

# A L<sup>A</sup>T<sub>E</sub>X Package for the frontmatter and title page \*†

Arthur Ogawa ‡

January 18, 2019

This file embodies the `ltxfront` package, the implementation and its user documentation.

The distribution point for this work is <https://journals.aps.org/revtex>, which contains fully unpacked, prebuilt runtime files and documentation.

The `ltxfront` package was commissioned by the American Physical Society and is distributed under the terms of the L<sup>A</sup>T<sub>E</sub>X Project Public License 1.3c, the same license under which all the portions of L<sup>A</sup>T<sub>E</sub>X itself is distributed. Please see <http://ctan.tug.org/macros/latex/base/lppl.txt> for details.

To use this document class, you must have a working T<sub>E</sub>X installation equipped with L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> and possibly `pdftex` and Adobe Acrobat Reader or equivalent.

To install, retrieve the distribution, unpack it into a directory on the target computer, and move the file `ltxfront.sty` into a location in your filesystem where it will be found by L<sup>A</sup>T<sub>E</sub>X.

To use, read the user documentation `ltxfront.pdf`.

## Contents

<b>1</b>	<b>Processing Instructions</b>	<b>3</b>
1.1	Build Instructions . . . . .	3
1.2	Change Log . . . . .	4
1.3	Bill of Materials . . . . .	4
1.3.1	Primary Source . . . . .	4
1.3.2	Generated by <code>latex ltxfront.dtx</code> . . . . .	4
1.3.3	Generated by <code>tex ltxfront.dtx</code> . . . . .	4
1.3.4	Documentation . . . . .	4
1.3.5	Auxiliary . . . . .	4
<b>2</b>	<b>Code common to all modules</b>	<b>4</b>

---

\*This file has version number 4.2c, last revised 2019/01/18/14:29:48.

†Version 4.2c © 2019 American Physical Society

‡[mailto:arthur\\_ogawa@sbcglobal.net](mailto:arthur_ogawa@sbcglobal.net)

<b>3</b>	<b>The doc module doc</b>	<b>5</b>
3.1	The Preamble . . . . .	5
3.1.1	Docstrip and info directives . . . . .	6
3.2	The “Read Me” File . . . . .	6
3.3	A Gnuly Created Sample Document . . . . .	9
3.4	The Document Body . . . . .	10
<b>4</b>	<b>Using this package</b>	<b>10</b>
4.1	Invoking the package . . . . .	10
<b>5</b>	<b>Frontmatter commands</b>	<b>11</b>
<b>6</b>	<b>The Front Matter</b>	<b>11</b>
6.1	Setting the title . . . . .	11
6.2	Specifying a date . . . . .	11
6.3	Specifying authors and affiliations . . . . .	11
6.3.1	Collaborations . . . . .	13
6.3.2	Footnotes for authors, collaborations, affiliations or title . . . . .	13
6.3.3	Specifying first names and surnames . . . . .	15
6.4	The abstract . . . . .	15
6.5	PACS codes . . . . .	15
6.6	Keywords . . . . .	15
6.7	Institutional report numbers . . . . .	16
6.8	maketitle . . . . .	16
<b>7</b>	<b>Compatibility with L<sup>A</sup>T<sub>E</sub>X’s Required Packages</b>	<b>16</b>
<b>8</b>	<b>Implementation of package</b>	<b>16</b>
8.1	Beginning of the package DOCSTRIP module . . . . .	16
8.2	Prerequisites . . . . .	16
8.3	Options . . . . .	17
8.4	Process package options . . . . .	18
<b>9</b>	<b>Kernel of Front Matter</b>	<b>18</b>
9.1	Initialization . . . . .	19
9.2	The title command . . . . .	19
9.3	The author, affiliation, and related commands . . . . .	19
9.3.1	Commands for author or collaboration . . . . .	20
9.3.2	Commands for affiliation . . . . .	23
9.3.3	Commands for auxiliary information . . . . .	30
9.3.4	Procedures for author, collaboration, and affiliation . . . . .	31
9.4	The keywords command . . . . .	34
9.5	The \date command and related commands . . . . .	35
9.6	The pacs command . . . . .	35
9.7	The \preprint command . . . . .	35
9.8	The abstract environment . . . . .	36

9.9	Formatting the title block . . . . .	39
9.9.1	Authors and affiliations in superscriptaddress style . . . . .	42
9.9.2	Authors and affiliations in groupedaddress style . . . . .	47
<b>10</b>	<b>Package defaults</b>	<b>52</b>
10.1	Printing out the “list-of” elements . . . . .	55
10.2	Syntax switch . . . . .	56
10.3	End of the <code>ltxfront</code> kernel . . . . .	57
10.4	Remainder of the <code>ltxfront</code> package . . . . .	57

# 1 Processing Instructions

The package file `ltxfront.sty` is generated from this file, `ltxfront.dtx`, using the `DOCSTRIP` facility of L<sup>A</sup>T<sub>E</sub>X via `tex ltxfront.dtx`. The typeset documentation that you are now reading is generated from the same file by typesetting it with L<sup>A</sup>T<sub>E</sub>X or `pdftex` via `latex ltxfront.dtx` or `pdflatex ltxfront.dtx`.

## 1.1 Build Instructions

You may bootstrap this suite of files solely from `ltxfront.dtx`. Prepare by installing L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> (and either `tex` or `pdftex`) on your computer, then carry out the following steps:

1. Within an otherwise empty directory, typeset `ltxfront.dtx` with T<sub>E</sub>X or `pdftex`, thereby generating the package file `ltxfront.sty`.
2. Next typeset `ltxfront.dtx` with L<sup>A</sup>T<sub>E</sub>X or `pdflatex`; you will obtain the typeset documentation you are now reading, along with the sample document `ltxfront-test.tex` and the file `README-LTXFRONT`.

Note: you will have to run L<sup>A</sup>T<sub>E</sub>X, then `makeindex -s gind.ist ltxfront.idx`, then `makeindex -s gglo.ist -o ltxfront.gls ltxfront.glo`, then L<sup>A</sup>T<sub>E</sub>X again in order to obtain a valid index and table of contents.

3. Now typeset `ltxfront.dtx` with T<sub>E</sub>X (not L<sup>A</sup>T<sub>E</sub>X), thereby generating the package file `ltxfront.sty`.
4. Install the following files into indicated locations within your TDS-compliant `texmf` tree (you may need root access):
  - `$TEXMF/tex/latex/revtex/ltxfront.sty`
  - `$TEXMF/source/latex/revtex/ltxfront.dtx`
  - `$TEXMF/doc/latex/revtex/ltxfront.pdf`

where `$TEXMF/` stands for `texmf-local/`, or some other `texmf` tree in your installation.

5. Run `mktexlsr` on `$TEXMF/` (you may need root access).

6. Build and installation are now complete; now put a `\usepackage{ltxfront}` in your document preamble!

## 1.2 Change Log

### 1.3 Bill of Materials

Following is a list of the files in this distribution arranged according to provenance.

#### 1.3.1 Primary Source

One single file generates all.

`ltxfront.dtx`

#### 1.3.2 Generated by `latex ltxfront.dtx`

Typesetting the source file under  $\text{\LaTeX}$  generates the readme and the documentation.

`README-LTXFRONT`

#### 1.3.3 Generated by `tex ltxfront.dtx`

Typesetting the installer generates the package files.

`ltxfront.sty`

#### 1.3.4 Documentation

The following are the online documentation:

`ltxfront.pdf`

#### 1.3.5 Auxiliary

The following are auxiliary files generated in the course of running  $\text{\LaTeX}$ :

`ltxfront.aux ltxfront.idx ltxfront.ind ltxfront.log ltxfront.toc`

## 2 Code common to all modules

We want to require only one place in this file where the version number is stated, and we also want to ensure that the version number is embedded into every generated file.

Now we declare that these files can only be used with  $\text{\LaTeX 2}_\epsilon$ . An appropriate message is displayed if a different  $\text{\TeX}$  format is used.

```
1 %<*doc|package>
2 \NeedsTeXFormat{LaTeX2e}[1995/12/01]%
3 %</doc|package>
```

As desired, the following modules all take common version information:

```
4 %<kernel&!package&!doc>\typeout{%
5 %<*package|doc>
6 \ProvidesFile{%
7 %</package|doc>
8 %<*kernel|package|doc>
9 ltxfront%
10 %</kernel|package|doc>
11 %<*doc>
12 .dtx%
13 %</doc>
14 %<package>.sty%
15 %<*package|doc>
16 }%
17 %</package|doc>
```

The following line contains, for once and for all, the version and date information. By various means, this information is reproduced consistently in all generated files and in the typeset documentation.

```
18 %<*doc|package|kernel>
19 %<version>
20 [2019/01/18/14:29:48 4.2c frontmatter package (AO,DPC,MD)]% \fileversion
21 %</doc|package|kernel>
22 %<kernel&!package&!doc>}%
```

### 3 The doc module doc

This module, consisting of the present section, typesets the programmer’s documentation, generating the README-LTXFRONT and sample document as needed.

Because the only uncommented-out lines of code at the beginning of this file constitute the doc module itself, we can simply typeset the .dtx file directly, and there is thus rarely any need to generate the “doc” DOCSTRIP module. Module delimiters are nonetheless required so that this code does not find its way into the other modules.

The `\end{document}` command concludes the typesetting run.

```
23 %<*doc>
```

#### 3.1 The Preamble

The programmers documentation is formatted with the `ltxdoc` class with local customizations, and with the usual code line indexing.

```
24 \documentclass{ltxdoc}
25 \RequirePackage{ltxdocext}%
26 \let\url\undefined
27 \RequirePackage[colorlinks=true,linkcolor=blue]{hyperref}%
28 %\expandafter\ifx\csname package@font\endcsname\@undefined\else
29 % \expandafter\RequirePackage\expandafter{\csname package@font\endcsname}%
```

```

30 %\fi
31 \CodelineIndex\EnableCrossrefs % makeindex -s gind.ist ltxfront
32 \RecordChanges % makeindex -s gglo.ist -o ltxfront.gls ltxfront.glo

```

### 3.1.1 Docstrip and info directives

We use so many DOCSTRIP modules that we set the `StandardModuleDepth` counter to 1.

```
33 \setcounter{StandardModuleDepth}{1}
```

The following command retrieves the date and version information from this file.

```
34 \expandafter\GetFileInfo\expandafter{\jobname.dtx}%
```

## 3.2 The “Read Me” File

As promised above, here is the contents of the “Read Me” file. That file serves a double purpose, since it also constitutes the beginning of the programmer’s documentation. What better thing, after all, to have appear at the beginning of the typeset documentation?

A good discussion of how to write a ReadMe file can be found in Engst, Tonya, “Writing a ReadMe File? Read This” *MacTech* October 1998, p. 58.

Note the appearance of the `\StopEventually` command, which marks the dividing line between the user documentation and the programmer documentation.

The usual user will not be asked to do a full build, not to speak of the bootstrap. Instructions for carrying these processes begin the programmer’s manual.

```

35 \begin{filecontents*}{README-LTXFRONT.tex}
36 \title{%
37 A \LaTeX\ Package for\the frontmatter and title page%
38 \thanks{%
39 This file has version number \fileversion,
40 last revised \filedate.%
41 }%
42 \thanks{%
43 Version \fileversion\ \copyright\ 2019 American Physical Society
44 }%
45 }%
46
47 \author{%
48 Arthur Ogawa%
49 \thanks{\texttt{mailto:arthur\_ogawa at sbcglobal.net}}%
50 }%
51 %\iffalse
52 % For version number and date,
53 % search on "\fileversion" in the .dtx file,
54 % or see the end of the README-LTXFRONT file.
55 %\fi
56 \maketitle
57

```

58 This file embodies the `\classname{ltxfront}` package,  
59 the implementation and its user documentation.  
60

61 The distribution point for this work is  
62 `\url{https://journals.aps.org/revtex}`,  
63 which contains fully unpacked, prebuilt runtime files and documentation.  
64

65 The `\classname{ltxfront}` package was commissioned by the American Physical Society  
66 and is distributed under the terms of the `\LaTeX` Project Public License 1.3c,  
67 the same license under which all the portions of `\LaTeX` itself is distributed.  
68 Please see `\url{http://ctan.tug.org/macros/latex/base/lppl.txt}` for details.  
69

70 To use this document class, you must have a working  
71 `\TeX` installation equipped with `\LaTeXe`  
72 and possibly `pdftex` and Adobe Acrobat Reader or equivalent.  
73

74 To install, retrieve the distribution,  
75 unpack it into a directory on the target computer,  
76 and move the file `\file{ltxfront.sty}`  
77 into a location in your filesystem where it will be found by `\LaTeX`.  
78

79 To use, read the user documentation `\file{ltxfront.pdf}`.  
80

81 `\tableofcontents`  
82

83 `\section{Processing Instructions}`  
84

85 The package file `\file{ltxfront.sty}`  
86 is generated from this file, `\file{ltxfront.dtx}`,  
87 using the `{\sc docstrip}` facility of `\LaTeX`  
88 via `|tex ltxfront.dtx|`.  
89 The typeset documentation that you are now reading is generated from  
90 the same file by typesetting it with `\LaTeX` or `pdftex`  
91 via `|latex ltxfront.dtx|` or `|pdflatex ltxfront.dtx|`.  
92

93 `\subsection{Build Instructions}`  
94

95 You may bootstrap this suite of files solely from `\file{ltxfront.dtx}`.  
96 Prepare by installing `\LaTeXe` (and either `tex` or `pdftex`) on your computer,  
97 then carry out the following steps:  
98 `\begin{enumerate}`  
99

100 `\item`  
101 Within an otherwise empty directory,  
102 typeset `\file{ltxfront.dtx}` with `\TeX` or `pdftex`,  
103 thereby generating the package file `\file{ltxfront.sty}`.  
104

105 `\item`  
106 Next typeset `\file{ltxfront.dtx}` with `\LaTeX` or `pdflatex`;  
107 you will obtain the typeset documentation you are now reading,

```

108 along with
109 the sample document \file{ltxfront-test.tex}
110 and the file \file{README-LTXFRONT}.
111
112 Note: you will have to run \LaTeX, then
113 \file{makeindex} \texttt{-s gind.ist ltxfront.idx}, then
114 \file{makeindex} \texttt{-s gglo.ist -o ltxfront.gls ltxfront.glo}, then
115 \LaTeX\ again in order to obtain a valid index and table of contents.
116 \item
117 Now typeset \file{ltxfront.dtx} with \TeX (not \LaTeX),
118 thereby generating the package file \file{ltxfront.sty}.
119 \item
120 Install the following files into indicated locations within your
121 TDS-compliant \texttt{texmf} tree (you may need root access):
122 \begin{itemize}
123 \item
124 \file{\$TEXMF/}\file{tex/}\file{latex/}\file{revtex/}\classname{ltxfront.sty}
125 \item
126 \file{\$TEXMF/}\file{source/}\file{latex/}\file{revtex/}\classname{ltxfront.dtx}
127 \item
128 \file{\$TEXMF/}\file{doc/}\file{latex/}\file{revtex/}\classname{ltxfront.pdf}
129 \end{itemize}
130 where \file{\$TEXMF/} stands for \file{texmf-local/}, or some other \texttt{texmf} tree
131 in your installation.
132 \item
133 Run \texttt{mktexlsr} on \file{\$TEXMF/} (you may need root access).
134 \item
135 Build and installation are now complete;
136 now put a \cmd\usepackage\texttt{\{ltxfront\}} in your document preamble!
137 \end{enumerate}
138
139 \subsection{Change Log}
140 \changes{4.0b}{1999/06/20}{Change handling of \cs{@author} to prevent parsing problems. (Helps
141 \changes{4.0b}{1999/06/20}{Gathered all code for the abstract environment together and abstract
142 \changes{4.0b}{1999/06/20}{Separate option now controls production}
143 \changes{4.0b}{1999/06/20}{Set \cs{footnote} to \LaTeX\ standard version for this scope.}
144 \changes{4.0c}{1999/11/13}{grid changes with ltxgrid}
145 \changes{4.0d}{1998/01/31}{\cs{comma@space}}
146 \changes{4.0d}{1998/01/31}{change name from \cs{tempaffiliation}}
147 \changes{4.0d}{1998/01/31}{hookify}
148 \changes{4.0d}{1998/01/31}{If empty \cs{abstractname} omit spacing}
149 \changes{4.0d}{1998/01/31}{PACS boxed to width of abstract}
150 \changes{4.0e}{2000/09/20}{Keywords on same footing as PACS}
151 \changes{4.0e}{2000/09/20}{Warn if no production}
152 \changes{4.0f}{2001/02/13}{Actually append \cs{AU@grp} to \cs{@AAClist}.}
153 \changes{4.0g}{1998/06/10}{multiple preprint commands}
154 \changes{4.0h}{1998/06/25}{Changes to support groups of affiliations}
155 \changes{4.0h}{1998/06/25}{new hook}
156 \changes{4.0l}{1998/09/01}{add optional argument handling back}
157 \changes{4.0l}{1998/09/01}{Changes to add collaboration processing, which now means doing comma

```



```

158 \changes{4.0l}{1998/09/01}{macro added}
159 \changes{4.1a}{2008-06-30}{(AO, 443) create an entry in the PDF bookmarks for the abstract via
160 \changes{4.1a}{2008-07-01}{(AO, 496) remove default prependation to \cs{email}.}
161 \changes{4.1a}{2008-07-01}{(AO, 496) remove default prependation to \cs{homepage}.}
162 \changes{4.1a}{2008/06/16}{removed code that had been commented out}
163 \changes{4.1b}{2008/06/29}{(AO, 455) provide a \texttt{description} environment for the abstract
164 \changes{4.1b}{2008/08/02}{(AO, 486) \cs{collaboration} to work with \texttt{groupedaddress}}
165 \changes{4.1b}{2008/08/02}{Change csname \cs{AFS@opr} to \cs{AFG@opr}}
166 \changes{4.1b}{2008/08/02}{Make the syntax of this package switch optional}%
167 \changes{4.1b}{2008/08/02}{Provide for diagnostics via \cs{frontmatterverbose@sw}}
168 \changes{4.1b}{2008/08/02}{Use \cs{@affil@def} for all access to \cs{@affil}}
169 \changes{4.1b}{2008/08/02}{Use \cs{frontmatter@init} as the vehicle for all initialization}
170 \changes{4.1b}{2008/08/04}{(AO, 443) PDF Bookmark for title page}%
171 \changes{4.1b}{2008/08/04}{Provide facility \cs{affil@cutoff}: if there is only one affiliation
172 \changes{4.1c}{2008/08/04}{Hyperref compatibility: take \cs{maketitle} back.}
173 \changes{4.1c}{2008/08/04}{Parametrize the production after the address}%
174 \changes{4.1c}{2008/08/04}{Produce PACS, keywords only if specified}%
175 \changes{4.1d}{2008/10/17}{(AO, 410) compare \cs{@tempa} to \cs{blankaffiliation}.}%
176 \changes{4.1f}{2009/07/07}{(AO, 515) Use \cs{set@footnotefont}, which is defined in ltxutil}
177 \changes{4.1g}{2009/10/05}{(AO, 535) Robustify \cs{footnote} while \cs{class@warn} is expanding
178 \changes{4.1g}{2009/10/05}{(AO, 539) Robustify \cs{class@warn} against TOC processing}%
179 \changes{4.1i}{2009/10/18}{(AO, 540) More runtime diagnostics via \cs{frontmatterverbose@sw}}
180 \changes{4.1i}{2009/10/23}{(AO, 540) Rationalize code that appends to \cs{AF@grp}}
181 \changes{4.1i}{2009/10/23}{(AO, 540) Optimize boolean logic }
182 \changes{4.1i}{2009/10/23}{(AO, 544) Class options runinaddress and unsortedaddress stopped wor
183 \changes{4.1m}{2009/11/20}{(AO, 563) \cs{collaboration} to work with both \texttt{groupedaddress}
184 \changes{4.1m}{2009/11/20}{(AO, 563) More complete runtime diagnostics via \cs{frontmatterverbo
185 \changes{4.1m}{2009/11/20}{(AO, 563) Let \cs{@collaboration@gobble} parse all three arguments o
186 \changes{4.1n}{2009/11/30}{(AO, 567) Handle \cs{noaffiliation} case}
187 \changes{4.1n}{2009/11/30}{(AO, 567) Handle punctuation for \classoption{superscriptaddress} st
188 \changes{4.1n}{2009/12/03}{(AO) remove http:// from \cs{href} call}
189 \changes{4.1n}{2010/01/02}{(AO, 571) Interface \cs{set@footnotewidth} for determining the set w
190 \changes{4.1n}{2010/01/02}{(AO, 571) allow split after last line of footnote}%
191 \changes{4.1n}{2010/01/02}{(AO, 572) Independent footnote counter for title block. Abstract foo
192 \changes{4.2a}{2018/12/12}{(MD) Updated name of README file and use standard fonts when typeset
193
194 \end{filecontents*}

```

### 3.3 A Gnuly Created Sample Document

```

195 \begin{filecontents*}{ltxfront-test.tex}
196 \documentclass{article}% use any combination of titlepage, twocolumn
197 \usepackage[]{}{ltxfront}% use inactive to turn off features
198 \begin{document}
199
200 \title{Gnu Veldt Cuisine}
201
202 \author{G. Picking}
203 \thanks{Supported by a grant from the GSF.}
204 \affiliation{Acme Kitchen Products}

```

```

205
206 \author{R. Dillo}
207 \thanks{On leave during 1985.}
208 \affiliation{Cordon Puce School}
209
210 \date{24 July 1984}%
211 \revised{5 January 1985}%
212
213 \maketitle
214
215 \section{Introduction}%
216 We reproduce most features of \LaTeX\ manual, Figure C.4, p. 183.
217 \end{document}
218 \end{filecontents*}

```

### 3.4 The Document Body

Here is the document body, containing only a `\DocInput` directive—referring to this very file. This very cute self-reference is a common `ltxdoc` idiom.

```

219 \begin{document}%
220 \expandafter\DocInput\expandafter{\jobname.dtx}%
221 % ^^A\PrintChanges
222 \end{document}
223 %</doc>

```

## 4 Using this package

Once this package is installed on your filesystem, you can employ it in adding functionality to  $\text{\LaTeX}$  by invoking it in your document or document class.

### 4.1 Invoking the package

In your document, you can simply call it up in your preamble:

```

%\documentclass{book}%
%\usepackage{ltxfront}%
%\begin{document}
%<your document here>
%\end{document}

```

However, the preferred way is to invoke this package from within your customized document class:

```

%\NeedsTeXFormat{LaTeX2e}[1995/12/01]%
%\ProvidesClass{myclass}%
%\RequirePackage{ltxfront}%
%\LoadClass{book}%
%<class customization commands>
%\endinput

```

Once loaded, the package gives you access to certain procedures, usually to be invoked by a  $\LaTeX$  command or environment.

## 5 Frontmatter commands

A document using this package will use commands like  $\backslash$ title,  $\backslash$ author,  $\backslash$ thanks, the `abstract` environment and  $\backslash$ maketitle. These are part of  $\LaTeX$ 's standard document classes. But this package supplements  $\LaTeX$  syntax with additional commands, such as  $\backslash$ affiliation,  $\backslash$ collaboration,  $\backslash$ email,  $\backslash$ homepage, and others.

Furthermore, you can have multiple  $\backslash$ author,  $\backslash$ affiliation, and  $\backslash$ collaboration commands, each with their own set of  $\backslash$ email,  $\backslash$ homepage, etc. commands.

## 6 The Front Matter

This section describes the macros this package provides for formatting the front matter of an article. The behavior and usage of these macros can be quite different from those provided in  $\LaTeX 2_{\epsilon}$ .

### 6.1 Setting the title

The title of the manuscript is simply specified by using the  $\backslash$ title{<title>} macro. A  $\backslash\backslash$  may be used to put a line break in a long title. The  $\backslash$ title command takes an optional argument, which you will use to give an abbreviated title for use in the running header.

### 6.2 Specifying a date

The  $\backslash$ date<date> command outputs the date on the manuscript. Using  $\backslash$ today will cause  $\LaTeX$  to insert the current date whenever the file is run:

```
 $\backslash$ date{ $\backslash$ today}
```

### 6.3 Specifying authors and affiliations

Commands specifying author and affiliation have been improved to save labor for authors and in production. Authors and affiliations are arranged into groupings called, appropriately enough, *author groups*. Each author group is a set of authors who share the same set of affiliations. Author names are specified with the  $\backslash$ author macro while affiliations (or addresses) are specified with the  $\backslash$ affiliation macro. Author groups are specified by sequences of  $\backslash$ author macros followed by  $\backslash$ affiliation macros. An  $\backslash$ affiliation macro applies to all previously specified  $\backslash$ author macros which don't already have an affiliation supplied.

For example, if Bugs Bunny and Roger Rabbit are both at Looney Tune Studios, while Mickey Mouse is at Disney World, the markup would be:

```
% \author{Bugs Bunny}
% \author{Roger Rabbit}
% \affiliation{Looney Tune Studios}
% \author{Mickey Mouse}
% \affiliation{Disney World}
```

The default is to display this as

Bugs Bunny and Roger Rabbit  
*Looney Tune Studios*  
Mickey Mouse  
*Disney World*

This layout style for displaying authors and their affiliations is chosen by selecting the class option `groupedaddress`. This option is the default for all APS journal styles, so it does not need to be specified explicitly. The other major way of displaying this information is to use superscripts on the authors and affiliations. This can be accomplished by selecting the class option `superscriptaddress`. To achieve the display

Bugs Bunny,<sup>1</sup> Roger Rabbit,<sup>1,2</sup> and Mickey Mouse<sup>2</sup>  
<sup>1</sup>*Looney Tune Studios*  
<sup>2</sup>*Disney World*

one would use the markup

```
% \author{Bugs Bunny}
% \affiliation{Looney Tune Studios}
% \author{Roger Rabbit}
% \affiliation{Looney Tune Studios}
% \affiliation{Disney World}
% \author{Mickey Mouse}
% \affiliation{Disney World}
```

Note that this package takes care of any commas and *and*'s that join the author names together and font selection, as well as any superscript numbering. Only the author names and affiliations should be given within their respective macros.

There is a third class option, `unsortedaddress`, for controlling author/affiliation display. The default `groupedaddress` will actually sort authors into the appropriate author groups if one chooses to specify an affiliation for each author. The markup:

```
% \author{Bugs Bunny}
% \affiliation{Looney Tune Studios}
% \author{Mickey Mouse}
% \affiliation{Disney World}
% \author{Roger Rabbit}
% \affiliation{Looney Tune Studios}
```

will result in the same display as for the first case given above even though Roger Rabbit is specified after Mickey Mouse. To avoid Roger Rabbit being moved into the same author group as Bugs Bunny, use the `\unsortedaddress` option instead. In general, it is safest to list authors in the order they should appear and specify affiliations for multiple authors rather than one at a time. This will afford the most independence for choosing the display option. Finally, it should be mentioned that the affiliations for the `\superscriptaddress` are presented and numbered in the order that they are encountered. This means that the order will usually follow the order of the authors. An alternative ordering can be forced by including a list of `\affiliation` commands before the first `\author` in the desired order. Then use the exact same text for each affiliation when specifying them for each author.

If an author doesn't have an affiliation, the `\noaffiliation` macro may be used in the place of an `\affiliation` macro.

### 6.3.1 Collaborations

A collaboration name can be specified with the `\collaboration` command. This is very similar to the `\author` command. It should appear at the end of an author group. The collaboration name will appear centered in parentheses. You may have multiple author groups, each with its own `\collaboration`. Because collaborations don't normally have affiliations, one needs to follow the `\collaboration` with `\noaffiliation`.

### 6.3.2 Footnotes for authors, collaborations, affiliations or title

Often one wants to specify additional information associated with an author, collaboration, or affiliation such as an e-mail address, an alternate affiliation, or some other ancillary information. This package introduces several new macros just for this purpose. They are:

- `\email[optional text]{e-mail address}`
- `\homepage[optional text]{URL}`
- `\altaffiliation[optional text]{affiliation}`
- `\thanks{miscellaneous text}`

In the first three, the *optional text* will be prepended before the actual information specified in the required argument. `\email` and `\homepage` each have a default text for their optional arguments ('Electronic address:' and 'URL:' respectively). The `\thanks` macro should only be used if one of the other three do not apply. Any author name can have multiple instances of these four commands. Note that unlike the `\affiliation` macro, these macros only apply to the `\author` that directly precedes it. Any `\affiliation` *must* follow the other author-specific macros. A typical usage might be as follows:

```
% \author{Bugs Bunny}
```

```

% \email[E-mail me at: ]{bugs@looney.com}
% \homepage[Visit: ]{http://looney.com/}
% \altaffiliation[Permanent address: ]
%           {Warner Brothers}
% \affiliation{Looney Tunes}

```

This would result in the footnote “E-mail me at: bugs@looney.com, Visit: http://looney.com/, Permanent address: Warner Brothers” being attached to Bugs Bunny. Note that:

- Only an e-mail address, URL, or affiliation should go in the required argument in the curly braces.
- The font is automatically taken care of.
- An explicit space is needed at the end of the optional text if one is desired in the output.
- Use the optional arguments to provide customized text only if there is a good reason to.

The `\collaboration`, `\affiliation`, or even `\title` can also have footnotes attached via these commands. If any ancillary data (`\thanks`, `\email`, `\homepage`, or `\altaffiliation`) are given in the wrong context (e.g., before any `\title`, `\author`, `\collaboration`, or `\affiliation` command has been given), then a warning is given in the TeX log, and the command is ignored.

Duplicate sets of ancillary data are merged, giving rise to a single shared footnote. However, this only applies if the ancillary data are identical: even the order of the commands specifying the data must be identical. Thus, for example, two authors can share a single footnote indicating a group e-mail address.

Duplicate `\affiliation` commands may be given in the course of the front matter, without the danger of producing extraneous affiliations on the title page. However, ancillary data should be specified for only the first instance of any particular institution’s `\affiliation` command; a later instance with different ancillary data will result in a warning in the TeX log.

It is preferable to arrange authors into sets. Within each set all the authors share the same group of affiliations. For each author, give the `\author` (and appropriate ancillary data), then follow this author group with the needed group of `\affiliation` commands.

If affiliations have been listed before the first `\author` macro to ensure a particular ordering, be sure that any later `\affiliation` command for the given institution is an exact copy of the first, and also ensure that no ancillary data is given in these later instances.

A journal may determine the placement of these ancillary information footnotes. One journal may put all such footnotes at the start of the bibliography while another journal may display them on the first page, as a footnote. One can override a journal style’s default behavior by specifying explicitly the class option `bibnotes` (puts the footnotes at the start of the bibliography) or `nobibnotes` (puts them on the first page).

### 6.3.3 Specifying first names and surnames

Many authors have names in which either the surname appears first or in which the surname is made up of more than one name. To ensure that such names are accurately captured for indexing and other purposes, the `\surname` macro should be used to indicate which portion of a name is the surname. Similarly, there is a `\firstname` macro as well, although usage of `\surname` should be sufficient. If an author's surname is a single name and written last, it is not necessary to use these macros. These macros do nothing but indicate how a name should be indexed. Here are some examples;

```
% \author{Andrew \surname{Lloyd Weber}}
% \author{\surname{Mao} Tse-Tung}
```

## 6.4 The abstract

An abstract for a paper is specified by using the `abstract` environment:

```
% \begin{abstract}
% Text of abstract
% \end{abstract}
```

Note that with this package, the abstract must be specified before the `\maketitle` command, and there is no need to embed it in an explicit minipage environment.

## 6.5 PACS codes

Some authors are asked to supply suggested PACS codes with their submissions. The `\pacs` macro is provided as a way to do this:

```
\pacs{23.23.+x, 56.65.Dy}
```

The actual display of the PACS numbers below the abstract is controlled by two class options: `showpacs` and `noshowpacs`. In particular, this is now independent of the `preprint` option. `showpacs` must be explicitly included in the class options to display the PACS codes.

## 6.6 Keywords

A `\keywords` macro may also be used to indicate keywords for the article.

```
\keywords{nuclear form; yrast level}
```

This will be displayed below the abstract and PACS (if supplied). Like PACS codes, the actual display of the the keywords is controlled by two classoptions: `showkeys` and `noshowkeys`. An explicit `showkeys` must be included in the `\documentclass` line to display the keywords.

## 6.7 Institutional report numbers

Institutional report numbers can be specified using the `\preprint` macro. These will be displayed in the upper lefthand corner of the first page. Multiple `\preprint` macros may be supplied (space is limited though, so only three or less may actually fit).

## 6.8 `maketitle`

After specifying the title, authors, affiliations, abstract, PACS codes, and report numbers, the final step for formatting the front matter of the manuscript is to issue the `\maketitle` command:

```
\maketitle
```

This command must follow all of the macros listed above. The macro will format the front matter in accordance with the various class options that were specified in the `\documentclass` line (either implicitly through defaults or explicitly).

## 7 Compatibility with L<sup>A</sup>T<sub>E</sub>X's Required Packages

Certain packages, usually ones written by members of the L<sup>A</sup>T<sub>E</sub>X Project itself, have been designated “required” and are distributed as part of standard L<sup>A</sup>T<sub>E</sub>X. These packages have been placed in a privileged position vis á vis the L<sup>A</sup>T<sub>E</sub>X kernel in that they override the definitions of certain kernel macros.

The `ltxfront` package will be incompatible with any package that redefines any of the kernel macros that `ltxfront` patches—if that package is loaded *after* `ltxfront`.

Hereinafter follows some notes on specific L<sup>A</sup>T<sub>E</sub>X packages.

## 8 Implementation of package

### 8.1 Beginning of the package `DOCSTRIP` module

```
224 %<*package>
225 \def\package@name{ltxfront}%
226 \expandafter\PackageInfo\expandafter{\package@name}{%
227 Title page macros for \protect\LaTeXe,
228 by A. Ogawa (arthur_ogawa at sbcglobal.net)%
229 }%
```

### 8.2 Prerequisites

This package requires the use of the `ltxutil` package.

```
230 \let\class@name\package@name
231 \RequirePackage{ltxutil}%
232 %</package>
```

End of the preliminary portion of the package code.



### 8.3 Options

Package options are treated in this portion of the dtx file. This portion is guarded by the `options docstrip` code so that it can be merged into a larger package gracefully.

```
233 %<*options>
```

Package option turns on diagnostics that trace its operation. `\frontmatterverbose@sw` is set false by default: we do not elect to monitor the workings of these procedures.

```
234 \DeclareOption{frontmatterverbose}{\@booleantrue\frontmatterverbose@sw}%
```

```
235 \@booleanfalse\frontmatterverbose@sw
```

Package option inactivates its syntax, for testing purposes.

```
236 \DeclareOption{inactive}{\@booleanfalse\frontmatter@syntax@sw}%
```

```
237 \@booleantrue\frontmatter@syntax@sw
```

`\runinaddress@sw` The flag `\runinaddress@sw` signifies that author addresses are to be set on the same line with the author.

The `runinaddress` option is the only one that sets `\runinaddress@sw` to true.

FIXME: this option pertains to cases where `\groupauthors@sw` is true, but is independant of `\@affils@sw`. So, it should be possible to assert both `runinaddress` and `unsortedaddress` as well as each separately.

```
238 \@booleanfalse\runinaddress@sw
```

`\@affils@sw` If `\@affils@sw` is false, an address is never recognised as “new” and is therefore always entered into the affiliation list, stopping groups of authors at the same address being consolidated into the same list: the address will be printed the same number of times it is entered.

The `unsortedaddress` option is the only one that sets `\@affils@sw` to false.

```
239 \@booleantrue\@affils@sw
```

`\groupauthors@sw` The flag `\groupauthors@sw` signifies that authors are to be grouped, affecting the meaning of `\@author@present`.

Clients will want to set `\groupauthors@sw` true if they want grouped addresses or either of the following variants: for unsorted addresses: set `\@affils@sw` false; for run-in addresses, set `\runinaddress@sw` true.

```
240 \@booleanfalse\groupauthors@sw
```

`groupedaddress` If we take the `groupedaddress` option as the default, then we can look upon the `superscriptaddress` option as simply turning `\groupauthors@sw` to `\false@sw`. `groupedaddress`, the default in APS journals, groups authors above a common address.

```
241 \DeclareOption{groupedaddress}{\clo@groupedaddress}%
```

```
242 \def\clo@groupedaddress{%
```

```
243 \@booleantrue\groupauthors@sw
```

```
244 \@booleantrue\@affils@sw
```

```
245 \@booleanfalse\runinaddress@sw
```

```
246 }%
```

`unsortedaddress` `unsortedaddress` is similar to `groupedaddress`, but turns off `\@affils@sw`, with the result that each address that is entered will be printed.

```
247 \DeclareOption{unsortedaddress}{\clo@unsortedaddress}%
248 \def\clo@unsortedaddress{%
249   \@booleantrue\groupauthors@sw
250   \@booleanfalse\@affils@sw
251   \@booleanfalse\runinaddress@sw
252 }%
```

`runinaddress` `runinaddress` is similar to `groupedaddress`, with the addition of the Boolean `\runinaddress@sw`, which causes the authors associated with a particular address group to be formatted in a paragraph instead of on separate lines.

```
253 \DeclareOption{runinaddress}{\clo@runinaddress}%
254 \def\clo@runinaddress{%
255   \@booleantrue\groupauthors@sw
256   \@booleantrue\@affils@sw
257   \@booleantrue\runinaddress@sw
258 }%
```

`superscriptaddress` `superscriptaddress` presents author affiliations as superscripts. Authors with like affiliations share the same superscript.

FIXME: if `\groupauthors@sw` is false, would not `\@affils@sw` be a don't care?

```
259 \DeclareOption{superscriptaddress}{\clo@superscriptaddress}%
260 \def\clo@superscriptaddress{%
261   \@booleanfalse\groupauthors@sw
262   \@booleantrue\@affils@sw
263   \@booleanfalse\runinaddress@sw
264 }%
265 %</options>
```

## 8.4 Process package options

```
266 %<*package>
267 \DeclareOption*{\OptionNotUsed}%
268 \ProcessOptions*
269 %</package>
```

## 9 Kernel of Front Matter

Here begins the kernel of the `frontmatter` package.

```
270 %<*kernel>
```

This package implements a new syntax for the title page of an article.

Special acknowledgment: this package uses concepts pioneered and first realized by William Baxter (<mailto:web@superscript.com>) in his SuperScript line of commercial typesetting tools, and which are used here with his permission.

These macros were first coded by David P. Carlisle for the American Physical society in the late 1990s as part of the development of REVTeX4. That development work was taken over by Arthur Ogawa, who is the author of this package.

## 9.1 Initialization

`\frontmatter@init` The `\frontmatter@init` procedure sets up all the registers for processing the title page. At present, this is done once, at `\class@documenthook` time. If multiple title pages are to be processed within a job, the initialization must be done for each.

```
271 \appdef\class@documenthook{\frontmatter@init}%
272 \let\frontmatter@init\empty
```

## 9.2 The title command

`\title` The author uses `\title` to specify the title. As in the AMS classes, `\title` has an optional argument specifying a short form for use in running heads.

`\do@title` Once the `\title` is given, you can specify any combination of `\thanks`, `\email`, `\homepage`, and `\altaffiliation` commands, applying to the title of the document.

Compatibility note: as with all of the standard L<sup>A</sup>T<sub>E</sub>X commands that we override, we make here a private definition and later switch it in if so required.

```
273 \newcommand\frontmatter@title[2] []{%
274 \def\@title{#2}%
275 \def\@shorttitle{#1}%
276 \let\@AF@join\@title@join
277 }%
```

`\@title` The arguments to `\title` are saved in these internal macros, which are set up to produce a warning if `\title` has not been used before `\maketitle`.

`\@shorttitle` Extensibility note: by using `\frontmatter@init` as the initialization procedure, we open the possibility of more than one title page per document.

```
278 \appdef\frontmatter@init{%
279 \def\@title{\class@warn{No title}}%
280 \let\@shorttitle\empty
281 \let\@title@aux\@title@aux@cleared
282 }%
```

```
\@title@join
\@title@join@
\@title@aux@cleared 283 \def\@title@join{\expandafter\@title@join@\@title@aux}%
284 \def\@title@join@#1#2{%
285 \def\@title@aux{\@join{\@separator}{#1}{#2}}%
286 }%
287 \def\@title@aux@cleared{}}%
```

## 9.3 The author, affiliation, and related commands

`\c@affil` `\c@affil` stores the affiliation numbers used for the superscript marks. The `\c@collab` `\newcounter` command defines `\theaffil`, which we never use: this counter is evaluated solely via `\the`.

`\c@collab` stores the collaboration numbers. The `\newcounter` command defines `\thecollab`. As with `\c@affil`, we never use `\thecollab`; only `\the` `\c@collab`.

Each counter must be cleared before frontmatter can be processed.

```
288 \newcounter{affil}%
289 \newcounter{collab}%
290 \appdef\frontmatter@init{%
291   \c@affil\z@
292   \c@collab\z@
293 }%
```

### 9.3.1 Commands for author or collaboration

`\author` Each author is given in a *separate* `\author` command. This is similar to the AMS classes, but the AMS also try to support the `article` class's `\and` command for putting multiple authors in one `\author` command. This package, however, does not support the `\and` command.

When the `\author` command is encountered, the previous author, author group, affiliation, and affiliation group are at an end. This procedure moves the previous author to the author group (`\move@AU`), moves the previous affiliation to the affiliation group (`\move@AF`), and terminates the previous author/affiliation group (`\move@AUAF`) as the case may require.

Compatibility note: as with all of the standard  $\text{\LaTeX}$  commands that we override, we make here a private definition and later switch it in if so required.

```
294 \newcommand\frontmatter@author{% implicit #1
295   \@author@def}% implicit #2
296 }%
```

`\collaboration` Specify the collaboration (given after a group of authors).

This procedure does exactly what `\author` does, and sets a flag signifying that the `\collaboration` command was given.

Note: earlier versions attempted to ignore the collaboration command if it was considered inappropriate. We must not do this, however, because it throws the state machine out of kilter.

```
297 \def\collaboration{% implicit #1
298   \@author@def{\@booleantrue\collaboration@sw}% implicit #2
299 }%
300 \appdef\frontmatter@init{%
301   \@booleanfalse\collaboration@sw
302 }%
```

`\@author@init` Here are all the access procedures for the author data structure.

`\@author@cleared` We define a flag value for `\@author(private)`, a procedure `\@author@gobble`  
`\@authorclear@sw` to read author data without any effect, a procedure for setting `\@author`  
`\@author@def` to the flag value (`\@author@init`), a Boolean to test against the flag value  
`\@author@join@` (`\@authorclear@sw`), and a procedure to add new ancillary data to the author.  
`\@author@join`

Note that expanding the functionality of the `\collaboration` command entailed changing the data structures, because now all authors must remember the collaboration they belong to. This change affects procedures `\@author@cleared` and `\@author@gobble`.

```

303 \def\@author@cleared{}{}%
304 \def\@author@gobble#1#2#3{}%
305 \def\@author@init{%
306   \let\@author\@author@cleared
307   \@booleanfalse\collaboration@sw
308 }%
309 \def\@authorclear@sw{\@ifx{\@author\@author@cleared}}%
310 \appdef\frontmatter@init{%
311   \@author@init
312 }%

313 \def\@author@def#1#2{%
314   \frontmatterverbose@sw{\typeout{\string\author\space\string\collaboration}}{}%
315   \move@AU\move@AF\move@AUAF
316   \let\@AF@join\@author@join
317   #1%
318   \def\@author{{#2}}%
319 }%
320 \def\@author@join@#1#2#3{%
321   \def\@author{{#1}}{\@join{\@separator}{#2}{#3}}%
322 }%
323 \def\@author@join{\expandafter\@author@join@\@author}%

```

`\move@AU` The `\move@AU` procedure moves the most recent author to the current author group `\AU@grp`, or the most recent collaboration to the collaboration group `\CO@grp`. If we recently did a `\@author@init`, this is a no-op.

To accommodate the expanded syntax of `\collaboration`, we add a new field to those associated with an author (collaborator), namely the value of the collaboration counter.

```

324 \def\move@AU{%
325   \@authorclear@sw{}{}%
326   \collaboration@sw{%
327     \advance\c@collab@ne
328     \@argswap{\CO@grp\CO@opr}%
329   }{}%
330   \@argswap{\AU@grp\AU@opr}%
331   }%
332   {%
333     \expandafter\@argswap@val
334     \expandafter{\@author}%
335     {\expandafter\@argswap@val\expandafter{\the\c@collab}{\add@AUCO@grp}}%
336   }%
337 }%
338 \@author@init
339 }%

```

The procedure `\add@AUCO@grp` adds to the specified group `#3` using the given intercolated operator `#4`, with arguments `{#1}` and `#2`, where the first argument is the collaboration number, the second is the author name, and the third the joined attributes of the author. For example, `\move@AU` could finally execute: `\appdef \AU@grp {\AU@opr{1}{A. Author}{}}`.

```
340 \def\add@AUCO@grp#1#2#3#4{%
341 \appdef#3{#4{#1}#2}%
342 \frontmatterverbose@sw{\say#3}{}%
343 }%
```

`\@author@finish` We define a procedure to terminate author/affiliation parsing just before the title block is formatted. We detect the case where an author group is not followed by an affiliation group.

We first finish off any author- or affiliation groups in progress. Next, we detect the case where there is an author group or a collaboration group in progress, but no affiliation group. If so, we emit a diagnostic message and act as if a `\noaffiliation` command had been given. In any case, we finish off any author/affiliation pair that may remain. Formatting of the title block may now proceed.

```
344 \def\@author@finish{%
345 \frontmatterverbose@sw{\typeout{\string\@author@finish}}{%
346 \move@AU\move@AF
347 \@ifx{\AU@grp\@empty}{%
348 \@ifx{\CO@grp\@empty}{%
349 }{%
350 \false@sw
351 }%
352 }{%
353 \@ifx{\AF@grp\@empty}{%
354 \beginingroup
355 \let\href\@secondoftwo
356 \let\AU@opr\@secondofthree
357 \let\CO@opr\@secondofthree
358 \let\footnote\@gobble
359 \@ifx{\CO@grp\@empty}{%
360 \class@warn{Assuming \string\noaffiliation\space for authors}%
361 \frontmatterverbose@sw{\say\AU@grp}%
362 }{%
363 \class@warn{Assuming \string\noaffiliation\space for collaboration}%
364 \frontmatterverbose@sw{\say\CO@grp}{}%
365 }%
366 \endgroup
367 \@affil@none\move@AF
368 }{%
369 }%
370 \move@AUAF
371 }%
372 \def\@secondofthree#1#2#3{#2}%
```

`\@join` The procedure `\@join` inserts a separator between two tokens, or, if the first token is nil, elides both that token and the separator.

```
373 \def\@join#1#2#3{%
374 \if@empty{#2}{#3}{#2#1#3}%
375 }%
376 \def\@separator{;\space}%
```

`\surname` No-op to allow better post-processing of the file.

```
\firstname 377 \let\surname\@firstofone
378 \let\firstname\@firstofone
```

`\and` The original L<sup>A</sup>T<sub>E</sub>X idea of using one `\author` command to capture all authors and their address just doesn't work with multiple authors possibly sharing addresses, so in this class disable `\and`.

```
379 \newcommand\frontmatter@and{\class@err{\protect\and\space is not supported}}
```

### 9.3.2 Commands for affiliation

`\affiliation` After each group of authors with the same address, give that address in `\affiliation`. If later in the list you have an author with the address of an earlier author, repeat the `\affiliation` command: the system will detect the equivalence and (if using the `superscriptaddress` option) only print the affiliation once, reusing the superscript marker.

When the `\affiliation` command is encountered, the current author and author group (if any) are at an end. Also, the current affiliation is at an end.

FIXME: changes to catcode required elsewhere now?

```
380 \def\cat@comma@active{\catcode'\,\active}%
381 {\cat@comma@active\gdef,{\active@comma}}%
382 \def\active@comma{,\penalty-300\relax}%
383 \newcommand\affiliation{%
384 \frontmatterverbose@sw{\typeout{\string\affiliation}}}%
385 \move@AU\move@AF
386 \begingroup
387 \cat@comma@active
388 \@affiliation
389 }%
390 \def\@affiliation#1{%
391 \endgroup
392 \let\@AF@join\@affil@join
393 \@affil@def{#1}%
394 }%
```

`\noaffiliation` User-level command `\noaffiliation` signifies that the current group of authors, or the current collaboration, has *no* affiliation.

This is implemented by acting as if the `\affiliation` command has been given, and using a flag value of `{\relax}`, which adds an element to the affiliation group of this author, albeit a nil one (`\AFF@opr {\relax} {}`).

Note that this procedure does substantially the same as `\@author@finish`. (the latter, If it does not bail out altogether, differs chiefly in providing diagnostics).

```
395 \newcommand\frontmatter@noaffiliation{%
396 \frontmatterverbose@sw{\typeout{\string\noaffiliation}}{}%
397 \move@AU\move@AF
```

In `\@author@finish`, there is code testing if an effective `\noaffiliation` should be executed now. Here, we do so in any case.

```
398 \@affil@none\move@AF
399 \move@AUAF
400 }%
401 \def\blankaffiliation{{}%}
```

`\@affil@cleared` Here are all the access procedures for the affiliation data structure. Note the similarity to those of the `\@author` data structure.

`\@affil@init` We define a flag value for `\@affil` (private), a procedure for setting `\@affil` to the flag value (`\@affil@init`), a Boolean to test against the flag value (`\@affilclear@sw`), a procedure `\@affil@def` to load a value into the structure, and a procedure to add new ancillary data to the affiliation.

```
\@affil@join@ 402 \def\@affil@cleared{{{}{}}}%
\@affil@join 403 \def\@affil@nil{{\relax}{}}%
404 \appdef\frontmatter@init{%
405 \@affil@init
406 }%
407 \def\@affil@none{%
408 \let\@affil\@affil@nil
409 }%
410 \def\@affil@init{%
411 \let\@affil\@affil@cleared
412 }%
413 \def\@affilclear@sw{\@ifx{\@affil\@affil@cleared}}%
414 \def\@affil@def#1{%
415 \def\@affil{#1}}%
416 }%
417 \def\@affil@join@#1#2#3{%
418 \def\@affil{#1}{\@join{\@separator}{#2}{#3}}%
419 }%
420 \def\@affil@join{\expandafter\@affil@join@\@affil}%}
```

`\move@AF` Move the most recent affiliation to the affiliation group. If we recently did a `\@affil@init`, this is a no-op.

Set the `\temp@sw` to false, then execute the affiliation list (`\@AFF@list`) with the intercolated operator set to `\@affil@match` and with `\@tempa` holding the first component of `\@affil`, the current affiliation, which is to be matched.

If the current affiliation has been seen before, then by side effect `\temp@sw` will be set to true, `\@tempc` will be set to the matching affiliation number, and `\@tempd` will be set to the affiliation's ancillary data. The Boolean `\@affils@sw` being false prevents this test from ever returning a "true" result.



This procedure uses \@tempa and sets \@tempc, \@tempd, and \@temp@sw.

```

421 \def\move@AF{%
422 \@affilclear@sw}{-%
423 \@booleanfalse\temp@sw
424 \let\@tempd\@empty
425 \@affils@sw{%
426   \expandafter\@affil@addr@def\expandafter\@tempa\@affil
427   \def\AFF@opr{\@affil@match\@tempa}%
428   \@AFF@list
429 }{\@temp@sw

```

True clause: This affiliation has been seen before. If ancillary data for the affiliation have been given, but are not identical to those seen before, give a warning.

```

430 {%
431   \expandafter\@affil@aux@def\expandafter\@tempb\@affil
432   \@ifx{\@tempb\@empty}{}{-%
433   \@ifx{\@tempb\@tempd}{}{-%
434   \class@warn{%
435     Ancillary information for \@tempa\space must not be different!
436     Please put all of it on the first instance%
437   }%
438 }%
439 }%
440 }%

```

False clause: This affiliation is a new one: increment the affiliation counter to get a unique affiliation ID and append the new ID, collaboration number, address, and auxiliary information to the list \@AFF@list.

Note that, with the expanded syntax of \@collaboration, we need to label each address with its associated collaboration.

Note also that the \@noaffiliation case is handled by recognizing the flag value \@affil@nil.

```

441 {%
442   \@ifx{\@affil\@affil@nil}{-%
443   \def\@tempc{0}%
444   \@argswap@val{0}%
445 }{%
446   \advance\c@affil\@one
447   \expandafter\def\expandafter\@tempc\expandafter{\the\c@affil}%
448   \expandafter\@argswap@val\expandafter{\the\c@affil}%
449 }%
450 {%
451   \expandafter\@argswap@val\expandafter{\the\c@collab}{-%
452   \expandafter\@argswap@val\expandafter{\@affil}{-%
453   \add@list@val@val@val\@AFF@list\AFF@opr
454   }%
455 }%
456 }%
457 }%

```

In either case, append this affiliation number to the `\AF@grp`, the affiliation group of the current author group. (It will later be considered for the `\@AFG@list`.) Then assign the flag value to `\@affil` that signifies it has been cleared.

```

458 \appdef@eval\AF@grp\@tempc
459 \frontmatterverbose@sw{\say\AF@grp}{-}%
460 \@affil@init
461 }%
462 }%
463 \def\@affil@addr@def#1#2#3{%
464 \def#1{#2}%
465 }%
466 \def\@affil@aux@def#1#2#3{%
467 \def#1{#3}%
468 }%

```

Procedure `\add@list@val@val@val` appends to the given list #1 using the given intercolated operator #2, with arguments #5 (affiliation ID) and #4 (collaboration ID) delimited by braces, and #3 bare. In the end, `\move@AF` may execute something like `\appdef \@AFF@list \AFF@opr {1} {3} {My Institution} {thanks}`.

```

469 \def\add@list@val@val@val#1#2#3#4#5{%
470 \appdef#1{#2{#5}{#4}#3}%
471 \frontmatterverbose@sw{\say#1}{-}%
472 }%

```

`\@affil@match` The procedure `\@affil@match` is the definition for `\AFF@opr`, the intercolated operator for `\@AFF@list`, when adding a new affiliation: it seeks a match with any earlier affiliation. When the affiliation list is executed, this procedure tests each entry against the ‘new’ affiliation (which has been stored in `\@tempa`) and its associated collaboration number (in `\@tempb`).

If `\groupauthors@sw` is true, then we require the collaboration ID to match, otherwise there is no such requirement, as is appropriate for the `superscriptaddress` class option.

Uses `\temp@sw`, `\@tempc`, and `\@tempd` to communicate back: if it returns `\temp@sw` true, a match was found. Then `\@tempc` will be the affiliation ID of the matching entry and `\@tempd` will be the auxiliary information of the matching entry.

```

473 \def\@affil@match#1#2#3#4#5{%
474 \temp@sw{-}%
475 \def\@tempifx{#4}%
476 \@ifx{\@tempifx#1}{-}%
477 \groupauthors@sw{%
478 \@ifnum{#3=\c@collab}{-}%
479 \true@sw
480 }{-}%
481 \false@sw
482 }%
483 }{-}%
484 \true@sw

```

```

485 }%
486 }-%
487 \false@sw
488 }%
489 {%
490 \@booleantrue\temp@sw
491 \def\@tempc{#2}%
492 \def\@tempd{#5}%
493 }-%
494 }%
495 }%
496 }%

```

`\move@AUAF` Append `\AF@grp`, `\AU@grp`, and `\CO@grp` to `\@AAC@list`, append `\CO@grp` to `\@AFG@list`, and merge any new `\AF@grp` to `\@AFG@list`.

The entire procedure is predicated on something non-trivial to move.

If both author group `\AU@grp` and affiliation group `\AF@grp` are nil, bail out.

No, try that again:

If we have seen any affiliations (`\AF@grp` is not nil), then it is time to move the current affiliation group and author group to `\@AAC@list`. If not, we are picking up authors into an author group, and we should bail out.

To extend this scheme to the new `\collaboration` abilities, we must append an affiliation `\AF@opr` to `\@AAC@list` also in the case where there was no affiliation specified (e.g., `\noaffiliation`). This affiliation will be a nil affiliation: `\AF@opr{0}`.

FIXME: only one of `\CO@grp` or `\AU@grp` is non-empty at this point, but this code does not appear to recognize this fact!

```

497 \def\move@AUAF{%
498 \frontmatterverbose@sw{\say\AU@grp\say\AF@grp\say\CO@grp}{}%
499 \@ifx{\AF@grp\@empty}{%
500 \@ifx{\@empty\CO@grp}{%

```

Both `\AF@grp` and `\CO@grp` are empty: nothing to do. But what if `\AU@grp` is not empty?

```

501 }-%

```

Append the nil `\AF@opr` to `\@AAC@list`.

```

502 \appdef \@AAC@list{\AF@opr{0}}%

```

Append the collaboration group to the `\@AAC@list`; no intercollated operator need be added, because we already have `\CO@opr`.

```

503 \appdef@e \@AAC@list{\CO@grp}%
504 \appdef@e \@AFG@list{\CO@grp}%
505 \let\CO@grp\@empty
506 }%
507 }-%

```

Append current affiliation group (expansion of `\AF@grp`) to the author/affiliation list (`\@AAC@list`), using `\AF@opr` as the intercollated operator; also append the current authors (`\AU@grp`).

```

508     \appdef     \@AAC@list{\AF@opr}%
509     \appdef@eval\@AAC@list{\AF@grp}%
510     \appdef@e   \@AAC@list{\AU@grp}%

```

Determine if either `\AU@grp` or `\CO@grp` is non-empty.

```

511     \@ifx{\@empty\AU@grp}{}%
512     \@ifx{\@empty\CO@grp}{%
513     }{%
514     \false@sw
515     }%

```

At this point, there is a Boolean in T<sub>E</sub>X's scanner; it will parse the following two brace-delimited sequences and select one of them.

```
516     {%
```

This branch is executed if and only if both `\AU@grp` and `\CO@grp` are empty.

```
517     }{%
```

There was either a `\author` or a `\collaboration` command. Append the current affiliation group to `\@AFG@list`, (only if it is not already there) using `\AFG@opr` as the intercolated operator.

Note that `\@AFG@list` is a list of *unique* affiliation *groups*, and building it entails an  $N^2$  computation.

```

518     \@booleanfalse\temp@sw
519     \def\AFG@opr{\x@match\AF@grp}%
520     \let\CO@opr\@author@gobble
521     \@AFG@list
522     \temp@sw}{}%
523     \appdef     \@AFG@list{\AFG@opr}%
524     \appdef@eval\@AFG@list{\AF@grp}%
525     }%

```

If the collaboration `\CO@grp` is not empty, we append it to the `\@AFG@list` and to the `\@AAC@list`.

```

526     \@ifx{\@empty\CO@grp}{}{%
527     \appdef@e   \@AAC@list{\CO@grp}%
528     \appdef@e   \@AFG@list{\CO@grp}%
529     \let\CO@grp\@empty
530     }%
531     }%

```

Finally, null out the macros holding the author group `\AU@grp`, affiliation group `\AF@grp`, and collaboration group `\CO@grp`.

```

532     \let\CO@grp\@empty
533     \let\AU@grp\@empty
534     \let\AF@grp\@empty
535     }%
536     \frontmatterverbose@sw{\say\@AAC@list\say\@AFG@list}{}%
537     }%

```

`\AU@grp` The control sequence name `\AU@grp` accumulates consecutive `\author` entries  
`\AF@grp` (with `\AU@opr` as the intercolated operator) and `\collaboration` entries (with  
`\@AAC@list`  
`\@AFG@list`  
`\@AFF@list`

`\CO@opr` as the intercolated operator). Ultimately, its contents are appended to the author/affiliation list `\@AAC@list` and to the list of affiliation groups, `\@AFG@list`. It must be initialized to `\@empty`.

Note on `\AU@opr` and `\CO@opr`: it is essential to treat these two operators to the greatest extent possible on an equal footing. Therefore we invariably assign values to the pair of them within the same procedure.

The macro `\AF@grp` accumulates affiliation IDs into an affiliation group. It must be empty at the beginning of the frontmatter.

The `\@AAC@list` macro accumulates authors (using `\AU@opr` as the intercolated operator), collaborations (using `\CO@opr` as the intercolated operator), and affiliations (using `\AF@opr` as the intercolated operator).

It must be empty at the beginning of the frontmatter.

The macro `\@AFG@list` accumulates affiliation groups, with `\AFG@opr` as the intercolated operator, and collaborations, with no intercolated operator. This token list is employed when and only when `groupedaddress` is in effect.

The macro `\@AFF@list` accumulates the list of affiliations with `\AFF@opr` as the intercolated operator. It must be empty at the beginning of the frontmatter.

Each `\affiliation` command in the document contributes to this list, as long as the argument of that instance of the command is “new”.

Each entry in this list consists of the `\AFF@opr` intercolated operator followed by three brace-delimited tokens, representing, in order:

1. the affiliation’s unique ID, a number increasing monotonically for each new entry,
2. the optional argument of the `\affiliation` command, the footnote to the affiliation, and
3. the text of the affiliation.

At various junctures in the code, the control sequence name `\AFF@opr` assumes the meaning of `\@affil@group`, `\affil@script`, or is defined to expand to `\@affil@match\@tempa`, and the list `\@AFF@list` is expanded.

For each of these lists, here is the syntax of an element and the usage of the associated operator:

```

%\AFF@list \AFF@opr{<affiliation ID>}{<collaboration ID>}{<address>}{<auxiliary info>} \frontmatter
%\AFG@list \AFG@opr{{<affiliation ID>}.{<affiliation ID>}} \frontmatter@author@produce@group
%\AAC@list \AF@opr{{<affiliation ID>}.{<affiliation ID>}} (\AU@opr|\CO@opr){<collaboration ID>}-
%
%\AFF@opr \@affil@match\@tempa \@affil@script \@affil@group
%\AFG@opr \affils@present@group \x@match\AF@grp
%\AF@opr \@gobble \@affilID@def \@affilID@count\AF@temp\@tempcnta \@affilID@match\AF@temp
%\AU@opr \@secondofthree \@author@count\@tempcnta \@author@present \@author@gobble \@author@count
%\CO@opr \@secondofthree \@collaboration@count\AU@temp\@tempcnta \@collaboration@present\AU@temp
%
%\CO@opr\@author@cleared
%
```

```

538 \appdef\frontmatter@init{%
539 \let\AU@grp\@empty
540 \let\CO@grp\@empty
541 \let\AF@grp\@empty
542 \let\@AAC@list\@empty
543 \let\@AFG@list\@empty
544 \let\@AFF@list\@empty
545 }%

```

`\@AF@join` The `\csname \@AF@join` is a procedure used within `\email`, `\homepage`, `\thanks`, and `\altaffiliation`. When such a command appears after an `\author`, `\collaboration`, or `\affiliationcommand`, the `\@AF@join` procedure appends the argument given to the appropriate macro. Its default is to give an error message, since these commands are legal only within the particular context mentioned.

```

546 \appdef\frontmatter@init{%
547 \let\@AF@join\@AF@join@error
548 }%
549 \def\@AF@join@error#1{%
550 \class@warn{%
551 \string\email, \string\homepage, \string\thanks, or \string\altaffiliation\space
552 appears in wrong context.
553 }%
554 }%

```

### 9.3.3 Commands for auxiliary information

`\email` Just tacks the email address on to the current author or affiliation.

```

555 \def\sanitize@url{%
556 \@makeother\%%
557 \@makeother\~%
558 \@makeother\_%
559 }%
560 \newcommand*\email[1] [] {\begingroup\sanitize@url\@email{#1}}%
561 \def\@email#1#2{%
562 \endgroup
563 \@AF@join{#1\href{mailto:#2}{#2}}%
564 }%

```

`\homepage` Just tacks the URL on to the current author or affiliation. Note: group opened in `\homepage` is closed in `\@homepage`.

```

565 \newcommand*\homepage[1] [] {\begingroup\sanitize@url\@homepage{#1}}%
566 \def\@homepage#1#2{%
567 \endgroup
568 \@AF@join{#1\href{#2}{#2}}%
569 }%

```

`\href` The document may load a package that defines either of these commands; if not, we give a default meaning.

```

570 \appdef\class@documenthook{%

```

```
571 \providecommand\href[1]{}%
572 }%
```

`\thanks` The operative version of `\thanks` appends an item to `\@author`, or `\@affil`'s auxiliary data.

```
573 \def\frontmatter@thanks{% implicit #1
574 \@AF@join
575 }%
```

`\altaffiliation` Implemented more or less like `\thanks` but shares the affiliation counter. Optional argument may be used to give explanatory text eg 'currently staying at' This will be placed before the address, if used in a footnote.

```
576 \newcommand*\altaffiliation[2] []{%
577 \@AF@join{#1#2}%
578 }%
```

### 9.3.4 Procedures for author, collaboration, and affiliation

Macros that list off, say, authors, and which require punctuation, like "A, B, and C", will use `\set@listcomma@list`, `\set@listcomma@count`, `\@listand`, and `\@listcomma`. These macros use the `\@tempcnta` register to keep track of how many items remain to be listed off.

At present, `\@author@present` will decrement that counter.

`\set@listcomma@list` Pop the author count for this collaboration.

`\set@listcomma@count` Note: by side effect, it assigns `\@listcomma`: for a list of length two, suppress comma addition.

```
579 \def\set@listcomma@list#1{%
580 \expandafter\@reset@ac\expandafter#1#1{0}\@reset@ac{%
581 \let\@listcomma\relax
582 }{%
583 \let\@listcomma\@listcomma@comma
584 }%
585 }%
586 \def\set@listcomma@count#1{%
587 \ifnum{#1}=\tw@}%
588 \let\@listcomma\relax
589 }{%
590 \let\@listcomma\@listcomma@comma
591 }%
592 }%
```

Does the actual pop operation, then generates a Boolean which selects one of the two assignments for `\@listcomma`.

Note: this procedure sets `\@tempcnta` to reflect the number of items to list off.

```
593 \def\@reset@ac#1#2#3\@reset@ac{%
594 \def#1{#3}%
595 \@tempcnta#2\relax
596 \ifnum{#2}=\tw@}%
```

597 }%

`\listand` Might need extending with penalties etc.

Note: this procedure expects `\@tempcnta` to reflect the current number of items remaining to list off.

```
598 \def\listand{\ifnum{\@tempcnta=\tw@}{\andname\space}{}}%
```

`\@listcomma` This definition, with `\@ne` puts a comma before and.

David Carlisle says: In UK English (at least) would have `\tw@` here, which would then implement the convention of indicating conjunction with comma or “and”, but not both.

Note: this procedure expects `\@tempcnta` to reflect the current number of items remaining to list off.

```
599 \def\@listcomma@comma{\ifnum{\@tempcnta>\@ne}{,}{}}%
```

```
600 \def\@listcomma@comma@UK{\ifnum{\@tempcnta>\tw@}{,}{}}%
```

`\@collaboration@gobble` This command will be the synonym of `\C0@grp` during the expansion of `\@AAC@list`, within `\affils@present@grouperexecution` (in a `groupedaddress` situation).

```
601 \def\@collaboration@gobble#1#2#3{}}%
```

`\doauthor` Main control over how authors are typeset.

`#1` is loaded by `\author`

`#2` is loaded by `\email`, `\homepage`, `\altaffiliation`, or `\thanks`

`#3` is the superscript affiliation, if at all used.

First, the author name is formatted, followed by a comma, then come any marks relating to affiliation (if present), then come the homepage URL and email address, if any, with appropriate punctuation.

Here, as in `\@affil@script`, the parameter `\affil@cutoff` controls whether the argument `#3` is produced as a superscript label connecting this author with its affiliation. (If there is only one affiliation on the title page, it makes sense to remove the superscript.)

Note that argument `#3` must effectively execute either `\aftergroup>true@sw` or `\aftergroup>false@sw` (exactly once!). In the case of `superscriptaddress`, this is done when `\@author@present@script` invokes `\doauthor` with its third argument containing `\@affil@present@script`. Otherwise (e.g., `groupedaddress`), `\doauthor` is invoked by `\@author@present@group` with its third argument containing `\@affil@present@group`.

```
602 \def\doauthor#1#2#3{%
```

```
603 \ignorespaces#1\unskip\@listcomma
```

```
604 \begingroup
```

```
605 #3%
```

At this point, we must have queued up a Boolean (either `\true@sw` or `\false@sw`).

```
606 \@ifempty{#2}{\endgroup}{}}{\endgroup{\comma@space}}{\frontmatter@footnote{#2}}%
```

```
607 \space \@listand
```

```
608 }%
```



`\x@match` Procedure `\x@match` is used as an alias `\AFG@opr`, probing the list of affiliation groups and setting a Boolean if a match is found to #1 (usually the current affiliation group).

Procedure `\y@match` is used as an alias to `\@TBN@opr`, probing the list of unique title block footnotes for a match to the footnote text: when the first match is found it sets a Boolean, defining `\@tempb` to the number of that footnote.

```

609 \def\x@match#1#2{%
610   \temp@sw{}{%
611     \def\@tempifx{#2}%
612     \@ifx{\@tempifx#1}{%
613       \@booleantrue\temp@sw
614     }{%
615     }%
616   }%
617 }%
618 \def\y@match#1#2#3{%
619   \temp@sw{}{%
620     \def\@tempifx{#3}%
621     \@ifx{\@tempifx#1}{%
622       \@booleantrue\temp@sw
623     \def\@tempb{#2}%
624   }{%
625   }%
626 }%
627 }%

```

`\frontmatter@footnote` Used when typesetting a title block footnote. (Item 290: merge duplicates.)

Maintains a list of frontmatter footnotes along with associated device. Uniquifies each footnote by traversing the list and looking for a match to the footnote text. If found, define `\@tempb` to the corresponding footnote device, and typeset it with `\@footnotemark`. If never found, manually increment footnote counter, determine the corresponding footnote device, and add footnote text and device to the list of frontmatter footnotes.

The list `\@FMN@list` is processed later in the title block production.

Note that this method of making footnotes runs afoul of `hyperref`'s patches of the L<sup>A</sup>T<sub>E</sub>X kernel. Therefore, we avoid trouble by refraining from using `\@footnotemark` and by creating our own hypertext links.

```

628 \def\frontmatter@footnote#1{%
629   \begingroup
630   \@booleanfalse\temp@sw
631   \def\@tempa{#1}%
632   \let\@tempb\@empty
633   \def\@TBN@opr{\y@match\@tempa}%
634   \@FMN@list

```

By side effect, `\@tempb` holds the value of the footnote counter if This note has been seen before. Otherwise `\temp@sw` is false.

```

635   \temp@sw{}%

```

```

636 \expandafter\frontmatter@footnotemark
637 \expandafter{\@tempb}%
638 }{%

```

Execute `\frontmatter@foot@mark`, feeding it the value of the current footnote counter. This call requires two levels of pre-expansion to accomplish, the first to expand `\csname`, the second to evaluate `\the`.

```

639 \stepcounter\@mpfn
640 \expandafter\expandafter
641 \expandafter\frontmatter@foot@mark
642 \expandafter\expandafter
643 \expandafter{%
644 \expandafter \the\csname c@\@mpfn\endcsname
645 }{\#1}%
646 }%
647 \endgroup
648 }%

```

Procedure `\frontmatter@foot@mark` lays down the footnote device (mark) and takes care of the footnote text. The latter merely involves queueing that text (along with its associated device) into `\@FMN@list`.

```

649 \def\frontmatter@foot@mark#1#2{%
650 \frontmatter@footnotemark{\#1}%
651 \g@addto@macro\@FMN@list{\@TBN@opr{\#1}{\#2}}%
652 }%
653 \appdef\frontmatter@init{%
654 \global\let\@FMN@list\@empty
655 }%

656 \def\frontmatter@footnotemark#1{%
657 \leavevmode
658 \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
659 \begingroup
660 \hyper@linkstart {link}{\frontmatter.#1}%
661 \csname c@\@mpfn\endcsname#1\relax
662 \def\@thefnmark{\frontmatter@thefootnote}%
663 \@makefnmark
664 \hyper@linkend
665 \endgroup
666 \ifhmode\spacefactor\@x@sf\fi
667 \relax
668 }%

```

## 9.4 The keywords command

```

\keywords Usual game, save text in a macro for processing by \maketitle.
\@keywords 669 \def\keywords#1{%
670 \aftermaketitle@chk{\keywords}%
671 \gdef\@keywords{\#1}%
672 }%

```

```

673 \appdef\frontmatter@init{%
674 \let\@keywords\@empty
675 }%

```

## 9.5 The `\date` command and related commands

`\date` Use the `\date` command to specify the document date, the `\received` command to specify the date received, `\revised` for date revised, `\accepted` for date accepted, `\revised` and `\published` for date published. Normally only `\date` to be used by author, `\accepted` the remainder used only by editors.

`\published` DPC: As for `\keywords`, but this time don't flag a warning if there is no revision date specified.

`\@date` In each case the user-level command defines the value of a macro which buffers the data entered by the user. For instance, `\received` causes `\@received` to acquire a value. The optional argument allows the user to override the text that will be typeset along with the date; the default value of that text is itself a localized macro.

```

676 \newcommand*\frontmatter@date[2][\Dated@name]{\def\@date{#1#2}}%
677 \def\@date{}%
678 \newcommand*\received[2][\Received@name]{\def\@received{#1#2}}%
679 \def\@received{}%
680 \newcommand*\revised[2][\Revised@name]{\def\@revised{#1#2}}%
681 \def\@revised{}%
682 \newcommand*\accepted[2][\Accepted@name]{\def\@accepted{#1#2}}%
683 \def\@accepted{}%
684 \newcommand*\published[2][\Published@name]{\def\@published{#1#2}}%
685 \def\@published{}%

```

## 9.6 The `\pacs` command

PACS, the Physics and Astronomy Classification Scheme.

```

\pacs As for \keywords
\@pacs
686 \def\pacs#1{%
687 \aftermaketitle@chk{\pacs}%
688 \gdef\@pacs{#1}%
689 }%
690 \appdef\frontmatter@init{%
691 \let\@pacs\@empty
692 }%

```

## 9.7 The `\preprint` command

```

693 \def\preprint#1{\gappdef\@preprint{\preprint{#1}}}%
694 \appdef\frontmatter@init{%
695 \let\@preprint\@empty
696 }%

```

## 9.8 The abstract environment

`\absbox`

```
697 \newbox\absbox
```

**abstract** Abstract, as in AMS classes, must be specified *before* `\maketitle`. It just saves everything up in `\absbox`. Note that the specifics of how the abstract is to be formatted are set by `\frontmatter@abstractwidth`, `\frontmatter@abstractheading`, and `\frontmatter@abstractfont`.

Here we wish to set the abstract into type but save it away in a box, much like the `\minipage` command does.

Note that the `\endabstract` portion of the environment begins with code that mostly duplicates `\endminipage`, but without the `\@iiparbox`. At the end, we simply transfer the contents of the `\minipage` into a box of our own.

However, we also have to end the gratuitous paragraph begun by `\minipage`. We ensure that no trace of that paragraph is left, by absorbing the `\parindent` box. Doing so destroys the paragraph entirely, except for the `\parskip` glue, and that morsel is pruned, because we are at the top of a page.

This mechanism is vulnerable, however, because anything placed into the horizontal list after the `\parindent` box will confound it. And that is exactly what happens under MikTeX: There seems to be an extension to the TeX standard operating here which inserts a `\special` at the beginning of every paragraph. (The mechanism is *not* based on `\everypar`.)

To accommodate this state of affairs, we have a new mechanism that offers an even more complete rewrite of `minipage`.

Note that an entry is made in the toc for the abstract; this has the side effect of creating a PDF bookmark for this element if the `hyperref` package is in effect.

```
698 \def\toclevel@abstract{1}%
699 \def\addcontents@abstract{%
700   \phantomsection
701   \expandafter\def\csname Parent0\endcsname{section*.2}%
702   \expandafter\@argswap@val\expandafter{\abstractname}{\addcontentsline{toc}{abstract}}%
703 }%
704 \newenvironment{frontmatter@abstract}{%
705   \aftermaketitle@chk{\begin{abstract}}%
706   \global\setbox\absbox\vbox\bgroup
707   \color@begingroup
708   \columnwidth\textwidth
709   \hsize\columnwidth
710   \@parboxrestore
```

The following line switches footnotes to the `mpfootnote` series. This action should rather be taken by `\frontmatter@abstractfont`, which can be customized. There is also an interaction with `\mini@note`.

```
711   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
712   \let\@footnotetext\frontmatter@footnotetext
713   \minipagefootnote@init
```

```

714 \let\set@listindent\set@listindent@
715 \let\@listdepth\@mplistdepth \@mplistdepth\z@
716 \let@environment{description}{frontmatter@description}%
717 \@minipagerestore
718 \@setminipage
719 \frontmatter@abstractheading
720 \frontmatter@abstractfont
721 \let\footnote\mini@note
722 \expandafter\everypar\expandafter{\the\everypar\addcontents@abstract\everypar{}}%
723 }{%
724 \par
725 \unskip
726 \minipagefootnote@here
727 \@minipagefalse %% added 24 May 89
728 \color@endgroup
729 \egroup
730 }%

```

`\frontmatter@footnotetext` We reimplement `\@mpfootnotetext` under a new name (so as not to be overridden by the `hyperref` package) and extend it to accomodate hyperrefs.

Note that this procedure is very like `ltxutil`'s `\mp@footnotetext`, except that it uses `\frontmatter@makefntext` instead of `\@makefntext`.

```

731 \long\def\frontmatter@footnotetext#1{%
732 \minipagefootnote@pick
733 \set@footnotefont
734 \set@footnotewidth
735 \@parboxrestore
736 \protected@edef\@currentlabel{\csname p@\@mpfn\endcsname\@thefnmark}%
737 \color@begingroup
738 \frontmatter@makefntext{%
739 \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox\vadjust{\vskip\z@skip}%
740 }%
741 \color@endgroup
742 \minipagefootnote@drop
743 }%

```

`\ltx@no@footwarn`

```

744 \def\ltx@no@footnote{%
745 \let\ltx@xfootnote\ltx@no@xfootnote\let\ltx@yfootnote\ltx@no@yfootnote
746 \let\ltx@xfootmark\ltx@no@xfootmark\let\ltx@yfootmark\ltx@no@yfootmark
747 \let\ltx@xfoottext\ltx@no@xfoottext\let\ltx@yfoottext\ltx@no@yfoottext
748 }%
749 \def\ltx@no@xfootnote[#1]#2{\ltx@no@footwarn\footnote}%
750 \def\ltx@no@yfootnote#1{\ltx@no@footwarn\footnote}%
751 \def\ltx@no@xfootmark[#1]{\ltx@no@footwarn\footnotemark}%
752 \def\ltx@no@yfootmark{\ltx@no@footwarn\footnotemark}%
753 \def\ltx@no@xfoottext[#1]#2{\ltx@no@footwarn\footnotetext}%
754 \def\ltx@no@yfoottext#1{\ltx@no@footwarn\footnotetext}%
755 \def\ltx@no@footwarn#1{%

```

```

756 \class@warn{%
757   The \string#1\space command is not legal on the title page;
758   using \string\thanks\space instead might suit you: consult the manual for details%
759 }%
760 }%

```

`\frontmatter@abstractheading` The default abstract head; journals will override this procedure.

```

761 \def\frontmatter@abstractheading{%
762   \begingroup
763   \centering\large
764   \abstractname
765   \par
766 \endgroup
767 }%

```

`\frontmatter@abstractfont` The default type specification for the body of the abstract. Journals will override this setting.

```

768 \def\frontmatter@abstractfont{}%

```

`frontmatter@description` Within the abstract, the description environment is defined as follows:

```

769 \newenvironment{frontmatter@description}{%
770   \list{}{%
771     \leftmargin\z@
772     \labelwidth\z@

% \itemindent-\leftmargin
%

773   \itemindent\z@
774   \let\makelabel\frontmatter@descriptionlabel
775 }%
776 }{%
777 \endlist
778 }%
779 \def\frontmatter@descriptionlabel#1{%
780   \hspace\labelsep
781   \normalfont\bfseries
782   #1:%
783 }%

```

`\frontmatter@abstractwidth` The default setting is the full text width; journals can override this setting.

```

784 \def\frontmatter@abstractwidth{\textwidth}

```

`\frontmatter@abstract@produce` This procedure determines how the abstract is incorporated into the title block. We split this out in anticipation of the need to lift the limitation that the abstract not break over pages.

If we are in preprint style, we provide an easy pagebreak point immediately above the abstract. This means that the abstract will either fit entirely on the

same page as the title block, or it will start a page of its own (which may itself break onto a subsequent page if necessary).

```

785 \def\frontmatter@abstract@produce{%
786   \par
787   \preprintsty@sw{%
788     \do@output@MVL{%
789       \vskip\frontmatter@preabstractspace
790       \vskip200\p@\@plus1fil
791       \penalty-200\relax
792       \vskip-200\p@\@plus-1fil
793     }%
794   }{%
795     \addvspace{\frontmatter@preabstractspace}%
796   }%
797   \begingroup
798     \dimen@ \baselineskip
799     \setbox\z@\vtop{\unvcopy\absbox}%
800     \advance\dimen@-\ht\z@\advance\dimen@-\prevdepth
801     \@ifdim{\dimen@>\z@}{\vskip\dimen@}{}%
802   \endgroup
803   \begingroup
804     \prep@absbox
805     \unvbox\absbox
806     \post@absbox
807   \endgroup
808   \@ifx{\@empty\mini@notes}{\mini@notes\par}%
809   \addvspace\frontmatter@postabstractspace
810 }%
811 \appdef\frontmatter@init{\let\mini@notes\@empty}%
812 \let\prep@absbox\@empty
813 \let\post@absbox\@empty

```

```

\frontmatter@preabstractspace Space above and space below abstract in title block
\frontmatter@postabstractspace 814 \def\frontmatter@preabstractspace{.5\baselineskip}
815 \def\frontmatter@postabstractspace{.5\baselineskip}

```

## 9.9 Formatting the title block

`titlepage` This is L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>'s article class version, with modifications.

```

816 \newenvironment{frontmatter@titlepage}{%
817   \twocolumn@sw{\onecolumngrid}{\newpage}%
818   \thispagestyle{titlepage}%
819   \setcounter{page}\@ne
820 }{%
821   \twocolumn@sw{\twocolumngrid}{\newpage}%
822   \twoside@sw{}{%
823     \setcounter{page}\@ne
824   }%
825 }%

```

`\maketitle` Put it all together to format the title block.

Gotcha! If you expand `\csnameundefined\endcsname`, you change the meaning of `\undefined` from “Undefined” to `\relax`. Watchout!

```
826 \def\frontmatter@maketitle{%
827   \@author@finish
828   \title@column\titleblock@produce
829   \suppressfloats[t]%
      Now save some memory.
830   \let\and\relax
831   \let\affiliation@gobble
832   \let\author@gobble
833   \let@AAC@list@empty
834   \let@AFF@list@empty
835   \let@AFG@list@empty
836   \let@AF@join@AF@join@error
837   \let@email@gobble
838   \let@address@empty
839   \let\maketitle\relax
840   \let\thanks@gobble
841   \let\abstract@undefined\let\endabstract@undefined
842   \titlepage@sw{%
843     \vfil
844     \clearpage
845   }{%
846 }%
```

`\maketitle@Hy` We provide the means to take back the definition of `\maketitle` from `hyperref`: we do not need its help in providing hypertext services in the title page.

```
847 \def\maketitle@Hy{%
848   \let\Hy@saved@footnotemark@footnotemark
849   \let\Hy@saved@footnotetext@footnotetext
850   \let@footnotemark\H@footnotemark
851   \let@footnotetext\H@footnotetext
852   \ifnextchar [%]
853     \Hy@maketitle@optarg
854     {%
855       \HyOrg@maketitle
856       \Hy@maketitle@end
857     }%
858 }%
859 \appdef\class@documenthook{%
860   \ifx{\maketitle\maketitle@Hy}{%
861     \class@info{Taking \string\maketitle\space back from hyperref}%
862     \let\maketitle\frontmatter@maketitle
863   }{%
864   }%
865 }%
```

`\titleblock@produce` This procedure produces the title block.



It effectively executes inside a group and always returns us to vertical mode.

Note: we assume that it is OK to set the footnote counter to zero at this point.

Is this a safe assumption?

```
866 \def\titleblock@produce{%  
867 \begingroup
```

First, we restore the footnote mechanism to its default state, then we customize the way the footnote mark is produced and how it is formatted.

```
868 \ltx@footnote@pop
```

The following three lines establish an independent footnote counter for use in the title block.

```
869 \def\@mpfn{mpfootnote}%  
870 \def\thempfn{\thempfootnote}%  
871 \c@mpfootnote\z@  
872 \let\@makefnmark\frontmatter@makefnmark  
873 \frontmatter@setup  
874 \thispagestyle{titlepage}\label{FirstPage}%
```

Produce the title:

```
875 \frontmatter@title@produce
```

Produce the author list:

```
876 \groupauthors@sw{%  
877 \frontmatter@author@produce@group  
878 }{%  
879 \frontmatter@author@produce@script  
880 }%
```

Produce the dates:

```
881 \frontmatter@RRAPformat{%  
882 \expandafter\produce@RRAP\expandafter{\@date}%  
883 \expandafter\produce@RRAP\expandafter{\@received}%  
884 \expandafter\produce@RRAP\expandafter{\@revised}%  
885 \expandafter\produce@RRAP\expandafter{\@accepted}%  
886 \expandafter\produce@RRAP\expandafter{\@published}%  
887 }%
```

Produce the abstract, PACS, and keywords, and end any paragraph.

```
888 \frontmatter@abstract@produce  
889 \@ifx@empty\@pacs{}{%  
890 \@pacs@produce\@pacs  
891 }%  
892 \@ifx@empty\@keywords{}{%  
893 \@keywords@produce\@keywords  
894 }%  
895 \par  
896 \frontmatter@finalspace  
897 \endgroup  
898 }%
```

`\frontmatter@title@produce` The default formatting procedure for the article title. This procedure should take care of the vertical space below the title.

Clients may override this procedure, but it is more likely that they will use the hooks `\frontmatter@title@above`, `\frontmatter@title@format`, and `\frontmatter@title@below`.

```

899 \def\toclevel@title{0}%
900 \def\frontmatter@title@produce{%
901   \begingroup
902   \frontmatter@title@above
903   \frontmatter@title@format
904   \@title
905   \unskip
906   \phantomsection\expandafter\@argswap@val\expandafter{\@title}{\addcontentsline{toc}{title}}%
907   \@ifx{\@title@aux\@title@aux@cleared}{-}{%
908     \expandafter\frontmatter@footnote\expandafter{\@title@aux}%
909   }%
910   \par
911   \frontmatter@title@below
912 \endgroup
913 }%
914 \appdef\let@mark{\let\\relax}%

```

`\frontmatter@title@above` The default values for formatting specifications for the article title.

`\frontmatter@title@format` The procedure `\frontmatter@title@above` should take care of the vertical space above the title; `\frontmatter@title@below` below. The procedure `\frontmatter@title@format` should invoke any font switches, etc. that may apply to the title.

Journals will override this procedure.

```

915 \def\frontmatter@title@above{}%
916 \def\frontmatter@title@format{}%
917 \def\frontmatter@title@below{\addvspace{\baselineskip}}%

```

### 9.9.1 Authors and affiliations in superscriptaddress style

`\frontmatter@author@produce@script` `\frontmatter@author@produce@script` is an alias for `\frontmatter@author@produce`, the procedure that formats the author/affiliation list.

In this case, the authors affiliations are being superscripted (class option `superscriptaddress`).

This procedure must ensure that any paragraph that it starts gets ended finally.

```

918 \def\frontmatter@author@produce@script{%
919   \begingroup
920   \let\@author@present\@author@present@script
921   \frontmatterverbose@sw{\typeout{\string\frontmatter@author@produce@script:}}\say\@AAC@list\s

```

When the author/affiliation list `\@AAC@list` is executed, `\@tempcnt` counts each author, and `\@tempa` stores a list of author indices for the current collaboration.

Note: this procedure uses `\AU@temp` to hold a list of author counts for each collaboration. Note: also uses `\@tempcnta` to communicate between procedures. This usage of `\@tempcnta` is OK, because the expansion of `\@AAC@list` will trigger only the execution of our own procedures.

```
922 \let\AU@temp\@empty
923 \@tempcnta\z@
```

We wish to know how many authors are in each collaboration, for the purposes of `\listcomma` and `\listand`.

Here we assign values for the intercolated operators appearing within `\@AAC@list`, then execute that macro, registering the authors in each collaboration. Afterwards, clean up by emulating an empty collaboration.

The result, stored in `\AU@temp`, is a list of brace-delimited tokens, each a number representing the number of authors in that collaboration.

```
924 \let\AF@opr \@gobble
925 \def\AU@opr{\@author@count\@tempcnta}%
926 \def\CO@opr{\@collaboration@count\AU@temp\@tempcnta}%
927 \@AAC@list
928 \expandafter\CO@opr\@author@cleared
```

Next, present the authors and their affiliations, with the collaborations interleaved.

We assign values for the intercolated operators appearing within `\@AAC@list`. Next, assign `\@listcomma` based on the number of authors in the first collaboration. Then expand `\@AAC@list`.

Upon completion, there will be two space tokens following the last author because of `\listand` processing; remove them, then end the last paragraph.

FIXME: this juncture would be a good time to effectively eliminate the unused affiliations in `\@AFF@list`.

```
929 \begingroup
930 \frontmatter@authorformat
931 \let\AF@opr \@affilID@def
932 \let\AU@opr \@author@present
933 \def\CO@opr{\@collaboration@present\AU@temp}%
934 \set@listcomma@list\AU@temp
935 \@AAC@list
936 \unskip\unskip
937 \par
938 \endgroup
```

We now list out the affiliations in the order they appeared.

```
939 \begingroup
940 \frontmatter@above@affiliation@script
941 \let\AFF@opr \@affil@script
942 \@AFF@list
943 \frontmatter@footnote@produce
944 \par
945 \endgroup
946 \endgroup
947 }%
```

`\@author@count` This version of `\AU@opr` counts the number of authors it processes.

```

948 \def\@author@count#1{%
949 \advance#1\@ne
950 \@author@gobble
951 }%
```

`\@collaboration@present` Format a collaboration. Note that we immediately end the paragraph thus begun, because we only support `superscriptaddress`-style processing.

Note also that the execution of the production procedures takes place inside of a group; for reasons of protective programming.

Bug note: originally, the number of authors in each collaboration was stored in `\@tempa`, but this exposed us to procedures in our production that modified the meaning of `\@tempa`.

The last action done in this procedure is to assign `\@listcomma` based on the number of authors in the next collaboration.

```

952 \def\@collaboration@present#1#2#3#4{%
953 \par
954 \begingroup
955 \frontmatter@collaboration@above
956 \@affilID@def{ }%
957 \@tempcnta\z@
958 \@author@present{ }{\@ignorespaces#3\unskip}{#4}%
959 \par
960 \endgroup
961 \set@listcomma@list#1%
962 }%
963 \def\frontmatter@collaboration@above{ }%
```

`\@collaboration@count` Simply register each author in this collaboration. Note: `#1` is a `\csname` to hold the value, `#2` is the `\count` register holding the count of interest.

```

964 \def\@collaboration@count#1#2{%
965 \appdef@eval#1{\the#2}#2\z@
966 \@author@gobble
967 }%
```

`\@affilID@def` The `\@affilID@def` procedure, an alias of `\AF@opr` used during `\frontmatter@author@produce@script` processing, memorizes the affiliation in `\@affilID@temp`.

```

968 \def\@affilID@def{\def\@affilID@temp}
```

`\@affilID@temp` The macro `\@affilID@temp` is used to communicate between `\doauthor` and instances of `\AF@opr` within the author list.

```

969 \let\@affilID@temp\@empty
```

`\affils@present@script` An alias of `\AFG@opr`, this procedure is applied to elements in `\@AFG@list`. It builds `\@AFU@list`, a list of unique affiliations found within affiliation groups.

```

%\def\affils@present@script#1{%
% \get@affil#1{ }%
```

```

}%%
%\def\get@affil#1{%
% \def\@tempa{#1}%
% \ifx{\@empty\@tempa}{-}{%
% \booleanfalse\temp@sw
% \def\AF@opr{\x@match\@tempa}%
% \@AFU@list
% \temp@sw}{-}{%
% \appdef\@AFU@list{\AF@opr{#1}}%
% }%
% \get@affil
% }%
}%%
%
```

`\affil@present@script` An alias of `\AF@opr`, this procedure is applied to `\@AFU@list` when formatting the affiliations under `superscriptaddress`. It in turn applies `\affil@script` to `\@AFF@list`, thereby formatting the affiliation associated with this affiliation index.

Note: it traverses the `\@AFF@list` for each entry in the `\@AFU@list`, thereby making this portion of code execute in  $N^2$  time.

```

%\def\affil@present@script#1{%
% \def\@tempa{#1}%
% \begingroup
% \frontmatter@affiliationfont
% \let\AFF@opr \affil@script
% \@AFF@list
% \endgroup
}%%
%
```

`\affil@script` Alias of `\AFF@opr`, this procedure is applied to `\@AFF@list` when formatting the affiliations under `superscriptaddress`. It rejects all entries not matching the affiliation index held in `\@tempa`.

```

970 \def\affil@script#1#2#3{%
971 \def\@tempifx{#1}\@ifx{\@tempifx\@tempa}{%
972 \if@empty{#2}{-}{%
973 \par
974 \begingroup
975 \def\@thefnmark{#1}\@makefnmark\ignorespaces
976 #2%
977 \@if@empty{#3}{-}{\frontmatter@footnote{#3}}%
978 \par
979 \endgroup
980 }%
981 }-}%
982 }%
```

`\@affil@script` Alias of `\AFF@opr`: The affiliations are being formatted in the case where affiliations are being superscripted: make a list out of the affiliations with the numbers printed. Society-specific code can change the formatting by overriding the definition of `\frontmatter@affiliationfont`.

A client may choose to set `\affil@cutoff` to either `\@ne` (the default) or `\tw@`. The latter setting will enforce a rule to the effect that if a single affiliation appears on the title page, then no affiliation superscript at all is produced.

`#1`—Affiliation ID: a number; zero signifies a `\noaffiliation`.

`#2`—Collaboration ID

`#3`—Affiliation address

`#4`—Auxiliary info

```
983 \def \@affil@script#1#2#3#4{%
```

If the affiliation is the `\noaffiliation`, then we do nothing.

```
984 \@ifnum{#1=\z@}{-}{%
```

```
985 \par
```

```
986 \begingroup
```

```
987 \frontmatter@affiliationfont
```

```
988 \@ifnum{\c@affil<\affil@cutoff}{-}{%
```

```
989 \def \@thefnmark{#1}\@makefnmark
```

```
990 }%
```

```
991 \ignorespaces#3%
```

```
992 \@ifempty{#4}{-}{\frontmatter@footnote{#4}}%
```

```
993 \par
```

```
994 \endgroup
```

```
995 }%
```

```
996 }%
```

```
997 \let \@affil@cutoff \@ne
```

`\@author@present@script` This version of `\AU@opr` applies to the `superscriptaddress` class option. Need to add commas between groups of address numbers, which are passed in the macro `\@affilID@temp` in the form `{3}{4}{7}` if this set of authors is related to addresses, 3, 4 and 7.

```
998 \def \@author@present@script#1#2#3{%
```

```
999 \begingroup
```

```
1000 \gdef \comma@space{\textsuperscript{,\,}}%
```

```
1001 \doauthor{#2}{#3}{\@affil@present@script}%
```

```
1002 \endgroup
```

```
1003 \advance \@tempcnta \m@ne
```

```
1004 }%
```

```
1005 \def \@affilcomma#1#2{%
```

```
1006 \@ifx{\z@#1}{-}{%
```

```
1007 \@ifx{\relax#2}{-}{%
```

```
1008 \@affilcomma{#2}%
```

```
1009 }%
```

```
1010 }{%
```

```
1011 #1%
```

```
1012 \@ifx{\relax#2}{-}{%
```

```
1013 \@ifx{\z@#2}{-}{%
```

```

1014   \@affilcomma
1015   }{%
1016   ,\,\@affilcomma{#2}%
1017   }%
1018   }%
1019   }%
1020   }%

```

`\@affil@present@script` The `\@affil@present@script` procedure presents the reference to the affiliations in the case of superscript addresses.

The affiliations are presented to `\@affil@present@script` as a list of brace-delimited tokens, each containing the affiliation ID (a number). The list is terminated by the token `\relax`. The macro recurses (via tail recursion) until the terminating token is encountered. Note that an affiliation ID of zero signifies a `\noaffiliation`: references to the `\noaffiliation` are passed over.

Upon termination, the affiliations, if any, are produced via `\textsuperscript` and the appropriate boolean is queued via `\aftergroup`. The boolean signifies that a non-trivial affiliation was produced, so that the proper punctuation can be produced. The group in question is that opened and closed by the `\doauthor` procedure.

This procedure uses `\@tempa` as a scratch register.

```

1021 \def\@affil@present@script{%
1022   \let\@tempa\@empty
1023   \expandafter\@affil@present@script@\@affilID@temp\relax
1024 }%
1025 \def\@affil@present@script@#1{%
1026   \@ifx{\relax#1}{%
1027     \@ifx{\@tempa\@empty}{%
1028       \aftergroup\false@sw
1029     }{%
1030       \textsuperscript{\expandafter\@affilcomma\@tempa\relax\relax}%
1031       \aftergroup\true@sw
1032     }%
1033   }{%
1034     \@ifnum{#1=\z@}{\@appdef\@tempa{#1}}}%
1035   \@affil@present@script@
1036 }%
1037 }%
1038 \@provide\@author@parskip{\z@skip}%

```

### 9.9.2 Authors and affiliations in `groupedaddress` style

`\frontmatter@author@produce@group` Alias for `\frontmatter@author@produce` when `groupedaddress` is in effect, the procedure that formats the author/affiliation list.

In this case, the authors are being grouped above their shared addresses. How it works: the `\@AFG@list`, a list of all unique affiliation groups, is expanded. For each such group, all authors belonging to that group are formatted above the affiliation group. Collaborations are silently passed over.

This procedure must ensure that any paragraph that it starts gets ended finally.

```

1039 \def\frontmatter@author@produce@group{%
1040   \begingroup
1041   \let\@author@present\@author@present@group
1042   \frontmatter@author@format
1043   \frontmatterverbose@sw{\typeout{\string\frontmatter@author@produce@group:}\say\@AAC@list\sa
1044   \let\AU@temp\@empty
1045   \set@listcomma@list\AU@temp
1046   \def\CO@opr{\@collaboration@present\AU@temp}%
1047   \let\AFG@opr \affils@present@group
1048   \let\@listcomma\relax
1049   \@AFG@list
1050   \frontmatter@footnote@produce
1051   \par
1052   \endgroup
1053   \frontmatter@author@below
1054 }%
1055 \@provide\frontmatter@author@below{}%

```

`\affils@present@group` Alias of `\AFG@opr`, this procedure is used on every element of `\@AFG@list`, the list of unique affiliation groups, when formatting the author/affiliation block: It presents all authors sharing this affiliation group.

This procedure is called via the procedure `\frontmatter@author@produce@group`, an alias for `\frontmatter@author@produce`,

Its use pertains to the class options `groupedaddress`, `unsortedaddress`, and `runinaddress` (that is, `\groupauthors@sw` is true).

For each invocation, the entire `\@AAC@list` is run through (twice), making this process execute in  $N^2$  time. (Is this really true?)

```

1056 \def\affils@present@group#1{%

```

Count (using `\@tempcnta`) the authors in the author list whose affiliation group matches the argument `#1`.

Note that `\AF@temp` is used as a storage register for the argument of the procedure, for purposes of comparison using `\ifx`.

Note: here we use `\@tempcnta` to accumulate the number of authors in the current affiliation set.

QUERY: do the value of `\AF@temp` and `\AF@opr` need to persist?

QUERY: what value does `\AU@opr` have at this point? ANSWER: the `\@affilID@count` and `\@affilID@match` procedures assign meanings to `\AU@opr` and `\CO@opr` dynamically. We initialize the two operators to `\undefined` so as to catch the pathological cases.

```

1057   \begingroup
1058   \def\AF@temp{#1}%
1059   \@tempcnta\z@
1060   \let\AU@opr \undefined
1061   \let\CO@opr \undefined
1062   \def\AF@opr{\@affilID@count\AF@temp\@tempcnta}%

```



```

1063   \@AAC@list
      If there are no authors using this affiliation group, skip it.
1064   \@ifnum{\@tempcnta=\z@}\{ }\{ \%
      DPC: If we have a list of length two, need to locally zap a comma.
1065   \begingroup
1066   \frontmatter@above@affilgroup
1067   \set@listcomma@count\@tempcnta
      Run through the author list, presenting (with \@author@present) those authors
      whose affiliation matched the given one.
1068   \let\AU@opr \undefined
1069   \let\CO@opr \undefined
1070   \def\AF@opr{\@affilID@match\AF@temp}%
1071   \@AAC@list
1072   \endgroup
      Now that the authors have all been presented, present the affiliations, grouped.
      Note: \@tempcnta is the number of affiliations for this set of authors.
1073   \begingroup
1074   \par
1075   \frontmatter@above@affiliation
1076   \frontmatter@affiliationfont
1077   \let\\\frontmatter@addressnewline
1078   \@tempcnta\z@
1079   \@tfor\AF@temp:=#1\do{%
1080     \expandafter\@ifx\expandafter{\expandafter\z@\AF@temp}\{ }\{ \%
1081     \advance\@tempcnta\@ne
1082     }%
1083   }%
1084   \@ifnum{\@tempcnta=\tw@}\{ }\{ \%
1085     \let\@listcomma\relax
1086   }{ }\{ \%
1087   \def@after@address
1088   \runinaddress@sw{%
1089   }{ \%
1090   \tightenlines@sw{ }\{ \%
      The following line determines the space between affiliations in grouped mode
1091   \parskip\z@
1092   }%
1093   \appdef\after@address\par
1094   }%
1095   \let\AFF@opr \@affil@group
1096   \do@affil@fromgroup\@AFF@list#1\relax
1097   \endgroup
      End of branch handling authors.
1098   }%
1099   \par

```

```

1100 \endgroup
1101 }%

\def@after@address Each of these procedures assigns a meaning to \after@address, the first a useful
\def@after@address@empty default, the second a nil. A society or journal substyle may define it according to
its own requirements.
1102 \def\def@after@address{\def\after@address{\@listcomma\ \@listand}}%
1103 \def\def@after@address@empty{\let\after@address\@empty}%

\affilID@count The control sequence name \AF@opr is placed in the author/affiliation list
\affilID@match (\@AAC@list) for each (new) affiliation, with that affiliation as its argument.
The control sequence name \AF@opr is an alias of a procedure to count the
number of authors at a given affiliation (\@affilID@count) or to present the given
authors (\@affilID@match). These two procedures are executed when formatting
under the groupedaddress class option.
1104 \def\@affilID@count#1#2#3{%
1105 \def\@tempifx{#3}%
1106 \@ifx{\@tempifx#1}{%
1107 \def\AU@opr{\@author@count#2}%
1108 }{%
1109 \let\AU@opr \@author@gobble
1110 }%
1111 \let\CO@opr \@collaboration@gobble
1112 }%
1113 \def\@affilID@match#1#2{%
1114 \def\@tempifx{#2}%
1115 \@ifx{\@tempifx#1}{%
1116 \let\AU@opr \@author@present
1117 }{%
1118 \let\AU@opr \@author@gobble
1119 }%
1120 \let\CO@opr \@collaboration@gobble
1121 }%

\do@affil@fromgroup Executes iteratively: selects the next address ID to print, and then re-execute the
list of addresses to print the text of that address.
Note: an argument of \relax breaks out of this iteration, while an argument
of \z@ (\noaffiliation) is a no-op.
Note: we have created an  $N^2$  computing process. FIXME: can use hashing
instead, you know.
1122 \def\do@affil@fromgroup#1#2{%
1123 \@ifx{\relax#2}{}%
1124 \count#2\relax
1125 \@ifnum{\z@=\count#2}{#1}%
1126 \do@affil@fromgroup#1%
1127 }%
1128 }%

```

`\@affil@group` The alias of `\AFF@opr` when the list `\@AFF@list` is executed by `\do@affil@fromgroup` (`groupedaddress`): Print the address if its number matches `\count@` (value set by `\do@affil@fromgroup`).

Note that we detect a `\noaffiliation` here by the affiliation address being the same as the expansion of `\blankaffiliation`.

I made a mistake when coding the comparison to `\blankaffiliation`: it originally read `\@ifx{#3\blankaffiliation}`. This is obviously wrong once you think twice.

```

1129 \def \@affil@group#1#2#3#4{%
1130   \@ifnum{#1=\count@}{%
1131     \def \@tempa{#3}%
1132     \@ifx{\@tempa\blankaffiliation}{}%
1133     #3%
1134     \@ifempty{#4}{}%
1135     \frontmatter@footnote{#4}%
1136   }%
1137   \after@address
1138 }%
1139 \advance \@tempcnta \m@ne
1140 }{}%
1141 }%

```

`\@author@present@group` This version of `\AU@opr` applies to the `groupedaddress` class option and its ilk.

`\@affil@present@group` Traverse the list calling the `\doauthor` hook on each author.

The third argument of `\doauthor` fulfills the requirement that it queue up a boolean via `\aftergroup`.

```

1142 \def \@author@present@group#1#2#3{%
1143   \gdef \comma@space{\gdef \comma@space{\textsuperscript{,\,}}}%
1144   \doauthor{#2}{#3}{\@affil@present@group}%
1145   \advance \@tempcnta \m@ne
1146 }%
1147 \def \@affil@present@group{%
1148   \aftergroup \false@sw
1149 }%

```

`\@pacs@produce` PACS, keywords and dates.

```

1150 \def \@pacs@produce#1{%
1151   \showPACS@sw{%
1152     \begingroup
1153     \frontmatter@PACS@format
1154     \@pacs@name#1\par
1155     \endgroup
1156   }{%
1157     \@ifempty{#1}{}%
1158     \class@warn{\PACS@warn}%
1159   }%
1160 }%
1161 }%

```

```

1162 \def \PACS@warn{If you want your PACS to appear in your output, use document class option showpa

```

`\@keywords@produce`

```
1163 \def\@keywords@produce#1{%
1164 \showKEYS@sw{%
1165 \begingroup
1166 \frontmatter@keys@format
1167 \@keys@name#1\par
1168 \endgroup
1169 }{%
1170 \@ifempty{#1}-{%
1171 \class@warn{If you want your keywords to appear in your output, use document class option sh
1172 }%
1173 }%
1174 }%
```

`\frontmatter@footnote@produce@footnote` Produce any footnotes to appear at the bottom of the title page.

`\frontmatter@footnote@produce@endnote` If frontmatter footnotes have been redirected to the bibliography, this will be a no-op. ref.: `\present@bibnote`.

```
1175 \def\frontmatter@footnote@produce@footnote{%
1176 \let\@TBN@opr\present@FM@footnote
1177 \@FMN@list
1178 \global\let\@FMN@list\@empty
1179 }%

1180 \def\present@FM@footnote#1#2{%
1181 \begingroup
1182 \csname c@\@mpfn\endcsname#1\relax
1183 \def\@thefnmark{\frontmatter@thefootnote}%
1184 \frontmatter@footnotetext{#2}%
1185 \endgroup
1186 }%
1187 \def\frontmatter@footnote@produce@endnote{%
1188 }%
```

## 10 Package defaults

`\title@column` We give meanings to the Booleans and other macros that a client document class would set on our behalf.

```
\collaboration@sw 1189 \appdef\frontmatter@init{%
\frontmatter@footnote@produce 1190 \@ifundefined\title@column {\let\title@column\@empty}-{%
\do@output@MVL 1191 \@ifundefined\preprintsty@sw {\@booleanfalse\preprintsty@sw}-{%
\comma@space 1192 \@ifundefined\frontmatter@footnote@produce{\let\frontmatter@footnote@produce\frontmatter@foot
1193 \@ifundefined\do@output@MVL {\let\do@output@MVL\@firstofone}-{%
1194 \@ifundefined\comma@space {\let\comma@space\@empty}-{%
1195 }%
```

`\frontmatter@thefootnote` In the title block, the footnote symbol (used for alternate affiliation and sundry) is a `\@fnsymbol` or a lowercase letter, as selected by a boolean. The society or journal may override this definition.

```

1196 \def\frontmatter@thefootnote{%
1197 \altaffilletter@sw{\@alpha}{\@fnsymbol}{\csname c@\@mpfn\endcsname}%
1198 }%
1199 \ifx{\altaffilletter@sw\@undefined}{\@booleantrue\altaffilletter@sw}{}%

```

`\frontmatter@makefnmark` The default procedure for setting the footnote mark within the frontmatter. A client may override this procedure.

```

1200 \def\frontmatter@makefnmark{%
1201 \@textsuperscript{%
1202 \normalfont\@thefnmark
1203 }%
1204 }%

```

`\frontmatter@makefntext` This procedure is used when setting footnotes within the title block.

Note that we are `hyperref` friendly: we invoke `\Hy@raisedlink` so that this footnote becomes a hypertext anchor. If `hyperref` is not loaded, then `ltxutil` has our act covered via some stubs.

```

1205 \long\def\frontmatter@makefntext#1{%
1206 \parindent 1em
1207 \noindent
1208 \Hy@raisedlink{\hyper@anchorstart{frontmatter.\expandafter\the\csname c@\@mpfn\endcsname}\hyper
1209 \@makefnmark
1210 #1%
1211 }%

```

`\frontmatter@setup` The default type specification for the title page. Clients will override this definition.

```

1212 \def\frontmatter@setup{}%

```

`\frontmatter@RRAPformat` The procedure `\frontmatter@RRAPformat` is the formatting procedure for the `\punct@RRAP` “Received, Revised, etc.” block on the title page. The macro `\punct@RRAP` produces the punctuation between the entries. Clients will override these two definitions.

The procedure `\produce@RRAP` is fairly general, and should be used by most journal substyles.

The procedure `\frontmatter@RRAP@format` will be overridden by the journal.

```

1213 \def\frontmatter@RRAPformat#1{%
1214 \removelastskip
1215 \begingroup
1216 \frontmatter@RRAP@format
1217 #1\par
1218 \endgroup
1219 }%
1220 \def\punct@RRAP{; }%
1221 \def\produce@RRAP#1{%
1222 \@ifempty{#1}{}%
1223 \@ifvmode{\leavevmode}{\unskip\punct@RRAP\ignorespaces}%
1224 #1%

```

```

1225 }%
1226 }%

\frontmatter@authorformat The default type specification for the author list. Clients will override this defini-
tion. For one alternative, see the revtex4 document class.
1227 \def\frontmatter@authorformat{}%

\frontmatter@above@affilgroup The default amount of space above an affiliation group (in a \affils@present@group
production). Clients will override this definition.
1228 \def\frontmatter@above@affilgroup{}%

\frontmatter@above@affiliation The \frontmatter@above@affiliation is the amount of space above affiliations
matter@above@affiliation@script in the groupedaddress style, and \frontmatter@above@affiliation@script is
that for superscriptaddress. Clients will override this definition. For one alter-
native, see the revtex4 document class.
1229 \def\frontmatter@above@affiliation{}%
1230 \def\frontmatter@above@affiliation@script{}%

\frontmatter@affiliationfont The default type specification for the affiliation. Clients will override this defini-
tion. For one alternative, see the revtex4 document class, where several alterna-
tive definitions are made.
1231 \def\frontmatter@affiliationfont{\itshape\selectfont}%

\frontmatter@RRAP@format The default type specification for the dates. Clients will override this definition.
1232 \def\frontmatter@RRAP@format{}%

\frontmatter@PACS@format Specify the formatting of the title page PACS statement.
Journals will override.
1233 \def\frontmatter@PACS@format{}%

\frontmatter@keys@format Specify the formatting of the title page keywords statement.
Journals will override.
1234 \def\frontmatter@keys@format{}%

\frontmatter@finalspace
1235 \def\frontmatter@finalspace{\addvspace{18\p@}}

\frontmatter@addressnewline The definition of \\ for address handling. Default puts all the ‘lines’ on a run-in
line, separated by comma and space. DPC: was
space between lines of addresss.
1236 \def\frontmatter@addressnewline{%
1237 \@ifhmode{\skip@\lastskip\unskip\unpenalty\break\hskip\skip@}{}%
1238 % was: \vskip-.5ex
1239 }%

\frontmatter@preabstractspace Space above and space below abstract in title block
\frontmatter@postabstractspace 1240 \def\frontmatter@preabstractspace{5.5\p@}
1241 \def\frontmatter@postabstractspace{6.5\p@}

```

`\aftermaketitle@chk` Error-checking code.

```
1242 \def\aftermaketitle@chk#1{%
1243   \@ifx{\maketitle\relax}{%
1244     \class@err{\protect#1 must be used before \protect\maketitle}%
1245   }{%}%
1246 }%
```

`\ps@titlepage` Default page style for title page. Journals will override this procedure.

```
1247 \def\ps@titlepage{\ps@empty}%
```

FIXME: there is a limitation to the default meaning of `\@startpage`: the label “FirstPage” is only defined if the document has a `\maketitle` command.

```
1248 \def\volumeyear#1{\gdef\@volumeyear{#1}}%
1249 \def\@volumeyear{}%
1250 \def\volumentnumber#1{\gdef\@volumenumber{#1}}%
1251 \def\@volumenumber{}%
1252 \def\issuenumber#1{\gdef\@issuenumber{#1}}%
1253 \def\@issuenumber{}%
1254 \def\eid#1{\gdef\@eid{#1}}%
1255 \def\@eid{}%
1256 %
1257 \def\startpage#1{\gdef\@startpage{#1}\c@page#1\relax}%
1258 \def\@startpage{\pageref{FirstPage}}%
1259 \def\endpage#1{\gdef\@endpage{#1}}%
1260 \def\@endpage{\pageref{LastPage}}%
```

## 10.1 Printing out the “list-of” elements

FIXME: The `\appendix@toc` procedure should change the meaning of `\l@section` so that the `\sections` can be appropriately formatted, reflecting their status as appendices.

```
1261 \def\print@toc#1{%
1262   \begingroup
1263   \expandafter\section
1264   \expandafter*%
1265   \expandafter{%
1266     \csname#1name\endcsname
1267   }%
1268   \let\appendix\appendix@toc
1269   \@starttoc{#1}%
1270 \endgroup
1271 }%
1272 \def\appendix@toc{}%
```

`\Dated@name` These strings are used in the `\date`, et al. commands.

```
\Received@name 1273 \def\Dated@name{Dated }%
\Revised@name 1274 \def\Received@name{Received }%
\Accepted@name 1275 \def\Revised@name{Revised }%
\Published@name
```

```

1276 \def\Accepted@name{Accepted }%
1277 \def\Published@name{Published }%

```

Two commands require robustifying against harsh treatment when TOC entries are written out. Note that `\class@warn` is not user-level markup, but crops up when the `\title` command is missing from the document.

```

1278 \appdef\robustify@contents{%
1279 \let\thanks@gobble\let\class@warn@gobble
1280 \def\begin{string\begin}\def\end{string\end}%
1281 }%

```

## 10.2 Syntax switch

Shall we take over the syntax of L<sup>A</sup>T<sub>E</sub>X? By default, Yes, unless the Boolean below has already been defined. In the package version, this is controlled by an option; in the kernel version, the Boolean remains undefined.

`\maketitle` presents a complication. If this code will execute under `ltxgrid`, then its version of the procedure will do fine.

As a package, under `article`, things are different. In this case, what should be done depends upon `\if@titlepage`. If true, then just execute `\maketitle` as it stands. If false, then the `article` class has defined its wrapper procedure, and we should take over the meaning of `\@maketitle` instead.

Here is a list of the commands defined by this package that potentially override those of standard L<sup>A</sup>T<sub>E</sub>X: `\frontmatter@title`, `\frontmatter@author`, `\frontmatter@and`, `\frontmatter@thanks`, `\frontmatter@date`, `frontmatter@abstract`, `frontmatter@titlepage`, `\frontmatter@maketitle`.

The following code will perform that override, given the appropriate state of the Boolean.

```

1282 \@ifxundefined\frontmatter@syntax@sw{\@booleantrue\frontmatter@syntax@sw}{}%
1283 \frontmatter@syntax@sw{%
1284 \let\title \frontmatter@title
1285 \let\author \frontmatter@author
1286 \let\date \frontmatter@date
1287 \@ifxundefined\@maketitle{%
1288 \let\maketitle \frontmatter@maketitle
1289 \@booleantrue \titlepage@sw
1290 }{%
1291 \let\@maketitle \frontmatter@maketitle
1292 \prepdef\maketitle\@author@finish
1293 }%
1294 \let\noaffiliation \frontmatter\noaffiliation
1295 \let\thanks@latex \thanks
1296 \let\thanks \frontmatter@thanks
1297 \let\and@latex \and
1298 \let\and \frontmatter@and
1299 \let@environment{titlepage}{frontmatter@titlepage}%
1300 \let@environment{abstract}{frontmatter@abstract}%
1301 }{%

```



```

1302 \let\noaffiliation\@empty
1303 }%

```

### 10.3 End of the ltxfront kernel

The kernel portion of the ltxfront package is complete.

```

1304 %</kernel>

```

### 10.4 Remainder of the ltxfront package

Include any code here that ought to be incorporated into the package, but should not be part of the kernel.

```

1305 %<*package>

```

Define formatting as like as possible to the article class. Comands that prepare for \maketitle: \author, \title, \date.

Argument of the above three may contain \\ and \thanks; argument of \author may contain \and. The \\ may still work fine; \and will be ignored (warning); \thanks even though it should not reside in the argument, will work.

An explicit \footnoteshould be treated as if it were \thanks.

```

% \@maketitle: \frontmatter@footnote@produce
%

```

```

1306 \def\thanks@latex#1{%
1307 \footnotemark
1308 \expandafter\expandafter
1309 \expandafter\gappdef
1310 \expandafter\expandafter
1311 \expandafter\@thanks
1312 \expandafter\expandafter
1313 \expandafter{%
1314 \expandafter\expandafter
1315 \expandafter\footnotetext
1316 \expandafter\expandafter
1317 \expandafter[%
1318 \expandafter\the\csname c@\mpfn\endcsname]{#1}}%
1319 }%
1320 \@booleanfalse\altaffilletter@sw
1321 \@if@sw@if@titlepage\fi{\@booleantrue}{\@booleanfalse}\titlepage@sw
1322 \def\frontmatter@title@above{\newpage\null\vskip2em\relax}%
1323 \def\frontmatter@title@format{\centering\LARGE\let\thanks\thanks@latex}%
1324 \def\frontmatter@title@below{\vskip1.5em\relax}%
1325 \def\frontmatter@author@format{\centering\large\advance\baselineskip\p@\parskip1.5\p@\let\thank
1326 \def\frontmatter@author@below{\vskip 1em\relax}%
1327 \def\frontmatter@above@affiliation{%
1328 \def\frontmatter@above@affiliation@script{%
1329 \def\frontmatter@affiliationfont{\centering\itshape}%

```

```

1330 \def\frontmatter@RRAP@format{\centering\large}%
1331 \def\frontmatter@preabstractspace{1.5em}%

1332 \long\def\frontmatter@footnotetext{%
1333 \expandafter\expandafter
1334 \expandafter\footnotetext
1335 \expandafter\expandafter
1336 \expandafter[%
1337 \expandafter\the\csname c@\mpfn\endcsname]%
1338 }%
1339 \def\and@space{\ }%
1340 \def\andname{and}%

```

Implement the feature of the article class whereby there is a default `\@date`, left over from the  $\text{\LaTeX}$  kernel. This package does not have a default date. The user must put in an explicit command, like `\date{\today}`, which will restore  $\text{\LaTeX}$ 's default behavior.

What if the abstract environment falls after the `\maketitle` command? The environment will be undefined; the user will have to deal with this.

End of the package.

```
1341 %</package>
```

Here ends the programmer's documentation.