

# The `snapshot` package

American Mathematical Society

Version 2.14, 2020/06/17

## Contents

1	Introduction . . . . .	1
2	Graphics files . . . . .	3
3	Formalities . . . . .	3
4	Package options . . . . .	4
5	Implementation . . . . .	5
	5.1 Determining the engine . . . . .	7
	5.2 Package options . . . . .	11
6	Compatibility . . . . .	21
7	In conclusion . . . . .	21
8	And finally . . . . .	21
9	Finale . . . . .	24

## 1 Introduction

The `snapshot` package helps the owner of a  $\text{\LaTeX}$  document obtain a list of the external dependencies of the document, in a form that can be embedded at the top of the document. To put it another way, it provides a snapshot of the current processing context of the document, insofar as it can be determined from inside  $\text{\LaTeX}$ .

If a document contains such a dependency list, then it becomes possible to arrange that the document be processed always with the same versions of everything, in order to ensure the same output. This could be useful for someone wanting to keep a  $\text{\LaTeX}$  document on hand and consistently reproduce an identical DVI file from it, on the fly; or for someone wanting to shield a document during the final stages of its production cycle from unexpected side effects of routine upgrades to the  $\text{\TeX}$  system.

Normal usage of the `snapshot` package involves the following steps:

1. Add a `\RequirePackage` statement at the top of the document:

```
\RequirePackage{snapshot}  
\documentclass{article}  
...
```

2. Run  $\text{\LaTeX}$  on the document. This will produce a dependency list in a file `\jobname.dep`. (I.e., if the document name is `vermont.ltx`, the dependency list will be named `vermont.dep`.)
3. Insert the `.dep` file at the top of the document, before `\documentclass`. The following example shows what you would end up with for a document that used the `article` documentclass and the `graphicx` package:

```

\RequirePackage{snapshot}[1999/11/03]
\RequireVersions{
  *{application}{TeX}      {1990/03/25 v3.x}
  *{format}{LaTeX2e}      {1999/06/01 v2.e}
  *{package}{snapshot}    {1999/11/03 v1.03}
  *{class}{article}       {1999/01/07 v1.4a}
  *{file}{size10.clo}     {1999/01/07 v1.4a}
  *{package}{graphicx}    {1999/02/16 v1.0f}
  *{package}{keyval}      {1999/03/16 v1.13}
  *{package}{graphics}    {1999/02/16 v1.01}
  *{package}{trig}        {1999/03/16 v1.09}
  *{file}{graphics.cfg}{0000/00/00 v0.0}
  *{file}{dvips.def}      {1999/02/16 v3.0i}
}
\documentclass{article}
\usepackage{graphicx}
...

```

The package option `log` will cause the dependency list to appear in the  $\text{\LaTeX}$  log file instead of in a separate `.dep` file:

```
\RequirePackage[log]{snapshot}
```

Making the necessary arrangements to ensure that future  $\text{\LaTeX}$  runs of the document actually call in the specified versions is a separate problem. The `snapshot` package only provides a way to generate the dependency list. However, the `\RequireVersions` statement does record the given information in a form that can be accessed from within  $\text{\LaTeX}$ . (It is for this purpose that it is not simply a comment.) In principle a package could be set up so that a later version would automatically attempt to emulate an earlier version if an earlier version was specified—much as  $\text{\LaTeX}$  currently switches to 2.09 compatibility mode if it sees `\documentstyle` instead of `\documentclass`.

For maximum reliability font checksums should also be reported in the dependency list, but standard  $\text{\TeX}$  3.x does not provide direct access to font checksums for macro programmers. This information could be added by a separate script that scans the DVI file. (Certain nontrivial complications are possible, however.)

## 2 Graphics files

When a graphics file is read in by a  $\LaTeX$  document using the standard `\includegraphics` command, it gets a dummy version number string of

`Graphics file (type foo)`

where `foo` is typically `eps`. This is with the current version of the `graphics` package (at the time of this writing: 1999/02/16 v1.01). What this means in practice is that all graphics files will have their snapshot date and version number recorded as

`Graphics v0.0`

and will always compare equal (the string “Graphics” will be used in place of a date, but since comparison is done with `\ifx` it doesn’t make any real difference).

It would be possible, for `.eps` files at least, to read the `CreationDate` comment that is normally included in the file header and use that as the basis of comparison. Recording the bounding box numbers instead of a dummy version number is another possibility, which you can get with the `bbinfo` option (as of `snapshot` version 2.05).

## 3 Formalities

`\RequireVersions` The `\RequireVersions` command scans its argument for file names and associated version number information. The syntax of a version line for a particular file is

```
*{ file type }{ file name }{ version info }
```

In other words, the `*` character in this context is like a command that takes three arguments. The extension part of the file name should be omitted in the second argument, except when the file type is `file` (following the conventions of  $\LaTeX$ ’s `\ProvidesPackage` and `\ProvidesFile` commands). The most commonly used file types are as follows.

**class** A  $\LaTeX$  documentclass file.

**package** A  $\LaTeX$  package file.

**tfm** A  $\TeX$  font metric file. In this case the “version number” is the checksum, and unless you are using an extended version of  $\TeX$  this information is not accessible from inside  $\LaTeX$ , so it must be filled in by an outside process. By default, font metric files are not listed in the dependency list since the checksum info is not available. There is a package option `tfm` to turn on the logging of metric files. (Not yet implemented [mjd,1999/09/23])

**format** This is almost always `LaTeX2e`. The information comes from `\fmtname`. `Lambda`, `e $\LaTeX$` , and `pdf $\LaTeX$`  leave `\fmtname` unchanged, and although this may seem dubious at first sight, I guess they have little choice: the widespread use of `\NeedsTeXFormat` in existing package files makes them produce an error message if `\fmtname` is changed. (Maybe the thing to do would be to modify the definition of `\NeedsTeXFormat` as well.)

For a  $\LaTeX$  format that uses the Babel mechanism for preloading hyphenation patterns, the version number of `babel.def` that was used in building the format might be of interest. But at first glance there does not seem to be any easy way of dealing with that, and in the normal course of things, a document that relies on Babel features will also have a `\usepackage{babel}` statement at the top, and that will yield adequate (I think) information for the snapshot dependency list.

**application** With standard  $\TeX$  there is no reliable way to get the exact version number from inside  $\LaTeX$ . If a document is processed with one of the recent variants that address this deficiency (such as  $e\text{-}\TeX$ ,  $pdf\text{-}\TeX$  and Omega) the available version info is used, but otherwise a presumptive value of `1990/03/25 v3.x` is used (the official release date of  $\TeX$  3.0).

**file** None of the above: some other file of miscellaneous type, e.g., `.clo`, `.cfg`, `.tex`, or `.def`.

The `\RequireVersions` command can be given an optional “ident” argument, similar to the argument of a `\label` command. This is not used internally but it could be used to assign a label to particular groups of files in case that helps with external processing.

## 4 Package options

The list of options supported by the `snapshot` package is as follows:

<code>dep</code>	<code>date</code>
<code>log</code>	<code>version</code>
<code>tfm</code>	<code>major-version</code>
<code>warning</code>	<code>bbinfo</code>
<code>error</code>	<code>test</code>
<code>self-warning</code>	

**dep**, **log** Write file date and version information to `jobname.dep` or to the  $\LaTeX$  log file, respectively.

**error**, **warning**, **self-warning** If the `snapshot` package is invoked with the `error` option *and also* the document contains a `\RequireVersions` statement, then each subsequent `\ProvidesFile`, `\ProvidesPackage`, and `\ProvidesClass` statement will compare date and version number information with the corresponding information from the `\RequireVersions` statement and give an error message if a mismatch is detected. With the `warning` option you get warnings instead of errors. By default both the date and the version number are compared; this behavior can be modified, however, by giving additional options:

**date** compare only dates,

**version** compare only version numbers,

**major-version** use only the major version number when comparing.

The `self-warning` causes a warning to be given, instead of an error, if the `snapshot` package itself has a date or version mismatch.

**Note:** A file that doesn't have any sort of `\ProvidesFile` or `\ProvidesPackage` statement in it will show up in the dependency list, with a dummy date and version number of 0000/00/00 v0.0, but testing for a version mismatch with such a file is then infeasible.

**bbinfo** For files of type “graphic”, include bounding-box info as the “version number”. A normal date and version number are seldom available for such files, and  $\LaTeX$  does not attempt to read them, which means that the `snapshot` package could not obtain the information except by drastically modifying the low-level  $\LaTeX$  operations that read graphics file information—which seems overly risky.

**tfm** *Not implemented yet!* Include information about which  $\TeX$  font metric files are called by  $\LaTeX$ . This list is usually somewhat different from the list of fonts that are actually used in the output file (`.dvi` or `.pdf`), primarily because the setup for math formulas will normally preload font metric information for all the fonts from which  $\LaTeX$ 's basic math symbols are drawn (the symbols documented in the  $\LaTeX$  book), even if the document does not use symbols from all of those fonts.

**test** This option is for a special purpose. It drastically changes the action of the `\RequireVersions` command so that it does not merely record the information for later reference, but does a “trial load attempt” for each file in the list, and then stops the  $\LaTeX$  job as soon as the list is finished, without continuing any further.

By “trial load attempt” I mean that the `\RequireVersions` command will actually input each file, but with various bits redefined so that `\ProvidesPackage` and variants will execute `\endinput`—in other words, only the first few lines of each file will be read.

What this means, practically speaking, is that if you run such a test on a document, it gives a relatively quick check on two useful pieces of information without having to retypeset the entire document (which for some documents might be very tedious)

1. The actual location on your system from which the file will be loaded.
2. The date and version info from `\Provides...` line, if present.

Similar results could be obtained by a combination of `kpsewhich` and `grep`, but because the `snapshot test` option works through  $\LaTeX$ , it is a system-independent method.

Caveat: If a file does not contain any `\ProvidesSomething` line, it will be read in its entirety, which might lead to errors. Or if there is any weird stuff preceding the `\ProvidesWhatever` line. But for well-behaved files the option seemed useful enough to be worth implementing.

## 5 Implementation

Standard declaration of package name and date.

```
1 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2 \ProvidesPackage{snapshot}[2020/06/17 v2.14]
```

Calling the `snapshot` package in a document causes L<sup>A</sup>T<sub>E</sub>X to list the file names and versions in the T<sub>E</sub>X log or in a `.dep` file, so that the information may be easily copied into the document file. The list so generated is nothing more than a slight adaptation of the output from L<sup>A</sup>T<sub>E</sub>X’s `\listfiles` command; it puts essentially the same information into a slightly more structured form so that it will be easier to use.

For the standard mechanisms that are already built into L<sup>A</sup>T<sub>E</sub>X (e.g., the handling of the second optional argument of `\LoadClass`), the de facto “version number” is the *date* given in the optional argument of a `\ProvidesClass` or similar command. Even though most `\ProvidesWhatever` commands also give something that follows the usual form of version numbers—a string of the form `v2.3`—this is only a convention, not used internally by L<sup>A</sup>T<sub>E</sub>X, and the identification string of a random loaded file is not guaranteed to include it. The `snapshot` package copies both pieces of information if available; if the second piece is not present, a dummy number `0.0` is supplied. Similarly, files that don’t include any `\ProvidesWhatever` statement will get a dummy date of `0000/00/00`;

**T<sub>E</sub>X system administrators who want to ensure maximal accuracy of the snapshot information should therefore make it a practice to use `\ProvidesFile` in `.cfg` files and other local files that might have an impact on the output fidelity of their documents.**

```

\@xp A couple of shorthand forms.
\@nx 3 \let\@xp=\expandafter \let\@nx=\noexpand

\str@cmp A function to compare two strings and return FT or TT (for use with \if).
\str@equal 4 \def\str@cmp#1#2\str@cmp#3{%
5 \if #1#3\else F\@car\fi \str@cmp#2\str@cmp
6 }
7 \def\string@equal#1#2{%
8 \str@cmp#1\relax\str@cmp#2{\relax\@gobbletwo}\@nil TT%
9 }

\RequireVersions Optional argument of \RequireVersions allows assigning a name to a partic-
ular collection of files. This might be useful for setting a TEX inputs path.
10 \newcommand{\RequireVersions}[2] [] {}%
11
12 \renewcommand{\RequireVersions}[1] [] {}%
13 \def\snap@check{\snap@compare@versions}%
14 \toks@{#1}%
15 \afterassignment\snap@storem
16 \let\@let@token=
17 }
18
19 \@onlypreamble\RequireVersions

\snap@storem

```

```

20 \def\snap@storem{%
21   \ifx\@let@token\bgroup \else
22     \PackageError{snapshot}{Expected a ‘\@charlb’ character here}\@ehc
23     \@xp@gobblefour
24   \fi
25   \futurelet\@let@token\snap@branch
26 }

```

\snap@check

```
27 \let\snap@check\@gobble
```

\snap@finish

```
28 \def\snap@finish{\toks@\bgroup}
```

\snap@branch

```

29 \def\snap@branch{%
30   \ifx\@let@token\egroup
31     \@xp\snap@finish
32   \else\ifx\@let@token *%
33     \let\reserved@c\snap@store@version
34   \else\ifx\@let@token\@sptoken
35     \lowercase{\def\reserved@c} {\futurelet\@let@token\snap@branch}%
36   \else
37     \let\reserved@c\snap@store@error
38   \fi\fi\fi
39   \reserved@c
40 }

```

\snap@store@error

```

41 \def\snap@store@error#1{%
42   \PackageError{snapshot}{Expected ‘*’ here, not ‘#1’}\@ehc
43 }
44
45 \@onlypreamble\snap@store@error

```

\snap@store@version

```

46 \def\snap@store@version #1#2#3#4{%
47   \@xp\snap@store@b\csname snapx@#2\endcsname{#2}{#3}{#4}%
48 }
49
50 \@onlypreamble\snap@store@version

```

## 5.1 Determining the engine

\snapshotApplication

Check for the engines that are active in  $\text{T}_{\text{E}}\text{X}$ Live 2020. Note that in in modern versions of  $\text{T}_{\text{E}}\text{X}$ Live, there is no distinction between *etex*, *pdfetex*, and *pdftex*. I believe this code will distinguish between distinct legacy binaries, but I have not tested this thoroughly. Similar remarks apply to the  $\text{pT}_{\text{E}}\text{X}$  family.

```
51 \let\snapshotApplication\@empty
```

```

52
53 \ifx\OmegaVersion\@@undefined \else
54   \edef\snapshotApplication{%
55     \ifx\AlephVersion\@@undefined
56       {\Omega}\space\space\space
57       {0000/00/00 v\OmegaVersion}%
58     \else
59       {Aleph}\space\space\space

```

\AlephVersion appears to be incorrectly set to 0.0, so include the Omega version as well.

```

60       {0000/00/00 v\OmegaVersion-\AlephVersion}%
61     \fi
62   }%
63 \fi
64
65 \ifx\ptexversion\@@undefined\else
66   \edef\snapshotApplication{%
67     {pTeX}\space\space\space\space
68     {0000/00/00 v\number\ptexversion
69     \ifx\ptexminorversion\undefined \u \else.\number\ptexminorversion\fi
70     \ifx\ptexrevision\undefined \u \else\ptexrevision\fi}%
71   }%
72   \ifx\uptexversion\@@undefined
73     \ifx\epTeXversion\@@undefined\else
74       \edef\snapshotApplication{%
75         {epTeX}\space\space\space
76         {0000/00/00 v\number\epTeXversion}%
77       }%
78     \fi
79   \else
80     \ifx\epTeXversion\@@undefined
81       \edef\snapshotApplication{%
82         {upTeX}\space\space\space
83         {0000/00/00 v\number\uptexversion
84         \ifx\uptexrevision\undefined \u \else\uptexrevision\fi}%
85       }%
86     \else

```

There is no separate \euptexversion, so use a combination of the underlying upTeX and epTeX versions.

```

87       \edef\snapshotApplication{%
88         {eupTeX}\space\space
89         {0000/00/00 v\number\uptexversion
90         \ifx\uptexrevision\undefined \u \else\uptexrevision\fi
91         --\number\epTeXversion}%
92     }%
93   \fi
94 \fi
95 \fi

```



```

96
97 \ifx\snapshotApplication\@empty
98   \ifx\XeTeXversion\@undefined\else
99     \edef\snapshotApplication{%
100       {XeTeX}\space\space\space
101       {0000/00/00 v\number\XeTeXversion
102       \ifx\XeTeXrevision\undefined\else\XeTeXrevision\fi}%
103     }%
104   \fi
105 \fi
106
107 \ifx\snapshotApplication\@empty
108   \ifx\luatexversion\@undefined\else
109     \begingroup
110       \@tempcnta\luatexversion
111       \divide\@tempcnta by 100
112       \edef\@tempa{\the\@tempcnta}%
113       \multiply\@tempcnta by 100
114       \@tempcntb\luatexversion
115       \advance\@tempcntb by -\@tempcnta
116       \edef\@tempa{\@tempa.\the\@tempcntb.\luatexrevision}%
117       \xdef\snapshotApplication{%
118         {luaTeX}\space\space
119         {0000/00/00 v\@tempa}%
120       }%
121     \endgroup
122   \fi
123 \fi
124
125 \ifx\snapshotApplication\@empty
126   \ifx\pdftexversion\@undefined \else
127     \begingroup
128       \ifx\XeTeXversion\@undefined
129         \@tempwafalse
130       \else
131         \@tempwatrue
132       \fi
133       \@tempcnta\pdftexversion
134       \divide\@tempcnta by 100
135       \edef\@tempa{\the\@tempcnta}%
136       \multiply\@tempcnta by 100
137       \@tempcntb\pdftexversion
138       \advance\@tempcntb by -\@tempcnta

```

eTeX was folded into pdfTeX as of version 1.40.

```

139       \ifnum\@tempcntb > 39
140         \@tempwafalse
141       \fi
142       \edef\@tempa{\@tempa.\the\@tempcntb.\pdftexrevision}%
143       \xdef\snapshotApplication{%

```

```

144         \if@tempswa
145             {pdfTeX}\space
146         \else
147             {pdfTeX}\space\space
148         \fi
149         {0000/00/00 v\@tempa}%
150     }%
151 \endgroup
152 \fi
153 \fi
154
155 \ifx\snapshotApplication\@empty
156     \ifx\eTeXversion\@undefined \else
157         \edef\snapshotApplication{%
158             {eTeX}\space\space\space\space
159             {0000/00/00 v\number\eTeXversion
160             \ifx\eTeXrevision\undefined
161                 \ifx\eTeXminorversion\undefined\else.\number\eTeXminorversion\fi
162             \else
163                 \eTeXrevision
164             \fi
165             }%
166         }%
167     \fi
168 \fi

169 \ifx\snapshotApplication\@empty
170     \edef\snapshotApplication{%
171         {TeX}\space\space\space\space\space
172         {1990/03/25 v3.x}%
173     }%
174 \fi

175 \def\@fmtextension{fmt}
176 \def\@tfmextension{tfm}
177 \edef\snapx@package{.\@pkgextension}
178 \edef\snapx@class{.\@clsextension}
179 \edef\snapx@format{.\@fmtextension}
180 \edef\snapx@tfm{.\@tfmextension}
181 \long\def\snapx@ignore{}
182 \let\snapx@application=\snapx@ignore
183 \let\snapx@file=\@empty
184 \let\snapx@end\@end
185 \expandafter\let\csname snapx@-----\endcsname\snapx@end

```

`\snap@store@b` For a package named `foo.sty`, this function defines `\rqv@foo.sty` to hold the date and version information.

```

186 \def\snap@store@b#1#2#3#4{%
187   \ifx#1\snapx@end
188     \exp\snap@finish
189   \else
190     \ifx#1\relax \let#1\@empty\fi
191     \def\@tempa##1 ##2 ##3\@nil{##1 ##2}%
192     \ifx#1\snapx@application
193       \exp\xdef\csname rqv@#3#1\endcsname{\@tempa#4 v?..? \relax\@nil}%
194     \else
195       \xdef\rqv@list{\rqv@list{#3#1}}%
196       \exp\xdef\csname rqv@#3#1\endcsname{\@tempa#4 v?..? \relax\@nil}%
197       \snap@intest{#3}{#1}%
198       \ifx#1\snapx@format \snap@check{#3.fmt}\else

```

Test if current file is snapshot.sty. Need to pre-expand the extension part to ensure the test is correct.

```

199         \edef\@tempa{\@nx\string@equal{snapshot.sty}{#3#1}}%
200         \if\@tempa \snap@selfcheck \fi
201     \fi
202 \fi
203 \fi
204 \futurelet\@let@token\snap@branch
205 }
206
207 \@onlypreamble\snap@store@b

```

`\snap@write` Default setup is geared to write the dependency list to a `.dep` file. The option `log` means write it to the `TEX` log instead.

```

208 \def\snap@write{\immediate\write\snap@out}
209 \let\snap@out\sixt@@n % fallback, probably never used

```

## 5.2 Package options

```

210 \DeclareOption{dep}{%
211   \def\snap@write{\immediate\write\snap@out}%
212 }

213 \DeclareOption{log}{%
214   \let\snap@write\typeout
215 }

```

The purpose of the ‘test’ option is to support a separate testing procedure for resolving file names and pre-checking version numbers. See §8 for more information.

```

216 \let\snap@intest=\@gobbletwo
217 \DeclareOption{test}{\def\snap@intest{True}}

```

For each font used by a document, we would like to list the `.tfm` file name and checksum. If `TEX` provided a `\fontchecksum` primitive similar to `\fontname` that could be used to get the checksum of any font, it would just about be feasible to do this entirely from within `LATEX`. As a partial solution we could at

least generate the list of font file names, to make it easier for an external utility to add the checksums.

In practice, extracting font names and checksums from the .dvi file will probably work well enough, leaving no work to be done by the `snapshot` package in this area. But theoretically speaking the output of a document could be affected by font metric files that are loaded during L<sup>A</sup>T<sub>E</sub>X processing but that do not show up in the .dvi file.

```
218 \DeclareOption{tfm}{%
219   \typeout{Option 'tfm' not implemented yet [1999/09/23]}%
220 }
```

### Warnings and errors

```
\snap@mismatch@warning
\snap@mismatch 221 \def\snap@mismatch@warning#1#2#3{\PackageWarningNoLine{#1}{#2}}
222 \def\snap@mismatch{\snap@mismatch@warning}

223 \DeclareOption{error}{%
224   \def\snap@mismatch{\PackageError}%
225   \def\snap@selfcheck{\snap@selfcheck@a}%
226   \ifx\snap@select@empty \let\snap@select\snap@select@all \fi
227 }

228 \DeclareOption{warning}{%
229   \def\snap@mismatch{\snap@mismatch@warning}%
230   \def\snap@selfcheck{\snap@selfcheck@a}%
231   \ifx\snap@select@empty \let\snap@select\snap@select@all \fi
232 }
```

`\snap@select@all` Because the exact form of the version number is not mandated by L<sup>A</sup>T<sub>E</sub>X, just  
`\snap@select` take the first two “words” delimited by spaces. And take a little extra care to properly handle multiple spaces between the words.

```
233 \def\snap@select@all#1#2 #3#4 #5\@nil{#1#2 #3#4}
234 \let\snap@select@empty
```

`\snap@seldate` If the naming conventions seem a little peculiar here, it’s because I had to add  
`\snap@selversion` some pieces later that I didn’t think of initially, and I wanted to minimize the  
`\snap@selmajor` chances of compatibility problems for client packages [mjd,2002-11-04].

```
235 \def\snap@seldate#1#2 #3\@nil{#1#2}%
236 \def\snap@selversion#1#2 #3{\snap@select@version #3}%
237 \def\snap@selmajor#1#2 #3{\snap@select@major #3}%

238 \DeclareOption{date}{\let\snap@select=\snap@seldate}
```

`\snap@select@version`

```
239 \def\snap@select@version#1{%
240   \ifodd 0#11 \exp\snap@sva\@xp#1\else\exp\snap@select@version\fi
241 }
242 \def\snap@sva#1.#2 #3\@nil{#1.#2}
```

`\snap@select@major`

```

243 \def\snap@select@major#1{%
244   \ifodd 0#11 \xp\snap@svm\@xp#1\else\xp\snap@select@major\fi
245 }
246 \def\snap@svm#1.#2\@nil{#1}

247 \DeclareOption{version}{\let\snap@select\snap@selversion}
248 \DeclareOption{major-version}{\let\snap@select\snap@selmajor}

249 \def\snap@bbinfo{01}
250 \DeclareOption{bbinfo}{\def\snap@bbinfo{00}}
```

`\snap@splitter` Give this an inert definition, for the time being, until we are ready to do the split.

```

251 \let\snap@splitter=?
252 \AtBeginDocument{%
253   \xdef\@filelist{\@filelist\snap@splitter}%
254 }
```

`\snap@selfcheck`

```

255 \let\snap@selfcheck\@empty
256 \let\snap@selfcheck@a\@empty
```

The self-warning option would normally be used in conjunction with the error option.

```

257 \DeclareOption{self-warning}{%
258   \def\snap@selfcheck{%
259     \begingroup
260       \def\snap@mismatch{\snap@mismatch@warning}%
261       \snap@selfcheck@a
262     \endgroup
263   }
264 }

265 \ExecuteOptions{warning}
266 \ProcessOptions\relax
```

`\snap@restore@extensions` We need the following patch to make up for the fact that `\@pkgextension` and `\@clsextension` are marked in the L<sup>A</sup>T<sub>E</sub>X kernel as “only preamble”.

```

267 \edef\snap@restore@extensions{%
268   \def\@nx\@pkgextension{\@pkgextension}%
269   \def\@nx\@clsextension{\@clsextension}%
270 }
```

`\snap@pad` Pad filename strings out to 8+3 length so that the list will look pretty.

```

271 \def\snap@pad#1#2#3#4#5#6#7#8#9{\snap@pad@a{#1#2#3#4#5#6#7#8#9}}
272 \def\snap@pad@a#1#2#3#4#5\@nil{\snap@pad@b#1#2#3#4\space\@nil}
273 \def\snap@pad@b#1\space#2\@nil#3{\def#3{#2}}
```

```

\snap@trim@version First stage: discard leading spaces before the first and second nonspace strings
                    in the argument. Take the first nonspace string as the date. Since we only do
                    equal/not-equal testing on dates, it does not seem essential to test if it is really
                    a valid date string or not (yyyy/mm/dd).
274 \def\snap@trim@version#1#2 #3{#1#2 \snap@trim@b #3}
                    Second stage: Scan for a version number. In order to handle some idiosyncratic
                    cases, such as url.sty version 1.4, we can't simply take the second nonspace
                    string as the version number but need to look for a leading digit.
275 \def\snap@trim@b#1{\ifodd 0#11 v#1\@xp\snap@trim@c\fi \snap@trim@b}
                    Arg 1 here is \snap@trim@b, which we just need to discard.
276 \def\snap@trim@c#1#2 #3\@nil{#2}
277 \let\rqv@list=\@empty
                    If \fmtname.fmt is not already in the file list, add it.
278 \edef\@tempc#1\fmtname{#1\fmtname}\@tempc
279 \def\@tempa#1,\fmtname.fmt,#2#3\@nil{#2}
280 \edef\@tempb{\@nx\@tempa,\@filelist,\fmtname.fmt,}
281
282 \if ?\@tempb?\@nil
283   \edef\@filelist{\fmtname.fmt,\@filelist}%
284   \def\@tempc{LaTeX2e}%
285   \@xp\edef\csname ver@\fmtname.fmt\endcsname{%
286     \fmtversion\space
287     v\ifx\@tempc\fmtname 2.e\else ?.\? \fi
288   }%
289 \fi
                    Ensure that files get recorded.
290 \listfiles

\snap@doit
291 \def\snap@doit#1{%
292   \begingroup
293   \ifx\delimiter#1\delimiter \else
294     \filename@parse{#1}%
295     \let\@tempd\@empty
296     \ifx\filename@ext\relax
297       \def\@tempa{file}\def\@tempb{~~~}%
298     \else\ifx\filename@ext\@pkgextension
299       \def\@tempa{package}\let\@tempb\@empty
300     \else\ifx\filename@ext\@clsextension
301       \def\@tempa{class}\def\@tempb{~~}%
302     \else\ifx\filename@ext\@fmtextension
303       \def\@tempa{format}\def\@tempb{~}%
304     \else\ifx\filename@ext\@tfmextension
305       \def\@tempa{tfm}\def\@tempb{~~~~}%
306     \else
307       \def\@tempa{file}%

```

```

308         \edef\@tempd{\filename@ext}%
309         \def\@tempb{~~~}%
310         \fi\fi\fi\fi\fi
311         \@xp\let\@xp\@tempe
312         \csname ver@\filename@base %
313         \ifx\filename@ext\relax\else.\filename@ext\fi\endcsname
314         \ifx\@tempe\@empty \let\@tempe\relax \fi
315         \edef\@tempe{%
316         \ifx\@tempe\relax 0000/00/00 v0.0%
317         \else
318         \@xp\@xp\@xp\@snap@trim@version\@xp\@tempe\space v0.0 v0.0 \@nil
319         \fi
320         }%
321         \edef\@tempc{\filename@area\filename@base\@tempd}% full file name
322         \@xp\@snap@pad\@tempc\space~::~::~::~::~::~::~::~::~::~::~\@nil\@tempd
323         \let~\space
324         \snap@write{\space\space *{\@tempa}\@tempb{\@tempc}\@tempd{\@tempe}}%
325         \fi
326         \aftergroup\@snap@doit
327     \endgroup
328 }

```

\snap@bracify

```

329 \def\snap@bracify#1#2,{%
330 \ifx\@empty#1\expandafter\@gobble\else {#1#2}\fi \snap@bracify
331 }

```

\snap@splitter@a

```

332 \def\snap@splitter@a{%
333     \iffalse{\fi }}% close current file name, end definition
334     \xdef\specific@files{%
335         \iffalse}\fi
336         \specific@files
337         \expandafter\@gobble\string % discard one closing brace
338 }

```

\snap@fdcheck

```

339 \def\snap@fdcheck#1{%
340     \ifx\delimiter#1%
341         \@xp\@gobble
342     \else
343         \snap@fda#1\@empty.fd\@empty ?\@nil
344     \fi
345     \snap@fdcheck
346 }
347
348 \def\snap@fda#1.fd\@empty#2#3\@nil{%
349     \if ?#2%
350         \xdef\specific@files{\specific@files {#1}}%

```

```

351   \else
352     \xdef\general@files{\general@files {#1.fd}}%
353   \fi
354 }

\general@files
\specific@files 355 \let\general@files\@empty
                 356 \let\specific@files\@empty

\SpecialInput  The \SpecialInput command is related to the packages-only option. Apart
                from some ad hoc handling for .fd files that get loaded on demand, all files
                that are input after \begin{document} are put into the specific-files list, and
                all files before \begin{document} go into the general-files (packages-only) list.
                If there is a macro file for a book (say), that contains definitions specific to that
                book, and that is loaded in the preamble, loading it with \SpecialInput will
                cause it to go in the specific-files list.
357 \newcommand{\SpecialInput}[1]{%
358   \xdef\specific@files{\specific@files{#1}}%
359   \@input#1\relax
360 }

\@dofilelist  Our definition of \@dofilelist does not retain much resemblance to the original
              in the LATEX kernel.
361 \def\@dofilelist{%
362   \snap@restore@extensions
363   \xdef\general@files{\@xp\snap@bracify \@filelist \@empty,\@empty,}%
364   \let\snap@splitter\snap@splitter@a
365   \xdef\general@files{\general@files}%
366   \let\@tempa\specific@files
367   \global\let\specific@files\@empty
368   \@xp\snap@fdcheck\@tempa{\delimiter}%
369   \ifx\rqv@list\@empty \else
370     \rqv@compare@lists
371   \fi
372   \ifx\snap@write\typeout \else
373     \newwrite\snap@out
374     \immediate\openout\snap@out=\jobname.dep \relax
375   \fi
376   \snap@write{\string\RequireVersions\@charlb}%
377   \snap@write{\space\space *{application}%
378     \snapshotApplication
379   }%
380   \@xp\snap@doit\general@files{\delimiter\aftergroup\@gobble\@gobble}%
381   \ifx\specific@files\@empty \else
382     \snap@specific
383   \fi
384   \snap@write{\@charrb}%
385   \ifx\snap@write\typeout \else
386     \immediate\closeout\snap@out

```



```

387     \typeout{Dependency list written on \jobname.dep.}%
388     \fi
389 }

```

`\snap@specific`

```

390 \def\snap@specific{%
391     \snap@write{ \space *{-----}{Document-specific files:}{----}}%
392     \@xp\snap@doit\specific@files{\delimiter\aftergroup\@gobble\@gobble}%
393 }

```

`\rqv@condense` The `\rqv@compare@lists` function checks to see if any files are found only in the `RequireVersions` list or only in the `\general@files` list.

```

394 \def\rqv@condense#1{%
395     \@xp\ifx\csname ver@#1\endcsname\N \else
396         \edef\L{L{#1}}%
397         \@xp\let\csname ver@#1\endcsname=\N
398     \fi
399     \rqv@condense
400 }
401
402 \def\rqv@condend{\endcsname ?\fi
403     \@xp\@xp\@xp\@gobbletwo\csname @xp\iftrue}

```

`\rqv@overloaded`

```

404 \def\rqv@overloaded#1{%
405     \snap@mismatch{snapshot}{^^J%
406         File #1 loaded though not in \noexpand\RequireVersions list%
407     }\@ehc
408 }

```

`\rqv@notloaded`

```

409 \def\rqv@notloaded#1{%
410     \snap@mismatch{snapshot}{^^J%
411         File #1 [\csname rqv@#1\endcsname] required but not loaded%
412     }\@ehc
413 }

```

`\rqv@set`

```

\rqv@test 414 \def\rqv@set#1{\@xp\let\csname ver@#1\endcsname\N \rqv@set}
415 \def\rqv@test#1{\csname ver@#1\endcsname{#1}\rqv@test}

```

`\rqv@compare@lists`

```

416 \def\rqv@compare@lists{%
417     \begingroup
    Clear up duplicate file names (just in case) to avoid redundant warning messages.
    This should seldom be necessary in practice.
418     \def\N{1}%
419     \let\L\@empty

```

```

420     \exp\rqv@condense\rqv@list\rqv@condend
421     \global\let\rqv@list=\L
422     \def\N{2}%
423     \let\L\@empty
424     \exp\rqv@condense\general@files\rqv@condend
425     \global\let\general@files=\L

Let's make a shorthand for the code that terminates our recursion.
426     \def\T{\@firstoftwo{\endcsname\@empty\@gobbletwo}}%

Set all the loaded general files to an error function.
427     \let\N\rqv@overloaded \exp\rqv@set\general@files \T

Set all the required files to an ignore function.
428     \let\N\@gobble
429     \exp\rqv@set\rqv@list \T

Execute all the general files.
430     \exp\rqv@test\general@files{\endcsname\csname @gobbletwo}%

And now do essentially the same thing in the reverse direction.
431     \let\N\rqv@notloaded
432     \exp\rqv@set\rqv@list \T
433     \let\N\@gobble
434     \exp\rqv@set\general@files \T
435     \exp\rqv@test\rqv@list{\endcsname\csname @gobbletwo}%
436     \endgroup
437 }

Compensate for a bug in old versions of amsgen.sty. This is a little tricky.
Old version: \ver@amsgen=1996/10/29 v1.2b
New version: \ver@amsgen.sty=1999/11/30 v2.0
438 %\@namedef{ver@amsgen.sty}{1996/10/29 v1.2b}
439 \AtBeginDocument{%
440     \@ifundefined{ver@amsgen}{}{%
441         \exp\let\csname ver@amsgen.sty\@xp\endcsname
442             \csname ver@amsgen\endcsname
443     }%
444 }

```

`\ProvidesFile` Because `\ProvidesFile` is used in `.fd` files which are normally read with special catcodes, there tend to be problems with whitespace characters being erroneously lost from the second argument. Since we have to put in a `\snap@check` call anyway, while we're at it let's fix a bug of this type that affected some older versions of L<sup>A</sup>T<sub>E</sub>X.

```

445 \def\ProvidesFile#1{%
446     \def\snap@checker{\snap@check{#1}}%
447     \begingroup
448         \aftergroup\snap@checker
449     \catcode'\ 10

```

Added guards from 2001/06/01 version of L<sup>A</sup>T<sub>E</sub>X. These are necessary because, for example, `inputenc` sets `\endlinechar` to a large (nonvalid character) value when reading input encoding files. The guards prevent an “invalid character” error.

```

450     \ifnum\endlinechar < 256
451         \ifnum \endlinechar>\m@ne
452             \catcode\endlinechar 10
453         \fi
454     \fi
455     \@makeother\%
456     \@makeother\&%
457     \kernel@ifnextchar[{\snap@providesfile{#1}}{\snap@providesfile{#1}[]}]%
458 }

```

`\snap@graphic@test` Normally the string found in the second arg of `\ProvidesFile` (for a nongraphic file) would begin with the usual date string. The `\includegraphics` command, however, begins the second arg with `Graphic file` instead. This test therefore just checks if the first two letters are `Gr`; this is enough, ordinarily, for us to conclude that we are dealing with a graphic file.

```

459 \def\snap@graphic@test#1#2#3\@nil{r\if G#1#2\else X\fi}

```

`\snap@providesfile`

```

460 \def\snap@providesfile#1[#2]{%
461     \wlog{File: #1 #2}%

```

Adopt a suggestion made by user `egrep` on the T<sub>E</sub>X stack exchange (<https://tex.stackexchange.com/questions/508985>) but without the use of the `\expanded` extension. This makes it more likely that `snapshot` can deal with macros inside the optional arguments of `\Provides...` commands.

```

462     \edef\@tempa{#2}%
463     \if\@xp\snap@graphic@test\@tempa @@\@nil
464         \snap@record@graphic#1\relax #2 (type ??)\@nil
465     \else
466         \@xp\xdef\csname ver@#1\endcsname{#2}%
467     \fi
468     \endgroup
469 }

```

This is what `\includegraphics` does to record graphic file information.

```

\@providesfile #1[#2]->
\wlog {File: #1 #2}\expandafter \xdef \csname ver@#1\endcsname {#2}
\endgroup
#1<-\Gin@base \Gin@ext
#2<-Graphic file (type eps)

```

`\snap@record@graphic` Check the graphics info.

```

470 \def\snap@record@graphic#1\relax #2(type #3)#4\@nil{%
471     \expandafter\xdef\csname ver@#1\endcsname{%

```

```

472     Graphic%
473     \if\snap@bbinfo :bb=\Gin@llx/\Gin@lly/\Gin@urx/\Gin@ury\fi
474     \space v0.0%
475 }%
476 }

```

\@pr@videpackage

```

477 \def\@pr@videpackage [#1]{%
478   \expandafter\xdef\csname ver@\@currname.\@current\endcsname{#1}%
479   \ifx\@current\@clsextension
480     \typeout{Document Class: \@gtempa\space#1}%
481   \else
482     \wlog{Package: \@gtempa\space#1}%
483   \fi
484   \snap@check{\@currname.\@current}%
485 }

```

\snap@selfcheck@a

```

486 \def\snap@selfcheck@a{\snap@check{snapshot.sty}}

```

\@nofmt

```

487 \def\@nofmt#1.fmt.#2 {#1 }

```

\snap@mismatch@a

```

488 \def\snap@mismatch@a#1#2#3{%
489   \snap@mismatch{snapshot}{^^J%
490     \space\space Required version #2 of \@nofmt#1.fmt. and^^J%
491     \space\space provided version #3 do not match%
492   }\@ehc
493 }

```

\snap@compare@versions When comparing \rqv@foo.sty (information from a previous L<sup>A</sup>T<sub>E</sub>X run) with \ver@foo.sty (information from current run), we first call \snap@trim@version on the latter to clear away any idiosyncrasies in the contents.

```

494 \def\snap@compare@versions#1{%
495   \begingroup
496     \ifundefined{rqv#1}{%
497       \edef\0{\csname rqv#1\endcsname}%
498       \edef\1{\csname ver#1\endcsname}%
499       \edef\1{\@xp\snap@trim@version\1 v0.0 v0.0 \@nil}%
500       \edef\@tempa{\@xp\snap@select\0 v0.0 v0.0 \@nil}%
501       \edef\@tempb{\@xp\snap@select\1 v0.0 v0.0 \@nil}%
502       \ifx\@tempa\@tempb \else
503         \edef\@tempd{\@nx\snap@mismatch@a{#1}{\@tempa}{\@tempb}}%
504         \@xp\@tempd
505       \fi
506     }%
507   \endgroup

```

When the `test` option is in effect, jump out of the current file instead of continuing.

```
508   \snap@test@abort
509 }
```

```
\snap@test@abort
```

```
510 \let\snap@test@abort=\@empty
```

## 6 Compatibility

Suppose that I have a  $\text{\LaTeX}$  document containing a `\RequireVersions` statement generated by `snapshot` and I send this to my colleague who, we believe, has a  $\text{\LaTeX}$  setup that is for our purposes identical. Suppose that our belief is erroneous in the following way: My colleague has a newer version of `snapshot` and a newer version of one of the affected files.

Here is what we **don't** want to happen: That the differing version of `snapshot` would cause the other differing file to be accepted without demur.

Conversely, if it is I who have the newer version of `snapshot`, the main concern is that some difference in the contents of the `\RequireVersions` statement would lead to an error when my colleague attempts to process the document.

## 7 In conclusion

```
511 \ifx\snap@select\@empty
512   \let\snap@compare@versions\@gobble
513   \let\snap@check\@gobble
514 \fi
```

Fallback for a command that is sometimes used in AMS journal production.

```
515 \providecommand{\controldates}[1]{}
```

## 8 And finally ...

If the embedded `\RequireVersions` data in a  $\text{\LaTeX}$  document is extracted to a separate file, and

```
\RequirePackage[test]{snapshot}
```

is added at the top, then the file can be run as a small separate  $\text{\LaTeX}$  job that will, among other things, produce in the log file a nice list of fully resolved file names—sort of a limited, but system-independent variant of the `kpsewhich` idea.

```
516 \ifx\snap@intest\@gobbletwo \endinput \fi
```

Some old, ill-behaved packages might throw in a `\makeatother` at the end which can cause problems for the next file that comes along when testing.

```
517 \def\restore@some@catcodes{}
518 \def\save@some@catcodes{%
519   \edef\restore@some@catcodes{%
520     \catcode\number'\@=\number\catcode'\@
521     \catcode\number'\="=\number\catcode'\"
```

```

522     \catcode\number'\^=\number\catcode'\^
523     \catcode\number'\_=\number\catcode'\_
524     \relax
525   }%
526 }

```

Some typical calls of `\snap@intest`:

```

\snap@intest{LaTeX2e}{\snapx@format}
\snap@intest{snapshot}{\snapx@package}
\snap@intest{mcom-1}{\snapx@class}
\snap@intest{amsmath}{\snapx@package}
\snap@intest{umsa.fd}{\snapx@file}
\snap@intest{pictex}{\snapx@file}

```

The extant public versions of the following files (in the teTeX distribution, at least) are known to be problematic when we are trying to read a `\ProvidesWhatever` line from the top of the file: `psfig.sty`, `pictex.sty`, `pictex.tex`, `epic.sty`, `amstex.sty`, `xy.tex`. Either (a) they don't have a `\ProvidesWhatever` line at all, or (b) they include some code before the `\ProvidesWhatever` line that makes some assumption true in normal processing but false in snapshot-test processing. E.g., there is code in `amstex.sty` that assumes `\documentstyle` or `\documentclass` was already executed and that the `\if@compatibility` switch got set accordingly.

`\snap@intest`

```

527 \def\snap@intest#1#2{%
528   \message{^^J}%
529   \begingroup
530     \edef\O{#1#2}%
531     \def\9{latex209.def}%
532     \ifx\O\9\global\@compatibilitytrue \fi
533     \ifx#2\snapx@format

```

If `arg1 + arg2 = "LaTeX2e.fmt"`, the calling function `\snap@storeb` will run `\snap@check` separately. This is a crude way of making things work in that case without much extra trouble.

```

534       \def\snap@test@abort{\endgroup}%
535     \else
536       \edef\N{%
537         \noexpand\snap@intest@b{#1#2}%
538         {#1}{\@xp@gobble#2\@empty}%
539         {\cname rqv@#1#2\endcsname}}%
540     \expandafter\endgroup\N
541   \fi
542 }

```

`\snap@intest@b`

```

543 \def\snap@intest@b#1#2#3#4{%
544   \def\@currname{#2}%
545   \def\@currtext{#3}%

```

```

546 \begingroup
547 \lccode'\/'\0\relax\lowercase{\endgroup
548 \ifnum\snap@seldate#4 00 0\@nil>\z@
549 }% matches \lowercase
550 \save@some@catcodes
551 \@@input #1 \relax
552 \restore@some@catcodes
553 \else
554 \snap@specialtest{#1}{#4}%
555 \fi
556 }

```

\snap@specialtest

```

557 \def\snap@specialtest#1#2{%
558 \fake@input{#1}%
559 }

```

\fake@input

```

560 \def\fake@input#1{%
561 \begingroup
562 % Ensure that outer \foo or unmatched braces don't trip us up
563 \catcode'\=12
564 \catcode'\{=12
565 \catcode'\}=12
566 \endinput

```

Note that these definitions of \G and \? are local, and recall that one-letter cs names don't use up hash table entries.

```

567 \def\G{\@car\endgroup}%
568 \expandafter\futurelet\expandafter\?\expandafter\G\@@input#1 \relax\@nil
569 }

```

```

570 \let\snap@test@abort=\endinput
571 \let\snap@selfcheck=\@empty

```

\snap@finish There's an extra close-brace left hanging around at the end, but I guess we don't care.

```

572 \def\snap@finish{%%
573 \endgroup
574 \message{^^J}%
575 \def\X##1{##1,\X}%
576 \edef\@filelist{\@xp\X\rqv@list{\@gobbletwo}}%
577 \def\X##1,?{##1}\edef\@filelist{\@xp\X\@filelist ?}%
578 \@dofilelist
579 \@@end
580 }

```

```

581 \def\snap@mismatch#1#2#3{}

```

Problematic: `xy.sty`, because it calls `xy.tex` before it calls `\ProvidesPackage`.  
And `pictex.tex` because it doesn't use `\ProvidesFile` at all.

```
582 \renewcommand{\RequireVersions}[2] [] {%
583   \begingroup
584     \makeatletter
585     \def\snap@check{\snap@compare@versions}%%
586     \let\snapx@tfm=\snap@ignore
```

This seems to help, with `english.lda` for example, to prevent an endless loop when attempting to load `babel.def`.

```
587     \def\ProvidesLanguage##1{\ProvidesFile{##1.lda}}%
588     \iffalse{\fi \futurelet\@let@token\snap@branch #2}%
589   \endgroup
590 }
```

## 9 Finale

The usual `\endinput` to ensure that random garbage at the end of the file doesn't get copied by `docstrip`.

```
591 \endinput
```

## Index

Numbers written in *italics* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

	<b>Symbols</b>				
<code>\"</code>	..... 521	<code>\fmettextension</code>	... \	..... 563	
<code>\&amp;</code>	..... 456	..... 175, 179, 302	<code>\{</code>	..... 564	
<code>\@end</code>	..... 184, 579	<code>\gobblefour</code>	..... 23	<code>\}</code>	..... 565
<code>\@input</code>	. 359, 551, 568	<code>\@tempa</code>	.... 480, 482	<code>\^</code>	..... 522
<code>\@undefined</code>	.....	<code>\makeother</code>	.. 455, 456	<code>\_</code>	..... 523
... 53, 55, 65,		<code>\nofmt</code>	..... <u>487</u> , 490		
72, 73, 80, 98,		<code>\nx</code>	..... <u>3</u>	<b>Numbers</b>	
108, 126, 128, 156		<code>\onlypreamble</code>	... 19, 45, 50, 207	<code>\0</code>	497, 500, 530, 532, 547
<code>\@car</code>	..... 5, 567	<code>\pkgextension</code>	... 177, 268, 298	<code>\1</code>	..... 498, 499, 501
<code>\@charlb</code>	..... 22, 376	<code>\pr@videpackage</code>	. <u>477</u>	<code>\9</code>	..... 531, 532
<code>\@charrb</code>	..... 384	<code>\sptoken</code>	..... 34	<code>\_</code>	..... 449
<code>\@clsextextension</code>	... 178, 269, 300, 479	<code>\tempd</code>	295, 308, 321,		
<code>\@compatibilitytrue</code>	532	322, 324, 503, 504		<b>A</b>	
<code>\@current</code>	.....	<code>\tempe</code>	..... 311,	<code>\afterassignment</code>	.. 15
478, 479, 484, 545		314–316, 318, 324		<code>\aftergroup</code>	.....
<code>\@currname</code>	478, 484, 544	<code>\tempswafalse</code>	129, 140	326, 380, 392, 448	
<code>\@dofilelist</code>	. <u>361</u> , 578	<code>\tempswatru</code>	.... 131	<code>\AlephVersion</code>	.. 55, 60
<code>\@filelist</code>	253, 280,	<code>\tfmextension</code>	... 176, 180, 304	<code>\AtBeginDocument</code>	.
283, 363, 576, 577		<code>\xp</code>	..... <u>3</u>	..... 252, 439	
<code>\@firstoftwo</code>	..... 426			<b>B</b>	
		<code>\bgroup</code>	..... 21, 28		



- C**
- \controldates . . . . 515
- D**
- \DeclareOption 210, 213, 217, 218, 223, 228, 238, 247, 248, 250, 257
  - \delimiter . . . 293, 340, 368, 380, 392
  - \divide . . . . 111, 134
- E**
- \egroup . . . . . 30
  - \endlinechar . . 450–452
  - \epTeXversion . . . . . 73, 76, 80, 91
  - \eTeXminorversion 161
  - \eTeXrevision 160, 163
  - \eTeXversion . . . . . 128, 156, 159
  - \ExecuteOptions . . 265
- F**
- \fake@input . . 558, 560
  - \filename@area . . 321
  - \filename@base 312, 321
  - \filename@ext . . . . 296, 298, 300, 302, 304, 308, 313
  - \filename@parse . . 294
  - \fmtname . . . . 278–280, 283, 285, 287
  - \fmtversion . . . . . 286
  - \foo . . . . . 562
- G**
- \G . . . . . 567, 568
  - \general@files . . . . 352, 355, 363, 365, 380, 424, 425, 427, 430, 434
  - \Gin@llx . . . . . 473
  - \Gin@lly . . . . . 473
  - \Gin@urx . . . . . 473
  - \Gin@ury . . . . . 473
- I**
- \if@tempwa . . . . . 144
- K**
- \kernel@ifnextchar 457
- L**
- \L 396, 419, 421, 423, 425
  - \lccode . . . . . 547
  - \listfiles . . . . . 290
  - \lowercase 35, 547, 549
  - \luatexrevision . . 116
  - \luatexversion . . . . . 108, 110, 114
- M**
- \m@ne . . . . . 451
  - \message . . . . 528, 574
  - \multiply . . . 113, 136
- N**
- \N 395, 397, 414, 418, 422, 427, 428, 431, 433, 536, 540
  - \NeedsTeXFormat . . . . 1
  - \number . 68, 69, 76, 83, 89, 91, 101, 159, 161, 520–523
- O**
- \OmegaVersion 53, 57, 60
- P**
- \PackageError 22, 42, 224
  - \PackageWarningNoLine . . . . . 221
  - \pdftexrevision . . 142
  - \pdftexversion . . . . . 126, 133, 137
  - \ProcessOptions . . 266
  - \ProvidesFile 445, 587
  - \ProvidesLanguage 587
  - \ProvidesPackage . . . 2
  - \ptexminorversion . 69
  - \ptexrevision . . . . 70
  - \ptexversion . . . 65, 68
- R**
- \RequireVersions . . . . 10, 376, 406, 582
  - \reserved@c . . . . . 33, 35, 37, 39
  - \restore@some@catcodes . . . . 517, 519, 552
- S**
- \save@some@catcodes . . . . . 518, 550
  - \sixt@on . . . . . 209
  - \snap@bbinfo . . . . . 249, 250, 473
  - \snap@bracify 329, 363
  - \snap@branch . . . . . 25, 29, 204, 588
  - \snap@check . . . . 13, 27, 198, 446, 484, 486, 513, 585
  - \snap@checker 446, 448
  - \snap@compare@versions . 13, 494, 512, 585
  - \snap@doit 291, 380, 392
  - \snap@fda . . . 343, 348
  - \snap@fdcheck 339, 368
  - \snap@finish . . . . . 28, 31, 188, 572
  - \snap@graphic@test . . . . . 459, 463
  - \snap@ignore . . . . 586
  - \snap@intest . . 197, 216, 217, 516, 527
  - \snap@intest@b 537, 543
  - \snap@mismatch 221, 224, 229, 260, 405, 410, 489, 581
  - \snap@mismatch@a . . . . . 488, 503
  - \snap@mismatch@warning . . . . 221, 229, 260
  - \snap@out 208, 209, 211, 373, 374, 386
- \rqv@compare@lists** . . . . . 370, 416
- \rqv@condend** . . . . . 402, 420, 424
- \rqv@condense** . . . . . 394, 420, 424
- \rqv@list** 195, 277, 369, 420, 421, 429, 432, 435, 576
- \rqv@notloaded** 409, 431
- \rqv@overloaded** 404, 427
- \rqv@set** . . . . . 414, 427, 429, 432, 434
- \rqv@test** 414, 430, 435

- \snap@pad ... [271](#), [322](#)
  - \snap@pad@a .. [271](#), [272](#)
  - \snap@pad@b .. [272](#), [273](#)
  - \snap@providesfile
    - ..... [457](#), [460](#)
  - \snap@record@graphic
    - ..... [464](#), [470](#)
  - \snap@restore@extensions
    - ..... [267](#), [362](#)
  - \snap@seldate ...
    - ... [235](#), [238](#), [548](#)
  - \snap@select .....
    - ... [226](#), [231](#),
    - [233](#), [238](#), [247](#),
    - [248](#), [500](#), [501](#), [511](#)
  - \snap@select@all .
    - ... [226](#), [231](#), [233](#)
  - \snap@select@major
    - ..... [237](#), [243](#)
  - \snap@select@version
    - ..... [236](#), [239](#)
  - \snap@selfcheck ..
    - ... [200](#), [225](#),
    - [230](#), [255](#), [258](#), [571](#)
  - \snap@selfcheck@a
    - ..... [225](#),
    - [230](#), [256](#), [261](#), [486](#)
  - \snap@selmajor [235](#), [248](#)
  - \snap@selversion .
    - ..... [235](#), [247](#)
  - \snap@specialtest
    - ..... [554](#), [557](#)
  - \snap@specific [382](#), [390](#)
  - \snap@splitter [251](#), [364](#)
  - \snap@splitter@a .
    - ..... [332](#), [364](#)
  - \snap@store@b . [47](#), [186](#)
  - \snap@store@error
    - ..... [37](#), [41](#)
  - \snap@store@version
    - ..... [33](#), [46](#)
  - \snap@storem ... [15](#), [20](#)
  - \snap@sva ... [240](#), [242](#)
  - \snap@svm ... [244](#), [246](#)
  - \snap@test@abort .
    - 508, [510](#), [534](#), [570](#)
  - \snap@trim@b . [274](#), [275](#)
  - \snap@trim@c . [275](#), [276](#)
  - \snap@trim@version
    - ... [274](#), [318](#), [499](#)
  - \snap@write .....
    - [208](#), [211](#), [214](#),
    - [324](#), [372](#), [376](#),
    - [377](#), [384](#), [385](#), [391](#)
  - \snapshotApplication
    - ..... [51](#), [378](#)
  - \snapx@application
    - ..... [182](#), [192](#)
  - \snapx@class ..... [178](#)
  - \snapx@end [184](#), [185](#), [187](#)
  - \snapx@file ..... [183](#)
  - \snapx@format ....
    - ... [179](#), [198](#), [533](#)
  - \snapx@ignore [181](#), [182](#)
  - \snapx@package ... [177](#)
  - \snapx@tfm .. [180](#), [586](#)
  - \SpecialInput .... [357](#)
  - \specific@files ..
    - ... [334](#), [336](#),
    - [350](#), [355](#), [358](#),
    - [366](#), [367](#), [381](#), [392](#)
  - \str@cmp ..... [4](#)
  - \str@equal ..... [4](#)
  - \string@equal .. [7](#), [199](#)
- T**
- \T [426](#), [427](#), [429](#), [432](#), [434](#)
  - \typeout . [214](#), [219](#),
  - [372](#), [385](#), [387](#), [480](#)
- U**
- \u ..... [69](#), [70](#), [84](#), [90](#)
  - \undefined [69](#), [70](#), [84](#),
  - [90](#), [102](#), [160](#), [161](#)
  - \uptexrevision . [84](#), [90](#)
  - \uptexversion [72](#), [83](#), [89](#)
- W**
- \wlog ..... [461](#), [482](#)
- X**
- \X ..... [575](#)–[577](#)
  - \XeTeXrevision ... [102](#)
  - \XeTeXversion . [98](#), [101](#)