
502        <-6>    ecbl0500
503        <6-7>   ecbl0600
504        <7-8>   ecbl0700
505        <8-9>   ecbl0800
506        <9-10>  ecbl0900
```

```
507        <10-12> ecbl1000
508        <12->   ecbl1200
509      }{}
510 \DeclareFontShape{T1}{cmr}{bx}{it}{
511        <-8>    ecbi0700
512        <8-9>   ecbi0800
513        <9-10>  ecbi0900
514        <10-12> ecbi1000
515        <12->   ecbi1200
516      }{}
517 \DeclareFontShape{T1}{cmr}{bx}{sc}{
518        <-6>    ecxc0500
519        <6-7>   ecxc0600
520        <7-8>   ecxc0700
521        <8-9>   ecxc0800
522        <9-10>  ecxc0900
523        <10-12> ecxc1000
524        <12->   ecxc1200
525      }{}
526 %
```

CM Sans

```
527 \DeclareFontFamily{T1}{cmss}{}
528 \DeclareFontShape{T1}{cmss}{m}{n}{
529        <-9>    ecss0800
530        <9-10>  ecss0900
531        <10-12> ecss1000
532        <12-17> ecss1200
533        <17->   ecss1728
534      }{}
535 \DeclareFontShape{T1}{cmss}{m}{sl}{
536        <-9>    ecsi0800
537        <9-10>  ecsi0900
538        <10-12> ecsi1000
539        <12-17> ecsi1200
540        <17->   ecsi1728
541      }{}
542 \DeclareFontShape{T1}{cmss}{m}{it}
543        {<->ssub*cmss/m/sl}{}
544 \DeclareFontShape{T1}{cmss}{m}{sc}
545        {<->sub*cmr/m/sc}{}
546 \DeclareFontShape{T1}{cmss}{sbc}{n}{
547        <->     ecssdc10
548      }{}
549 \DeclareFontShape{T1}{cmss}{bx}{n}{
550        <-10>   ecsx0900
551        <10->   ecsx1000
552      }{}
553 \DeclareFontShape{T1}{cmss}{bx}{sl}{
554        <-10>   ecso0900
555        <10->   ecso1000
556      }{}
557 \DeclareFontShape{T1}{cmss}{bx}{it}
558        {<->ssub*cmss/bx/sl}{}
```

The following substitutions are not provided in the default `.fd` files. I have included them, so that you can easily use the EC fonts with the default bold series being `b` rather than `bx`.

```
559 \DeclareFontShape{T1}{cmss}{b}{n}
560        {<->ssub*cmss/bx/n}{}
561 \DeclareFontShape{T1}{cmss}{b}{sl}
562        {<->ssub*cmss/bx/sl}{}
```

```
563 \DeclareFontShape{T1}{cmss}{b}{it}
564       {<->ssub*cmss/bx/sl}{}
```

CM Typewriter

```
565 \DeclareFontFamily{T1}{cmtt}{\hyphenchar \font\m@ne}
566 \DeclareFontShape{T1}{cmtt}{m}{n}{
567       <-9>    ectt0800
568       <9-10>  ectt0900
569       <10-12> ectt1000
570       <12-17> ectt1200
571       <17->   ectt1728
572     }{}
573 \DeclareFontShape{T1}{cmtt}{m}{it}{
574       <-9>    ecit0800
575       <9-10>  ecit0900
576       <10-12> ecit1000
577       <12-17> ecit1200
578       <17->   ecit1728
579     }{}
580 \DeclareFontShape{T1}{cmtt}{m}{sl}{
581       <-9>    ecst0800
582       <9-10>  ecst0900
583       <10-12> ecst1000
584       <12-17> ecst1200
585       <17->   ecst1728
586     }{}
587 \DeclareFontShape{T1}{cmtt}{m}{sc}{
588       <-9>    ectc0800
589       <9-10>  ectc0900
590       <10-12> ectc1000
591       <12-17> ectc1200
592       <17->   ectc1728
593     }{}
594 \DeclareFontShape{T1}{cmtt}{bx}{n}
595       {<->sub * cmtt/m/n}{}
596 \DeclareFontShape{T1}{cmtt}{bx}{it}
597       {<->sub * cmtt/m/it}{}
598 \DeclareFontShape{T1}{cmtt}{bx}{sl}
599       {<->sub * cmtt/m/sl}{}
```

Substitutions not provided in the default `.fd` files:

```
600 \DeclareFontShape{T1}{cmtt}{b}{n}
601       {<->sub * cmtt/m/n}{}
602 \DeclareFontShape{T1}{cmtt}{b}{it}
603       {<->sub * cmtt/m/it}{}
604 \DeclareFontShape{T1}{cmtt}{b}{sl}
605       {<->sub * cmtt/m/sl}{}
```

CM Typewiter (var.)

```
606 \DeclareFontFamily{T1}{cmvtt}{}
607 \DeclareFontShape{T1}{cmvtt}{m}{n}{
608       <-9>    ecvt0800
609       <9-10>  ecvt0900
610       <10-12> ecvt1000
611       <12-17> ecvt1200
612       <17->   ecvt1728
613     }{}
614 \DeclareFontShape{T1}{cmvtt}{m}{it}{
615       <-9>    ecvi0800
616       <9-10>  ecvi0900
617       <10-12> ecvi1000
618       <12-17> ecvi1200
```

```
619        <17->    ecvi1728
620      }{}
```

5.11.3 TS1 encoding

CM Roman

```
621 \DeclareFontFamily{TS1}{cmr}{\hyphenchar\font\m@ne}
622 \DeclareFontShape{TS1}{cmr}{m}{n}{
623        <-6>     tcrm0500
624        <6-7>    tcrm0600
625        <7-8>    tcrm0700
626        <8-9>    tcrm0800
627        <9-10>   tcrm0900
628        <10-12>  tcrm1000
629        <12-17>  tcrm1200
630        <17->    tcrm1728
631      }{}
632 \DeclareFontShape{TS1}{cmr}{m}{sl}{
633        <-6>     tcsl0500
634        <6-7>    tcsl0600
635        <7-8>    tcsl0700
636        <8-9>    tcsl0800
637        <9-10>   tcsl0900
638        <10-12>  tcsl1000
639        <12-17>  tcsl1200
640        <17->    tcsl1728
641      }{}
642 \DeclareFontShape{TS1}{cmr}{m}{it}{
643        <-8>     tcti0700
644        <8-9>    tcti0800
645        <9-10>   tcti0900
646        <10-12>  tcti1000
647        <12-17>  tcti1200
648        <17->    tcti1728
649      }{}
650 \DeclareFontShape{TS1}{cmr}{m}{ui}{
651        <-8>     tcui0700
652        <8-9>    tcui0800
653        <9-10>   tcui0900
654        <10-12>  tcui1000
655        <12-17>  tcui1200
656        <17->    tcui1728
657      }{}
658 \DeclareFontShape{TS1}{cmr}{b}{n}{
659        <-6>     tcrb0500
660        <6-7>    tcrb0600
661        <7-8>    tcrb0700
662        <8-9>    tcrb0800
663        <9-10>   tcrb0900
664        <10-12>  tcrb1000
665        <12-17>  tcrb1200
666        <17->    tcrb1728
667      }{}
668 \DeclareFontShape{TS1}{cmr}{bx}{n}{
669        <-6>     tcbx0500
670        <6-7>    tcbx0600
671        <7-8>    tcbx0700
672        <8-9>    tcbx0800
673        <9-10>   tcbx0900
674        <10-12>  tcbx1000
675        <12->    tcbx1200
676      }{}
```

```
677 \DeclareFontShape{TS1}{cmr}{bx}{sl}{
678         <-6>     tcbl0500
679         <6-7>    tcbl0600
680         <7-8>    tcbl0700
681         <8-9>    tcbl0800
682         <9-10>   tcbl0900
683         <10-12>  tcbl1000
684         <12->    tcbl1200
685       }{}
686 \DeclareFontShape{TS1}{cmr}{bx}{it}{
687         <-8>     tcbi0700
688         <8-9>    tcbi0800
689         <9-10>   tcbi0900
690         <10-12>  tcbi1000
691         <12->    tcbi1200
692       }{}
```

CM Sans

```
693 \DeclareFontFamily{TS1}{cmss}{\hyphenchar\font\m@ne}
694 \DeclareFontShape{TS1}{cmss}{m}{n}{
695         <-9>     tcss0800
696         <9-10>   tcss0900
697         <10-12>  tcss1000
698         <12-17>  tcss1200
699         <17->    tcss1728
700       }{}
701 \DeclareFontShape{TS1}{cmss}{m}{it}
702       {<->ssub*cmss/m/sl}{}
703 \DeclareFontShape{TS1}{cmss}{m}{sl}{
704         <-9>     tcsi0800
705         <9-10>   tcsi0900
706         <10-12>  tcsi1000
707         <12-17>  tcsi1200
708         <17->    tcsi1728
709       }{}
710 \DeclareFontShape{TS1}{cmss}{sbc}{n}{
711         <->      tcssdc10
712       }{}
713 \DeclareFontShape{TS1}{cmss}{bx}{n}{
714         <-10>    tcsx0900
715         <10->    tcsx1000
716       }{}
717 \DeclareFontShape{TS1}{cmss}{bx}{sl}{
718         <-10>    tcso0900
719         <10->    tcso1000
720       }{}
721 \DeclareFontShape{TS1}{cmss}{bx}{it}
722       {<->ssub*cmss/bx/sl}{}
```

Substitutions not provided in the default `.fd` files:

```
723 \DeclareFontShape{TS1}{cmss}{b}{n}
724       {<->ssub*cmss/bx/n}{}
725 \DeclareFontShape{TS1}{cmss}{b}{sl}
726       {<->ssub*cmss/bx/sl}{}
727 \DeclareFontShape{TS1}{cmss}{b}{it}
728       {<->ssub*cmss/bx/sl}{}
```

CM Typewriter

```
729 \DeclareFontFamily{TS1}{cmtt}{\hyphenchar \font\m@ne}
730 \DeclareFontShape{TS1}{cmtt}{m}{n}{
731         <-9>     tctt0800
732         <9-10>   tctt0900
```

```
733          <10-12> tctt1000
734          <12-17> tctt1200
735          <17->   tctt1728
736      }{}
737 \DeclareFontShape{TS1}{cmtt}{m}{it}{
738          <-9>    tcit0800
739          <9-10>  tcit0900
740          <10-12> tcit1000
741          <12-17> tcit1200
742          <17->   tcit1728
743      }{}
744 \DeclareFontShape{TS1}{cmtt}{m}{sl}{
745          <-9>    tcst0800
746          <9-10>  tcst0900
747          <10-12> tcst1000
748          <12-17> tcst1200
749          <17->   tcst1728
750      }{}
751 \DeclareFontShape{TS1}{cmtt}{bx}{n}
752          {<->sub * cmtt/m/n}{}
753 \DeclareFontShape{TS1}{cmtt}{bx}{it}
754          {<->sub * cmtt/m/it}{}
755 \DeclareFontShape{TS1}{cmtt}{bx}{sl}
756          {<->sub * cmtt/m/sl}{}
```

Substitutions not provided in the default `.fd` files:

```
757 \DeclareFontShape{TS1}{cmtt}{b}{n}
758          {<->sub * cmtt/m/n}{}
759 \DeclareFontShape{TS1}{cmtt}{b}{it}
760          {<->sub * cmtt/m/it}{}
761 \DeclareFontShape{TS1}{cmtt}{b}{sl}
762          {<->sub * cmtt/m/sl}{}
```

CM Typewriter (var.)

```
763 \DeclareFontFamily{TS1}{cmvtt}{}
764 \DeclareFontShape{TS1}{cmvtt}{m}{n}{
765          <-9>    tcvt0800
766          <9-10>  tcvt0900
767          <10-12> tcvt1000
768          <12-17> tcvt1200
769          <17->   tcvi1728
770      }{}
771 \DeclareFontShape{TS1}{cmvtt}{m}{it}{
772          <-9>    tcvi0800
773          <9-10>  tcvi0900
774          <10-12> tcvi1000
775          <12-17> tcvi1200
776          <17->   tcvi1728
777      }{}
```

5.11.4   OT1 encoding

CM Roman

```
778 \DeclareFontFamily{OT1}{cmr}{\hyphenchar\font45 }
779 \DeclareFontShape{OT1}{cmr}{m}{n}{
780          <-6>    cmr5
781          <6-7>   cmr6
782          <7-8>   cmr7
783          <8-9>   cmr8
784          <9-10>  cmr9
785          <10-12> cmr10
786          <12-17> cmr12
```

```
787          <17->    cmr17
788      }{}
789 \DeclareFontShape{OT1}{cmr}{m}{sl}{
790          <-9>     cmsl8
791          <9-10>   cmsl9
792          <10-12>  cmsl10
793          <12->    cmsl12
794      }{}
795 \DeclareFontShape{OT1}{cmr}{m}{it}{
796          <-8>     cmti7
797          <8-9>    cmti8
798          <9-10>   cmti9
799          <10-12>  cmti10
800          <12->    cmti12
801      }{}
802 \DeclareFontShape{OT1}{cmr}{m}{sc}{
803          <->      cmcsc10
804      }{}
805 \DeclareFontShape{OT1}{cmr}{m}{ui}{
806          <->      cmu10
807      }{}
808 \DeclareFontShape{OT1}{cmr}{b}{n}{
809          <->      cmb10
810      }{}
811 \DeclareFontShape{OT1}{cmr}{bx}{n}{
812          <-6>     cmbx5
813          <6-7>    cmbx6
814          <7-8>    cmbx7
815          <8-9>    cmbx8
816          <9-10>   cmbx9
817          <10-12>  cmbx10
818          <12->    cmbx12
819      }{}
820 \DeclareFontShape{OT1}{cmr}{bx}{sl}{
821          <->      cmbxsl10
822      }{}
823 \DeclareFontShape{OT1}{cmr}{bx}{it}{
824          <->      cmbxti10
825      }{}
826 \DeclareFontShape{OT1}{cmr}{bx}{ui}
827      {<->sub*cmr/m/ui}{}
```

CM Sans

```
828 \DeclareFontFamily{OT1}{cmss}{\hyphenchar\font45 }
829 \DeclareFontShape{OT1}{cmss}{m}{n}{
830          <-9>     cmss8
831          <9-10>   cmss9
832          <10-12>  cmss10
833          <12-17>  cmss12
834          <17->    cmss17
835      }{}
836 \DeclareFontShape{OT1}{cmss}{m}{it}
837      {<->sub*cmss/m/sl}{}
838 \DeclareFontShape{OT1}{cmss}{m}{sl}{
839          <-9>     cmssi8
840          <9-10>   cmssi9
841          <10-12>  cmssi10
842          <12-17>  cmssi12
843          <17->    cmssi17
844      }{}
845 \DeclareFontShape{OT1}{cmss}{m}{sc}
846      {<->sub*cmr/m/sc}{}
```

```
847 \DeclareFontShape{OT1}{cmss}{m}{ui}
848     {<->sub*cmr/m/ui}{}
849 \DeclareFontShape{OT1}{cmss}{sbc}{n}{
850        <->      cmssdc10
851     }{}
852 \DeclareFontShape{OT1}{cmss}{bx}{n}{
853        <->      cmssbx10
854     }{}
855 \DeclareFontShape{OT1}{cmss}{bx}{ui}
856     {<->sub*cmr/bx/ui}{}
```

CM Typewriter

```
857 \DeclareFontFamily{OT1}{cmtt}{\hyphenchar \font\m@ne}
858 \DeclareFontShape{OT1}{cmtt}{m}{n}{
859        <-9>    cmtt8
860        <9-10>  cmtt9
861        <10-12> cmtt10
862        <12->   cmtt12
863     }{}
864 \DeclareFontShape{OT1}{cmtt}{m}{it}{
865        <->      cmitt10
866     }{}
867 \DeclareFontShape{OT1}{cmtt}{m}{sl}{
868        <->      cmsltt10
869     }{}
870 \DeclareFontShape{OT1}{cmtt}{m}{sc}{
871        <->      cmtcsc10
872     }{}
873 \DeclareFontShape{OT1}{cmtt}{m}{ui}
874     {<->ssub*cmtt/m/it}{}
875 \DeclareFontShape{OT1}{cmtt}{bx}{n}
876     {<->ssub*cmtt/m/n}{}
877 \DeclareFontShape{OT1}{cmtt}{bx}{it}
878     {<->ssub*cmtt/m/it}{}
879 \DeclareFontShape{OT1}{cmtt}{bx}{ui}
880     {<->ssub*cmtt/m/it}{}
```

CM Typewriter (var.)

```
881 \DeclareFontFamily{OT1}{cmvtt}{\hyphenchar\font45 }
882 \DeclareFontShape{OT1}{cmvtt}{m}{n}{
883        <->      cmvtt10
884     }{}
885 \DeclareFontShape{OT1}{cmvtt}{m}{it}{
886        <->      cmvtti10
887     }{}
```

5.11.5   OML and OMS encoded math fonts

```
888 \DeclareFontFamily{OML}{cmm}{\skewchar\font127 }
889 \DeclareFontShape{OML}{cmm}{m}{it}{
890        <-6>    cmmi5
891        <6-7>   cmmi6
892        <7-8>   cmmi7
893        <8-9>   cmmi8
894        <9-10>  cmmi9
895        <10-12> cmmi10
896        <12->   cmmi12
897     }{}
898 \DeclareFontShape{OML}{cmm}{b}{it}{<-6>cmmib5<6-8>cmmib7<8->cmmib10}{}
899 \DeclareFontShape{OML}{cmm}{bx}{it}
900     {<->ssub*cmm/b/it}{}
```

```
901 \DeclareFontFamily{OMS}{cmsy}{\skewchar\font48 }
902 \DeclareFontShape{OMS}{cmsy}{m}{n}{
903          <-6>    cmsy5
904          <6-7>   cmsy6
905          <7-8>   cmsy7
906          <8-9>   cmsy8
907          <9-10>  cmsy9
908          <10->   cmsy10
909       }{}
910 \DeclareFontShape{OMS}{cmsy}{b}{n}{<-6>cmbsy5<6-8>cmbsy7<8->cmbsy10}{}
```

5.11.6   LATEX symbols

```
911 \DeclareFontFamily{U}{lasy}{}
912 \DeclareFontShape{U}{lasy}{m}{n}{
913          <-6>    lasy5
914          <6-7>   lasy6
915          <7-8>   lasy7
916          <8-9>   lasy8
917          <9-10>  lasy9
918          <10->   lasy10
919       }{}
920 \DeclareFontShape{U}{lasy}{b}{n}{
921          <-10>   ssub * lasy/m/n
922          <10->   lasyb10
923       }{}
924 \endgroup
925 ⟨/fix − cm⟩
```