

The nag package*

Ulrich Michael Schwarz[†]

July 30, 2009

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 All the same, there are commands, classes and packages which are outdated and superseded.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

| | | |
|----------|---|-----------|
| 1 | User-side considerations. | 2 |
| 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 |
| 2 | Author-side considerations and implementation. | 10 |
| 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 |
| 3 | Switch vs. Environment | 16 |
| 4 | Compatibility issues | 18 |
| 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 | 23 |
| 4.6 | Version control packages | 23 |

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

[†]ulmi@absatzen.de

1 User-side considerations.

1.1 Installation.

Process `nag.ins` with \LaTeX to obtain some files: `nag.sty` and `nag-l2tabu.cfg` et al. must go to a place where \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 \LaTeX file you want to use them on may work under many circumstances.) You can, as usual, run \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`, `mathppl`, `euler` and `utopia`, and of the outdated class `sclrttr`.
- Figures and tables without caption (this is not technically in `l2tabu`, but the people who have floats without captions tend to ask “Why is \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@doubleddollar{$$}%$$
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@maybeispmath{%
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@maybeispmath}%
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{scrletter}{the scrletter2 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,caption@xlabel}%
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 [2009/07/04 v0.62alpha2 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 \newif\ifnag@allfloatpositionsfailed
219 \newcommand\nag@allfloatsizechecks{}%
220 \newcommand\nag@onefloatsizecheck[2]{%
221 % #1 is size fraction of textheight,
222 % #2 is position to say in warning.
223 \ifdim \ht\@currbox>#1\textheight
224 \@tempdima -#1\textheight
225 \advance \@tempdima \ht\@currbox
226 \PackageInfo{nag}{Float too large for #2 by \the\@tempdima}%
227 % note we do not truncate.
228 % also, it's too late to add "p" now.
229 \else
230 \nag@allfloatpositionsfailedfalse
231 \fi
232 }
233 % \@currbox is current float box,
234 % \@fps is the current list of float specifiers.
235 \renewcommand\@largefloatcheck{%
236 \ifdim \ht\@currbox>\textheight
237 \@tempdima -\textheight
238 \advance \@tempdima \ht\@currbox
239 \latex@warning {Float too large for page by \the\@tempdima}%
240 \ht\@currbox \textheight
241 \fi
242 %% the preceding is the original check.
243 \nag@dofloatsizechecktrue
244 \nag@allfloatpositionsfailedtrue
245 \def\nag@allfloatsizechecks{}%
246 \@xa\@xa\@xa\@tfor\@xa\@xa\@xa\nag@fltsz@tmp\@xa\@xa\@xa:\@xa\@xa\@xa=\csname @fps\endcsname
247 \ifx\nag@fltsz@tmp\relax
248 \nag@dofloatsizecheckfalse
249 \fi

```



```

250 \if\nag@fltsz@tmp !
251 \nag@dofloatsizecheckfalse
252 \else
253 \if\nag@fltsz@tmp t
254 \g@addto@macro\nag@allfloatsizechecks
255 {\nag@onefloatsizecheck{\topfraction}{top of page}}%
256 \else
257 \if\nag@fltsz@tmp b
258 \g@addto@macro\nag@allfloatsizechecks
259 {\nag@onefloatsizecheck{\bottomfraction}{bottom of page}}%
260 \else
261 \if\nag@fltsz@tmp p
262 \nag@allfloatpositionsfailedfalse
263 \fi
264 \fi
265 \fi
266 \fi
267 }%
268 \ifnag@dofloatsizecheck
269 \nag@allfloatsizechecks
270 \ifnag@allfloatpositionsfailed
271 \nag@warn{All float specifiers '\@fps' won't work}%
272 \fi
273 \fi
274 }%

```

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.

```

275 \def\@input#1{%
276 \IfFileExists{#1}{\@input\@filef@und}{%
277 \typeout{No file #1.}
278 \@latex@warning{File '#1' not found}
279 }{The file '#1' was requested but not found }
280 \protected@edef\nag@nofile{File '#1' requested, but not found}%
281 \@xa\AtEndDocument\@xa{%
282 \@xa\@latex@info@no@line\@xa{%
283 \nag@nofile
284 }%
285 }%
286 }}%
287 %
288 \def\@input@#1{\InputIfFileExists{#1}}{%
289 \typeout{No file #1.}
290 \@latex@warning{File '#1' not found}
291 {The file '#1' was requested but not found }
292 \edef\nag@nofile{File '#1' requested, but not found}%
293 \@xa\AtEndDocument\@xa{%
294 \@xa\@latex@info@no@line\@xa{%
295 \nag@nofile

```

```

296     }%
297   }%
298 }}%
299 %

```

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.

```

300 \NeedsTeXFormat{LaTeX2e}
301 \ProvidesPackage{nag}[2009/07/30 0.62 warning about old commands (ulmi)]
302 \let\@xa\expandafter
303 \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.

```

304 \newcounter{nag@c}
305 \renewcommand\thenag@c{\roman{nag@c}}%
306 \setcounter{nag@c}{1}%
307 \begingroup
308   \let\@addtoreset\@gobbletwo
309   \newcounter{nag@sins}%
310 \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 α_2 on, nag is more robust about not defining commands that are not there. Now, they're not even relaxed.

```

311 \newcommand\nag@ifundefined[1]{%
312   \begingroup
313   \@ifundefined{#1}{\endgroup\@firstoftwo}{\endgroup\@secondoftwo}%
314 }

```

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.

```

315 \newcommand\nag@prepend[2]{%

```

```

316 \nag@ifundefined{#1}{%
317 % if it doesn't exist, don't do anything.
318 \PackageInfo{nag}{%
319   Command \@backslashchar#1\space not defined, skipping amendment%
320 }%
321 }{%
322 \nag@ifundefined{#1 }{%
323 \let\nag@maybespace\@empty
324 }{%
325 \let\nag@maybespace\space
326 %\PackageInfo{nag}{%
327 % Command \@backslashchar#1\space appears robust\MessageBreak
328 % Modifying '@backslashchar#1\space' instead.
329 }%
330 }%
331 \@xa\let
332 \csname nag@@#1@\thenag@c\@xa\endcsname
333 \csname #1\nag@maybespace\endcsname
334 \@xa\DeclareRobustCommand\csname nag@@warning@\thenag@c\@xa\endcsname{%
335 #2%
336 }%
337 \@xa\nag@pr@p@end\csname #1\nag@maybespace\@xa\endcsname
338 \csname nag@@#1@\thenag@c\@xa\endcsname
339 \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?

```

340 \stepcounter{nag@c}%
341 }%
342 }
343 \newcommand\nag@pr@p@end[3]{%
344 \def#1{#3#2}%
345 }

```

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

```

346 \DeclareRobustCommand\nag@warn{%
347 \addtocounter{nag@sins}{1}%
348 \PackageWarning{nag}%
349 }
350 \DeclareRobustCommand\nag@warnNoLine{%
351 \addtocounter{nag@sins}{1}%
352 \PackageWarningNoLine{nag}%
353 }

```

2.2 Obsoletifying 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.

```

354 \newcommand\ObsoleteCS[3][obsolete]{%
355   \AtBeginDocument{%
356     \nag@prepend{#2}{%
357       \nag@warn{%
358 Command \@backslashchar#2 is #1.
359 \MessageBreak
360 Use #3 instead}%
361   }%
362 }%
363 }

```

`\ObsoleteEnv`

```

364 \newcommand\ObsoleteEnv[3][obsolete]{%
365   \AtBeginDocument{%
366     \nag@prepend{#2}{%
367       \nag@warn{%
368 Environment #2 is #1.
369 \MessageBreak
370 Use #3 instead}%
371   }%
372 }%
373 }

```

2.3 Obsolete 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`).

```

374 \newcommand\nag@ifcsname[3]{%
375   \begingroup\@ifundefined{#1}{#3}{#2}\endgroup
376 }

```

Just because we can, use ϵ TeX' `\ifcsname` if we can. This bootstrapping gives me a big grin... Note we add an extra group for compatibility with the non- ϵ case.

```

377 \nag@ifcsname{ifcsname}{%
378   \renewcommand*\nag@ifcsname[3]{%
379     \begingroup
380     % assume it won't be there.
381     \let\tmp@a@\secondoftwo
382     \ifcsname #1\endcsname
383     % It still might be relax from some other test. Thanks to J\"org

```

```

384 % Sommer for finding this bug.
385 \expandafter\ifx\csname #1\endcsname\relax
386 \else
387 % it's there after all
388 \let\tmp@a\@firstoftwo
389 \fi
390 \fi
391 \tmp@a{#2}{#3}%
392 \endgroup
393 }%

```

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

```

394 \global\let\nag@ifcsname\nag@ifcsname
395 }{}

```

`\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.

```

396 \newcommand\ObsoletePackage[3][obsolete]{%
397 \AtEndDocument{%
398 % |\@clsextension| is onlypreamble, for some reason.
399 \nag@ifcsname{ver@#2.sty}{%
400 \nag@warnNoLine{%
401 Package #2 is #1.\MessageBreak
402 Use #3 instead}%
403 }{}%
404 }%
405 }

```

`\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.

```

406 \newcommand\ObsoleteClass[3][obsolete]{%
407 \AtEndDocument{%
408 % |\@clsextension| is onlypreamble, for some reason.
409 \nag@ifcsname{ver@#2.cls}{%
410 \nag@warnNoLine{%
411 Class #2 is #1.\MessageBreak
412 Use #3 instead}%
413 }{}%
414 }%
415 }

```

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.

```
416 \newif\ifnag@haslabel
417 \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.

```
418 \newcommand\nag@hackfloat[1]{%
419   \nag@prepend{#1}{%
420     \global\nag@haslabelfalse\global\nag@hascaptionfalse
421   }%
422   \nag@prepend{end#1}{%
423     \ifnag@hascaption\relax\else
424       \nag@warn%
425     {#1 with no \protect\caption}%
426     \fi
427     % labels outside floats shouldn't complain:
428     \global\nag@hascaptiontrue
429     % (we do this always because it needs to be global)
430   }%
431 }
```

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.

```
432 \AtBeginDocument{%
433 \g@addto@macro{\@preamblecmds}{%
```

```

434 \@for\labelprovider:=\nag@labels\do{%
435   \ifx\labelprovider\@empty\else
436     \nag@prepend{\labelprovider}%
437     {\nag@captioncheck\nag@haslabeltrue}%
438   \fi
439 }%
440 \@for\captionprovider:=\nag@captions\do{%
441   \ifx\captionprovider\@empty\else
442     \nag@prepend{\captionprovider}{\global\nag@hascaptiontrue}%
443   \fi
444 }%
445 }%
446 }
447 \newcommand\nag@captioncheck{%
448   \ifnag@hascaption\else
449     \nag@warn{\protect\label\space in float, but not after
450       \protect\caption}%
451   \fi
452 }

```

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.

```

453 \def\nag@floats{}
454 \def\nag@labels{}
455 \def\nag@captions{}

```

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

```

456 \newcommand\nag@floatsetup{%
457   \@for\flo:=\nag@floats\do{%
458     \ifx\flo\@empty\else
459       \@xa\nag@hackfloat\@xa{\flo}%
460     \fi
461   }%
462 }

```

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

```

463 \AtBeginDocument{%
464   \nag@floatsetup
465 }

```

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

```

466 \AtBeginDocument{%
467   \AtEndDocument{%
468     \ifnum\value{nag@sins}>0%
469       \PackageWarningNoLine{nag}{\arabic{nag@sins} complaints
470         in total}%

```

```

471 \else
472 \typeout{No complaints by nag.}%
473 \fi
474 }%
475 }

```

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:

```

476 \DeclareRobustCommand\nag@ifCurrentEnvironment[3]{%
477 \bgroup
478 \def\tmp@a{#1}%
479 \ifx\@currenenvir\tmp@a
480 #2%
481 \else
482 #3%
483 \fi
484 \egroup
485 }

```

And now, the two variations there are:

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

```

486 \newcommand\NotAnEnvironment[1]{%
487 \AtBeginDocument{%
488 \nag@prepend{#1}{%
489 \nag@ifCurrentEnvironment{#1}{%
490 \nag@warn{%
491 There is no environment ‘‘#1’’. \MessageBreak
492 Maybe you want a grouped \@backslashchar#1
493 }%
494 }{% OK case.
495 }%
496 }%
497 }%
498 }

```

`\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.

499% we need to maintain a stack of environments that are used in the
500% \foo...\endfoo way.
501 \newcommand\nag@envstack{\relax}
502
503 \DeclareRobustCommand\nag@beginenv[1]{%
504 % push a begin-entry onto the stack. Form is
505 % |{\foo{lineno}}| for environment foo.
506 \bgroup
507 \@xa\toks@\@xa{\nag@envstack}%
508 \xdef\nag@envstack{%
509 \@nx{%
510 \@xa\@nx\csname #1\endcsname
511 \@nx{\the\inputlineno\@nx}%
512 \@nx}%
513 \the\toks@
514 }%
515 \egroup
516 }
517 \DeclareRobustCommand\nag@endenv[1]{%
518 % extract the first entry.
519 \@xa\nag@endenv\nag@envstack\@nil #1\@nil
520 }
521
522 \def\nag@endenv#1#2\@nil #3\@nil{%
523 \def\tmp@a{#1}%
524 \def\tmp@b{\relax}%
525 \ifx\tmp@a\tmp@b
526 % This was the end-of-stack flag.
527 \nag@warn{'\@backslashchar end#3'} without matching
528 '\@backslashchar #3'}
529 \else
530 % We may assume this is a proper entry. See if the begin-token on
531 % the stack matches what |\nag@endenv| was passed.
532 \@xa\ifx\csname #3\@xa\endcsname\@firstoftwo #1%
533 %OK case, just pop the entry.
534 \gdef\nag@envstack{#2}%
535 \else
536 % error case
537 \nag@warn{%
538 You cannot close '\@xa\string\@firstoftwo #1' on line
539 \@secondoftwo #1 with '\@backslashchar end#3'%
540 }%
541 % leave it on the stack. Some case of misnesting will always cause
542 % horrible amounts of follow-up errors. Also, scare them!
543 \fi
544 \fi
545 }

```

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

```
546 \AtEndDocument{%
547   \@xa\@tfor\@xa\looseends\@xa:\@xa=\nag@envstack\do{%
548     \@xa\ifx\looseends\relax\else
549       \nag@warnNoLine{Unmatched
550         ‘\@xa\@xa\@xa\string\@xa\@firstoftwo\looseends’
551         command on line
552         \@xa\@xa\@xa\string\@xa\@secondoftwo\looseends%
553       }%
554     \fi
555   }%
556 }
```

Now, the user-side command is easy.

```
557 \newcommand\NotASwitch[1]{%
558   \AtBeginDocument{%
559     \nag@prepend{#1}{%
560       \nag@beginenv{#1}%
561     }%
562     \nag@prepend{end#1}{%
563       \nag@endenv{#1}%
564     }%
565   }%
566 }
```

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.

```
567 \AtBeginDocument{%
568   \nag@ifcsname{ver@subfig.sty}{%
569     \PackageInfo{nag}{Attempting subfig hack\@gobble}%
570     \nag@maybehacksubfig
```

```

571 }{%
572 }%
573 }
574 \def\nag@maybehacksubfig{%
575 %
576 % of course, i need to touch the single longest definition in
577 % subfig.sty, to amend one single command...
578 %
579 % The definition is taken from subfig.sty 1.3 dated 2005/07/05 by
580 % S.D. Chochran, where it is called sf@@@subfloat, and appears here
581 % under the conditions of section 6 of the LPPL 1.3. The subfig
582 % package is available on a CTAN mirror near you.
583 %
584 \long\def\nag@@original@sfsf@@@subfloat##1[##2][##3]##4{%
585 \ifundefined{FBsc@max}{%
586 }{%
587 \FB@readaux{\let\FBsuboheight\relax}%
588 }%
589 \@tempcnta=\@ne
590 \if@minipage
591 \@tempcnta=\z@
592 \else\ifdim \lastskip=\z@ \else
593 \@tempcnta=\tw@
594 \fi\fi
595 \ifmaincaptiontop
596 \sf@top=\sf@nearskip
597 \sf@bottom=\sf@farskip
598 \else
599 \sf@top=\sf@farskip
600 \sf@bottom=\sf@nearskip
601 \fi
602 \leavevmode
603 \setbox\@tempboxa \hbox{%
604 ##4}%
605 \@tempdima=\wd\@tempboxa
606 \ifundefined{FBsc@max}{%
607 }{%
608 \global\advance\Xhsize-\wd\@tempboxa
609 \dimen@=\ht\@tempboxa
610 \advance\dimen@\dp\@tempboxa
611 \ifdim\dimen@>\FBso@max
612 \global\FBso@max\dimen@
613 \fi
614 }%
615 \vtop\bgroup
616 \vbox\bgroup
617 \ifcase\@tempcnta
618 \@minipagefalse
619 \or
620 \vskip\sf@top

```

```

621     \or
622     \ifdim \lastskip=\z@ \else
623     \@tempskipb\sf@top\relax\@xaddvskip
624     \fi
625     \fi
626     \sf@ifpositiontop{%
627     \ifx \@empty##3\relax \else
628     \sf@subcaption{##1}{##2}{##3}%
629     \vskip\sf@capskip
630     \vskip\sf@captopadj
631     \fi\egroup
632     \hrule width0pt height0pt depth0pt
633     \box\@tempboxa
634     }{%
635     \@ifundefined{FBsc@max}{%
636     \box\@tempboxa
637     }{%
638     \ifx\FBsuboheight\relax
639     \box\@tempboxa
640     \else
641     \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
642     \fi}%
643     \egroup
644     \ifx \@empty##3\relax \else
645     \vskip\sf@capskip
646     \hrule width0pt height0pt depth0pt
647     \sf@subcaption{##1}{##2}{##3}%
648     \fi
649     }%
650     \vskip\sf@bottom
651     \egroup
652     \@ifundefined{FBsc@max}{%
653     }{%
654     \addtocounter{FRobj}{-1}%
655     \ifnum\c@FRobj=0\else
656     \subfloatrowsep
657     \fi
658     }%
659     \ifmaincaptiontop\else
660     \global\advance\@nameuse{c@\@captype}\m@ne
661     \fi
662     \endgroup\ignorespaces}%
663 %
664 \expandafter\ifx\c@name sf@@@subfloat\endcsname\nag@@original@@sf@@@subfloat
665 % yup, that's it.
666 \PackageInfo{nag}{OK, equivalent to subfig 1.3, redefining
667 \backslashchar sf@@@subfloat@gobble}%
668 \global\long\def\sf@@@subfloat##1[##2][##3]##4{%
669 \@ifundefined{FBsc@max}{%
670 }{%

```

```

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

```

```

721     \fi\egroup
722     \hrule width0pt height0pt depth0pt
723     \box\@tempboxa
724   }{%
725     \@ifundefined{FBsc@max}{%
726       \box\@tempboxa
727     }{%
728       \ifx\FBsuboheight\relax
729         \box\@tempboxa
730       \else
731         \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
732       \fi}%
733   \egroup
734   \ifx \@empty##3\relax \else
735     \vskip\sf@capskip
736     \hrule width0pt height0pt depth0pt
737     \sf@subcaption{##1}{##2}{##3}%
738   \fi
739   }%
740   \vskip\sf@bottom
741 \egroup
742 \@ifundefined{FBsc@max}{%
743 }{%
744   \addtocounter{FRObj}{-1}%
745   \ifnum\c@FRObj=0\else
746     \subfloatrowsep
747   \fi
748 }%
749 \ifmaincaptiontop\else
750   \global\advance\@nameuse{c@\@captype}\m@ne
751 \fi
752 \endgroup\ignorespaces}%
753 \else
754   \PackageInfo{nag}{Not redefining sf@@@subfloat, it looks odd@gobble}
755 \fi
756 }

```

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`.

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

4.4 The topcapt package and the subfig package

nagdemo exhibits 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.

```
758 \DeclareOption*{%
759   \InputIfFileExists{nag-\CurrentOption.cfg}{%
760     \PackageInfo{nag}{%
761       Loaded nag-\CurrentOption.cfg
762     }
763   }{%
764     \InputIfFileExists{\CurrentOption.nag}{%
765       \PackageWarningNoLine{nag}{%
766         Loaded old-style config file \CurrentOption.nag.\MessageBreak
767         Consider renaming the file to nag-\CurrentOption.cfg
768       }%
769     }{%
770       \PackageWarningNoLine{nag}{Required ruleset
771         \CurrentOption, and it wasn't there}
772     }%
773   }
774 }
775 \ProcessOptions*
```

Change History

0.1

0.2

General: First official version. 1
General: Added abort.nag, suggested
by Michael Zedler 1

| | | | | | |
|--|--|--|--|--|--|
| <code>\@latex@info@no@line</code> | 282, 294 | | | | |
| <code>\@latex@warning</code> | 239, 278, 290 | | | | |
| <code>\@minipagefalse</code> | 618, 708 | | | | |
| <code>\@nameuse</code> | 660, 750 | | | | |
| <code>\@preamblecmds</code> | 433 | | | | |
| <code>\@secondoftwo</code> | ... 20, 25, 313, 381, 539, 552 | | | | |
| <code>\@sptoken</code> | 49 | | | | |
| <code>\@svnBeginRead</code> | 79 | | | | |
| <code>\@tempboxa</code> | 603, 605, 608–610, 633, 636, 639, 641, 687, 692, 695– 697, 723, 726, 729, 731 | | | | |
| <code>\@tfor</code> | 246, 547 | | | | |
| <code>\@typeset@protect</code> | ... 24 | | | | |
| <code>\@undefined</code> | 165 | | | | |
| <code>\@xa</code> | 216, 246, 281, 282, 293, 294, 302, 331, 332, 334, 337–339, 459, 507, 510, 519, 532, 538, 547, 548, 550, 552 | | | | |
| <code>\@xaddvskip</code> | 623, 713 | | | | |
| <code>\[</code> | 56 | | | | |
| <code>\]</code> | 56 | | | | |
| <code>_</code> | 72 | | | | |
| A | | | | | |
| <code>\active</code> | 44, 46, 72 | | | | |
| <code>\advance</code> | 225, 238, 608, 610, 660, 695, 697, 750 | | | | |
| <code>\AtBeginDocument</code> | 43, 44, 67, 355, 365, 432, 463, 466, 487, 558, 567 | | | | |
| <code>\AtEndDocument</code> | . 281, 293, 397, 407, 467, 546 | | | | |
| B | | | | | |
| <code>\bfseries</code> | 102 | | | | |
| <code>\bgroup</code> | 45, 48, 71, 477, 506, 615, 616, 702, 706 | | | | |
| <code>\binom</code> | 120 | | | | |
| <code>\bottomfraction</code> | 259 | | | | |
| <code>\box</code> | 633, 636, 639, 641, 723, 726, 729, 731 | | | | |
| C | | | | | |
| <code>\c@FRobj</code> | 655, 745 | | | | |
| <code>\caption</code> | 425, 450 | | | | |
| <code>\captionprovider</code> | 440–442 | | | | |
| <code>\catcode</code> | 44, 46, 72, 73, 79, 84, 88, 92 | | | | |
| <code>\changes</code> | 162, 171, 172 | | | | |
| D | | | | | |
| <code>\DeclareRobustCommand</code> | ... 206, 210, 334, 346, 350, 476, 503, 517 | | | | |
| <code>\dimen@</code> | . 609–612, 696–699 | | | | |
| E | | | | | |
| <code>\edef</code> | 292 | | | | |
| <code>\egroup</code> | 58, 60, 66, 484, 515, 631, 643, 651, 721, 733, 741 | | | | |
| F | | | | | |
| <code>\FB@readaux</code> | 587, 671 | | | | |
| <code>\FBafil</code> | 641, 731 | | | | |
| <code>\FBbfil</code> | 641, 731 | | | | |
| <code>\FBso@max</code> | 611, 612, 698, 699 | | | | |
| <code>\FBsuboheight</code> | . 587, 638, 641, 671, 728, 731 | | | | |
| <code>\fontcmd</code> | 188, 190, 192, 195 | | | | |
| <code>\frac</code> | 118–120 | | | | |
| G | | | | | |
| <code>\g@addto@macro</code> | 79, 161, 173, 254, 258, 433, 757 | | | | |
| <code>\gdef</code> | 47, 534 | | | | |
| <code>\global</code> | . 65, 394, 420, 428, 442, 608, 612, 660, 668, 695, 699, 750 | | | | |
| H | | | | | |
| <code>\hbox</code> | 603, 687 | | | | |
| I | | | | | |
| <code>\if@minipage</code> | ... 590, 674 | | | | |
| <code>\IfFileExists</code> | 276 | | | | |
| <code>\ifinner</code> | 19 | | | | |
| <code>\ifmaincaptiontop</code> | 595, 659, 679, 749 | | | | |
| <code>\ifmmode</code> | 51 | | | | |
| <code>\ifnag@allfloatpositionsfailed</code> | 218, 270 | | | | |
| <code>\ifnag@dofloatsizecheck</code> | 217, 268 | | | | |
| <code>\ifnag@hascaption</code> | 417, 423, 448 | | | | |
| <code>\ifnag@haslabel</code> | 416 | | | | |
| <code>\ignorespaces</code> | .. 662, 752 | | | | |
| <code>\inputlineno</code> | 511 | | | | |
| <code>\itshape</code> | 104 | | | | |
| J | | | | | |
| <code>\justenv</code> | 201, 202 | | | | |
| <code>\justsw</code> | 197, 199 | | | | |
| L | | | | | |
| <code>\label</code> | 449 | | | | |
| <code>\labelprovider</code> | . 434–436 | | | | |
| <code>\lastskip</code> | 592, 622, 676, 712 | | | | |
| <code>\leavevmode</code> | 602, 686 | | | | |
| <code>\long</code> | 584, 668 | | | | |
| <code>\looseends</code> | 547, 548, 550, 552 | | | | |
| M | | | | | |
| <code>\m@ne</code> | 660, 750 | | | | |
| <code>\mathcal</code> | 116 | | | | |
| <code>\message</code> | 28 | | | | |
| N | | | | | |
| <code>\nag@@original@@sf@@@subfloat</code> | 584, 664 | | | | |
| <code>\nag@allfloatpositionsfailedfalse</code> | 230, 262 | | | | |
| <code>\nag@allfloatpositionsfailedtrue</code> | 244 | | | | |
| <code>\nag@allfloatsizechecks</code> | 219, 245, 254, 258, 269 | | | | |
| <code>\nag@beginenv</code> | .. 503, 560 | | | | |
| <code>\nag@captioncheck</code> | 437, 447 | | | | |
| <code>\nag@captions</code> | 173, 440, 455 | | | | |
| <code>\nag@dofloatsizecheckfalse</code> | 248, 251 | | | | |
| <code>\nag@dofloatsizechecktrue</code> | 243 | | | | |
| <code>\nag@doubledollar</code> | 12, 55, 58 | | | | |
| <code>\nag@end@nv</code> | 519, 522 | | | | |
| <code>\nag@endenv</code> | . 517, 531, 563 | | | | |

| | | |
|--|---|--|
| <code>\nag@envstack</code> .. 501, 507, 508, 519, 534, 547 | <code>\newcommand</code> 219, 220, 311, 315, 343, 354, 364, 374, 396, 406, 418, 447, 456, 486, 501, 557, 757 | 141, 144, 151, 155, 156, 159, 175, 181, 185 |
| <code>\nag@expanding@voodoo</code> 14, 32 | <code>\newcounter</code> 304, 309 | <code>\scshape</code> 108 |
| <code>\nag@floats</code> . 453, 457, 757 | <code>\newif</code> .. 217, 218, 416, 417 | <code>\sectioning</code> 178, 179 |
| <code>\nag@floatsetup</code> 456, 464 | <code>\noexpand</code> 303 | <code>\setbox</code> 603, 687 |
| <code>\nag@fltsz@tmp</code> . 246, 247, 250, 253, 257, 261 | <code>\NotAnEnvironment</code> 176, 179, 190, 195, 199, 486 | <code>\setcounter</code> 306 |
| <code>\nag@hackfloat</code> . 418, 459 | <code>\NotASwitch</code> . 202, 499, 557 | <code>\sf@@@subfloat</code> 668 |
| <code>\nag@hascaptionfalse</code> 420 | O | <code>\sf@bottom</code> 597, 600, 650, 681, 684, 740 |
| <code>\nag@hascaptiontrue</code> 417, 428, 442 | <code>\ObsoleteClass</code> . 124, 406 | <code>\sf@capskip</code> 629, 645, 719, 735 |
| <code>\nag@haslabelfalse</code> . 420 | <code>\ObsoleteCS</code> 101, 103, 105, 107, 109, 111, 113, 115, 119, 120, 122, 354 | <code>\sf@captopadj</code> .. 630, 720 |
| <code>\nag@haslabeltrue</code> .. 437 | <code>\ObsoleteEnv</code> ... 184, 364 | <code>\sf@farskip</code> 597, 599, 681, 683 |
| <code>\nag@ifcscname</code> 374, 377, 378, 394, 399, 409, 568 | <code>\ObsoletePackage</code> 8, 9, 126– 128, 130, 132, 133, 135, 136, 138, 140, 143, 145, 147, 149, 152, 154, 157, 158, 396 | <code>\sf@ifpositiontop</code> 626, 716 |
| <code>\nag@ifCurrentEnvironment</code> 476, 489 | P | <code>\sf@nearskip</code> 596, 600, 680, 684 |
| <code>\nag@ifundefined</code> 311, 316, 322 | <code>\PackageError</code> .. 208, 212 | <code>\sf@subcaption</code> 628, 647, 718, 737 |
| <code>\nag@labels</code> . 161, 434, 454 | <code>\PackageInfo</code> 76, 80, 83, 87, 91, 226, 318, 326, 569, 666, 754, 760 | <code>\sf@top</code> . 596, 599, 620, 623, 680, 683, 710, 713 |
| <code>\nag@maybe@dispmath</code> 33, 47 | <code>\protected@edef</code> 280 | <code>\sffamily</code> 110 |
| <code>\nag@maybedispmath</code> 16, 65 | <code>\providecommand</code> 43 | <code>\sloppy</code> 98 |
| <code>\nag@maybehacksubfig</code> 570, 574 | <code>\ProvidesFile</code> 1, 186, 204, 214 | <code>\slshape</code> 112 |
| <code>\nag@maybespace</code> 323, 325, 333, 337 | <code>\ProvidesPackage</code> ... 301 | <code>\string</code> .. 56, 538, 550, 552 |
| <code>\nag@nofile</code> 280, 283, 292, 295 | R | <code>\subfloatrowsep</code> 656, 746 |
| <code>\nag@onefloatsizecheck</code> 220, 255, 259 | <code>\RCS</code> 71 | T |
| <code>\nag@pr@p@nd</code> ... 337, 343 | <code>\renewcommand</code> 71, 235, 305, 378 | <code>\texorpdfstring</code> .. 17, 43 |
| <code>\nag@prepend</code> 164, 311, 356, 366, 419, 422, 436, 442, 488, 559, 562 | <code>\rmfamily</code> 106 | <code>\textbf</code> 102 |
| <code>\nag@quark</code> 49 | <code>\roman</code> 305 | <code>\textheight</code> 223, 224, 236, 237, 240 |
| <code>\nag@singledollar</code> .. . 13, 14, 22, 35, 40, 60 | S | <code>\textit</code> 104 |
| <code>\nag@warn</code> 54, 166, 206, 271, 346, 357, 367, 424, 449, 490, 527, 537 | <code>\S</code> . 7, 10, 11, 97, 98, 100, 117, 121, 123, 125, 129, 131, 134, 139, | <code>\textrm</code> 106 |
| <code>\nag@warnNoLine</code> 210, 350, 400, 410, 549 | | <code>\textsc</code> 108 |
| <code>\NagDeclareFloat</code> 160, 757 | | <code>\textsl</code> 112 |
| <code>\NeedsTeXFormat</code> 300 | | <code>\texttt</code> 114 |
| | | <code>\thenag@c</code> 305, 332, 334, 338, 339 |
| | | <code>\toks@</code> 507, 513 |
| | | <code>\topfraction</code> 255 |
| | | <code>\ttfamily</code> 114 |
| | | X |
| | | <code>\xdef</code> 508 |
| | | <code>\Xhsize</code> 608, 695 |