

# The nag package<sup>\*</sup>

Ulrich Michael Schwarz<sup>†</sup>

February 10, 2008

## Abstract

Old habits die hard. All the same, there are commands, classes and packages which are outdated and superseded. `nag` provides routines to warn the user about the use of those. As an example, we provide an extension that detects many of the “sins” described in `l2tabu`.

## Contents

<b>1</b>	<b>User-side considerations.</b>	<b>2</b>
1.1	Installation. . . . .	2
1.2	Usage. . . . .	2
1.3	Known bugs . . . . .	2
1.4	<code>nag-l2tabu.cfg</code> . . . . .	2
1.5	<code>nag-orthodox.cfg</code> . . . . .	7
1.6	<code>nag-abort.cfg</code> . . . . .	7
1.7	<code>nag-experimental.cfg</code> . . . . .	8
<b>2</b>	<b>Author-side considerations and implementation.</b>	<b>9</b>
2.1	Low-level tools. . . . .	10
2.2	Obsoletifying commands. . . . .	11
2.3	Obsoletifying packages and classes. . . . .	12
2.4	Common float errors and no-nos. . . . .	13
<b>3</b>	<b>Switch vs. Environment</b>	<b>15</b>
<b>4</b>	<b>Compatibility issues</b>	<b>18</b>
4.1	The caption package . . . . .	18
4.2	The subfig package . . . . .	18
4.3	The float package . . . . .	22
4.4	The topcapt package and the subfig package . . . . .	22
4.5	The rotating package . . . . .	22
4.6	Version control packages . . . . .	22

---

<sup>\*</sup>This document corresponds to `nag` 0.61, dated 2008/02/10. Other versions can be found at <http://absatzen.de/>

<sup>†</sup>[ulmi@absatzen.de](mailto:ulmi@absatzen.de)

## 1 User-side considerations.

### 1.1 Installation.

Process `nag.ins` with  $\text{\LaTeX}$  to obtain some files: `nag.sty` and `nag-l2tabu.cfg` et al. must go to a place where  $\text{\LaTeX}$  will find them, like the local TEXMF tree. (If all else fails and you need it to work *right now*, having them in the same directory as the  $\text{\LaTeX}$  file you want to use them on may work under many circumstances.) You can, as usual, run  $\text{\LaTeX}$  on `nag.dtx` to obtain this documentation, including the implementation docs. (This is recommended if you plan to extend `nag` to handle your own packages.) `nagdemo.tex` is a horrible document that will show you many of the warnings that `nag` can generate.

### 1.2 Usage.

Add the following to the beginning your main document (Comments and `\listfiles` can be safely left before it, though):

```
\RequirePackage[l2tabu, orthodox]{nag}
```

This will check for many common mistakes, and give some hints on what to use instead. However, you should always refer to `l2tabu` for a more detailed explanation of the whats and whys: it gives more information than can be possibly pressed into two lines of error message. Orthodox checks for pitfalls that are not technically incorrect. If you know what you're doing, omit `orthodox`.

### 1.3 Known bugs

currently none.

### 1.4 `nag-l2tabu.cfg`

In a nutshell, `nag-l2tabu.cfg` detects the following:

- Usage of the 2.09-style font commands `\it`, `\bf`, `\rm`, `\sc`, `\sl`, `\tt` and `\cal`.
- Usage of `\centerline`.
- Usage of the outdated packages `epsfig`, `psfig`, `epsf`, `doublespace`, `fancyheadings`, `scrpage`, `umlaut`, `isolatin`, `isolatin1`, `t1enc`, `caption2`, `psfonts`, `mathptm`, `times`, `palatino`, `mathpple`, `euler` and `utopia`, and of the outdated class `scrlltr`.
- Figures and tables without caption (this is not technically in `l2tabu`, but the people who have floats without captions tend to ask “Why is  $\text{\LaTeX}$  moving my pictures away from where I put them?”), labels within floats that do not reference the caption, and usage of the `center` environment within floats.

It is beyond the possibilities of this package to detect things like use of  $\TeX$  assignment syntax, or direct change of paper parameters, or reliable detection of user-issued `\sloppy`. `eqnarray` is handled as of 0.60alpha4, and there is code for `$$` in experimental since 0.60alpha4, which has been moved to `l2tabu` in 0.60.

**Be warned, that this package will possibly balk at legitimate use, and not find illegitimate use in all cases. It is a tool, not a replacement for study of `l2tabu`.**

```

1 \ProvidesFile{nag-l2tabu.cfg}
2       [2008/01/27 v2.1alpha1 l2tabu rules for nag.sty (ulmi)]
3 %%
4 %% The sins.
5 %%
6 %% Section numbers refer to l2tabuen 1.7 revised/enlarged dated 2004OCT24
7 %% \S 1.1
8 \ObsoletePackage{a4wide}{the \lq a4paper\rq\space class option}
9 \ObsoletePackage{a4}{the \lq a4paper\rq\space class option}
10 %% \S 1.2--1.5 cannot reasonably be checked programmatically
11 %% \S 1.6

```

Hacking galore ahead! We will make the dollar active. Since unlike `onlyamsmath`, we do not change the user's command to  $\TeX$  or `amsmath` commands, we need to store the old double dollar sequence as well as the single dollar.

```

12 \def\nag@doubledollar{$$}%$$
13 \def\nag@singledollar{$}%$

```

This is used to hide our redefinition in unprotected expanding context. This should not happen: you are expected to *always* use protected means of expansion in  $\TeX$ , but fecal matter happens. See below for a good trick to distinguish expansion from executing context.

```

14 \def\nag@expanding@voodoo#1#2#3{\relax\relax\nag@singledollar}
15
16 \def\nag@maybedispmath{%
17   \texorpdfstring{%
18     %% in TeX context, do tricky stuff.
19     \ifinner\expandafter\@firstoftwo
20     \else\expandafter\@secondoftwo\fi
21     {%% in inner mode, $$ is an empty formula, so no testing wanted.
22       \nag@singledollar}%
23   }%%
24   \ifx\protect\@typeset@protect\expandafter\@firstoftwo
25   \else\expandafter\@secondoftwo\fi
26   {%% normal case: looks like typesetting
27     %% protect against strictly expanding context
28     %% like TeX' \message: the first expanding voodoo will expand,
29     %% removing the rest, inserting \relax\relax$ instead. This is
30     %% not totally transparent, but \let\relax\relax is as close
31     %% to a no-op as we can get.
32     \let\nag@expanding@voodoo\nag@expanding@voodoo
33     \protect\nag@maybe@dispmath}%
34   }%% some other case, hide ourselves

```

```

35     \nag@singledollar}%
36   }%
37 }{%
38   %% in pdf context, just be a math shift. This creates the "math
39   %% shift not allowed" warnings we all love.
40   \nag@singledollar
41 }%
42 }

```

If the user doesn't load hyperref, we have to fake its `\texorpdfstring` command. Note that this will break any package that is foolish enough to detect hyperref by testing for definedness of `\texorpdfstring`.

```

43 \AtBeginDocument{\providecommand\texorpdfstring{\@firstoftwo}}
44 \AtBeginDocument{\catcode'\$active}%\$

```

Now, the proper testing. (Yes, the above is just the technicalities.) We use the kernel's `\@ifnextchar` to look for a possible second dollar. Note however, this would allow skipping of spaces between them, and `$_$` is not a displayed equation start in  $\TeX$ . We work around this by re\letting `\@sptoken` to something that cannot legally appear in the source.

```

45 \bgroup
46   \catcode'\$active%$
47   \gdef\nag@maybe@dispmath{%
48     \bgroup
49     \let\@sptoken\nag@quark% prevent skipping of spaces
50     \@ifnextchar${}%$%
51     \ifmmode
52       % we already warned upon entering.
53     \else
54       \nag@warn{%
55         \nag@doubledollar...\nag@doubledollar\space is obsolete.\MessageBreak
56         Use \string\[\...\string\] et al. instead}%
57     \fi
58     \egroup\expandafter\nag@doubledollar\@gobble
59   }{%
60     \egroup\nag@singledollar
61   }%
62 }
63 % we do the assignment here, which means any package that redefines
64 % \$ as well will silently disable us. This is a feature.
65 \global\let$\nag@maybe@dispmath%$
66 \egroup

```

new in 2.1alpha1: more compat testing. Version control keywords are dollar-delimited. all five implementations get it wrong.

```

67 \AtBeginDocument{%
68   \@ifpackageloaded{rcs}{%
69     % this redefinition is functionally equivalent,
70     % but does not share actual code.
71     \renewcommand\RCS{\bgroup%

```

```

72     \catcode'\_ =\active
73     \catcode'\$=3 % this line added for compatibility.
74     \csname RCS_get_argument\endcsname
75 }
76 \PackageInfo{nag}{rcs.sty hack applied}%
77 }{}%
78 \@ifpackageloaded{svninfo}{%
79   \g@addto@macro\@svnBeginRead{\catcode'\$ 3}%
80   \PackageInfo{nag}{svninfo.sty hack applied}%
81 }{}%
82 \@ifpackageloaded{svn}{%
83   \PackageInfo{nag}{svn.sty is broken: disabling dollar check}%
84   \catcode'\$ 3
85 }{}%
86 \@ifpackageloaded{rcsinfo}{%
87   \PackageInfo{nag}{rcsinfo.sty is broken: disabling dollar check}%
88   \catcode'\$ 3
89 }{}%
90 \@ifpackageloaded{pgf}{%
91   \PackageInfo{nag}{pgf.sty is broken: disabling dollar check}%
92   \catcode'\$ 3
93 }{}%
94 }
95
96
97%% \S 1.7 cannot reasonably be checked programmatically
98%% \S 1.8 \sloppy is called by parbox, among others, and would
99%% give many spurious warnings.
100%% \S 2.1.1
101\ObsoleteCS[an old LaTeX 2.09 command]{bf}
102    {\protect\bfseries\space or \protect\textbf}
103\ObsoleteCS[an old LaTeX 2.09 command]{it}
104    {\protect\itshape\space or \protect\textit}
105\ObsoleteCS[an old LaTeX 2.09 command]{rm}
106    {\protect\rmfamily\space or \protect\textrm}
107\ObsoleteCS[an old LaTeX 2.09 command]{sc}
108    {\protect\scshape\space or \protect\textsc}
109\ObsoleteCS[an old LaTeX 2.09 command]{sf}
110    {\protect\sffamily\space or \protect\textsf}
111\ObsoleteCS[an old LaTeX 2.09 command]{sl}
112    {\protect\slshape\space or \protect\textsl}
113\ObsoleteCS[an old LaTeX 2.09 command]{tt}
114    {\protect\ttfamily\space or \protect\texttt}
115\ObsoleteCS[an old LaTeX 2.09 command]{cal}
116    {\protect\mathcal}% Hmm, this is not in l2tabu?
117%% \S 2.1.2
118%% Gone with 1.8 because this never worked for the kernel \frac anyway.
119%% \ObsoleteCS[TeX]{over}{\protect\frac}
120%% \ObsoleteCS[TeX]{choose}{\protect\frac\space or amsmath's \protect\binom}
121%% \S 2.1.3

```

```

122 \ObsoleteCS[TeX]{centerline}{\protect\centering\space or center environment}
123 %% \S 2.2.1
124 \ObsoleteClass{scrlettr}{the scrlettr2 package}
125 %% \S 2.2.2
126 \ObsoletePackage{epsf}{the graphicx package}
127 \ObsoletePackage{psfig}{the graphicx package}
128 \ObsoletePackage[deprecated]{epsfig}{the graphicx package directly}
129 %% \S 2.2.3
130 \ObsoletePackage{doublespace}{the setspace package}
131 %% \S 2.2.4
132 \ObsoletePackage{fancyheadings}{the fancyhdr or scrpage2 packages}
133 \ObsoletePackage{scrpage}{the scrpage2 package}
134 %% \S 2.2.5
135 \ObsoletePackage{isolatin}{the inputenc package with option latin1}
136 \ObsoletePackage{umlaut}{the inputenc package with suitable option
137   (latin1, utf8 ...)}
138 \ObsoletePackage{isolatin1}{the inputenc package with option latin1}
139 %% \S 2.2.6
140 \ObsoletePackage{tlenc}{the fontenc package with option T1}
141 %% \S 2.2.7 we don't check for bst yet.
142 %% (This is in l2tabu 1.8)
143 \ObsoletePackage{caption2}{the caption package v3.0 or later}
144 %% \S 2.3.1-3
145 \ObsoletePackage{times}
146           {the mathptmx, helvet (option scaled=.9), courier packages}
147 \ObsoletePackage{pslatex}
148           {the mathptmx, helvet (option scaled=.9), courier packages}
149 \ObsoletePackage{mathptm}
150           {the mathptmx package}
151 %% \S 2.3.4-5
152 \ObsoletePackage{palatino}
153           {the mathpazo, helvet (option scaled=.95), courier packages}
154 \ObsoletePackage{mathpple}{the mathpazo package}
155 %% \S 2.3.6 can't be checked
156 %% \S 2.3.7
157 \ObsoletePackage{euler}{the eulervm package}
158 \ObsoletePackage{utopia}{the fourier package}
159 %% \S 3.1
160 \NagDeclareFloat{figure}\NagDeclareFloat{table}%
161 \g@addto@macro\nag@labels{,label}%
162 % \changes{0.60}{2007/03/31}{alternate center-in-float check, doesn't
163 % take up as many macro names}
164 \nag@prepend{endcenter}{%
165   \ifx\@capytype\undefined\else
166     \nag@warn{\lq center\rq\space environment in \@capytype.\MessageBreak
167       Maybe you want \protect\centering\space instead}%
168   \fi
169 }%
170 %% The latter two are used by KOMA-Script, the last by hypcap.
171 % \changes{0.53}{2007/03/21}{hypcap support. (H.G.Krauth\'auser)}

```

```

172% \changes{0.53}{2007/03/21}{topcapt support.}
173 \g@addto@macro\nag@captions{,caption,captionabove,captionbelow,hc@caption,topcaption}%
174
175%% \S 3.2
176 \NotAnEnvironment{appendix}%
177%% In the same vein:
178 \@for\sectioning:=frontmatter,mainmatter,backmatter\do{%
179   \expandafter\NotAnEnvironment\expandafter{\sectioning}%
180 }
181%% \S 3.3
182%% It's more trouble than it's worth to have another warning for
183%% align*, since it passes through align.
184 \ObsoleteEnv{eqnarray}{amsmath's align}
185%% \S 3.4 -- nothing to be done --

```

## 1.5 nag-orthodox.cfg

nag-orthodox.cfg warns about usage that is not technically incorrect, but will mostly do things an unwary user may not expect. This includes in particular the usage of font size and style switches as environments (line spacing will be off if the environment does not contain a trailing \par, spurious spaces might occur since the switches don't \ignorespaces), and, conversely, the usage of center etc. environments as unclosed switches. (Detection of the latter might still be somewhat brittle.)

```

186 \ProvidesFile{nag-orthodox.cfg}
187       [2006/04/19 v1.8 strict rules for nag.sty (ulmi)]
188 \@for\fontcmd:=tiny,small,footnotesize,normalsize,large,Large,%
189       LARGE,huge,Huge\do{%
190   \expandafter\NotAnEnvironment\expandafter{\fontcmd}%
191 }%
192 \@for\fontcmd:=sffamily,rmfamily,ttfamily,%
193       bfseries,mdseries,scshape,%
194       itshape,upshape\do{%
195   \expandafter\NotAnEnvironment\expandafter{\fontcmd}%
196 }%
197 \@for\justsw:=centering,raggedleft,raggedright,%
198       RaggedLeft,RaggedRight\do{%
199   \expandafter\NotAnEnvironment\expandafter{\justsw}%
200 }
201 \@for\justenv:=center,flushleft,flushright\do{%
202   \expandafter\NotASwitch\expandafter{\justenv}%
203 }

```

## 1.6 nag-abort.cfg

Requesting this nag file will turn all complaints into errors.

```

204 \ProvidesFile{nag-abort.cfg}
205       [2007/11/10 v0.2 treat complaints as errors (ulmi)]
206 \DeclareRobustCommand\nag@warn[1]{%

```

```

207 \addtocounter{nag@sins}{1}%
208 \PackageError{nag}{#1}{#1}%
209 }
210 \DeclareRobustCommand\nag@warnNoLine[1]{%
211 \addtocounter{nag@sins}{1}%
212 \PackageError{nag}{#1}{#1}%
213 }

```

## 1.7 nag-experimental.cfg

Functionality that needs more testing.

```

214 \ProvidesFile{nag-experimental.cfg}
215 [2008/02/02 v0.1d experimental additions to nag (ulmi)]
Check if a float that may be positioned b is actually small enough for bottomfraction
etc.
216 \let\@xa\expandafter
217 \newif\ifnag@dofloatsizecheck
218 \newcommand\nag@allfloatsizechecks{}%
219 \newcommand\nag@onefloatsizecheck[2]{%
220 % #1 is size fraction of textheight,
221 % #2 is position to say in warning.
222 \ifdim \ht\@currbox>#1\textheight
223 \@tempdima -#1\textheight
224 \advance \@tempdima \ht\@currbox
225 \PackageInfo{nag}{Float too large for #2 by \the\@tempdima}%
226 % note we do not truncate.
227 % also, it's too late to add "p" now.
228 \fi
229 }
230 % \@currbox is current float box,
231 % \@fps is the current list of float specifiers.
232 \renewcommand\@largefloatcheck{%
233 \ifdim \ht\@currbox>\textheight
234 \@tempdima -\textheight
235 \advance \@tempdima \ht\@currbox
236 \@latex@warning {Float too large for page by \the\@tempdima}%
237 \ht\@currbox \textheight
238 \fi
239 %% the preceding is the original check.
240 \nag@dofloatsizechecktrue
241 \def\nag@allfloatsizechecks{}%
242 \@xa\@tfor\@xa\nag@fltsz@tmp\@xa:\@xa=\@fps\do{%
243 \if\nag@fltsz@tmp !
244 \nag@dofloatsizecheckfalse
245 \else
246 \if\nag@fltsz@tmp t
247 \g@addto@macro\nag@allfloatsizechecks
248 {\nag@onefloatsizecheck{\topfraction}{top of page}}%
249 \else

```



```

250     \if\nag@fltsz@tmp b
251     \g@addto@macro\nag@allfloatsizechecks
252         {\nag@onefloatsizecheck{\bottomfraction}{bottom of page}}}%
253     \fi
254 \fi
255 \fi
256 }%
257 \ifnag@dofloatsizecheck\nag@allfloatsizechecks\fi
258 }%

```

More experimental code: warning about files that were requested but not there. The really important one would be a check for include (this is just a typeout in the kernel?!). But as it is, we get warnings that point out missing ToC, LoF etc.

```

259 \def\@input#1{%
260   \IfFileExists{#1}{\@input\@filef@und}{%
261     \typeout{No file #1.}
262     \@latex@warning{File ‘#1’ not found}
263     %{The file ‘#1’ was requested but not found }
264     \protected@edef\nag@nofile{File ‘#1’ requested, but not found}%
265     \@xa\AtEndDocument\@xa{%
266       \@xa\@latex@info@no@line\@xa{%
267         \nag@nofile
268       }%
269     }%
270 }}%
271 %
272 \def\@input@#1{\InputIfFileExists{#1}{}{%
273   \typeout{No file #1.}
274   \@latex@warning{File ‘#1’ not found}
275   {The file ‘#1’ was requested but not found }
276   \edef\nag@nofile{File ‘#1’ requested, but not found}%
277   \@xa\AtEndDocument\@xa{%
278     \@xa\@latex@info@no@line\@xa{%
279       \nag@nofile
280     }%
281   }%
282 }}%
283 %

```

## 2 Author-side considerations and implementation.

If you are a package or class author and want to extend the range of nag (or prevent nag from criticizing your macros), please see the description below, in sections 2.2 and following. It is probably wise to group new rules in a separate nag file: users can request nag files by passing their name as a package parameter, as shown above for the example of l2tabu.

## 2.1 Low-level tools.

Identify ourselves.

```
284 \NeedsTeXFormat{LaTeX2e}
285 \ProvidesPackage{nag}[2008/02/10 0.61 warning about old commands (ulmi)]
286 \let\@xa\expandafter
287 \let\@nx\noexpand
```

First of all, two counters we need. The first is used to generate running numbers for replacement macros, the latter is stepped for each complaint we have, so that the user gets a frighteningly high number, showing how sinful he or she is.

```
288 \newcounter{nag@c}
289 \renewcommand\thenag@c{\roman{nag@c}}%
290 \setcounter{nag@c}{1}%
291 \begingroup
292   \let\@addtoreset\@gobbletwo
293   \newcounter{nag@sins}%
294 \endgroup
```

`\nag@prepend` `\nag@prepend{<cs>}{<something>}`: Prepend *<something>* to the macro definition of *<cs>*.

In reality, we do call indirection: save old macro away, redefine macro to do the something, call old macro. (With thanks to Juergen Goebel, Heiko Oberdiek and Rolf Niepraschk (savesym))

From 0.60 $\alpha_2$  on, nag is more robust about not defining commands that are not there. Now, they're not even relaxed.

```
295 \newcommand\nag@ifundefined[1]{%
296   \begingroup
297   \ifundefined{#1}\endgroup\@firstoftwo\endgroup\@secondoftwo}%
298 }
```

Don't define the macro if it's not there. This confuses caption, which loads ragged2e AtBeginDocument, at which point, RaggedLeft et al. were already defined by us. ... but *do* log a message.

```
299 \newcommand\nag@prepend[2]{%
300   \nag@ifundefined{#1}{%
301     % if it doesn't exist, don't do anything.
302     \PackageInfo{nag}{%
303       Command \@backslashchar#1\space not defined, skipping amendment%
304     }%
305   }{%
306     \nag@ifundefined{#1 }{%
307       \let\nag@maybespace\@empty
308     }{%
309       \let\nag@maybespace\space
310       %\PackageInfo{nag}{%
311         % Command \@backslashchar#1\space appears robust\MessageBreak
312         % Modifying '@backslashchar#1\space' instead.
313         %}%
314     }%
315   }
```

```

315 \@xa\let
316 \csname nag@@#1@\thenag@c\@xa\endcsname
317 \csname #1\nag@maybespace\endcsname
318 \@xa\DeclareRobustCommand\csname nag@@warning@\thenag@c\@xa\endcsname{%
319 #2%
320 }%
321 \@xa\nag@pr@p@nd\csname #1\nag@maybespace\@xa\endcsname
322 \csname nag@@#1@\thenag@c\@xa\endcsname
323 \csname nag@@warning@\thenag@c\@xa\endcsname

```

Fun with scoping: one might think we can get away with a (non-local) `\advance\c@nag@c 1\relax` here. This would lead to less hashtable usage. Problem: if a `nag@@foo@17` macro ever escapes its scope, it might be bound to something else entirely. This might occur with some of the fancier table packages which use external files?

```

324 \stepcounter{nag@c}%
325 }%
326 }
327 \newcommand\nag@pr@p@nd[3]{%
328 \def#1{#3#2}%
329 }

```

`\nag@warn` All complaints to the user run through one of these two macros, with or without source line.

```

330 \DeclareRobustCommand\nag@warn{%
331 \addtocounter{nag@sins}{1}%
332 \PackageWarning{nag}%
333 }
334 \DeclareRobustCommand\nag@warnNoLine{%
335 \addtocounter{nag@sins}{1}%
336 \PackageWarningNoLine{nag}%
337 }

```

## 2.2 Obsoletefying commands.

(No, I do not think that is a proper word either.)

`\obsoleteCS` Usage: `\obsoleteCS[⟨reason⟩]{⟨CS⟩}{⟨suggestions⟩}` Mark `\⟨CS⟩` as obsolete. `⟨reason⟩` defaults to obsolete. When the macro is used anyway, the following warning is logged: Command `\⟨CS⟩` is `⟨reason⟩`. Use `⟨suggestions⟩` instead.

```

338 \newcommand\obsoleteCS[3][obsolete]{%
339 \AtBeginDocument{%
340 \nag@prepend{#2}{%
341 \nag@warn{%
342 Command \@backslashchar#2 is #1.
343 \MessageBreak
344 Use #3 instead}%
345 }%
346 }%
347 }

```

`\ObsoleteEnv`

```
348 \newcommand\ObsoleteEnv[3][obsolete]{%
349   \AtBeginDocument{%
350     \nag@prepend{#2}{%
351       \nag@warn{%
352 Environment #2 is #1.
353 \MessageBreak
354 Use #3 instead}%
355   }%
356 }%
357 }
```

## 2.3 Obsoletifying packages and classes.

Checking for packages and classes is done by looking for `ver@foo.sty`, which holds the version information that is also displayed by `\listfiles`. This means that we're out of luck if fontenc ever becomes obsolete, because that won't be detected.

First, define a macro to check if a control sequence is defined. Unlike `\@ifundefined`, this will not define the control sequence to `\relax`, but the arguments will be executed in a group. For our purposes, this doesn't matter, because we only give a warning (and `\addtocounter` already is `\global`).

```
358 \newcommand\nag@ifcsname[3]{%
359   \begingroup\@ifundefined{#1}{#3}{#2}\endgroup
360 }
```

Just because we can, use  $\epsilon\text{T}_{\text{X}}$  `\ifcsname` if we can. This bootstrapping gives me a big grin... Note we add an extra group for compatibility with the non- $\epsilon$  case.

```
361 \nag@ifcsname{ifcsname}{%
362   \renewcommand*\nag@ifcsname[3]{%
363     \begingroup
364       % assume it won't be there.
365       \let\tmp@a\@secondoftwo
366       \ifcsname #1\endcsname
367       % It still might be relax from some other test. Thanks to J\"org
368       % Sommer for finding this bug.
369       \expandafter\ifx\csname #1\endcsname\relax
370       \else
371         % it's there after all
372         \let\tmp@a\@firstoftwo
373       \fi
374     \fi
375     \tmp@a{#2}{#3}%
376   \endgroup
377 }
```

This way of escaping the grouping gives me an even bigger grin.

```
378 \global\let\nag@ifcsname\nag@ifcsname
379 }{}
```

`\ObsoletePackage` Usage: `\ObsoletePackage[<reason>]{<package>}{<alternative>}`. Mark *<package>* as obsolete. *<reason>* defaults to obsolete. If the *<package>* is used anyway, at the end of the compilation, the following warning will be displayed:  
Package *<package>* is *<reason>*. Use *<alternative>* instead.

```

380 \newcommand\ObsoletePackage[3][obsolete]{%
381   \AtEndDocument{%
382     |\@clsextension| is onlypreamble, for some reason.
383     \nag@ifcsname{ver@#2.sty}{%
384       \nag@warnNoLine{%
385         Package #2 is #1.\MessageBreak
386         Use #3 instead}%
387       }{}%
388     }%
389 }

```

`\ObsoleteClass` Usage: `\ObsoleteClass[<reason>]{<class>}{<alternative>}`. Mark *<class>* as obsolete. *<reason>* defaults to obsolete. If the *<class>* is used anyway, at the end of the compilation, the following warning will be displayed:  
Class *<class>* is *<reason>*. Use *<alternative>* instead.

```

390 \newcommand\ObsoleteClass[3][obsolete]{%
391   \AtEndDocument{%
392     |\@clsextension| is onlypreamble, for some reason.
393     \nag@ifcsname{ver@#2.cls}{%
394       \nag@warnNoLine{%
395         Class #2 is #1.\MessageBreak
396         Use #3 instead}%
397       }{}%
398     }%
399 }

```

## 2.4 Common float errors and no-nos.

We do the following:

- check for presence of a caption
- check for absence of the center environment
- check that a label comes only after a caption

First of all, we define two ifs to memorize whether we have a label and/or a caption in the float already. Package writers may want to set these manually behind nag's back. In this way, they can suppress possible warnings if they know what they're doing – we only check at the end of the float environment, which gives them plenty of time to call `\csname nag@haslabeltrue\endcsname` et al. (Thanks to Markus Kohm for pointing out this need.) We initialize `\nag@hascaption` to be true because since 0.60, `\label` always checks if it's after a caption, even outside of floats.

```

400 \newif\ifnag@haslabel
401 \newif\ifnag@hascaption\nag@hascaptiontrue

```

Now, to the work proper: as of 0.60, it is sufficient to set the label and caption flags to false. `\endcenter` now always checks if it is inside a float (looking at `\@capytype`). The label and caption commands are amended only once. This should be sufficient: captions are not handled by letting `\caption` to the proper command upon float entry, so we assume nobody redefines `\caption` at runtime, or they provide more entries to `\nag@captions`. Similar for `\label`, and we do not care about the flag setting outside of floats.

```

402 \newcommand\nag@hackfloat[1]{%
403   \nag@prepend{#1}{%
404     \global\nag@haslabelfalse\global\nag@hascaptionfalse
405   }%
406   \nag@prepend{end#1}{%
407     \ifnag@hascaption\relax\else
408       \nag@warn%
409     {#1 with no \protect\caption}%
410     \fi
411     % labels outside floats shouldn't complain:
412     \global\nag@hascaptiontrue
413     % (we do this always because it needs to be global)
414   }%
415 }

```

Add checks to all macros named by `\nag@labels` and `\nag@captions`, respectively. Scoping of presence-of-caption information: Well, maybe I should do it the way the kernel does, which means a label is just as local as `\refstepcounter's` `\@currentlabel` information as of v0.4. I think we can leave captions global. Big old hack: we do this at `\@preamblecmds`-time, which is after `\AtBeginDocument`, since `hyperref` loads `nameref` ABD, and `nameref` steps all over `label`. *Note:* We cannot use `\nag@prepend` for this, since it would break the `pkgindoc` package, which nobody has ever heard of, but it's in the kernel and relies on certain tokens being present in the expansion of `\@preamblecmds`. Now, you pretty much cannot get any later than this.

*Note:* we cannot exchange the order of the for loops here: if a cs generates both a label and a caption, it shouldn't get complained about.

```

416 \AtBeginDocument{%
417   \g@addto@macro{\@preamblecmds}{%
418     \@for\labelprovider:=\nag@labels\do{%
419       \ifx\labelprovider\@empty\else
420         \nag@prepend{\labelprovider}%
421         {\nag@captioncheck\nag@haslabeltrue}%
422       \fi
423     }%
424     \@for\captionprovider:=\nag@captions\do{%
425       \ifx\captionprovider\@empty\else
426         \nag@prepend{\captionprovider}{\global\nag@hascaptiontrue}%
427       \fi
428     }%
429   }%
430 }
431 \newcommand\nag@captioncheck{%

```

```

432 \ifnag@hascaption\else
433 \nag@warn{\protect\label\space in float, but not after
434 \protect\caption}%
435 \fi
436 }

```

Define the lists of commands that are floats, generate labels, and generate captions, respectively. We don't start with defined floats (that is for nag-l2tabu.cfg to set up). Since v0.52, we handle an empty name, so the lists may be empty. Also, no labels and captions are provided by default since v0.52. This has been moved to nag-l2tabu.cfg. See also \NagDeclareFloat, which is the user-level wrapper for new floats. Since there are no packages to define new caption or label commands on an user level, there is no wrapper for those.

```

437 \def\nag@floats{}
438 \def\nag@labels{}
439 \def\nag@captions{}

```

We call the above for each float environment named via \nag@floats:

```

440 \newcommand\nag@floatsetup{%
441 \@for\flo:=\nag@floats\do{%
442 \ifx\flo\@empty\else
443 \@xa\nag@hackfloat\@xa{\flo}%
444 \fi
445 }%
446 }

```

but only after all other packages get their chance to add to the list:

```

447 \AtBeginDocument{%
448 \nag@floatsetup
449 }

```

At the very end, we will display a running total of complaints.

```

450 \AtBeginDocument{%
451 \AtEndDocument{%
452 \ifnum\value{nag@sins}>0%
453 \PackageWarningNoLine{nag}{\arabic{nag@sins} complaints
454 in total}%
455 \else
456 \typeout{No complaints by nag.}%
457 \fi
458 }%
459 }

```

### 3 Switch vs. Environment

People often use switches as environments and vice versa. This is dangerous in because it tends to *almost* work. (Consider font size commands in particular, but also \centering vs. center environment.) As usual, “it’s not an error if you know what you’re doing”. In particular, it is perfectly valid code to use the \foo... \endfoo syntax.

So, `\NotASwitch` needs to trace the calls to `\foo` and see if they match with corresponding `\endfoos` with its own stack. This might still be brittle. Fortunately, it is currently only needed for `nag-orthodox`, where it checks for the justification environments.

First of all, a helper macro we hinge upon:

```
460 \DeclareRobustCommand\nag@ifCurrentEnvironment[3]{%
461   \bgroup
462   \def\tmp@a{#1}%
463   \ifx\@currenvir\tmp@a
464     #2%
465   \else
466     #3%
467   \fi
468   \egroup
469 }
```

And now, the two variations there are:

`\NotAnEnvironment` Usage: `\NotAnEnvironment{<command>}` Issue an error if the user calls `\begin{command}` and not `\command` directly.

```
470 \newcommand\NotAnEnvironment[1]{%
471   \AtBeginDocument{%
472     \nag@prepend{#1}{%
473       \nag@ifCurrentEnvironment{#1}{%
474         \nag@warn{%
475           There is no environment ‘‘#1’’. \MessageBreak
476           Maybe you want a grouped \@backslashchar#1
477         }%
478       }{% OK case.
479       }%
480     }%
481   }%
482 }
```

`\NotASwitch` is a bit more involved:

`\NotASwitch` Usage: `\NotASwitch{<command>}` Issue an error if the user calls `\command` and not `\begin{command}` and mis-nests calls or doesn't call `\endcommand` at all.

```
483 % we need to maintain a stack of environments that are used in the
484 % \foo...\endfoo way.
485 \newcommand\nag@envstack{\relax}
486
487 \DeclareRobustCommand\nag@beginenv[1]{%
488   % push a begin-entry onto the stack. Form is
489   % |\foo{lineno}| for environment foo.
490   \bgroup
491   \@xa\toks@\@xa{\nag@envstack}%
492   \xdef\nag@envstack{%
493     \@nx{%
494       \@xa\@nx\csname #1\endcsname
```



```

495     \@nx{\the\inputlineno\@nx}%
496     \@nx}%
497     \the\toks@
498     }%
499 \egroup
500 }
501 \DeclareRobustCommand\nag@endenv[1]{%
502 % extract the first entry.
503 \@xa\nag@endenv\nag@envstack\@nil #1\@nil
504 }
505
506 \def\nag@endenv#1#2\@nil #3\@nil{%
507   \def\tmp@a{#1}%
508   \def\tmp@b{\relax}%
509   \ifx\tmp@a\tmp@b
510     % This was the end-of-stack flag.
511     \nag@warn{'\@backslashchar end#3'} without matching
512     '\@backslashchar #3'}
513   \else
514     % We may assume this is a proper entry. See if the begin-token on
515     % the stack matches what |\nag@endenv| was passed.
516     \@xa\ifx\csname #3\@xa\endcsname\@firstoftwo #1%
517       %OK case, just pop the entry.
518       \gdef\nag@envstack{#2}%
519     \else
520       % error case
521       \nag@warn{%
522         You cannot close '\@xa\string\@firstoftwo #1' on line
523         \@secondoftwo #1 with '\@backslashchar end#3'%
524       }%
525       % leave it on the stack. Some case of misnesting will always cause
526       % horrible amounts of follow-up errors. Also, scare them!
527       \fi
528     \fi
529 }

```

At the end, we complain about all the entries that are still on the stack.

```

530 \AtEndDocument{%
531   \@xa\@tfor\@xa\looseends\@xa:\@xa=\nag@envstack\do{%
532     \@xa\ifx\looseends\relax\else
533       \nag@warnNoLine{Unmatched
534         '\@xa\@xa\@xa\string\@xa\@firstoftwo\looseends'
535         command on line
536         \@xa\@xa\@xa\string\@xa\@secondoftwo\looseends%
537       }%
538     \fi
539   }%
540 }

```

Now, the user-side command is easy.

```

541 \newcommand\NotASwitch[1]{%
542   \AtBeginDocument{%
543     \nag@prepend{#1}{%
544       \nag@beginenv{#1}%
545     }%
546     \nag@prepend{end#1}{%
547       \nag@endenv{#1}%
548     }%
549   }%
550 }

```

## 4 Compatibility issues

### 4.1 The caption package

Axel Sommerfeldt's caption package loads the ragged2e package AtBeginDocument (regardless of whether it is needed). This is too late for us to amend the \RaggedFoo commands with \NotAnEnvironment. Since v0.51 of nag, they will then be skipped (with information in the log). Earlier versions would fail because by time ragged2e was loaded, the commands were already defined by the amendment process. To make sure the commands *are* amended, load ragged2e explicitly yourself.

### 4.2 The subfig package

Starting with v0.52 of nag, we recognize the fact that the \subfloat command from Steven D. Cochran's subfig package is a caption-provider for its fourth argument. Earlier versions would flag use of \label as inappropriate. The current implementation works with versions close enough to v1.3 of subfig. Since the change is a one-liner, I hope it will be integrated into future versions of subfig.

```

551 \AtBeginDocument{%
552   \nag@ifcsname{ver@subfig.sty}{%
553     \PackageInfo{nag}{Attempting subfig hack\@gobble}%
554     \nag@maybehacksubfig
555   }{%
556   }%
557 }
558 \def\nag@maybehacksubfig{%
559   %
560   % of course, i need to touch the single longest definition in
561   % subfig.sty, to amend one single command...
562   %
563   % The definition is taken from subfig.sty 1.3 dated 2005/07/05 by
564   % S.D. Cochran, where it is called sf@@@subfloat, and appears here
565   % under the conditions of section 6 of the LPPL 1.3. The subfig
566   % package is available on a CTAN mirror near you.
567   %
568   \long\def\nag@@original@@sf@@@subfloat##1[##2][##3]##4{%

```

```

569 \ifundefined{FBsc@max}{%
570 }{%
571   \FB@readaux{\let\FBsuboheight\relax}%
572 }%
573 \@tempcnta=\@ne
574 \if@minipage
575   \@tempcnta=\z@
576 \else\ifdim \lastskip=\z@ \else
577   \@tempcnta=\tw@
578 \fi\fi
579 \ifmaincaptiontop
580   \sf@top=\sf@narskip
581   \sf@bottom=\sf@farskip
582 \else
583   \sf@top=\sf@farskip
584   \sf@bottom=\sf@narskip
585 \fi
586 \leavevmode
587 \setbox\@tempboxa \hbox{%
588   ##4}%
589 \@tempdima=\wd\@tempboxa
590 \ifundefined{FBsc@max}{%
591 }{%
592   \global\advance\Xhsize-\wd\@tempboxa
593   \dimen@=\ht\@tempboxa
594   \advance\dimen@\dp\@tempboxa
595   \ifdim\dimen@>\FBso@max
596     \global\FBso@max\dimen@
597   \fi
598 }%
599 \vtop\bgroup
600   \vbox\bgroup
601     \ifcase\@tempcnta
602       \@minipagefalse
603     \or
604       \vskip\sf@top
605     \or
606       \ifdim \lastskip=\z@ \else
607         \@tempskipb\sf@top\relax\@xaddvskip
608       \fi
609     \fi
610     \sf@ifpositiontop{%
611       \ifx \@empty##3\relax \else
612         \sf@subcaption{##1}{##2}{##3}%
613         \vskip\sf@capskip
614         \vskip\sf@captopadj
615       \fi\egroup
616       \hrule width0pt height0pt depth0pt
617       \box\@tempboxa
618     }{%

```

```

619         \@ifundefined{FBsc@max}{%
620             \box\@tempboxa
621         }{%
622             \ifx\FBsuboheight\relax
623                 \box\@tempboxa
624             \else
625                 \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
626             \fi}%
627         \egroup
628     \ifx \@empty##3\relax \else
629         \vskip\sf@capskip
630         \hrule width0pt height0pt depth0pt
631         \sf@subcaption{##1}{##2}{##3}%
632     \fi
633 }%
634 \vskip\sf@bottom
635 \egroup
636 \@ifundefined{FBsc@max}{%
637 }{%
638     \addtocounter{FRobj}{-1}%
639     \ifnum\c@FRobj=0\else
640         \subfloatrowsep
641     \fi
642 }%
643 \ifmaincaptiontop\else
644     \global\advance\@nameuse{c@\@captype}\m@ne
645 \fi
646 \endgroup\ignorespaces}%
647 %
648 \expandafter\ifx\csname sf@@@subfloat\endcsname\nag@@original@@sf@@@subfloat
649     % yup, that's it.
650     \PackageInfo{nag}{OK, equivalent to subfig 1.3, redefining
651         \@backslashchar sf@@@subfloat\@gobble}%
652     \global\long\def\sf@@@subfloat##1[##2][##3]##4{%
653         \@ifundefined{FBsc@max}{%
654         }{%
655             \FB@readaux{\let\FBsuboheight\relax}%
656         }%
657         \@tempcnta=\@ne
658         \if@minipage
659             \@tempcnta=\z@
660         \else\ifdim \lastskip=\z@ \else
661             \@tempcnta=\tw@
662         \fi\fi
663         \ifmaincaptiontop
664             \sf@top=\sf@nearskip
665             \sf@bottom=\sf@farskip
666         \else
667             \sf@top=\sf@farskip
668             \sf@bottom=\sf@nearskip

```

```

669 \fi
670 \leavevmode
671 \setbox\@tempboxa \hbox{%
672   %% ulmi: new 2007/02/25: #4 may contain label command
673   \csname nag@hascaptiontrue\endcsname
674   %% and that was it.
675   ##4}%
676 \@tempdima=\wd\@tempboxa
677 \@ifundefined{FBsc@max}{%
678 }{%
679   \global\advance\Xhsize-\wd\@tempboxa
680   \dimen@=\ht\@tempboxa
681   \advance\dimen@\dp\@tempboxa
682   \ifdim\dimen@>\FBso@max
683     \global\FBso@max\dimen@
684   \fi
685 }%
686 \vtop\bgroup
687   %% ulmi: new 2007/05/10: #2, #3 may contain label command
688   \csname nag@hascaptiontrue\endcsname
689   %% and that was it.
690   \vbox\bgroup
691     \ifcase\@tempcnta
692       \@minipagefalse
693     \or
694       \vskip\sf@top
695     \or
696       \ifdim \lastskip=\z@ \else
697         \@tempskipb\sf@top\relax\@xaddvskip
698       \fi
699     \fi
700     \sf@ifpositiontop{%
701       \ifx \@empty##3\relax \else
702         \sf@subcaption{##1}{##2}{##3}%
703         \vskip\sf@capskip
704         \vskip\sf@captopadj
705       \fi\egroup
706       \hrule width0pt height0pt depth0pt
707       \box\@tempboxa
708     }{%
709       \@ifundefined{FBsc@max}{%
710         \box\@tempboxa
711       }{%
712         \ifx\FBsuboheight\relax
713           \box\@tempboxa
714         \else
715           \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
716         \fi}%
717     \egroup
718   \ifx \@empty##3\relax \else

```

```

719         \vskip\sf@capskip
720         \hrule width0pt height0pt depth0pt
721         \sf@subcaption{##1}{##2}{##3}%
722     \fi
723 }%
724     \vskip\sf@bottom
725 \egroup
726 \@ifundefined{FBsc@max}{%
727 }{%
728     \addtocounter{FRobj}{-1}%
729     \ifnum\c@FRobj=0\else
730         \subfloatrowsep
731     \fi
732 }%
733 \ifmaincaptiontop\else
734     \global\advance\@nameuse{c@\@captype}\m@ne
735 \fi
736 \endgroup\ignorespaces}%
737 \else
738     \PackageInfo{nag}{Not redefining sf@@@subfloat, it looks odd\@gobble}
739 \fi
740 }

```

### 4.3 The float package

Sorry, there is no way for nag to automatically add new float types to check them for captions. However, since v0.52, there is an user-level command `\NagDeclareFloat` that will do the bookkeeping for you, i.e. after your call to `\newfloat`, you call `\NagDeclareFloat` with the first argument to `\newfloat`.

```

741 \newcommand*\NagDeclareFloat[1]{\g@addto@macro\nag@floats{,#1}}

```

### 4.4 The topcapt package and the subfig package

nagdemonstrates an error when `topcapt` and `subfig` are used together, i.e. `subfig` thinks the caption has not been stepped already. This is not a bug in nag.

### 4.5 The rotating package

`rotating` uses `\centerline` to place rotated floats. As far as I can see, the usage is legitimate there, and using `\centering` instead would change behaviour when the float's dimension are larger than the text body. (Currently, the height of the figure may exceed `\textwidth` without warning.) If this bothers you, go read the warning on p. 3 again.

### 4.6 Version control packages

Common version control systems like `rcs`, `cvs`, `svn` insert their keywords between dollar signs. Packages that parse these keywords define their commands and usually assume

catcode 3, which is not true if either onlyamsmath or nag is loaded. Special handling is introduced for rcs and svninfo. In case of rcsinfo, svn and pgf (yes, it's got internal VC handling that fails when \pgfuselibrary is used outside the preamble – thanks to Ralf Thöle for spotting this one), dollar checking is disabled.

## 5 Loading extensions

Finally, we deal with package options. This is simple: just try to input appropriate nag files.

```

742 \DeclareOption*{%
743   \InputIfFileExists{nag-\CurrentOption.cfg}{%
744     \PackageInfo{nag}{%
745       Loaded nag-\CurrentOption.cfg
746     }
747   }{%
748     \InputIfFileExists{\CurrentOption.nag}{%
749       \PackageWarningNoLine{nag}{%
750         Loaded old-style config file \CurrentOption.nag.\MessageBreak
751         Consider renaming the file to nag-\CurrentOption.cfg
752       }%
753     }{%
754       \PackageWarningNoLine{nag}{Required ruleset
755         \CurrentOption, and it wasn't there}
756     }%
757   }
758 }
759 \ProcessOptions*

```

## Change History

0.1		Handling command vs. environ-
General: First official version. . . . .	1	ment; bugfixes . . . . . 1
0.2		0.5
General: Added abort.nag, suggested		General: Handle the case that some-
by Michael Zedler . . . . .	1	body else relaxes the ver@-
Rephrased umlaut.sty warning,		commands. Stack-based No-
suggested by Patrick Happel. . . . .	1	tASwitch. . . . . 1
0.3		0.51
General: Fixed missing globals . . . . .	14	\nag@prepend: bugfix . . . . . 10
New ifdefined that won't relax the		0.52
commands . . . . .	1	General: Command NagDeclareFloat
0.4		added . . . . . 22
General: bugfix . . . . .	14	made eTeX-ifcstype more robust . 12
config file names changed to free		twiddle subfig's bowels . . . . . 18
extension . . . . .	23	\nag@prepend: info . . . . . 10

0.53	General: bugfix: more Robustness. (Jörg Sommer) . . . . .	16	0.61alpha1	General: roman counter (external file issue) . . . . .	10
0.54	\NotASwitch: bugfix: can't get around the token register. (Jörg Sommer) . . . . .	16	\nag@prepend: Extra indirection of warnings for robustness (upper- casing/LoF issues) . . . . .	10	
0.55	General: Some spaces crept in in 0.5 . . . . .	1	\nag@warn: Made robust. . . . .	11	
0.60	General: @preamblecmds . . . . .	14	0.61alpha2	\nag@prepend: Creep under existing robust cover . . . . .	10
	Captions/Labels now done only once, and not every time we enter a float . . . . .	14	0.61alpha4	General: sin counter should not be saved by include . . . . .	10
0.60alpha2	\nag@prepend: don't even relax un- known commands (J.Sommer) . . . . .	10	0.61alpha6	General: Compatibility w/ VCS pack- ages, pgf . . . . .	4

## Index

Numbers written in *italic* 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>	\@input@ . . . . .	272	491, 494, 503, 516,
\" . . . . .	\@largefloatcheck . . . . .	232	522, 531, 532, 534, 536
\\$ . . . . .	\@latex@info@no@line . . . . .	266, 278	\@xaddvskip . . . . .
\@@input . . . . .	\@latex@warning . . . . .	236, 262, 274	\[ . . . . .
\@addtoreset . . . . .	\@minipagefalse . . . . .	602, 692	\] . . . . .
\@backslashchar . . . . .	\@nameuse . . . . .	644, 734	\_ . . . . .
311, 312, 342, 476,	\@preamblecmds . . . . .	417	<b>A</b>
511, 512, 523, 651	\@secondoftwo . . . . .	20,	\active . . . . .
\@captype . . . . .	\@sptoken . . . . .	49	\advance . . . . .
\@clsextension . . . . .	\@svnBeginRead . . . . .	79	224, 235, 592,
\@currbox . . . . .	\@tempboxa . . . . .	587,	594, 644, 679, 681, 734
224, 230, 233, 235, 237	589, 592–594, 617,		\AtBeginDocument . . . . .
\@currenvir . . . . .	620, 623, 625,		. . . . .
\@empty . . . . .	671, 676, 679–		43, 44,
442, 611, 628, 701, 718	681, 707, 710, 713, 715		67, 339, 349, 416,
\@filef@und . . . . .	\@tfor . . . . .	242, 531	447, 450, 471, 542, 551
\@firstoftwo . . . . .	\@typeset@protect . . . . .	24	\AtEndDocument . . . . .
19, 24, 43,	\@undefined . . . . .	165	265,
297, 372, 516, 522, 534	\@xa . . . . .	216,	277, 381, 391, 451, 530
\@fps . . . . .	242, 265, 266, 277,		<b>B</b>
\@gobble . . . . .	278, 286, 315, 316,		\bfseries . . . . .
\@gobbletwo . . . . .	318, 321–323, 443,		102
\@ifnextchar . . . . .			\bgroup . . . . .
\@ifpackageloaded . . . . .			45, 48, 71, 461,
. . . . .			490, 599, 600, 686, 690
\@input . . . . .			\binom . . . . .
259			120
			\bottomfraction . . . . .
			252
			\box . . . . .
			617, 620, 623,
			625, 707, 710, 713, 715



<b>C</b>		
\c@FRobj .....	639, 729	
\caption .....	409, 434	
\captionprovider	424–426	
\catcode .....	44, 46,	
	72, 73, 79, 84, 88, 92	
\changes ....	162, 171, 172	
<b>D</b>		
\DeclareRobustCommand		
...	206, 210, 318,	
	330, 334, 460, 487, 501	
\dimen@ .	593–596, 680–683	
<b>E</b>		
\edef .....	276	
\egroup .....	58, 60,	
	66, 468, 499, 615,	
	627, 635, 705, 717, 725	
<b>F</b>		
\FB@readaux ....	571, 655	
\FBafil .....	625, 715	
\FBbfil .....	625, 715	
\FBso@max	595, 596, 682, 683	
\FBsuboheight ..	571,	
	622, 625, 655, 712, 715	
\fontcmd	188, 190, 192, 195	
\frac .....	118–120	
<b>G</b>		
\g@addto@macro	79, 161,	
	173, 247, 251, 417, 741	
\gdef .....	47, 518	
\global ..	65, 378, 404,	
	412, 426, 592, 596,	
	644, 652, 679, 683, 734	
<b>H</b>		
\hbox .....	587, 671	
<b>I</b>		
\if@minipage ...	574, 658	
\IfFileExists .....	260	
\ifinner .....	19	
\ifmaincaptiontop ..		
...	579, 643, 663, 733	
\ifmmode .....	51	
\ifnag@dofloatsizecheck		
.....	217, 257	
\ifnag@hascaption ..		
.....	401, 407, 432	
\ifnag@haslabel ....	400	
\ignorespaces ..	646, 736	
\inputlineno .....	495	
\itshape .....	104	
<b>J</b>		
\justenv .....	201, 202	
\justsw .....	197, 199	
<b>L</b>		
\label .....	433	
\labelprovider .	418–420	
\lastskip	576, 606, 660, 696	
\leavevmode ....	586, 670	
\long .....	568, 652	
\looseends .....		
...	531, 532, 534, 536	
<b>M</b>		
\m@ne .....	644, 734	
\mathcal .....	116	
\message .....	28	
<b>N</b>		
\nag@@original@@sf@@@subfloat		
.....	568, 648	
\nag@allfloatsizechecks		
.....	218,	
	241, 247, 251, 257	
\nag@beginenv ..	487, 544	
\nag@captioncheck ..		
.....	421, 431	
\nag@captions	173, 424, 439	
\nag@dofloatsizecheckfalse		
.....	244	
\nag@dofloatsizechecktrue		
.....	240	
\nag@doubledollar ..		
.....	12, 55, 58	
\nag@end@env ....	503, 506	
\nag@endenv .	501, 515, 547	
\nag@envstack ..	485,	
	491, 492, 503, 518, 531	
\nag@expanding@voodoo		
.....	14, 32	
\nag@floats .	437, 441, 741	
\nag@floatsetup	440, 448	
\nag@fltsz@tmp .....		
...	242, 243, 246, 250	
\nag@hackfloat .	402, 443	
\nag@hascaptionfalse	404	
\nag@hascaptiontrue		
.....	401, 412, 426	
\nag@haslabelfalse .	404	
\nag@haslabeltrue ..	421	
\nag@ifcsname	358, 361,	
	362, 378, 383, 393, 552	
\nag@ifCurrentEnvironment		
.....	460, 473	
\nag@ifundefined ...		
.....	295, 300, 306	
\nag@labels .	161, 418, 438	
\nag@maybe@dispmath		
.....	33, 47	
\nag@maybe@dispmath .		
.....	16, 65	
\nag@maybe@hacksubfig		
.....	554, 558	
\nag@maybespace ....		
...	307, 309, 317, 321	
\nag@nofile .....		
...	264, 267, 276, 279	
\nag@onefloatsizecheck		
.....	219, 248, 252	
\nag@pr@p@nd ...	321, 327	
\nag@prepend	164, 295,	
	340, 350, 403, 406,	
	420, 426, 472, 543, 546	
\nag@quark .....	49	
\nag@singledollar ..		
	13, 14, 22, 35, 40, 60	
\nag@warn ....	54, 166,	
	206, 330, 341, 351,	
	408, 433, 474, 511, 521	
\nag@warnNoLine	210,	
	334, 384, 394, 533	
\NagDeclareFloat	160, 741	
\NeedsTeXFormat ....	284	
\newcommand ....	218,	
	219, 295, 299, 327,	
	338, 348, 358, 380,	
	390, 402, 431, 440,	
	470, 485, 541, 741	
\newcounter ....	288, 293	
\newif .....	217, 400, 401	
\noexpand .....	287	

<code>\NotAnEnvironment</code> .. ..... 176, 179, 190, 195, 199, <u>470</u>	<b>R</b> <code>\RCS</code> ..... 71 <code>\renewcommand</code> ..... .... 71, 232, 289, 362 <code>\rmfamily</code> ..... 106 <code>\roman</code> ..... 289	<code>\sf@subcaption</code> ..... ... 612, 631, 702, 721 <code>\sf@top</code> . 580, 583, 604, 607, 664, 667, 694, 697 <code>\sffamily</code> ..... 110 <code>\sloppy</code> ..... 98 <code>\slshape</code> ..... 112 <code>\string</code> .. 56, 522, 534, 536 <code>\subfloatrowsep</code> 640, 730
<b>O</b> <code>\ObsoleteClass</code> . 124, <u>390</u> <code>\ObsoleteCS</code> ..... ... 101, 103, 105, 107, 109, 111, 113, 115, 119, 120, 122, <u>338</u> <code>\ObsoleteEnv</code> ... 184, <u>348</u> <code>\ObsoletePackage</code> ... ..... 8, 9, 126– 128, 130, 132, 133, 135, 136, 138, 140, 143, 145, 147, 149, 152, 154, 157, 158, <u>380</u>	<b>S</b> <code>\S</code> . 7, 10, 11, 97, 98, 100, 117, 121, 123, 125, 129, 131, 134, 139, 141, 144, 151, 155, 156, 159, 175, 181, 185 <code>\scshape</code> ..... 108 <code>\sectioning</code> .... 178, 179 <code>\setbox</code> ..... 587, 671 <code>\setcounter</code> ..... 290 <code>\sf@@@subfloat</code> ..... 652 <code>\sf@bottom</code> ..... 581, 584, 634, 665, 668, 724 <code>\sf@capskip</code> ..... ... 613, 629, 703, 719 <code>\sf@captopadj</code> .. 614, 704 <code>\sf@farskip</code> ..... ... 581, 583, 665, 667 <code>\sf@ifpositiontop</code> .. ..... 610, 700 <code>\sf@nearskip</code> ..... ... 580, 584, 664, 668	<b>T</b> <code>\texorpdfstring</code> .. 17, 43 <code>\textbf</code> ..... 102 <code>\textheight</code> .... 222, 223, 233, 234, 237 <code>\textit</code> ..... 104 <code>\textrm</code> ..... 106 <code>\textsc</code> ..... 108 <code>\textsf</code> ..... 110 <code>\textsl</code> ..... 112 <code>\texttt</code> ..... 114 <code>\thenag@c</code> ..... 289, 316, 318, 322, 323 <code>\toks@</code> ..... 491, 497 <code>\topfraction</code> ..... 248 <code>\ttfamily</code> ..... 114
<b>P</b> <code>\PackageError</code> .. 208, 212 <code>\PackageInfo</code> 76, 80, 83, 87, 91, 225, 302, 310, 553, 650, 738, 744 <code>\protected@edef</code> .... 264 <code>\providecommand</code> ..... 43 <code>\ProvidesFile</code> ..... .... 1, 186, 204, 214 <code>\ProvidesPackage</code> ... 285	<b>X</b> <code>\xdef</code> ..... 492 <code>\Xhsize</code> ..... 592, 679	