

The `nag` package*

Ulrich Michael Schwarz†

April 5, 2010

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 All the same, there are commands, classes and packages which are outdated and superseded.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.	12
2.3	Obsoletifying packages and classes.	12
2.4	Common float errors and no-nos.	14
3	Switch vs. Environment	16
4	Compatibility issues	18
4.1	The <code>caption</code> package	18
4.2	The <code>subfig</code> package	19
4.3	The <code>float</code> package	23
4.4	The <code>topcapt</code> package and the <code>subfig</code> package	23
4.5	The <code>rotating</code> package	23
4.6	Version control packages	23

*This document corresponds to `nag` 0.621, dated 2010/04/05. 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`, `tlenc`, `caption2`, `psfonts`, `mathptm`, `times`, `palatino`, `mathppl`, `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 \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           [2010/04/05 v2.1 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 \LaTeX 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 \LaTeX , 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 \def\nag@quark{\nag@quark}
46 \bgroup
47 \catcode'\active%$
48 \gdef\nag@maybe@dispmath{%
49     \bgroup
50     \let\@sptoken\nag@quark% prevent skipping of spaces
51     \ifnextchar${%$%
52         \ifmmode
53             % we already warned upon entering.
54         \else
55             \nag@warn{%
56                 \nag@doubledollar...\nag@doubledollar\space is obsolete.\MessageBreak
57                 Use \string\[\...\string\] et al. instead}%
58             \fi
59         \egroup\expandafter\nag@doubledollar@gobble
60     }{%
61         \egroup\nag@singledollar
62     }%
63 }
64 % we do the assignment here, which means any package that redefines
65 % \$ as well will silently disable us. This is a feature.
66 \global\let$\nag@maybe@dispmath%$
67 \egroup

```

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

```

68 \AtBeginDocument{%
69     \ifpackageloaded{rcs}{%
70         % this redefinition is functionally equivalent,

```

```

71 % but does not share actual code.
72 \renewcommand\RCS{\bgroup%
73   \catcode'\_ =\active
74   \catcode'\$=3 % this line added for compatibility.
75   \csname RCS_get_argument\endcsname
76 }
77 \PackageInfo{nag}{rcs.sty hack applied}%
78 }{}%
79 \@ifpackageloaded{svninfo}{%
80   \g@addto@macro\@svnBeginRead{\catcode'\$ 3 }%
81   \PackageInfo{nag}{svninfo.sty hack applied}%
82 }{}%
83 \@ifpackageloaded{svn}{%
84   \PackageInfo{nag}{svn.sty is broken: disabling dollar check}%
85   \catcode'\$ 3
86 }{}%
87 \@ifpackageloaded{rcsinfo}{%
88   \PackageInfo{nag}{rcsinfo.sty is broken: disabling dollar check}%
89   \catcode'\$ 3
90 }{}%
91 \@ifpackageloaded{pgf}{%
92   \PackageInfo{nag}{pgf.sty is broken: disabling dollar check}%
93   \catcode'\$ 3
94 }{}%
95 }
96
97
98 %% \S 1.7 cannot reasonably be checked programmatically
99 %% \S 1.8 \sloppy is called by parbox, among others, and would
100 %% give many spurious warnings.
101 %% \S 2.1.1
102 \ObsoleteCS[an old LaTeX 2.09 command]{bf}
103   {\protect\bfseries\space or \protect\textbf}
104 \ObsoleteCS[an old LaTeX 2.09 command]{it}
105   {\protect\itshape\space or \protect\textit}
106 \ObsoleteCS[an old LaTeX 2.09 command]{rm}
107   {\protect\rmfamily\space or \protect\textrm}
108 \ObsoleteCS[an old LaTeX 2.09 command]{sc}
109   {\protect\scshape\space or \protect\textsc}
110 \ObsoleteCS[an old LaTeX 2.09 command]{sf}
111   {\protect\sffamily\space or \protect\textsf}
112 \ObsoleteCS[an old LaTeX 2.09 command]{sl}
113   {\protect\slshape\space or \protect\textsl}
114 \ObsoleteCS[an old LaTeX 2.09 command]{tt}
115   {\protect\ttfamily\space or \protect\texttt}
116 \ObsoleteCS[an old LaTeX 2.09 command]{cal}
117   {\protect\mathcal}% Hmm, this is not in l2tabu?
118 %% \S 2.1.2
119 %% Gone with 1.8 because this never worked for the kernel \frac anyway.
120 %% \ObsoleteCS[TeX]{over}{\protect\frac}

```

```

121 %% \ObsoleteCS[TeX]{choose}{\protect\frac\space or amsmath's \protect\binom}
122 %% \S 2.1.3
123 \ObsoleteCS[TeX]{centerline}{\protect\centering\space or center environment}
124 %% \S 2.2.1
125 \ObsoleteClass{scrletter}{the scrletter2 package}
126 %% \S 2.2.2
127 \ObsoletePackage{epsf}{the graphicx package}
128 \ObsoletePackage{psfig}{the graphicx package}
129 \ObsoletePackage[deprecated]{epsfig}{the graphicx package directly}
130 %% \S 2.2.3
131 \ObsoletePackage{doublespace}{the setspace package}
132 %% \S 2.2.4
133 \ObsoletePackage{fancyheadings}{the fancyhdr or scrpage2 packages}
134 \ObsoletePackage{scrpage}{the scrpage2 package}
135 %% \S 2.2.5
136 \ObsoletePackage{isolatin}{the inputenc package with option latin1}
137 \ObsoletePackage{umlaut}{the inputenc package with suitable option
138   (latin1, utf8 ...)}
139 \ObsoletePackage{isolatin1}{the inputenc package with option latin1}
140 %% \S 2.2.6
141 \ObsoletePackage{tlenc}{the fontenc package with option T1}
142 %% \S 2.2.7 we don't check for bst yet.
143 %% (This is in l2tabu 1.8)
144 \ObsoletePackage{caption2}{the caption package v3.0 or later}
145 %% \S 2.3.1-3
146 \ObsoletePackage{times}
147   {the mathptmx, helvet (option scaled=.9), courier packages}
148 \ObsoletePackage{pslatex}
149   {the mathptmx, helvet (option scaled=.9), courier packages}
150 \ObsoletePackage{mathptm}
151   {the mathptmx package}
152 %% \S 2.3.4-5
153 \ObsoletePackage{palatino}
154   {the mathpazo, helvet (option scaled=.95), courier packages}
155 \ObsoletePackage{mathpple}{the mathpazo package}
156 %% \S 2.3.6 can't be checked
157 %% \S 2.3.7
158 \ObsoletePackage{euler}{the eulervm package}
159 \ObsoletePackage{utopia}{the fourier package}
160 %% \S 3.1
161 \NagDeclareFloat{figure}\NagDeclareFloat{table}%
162 \g@addto@macro\nag@labels{,label,caption@xlabel}%
163 % \changes{0.60}{2007/03/31}{alternate center-in-float check, doesn't
164 % take up as many macro names}
165 \nag@prepend{endcenter}{%
166   \ifx\@capytype\undefined\else
167     \nag@warn{\lq center\rq\space environment in \@capytype.\MessageBreak
168       Maybe you want \protect\centering\space instead}%
169   \fi
170 }%

```

```

171 %% The latter two are used by KOMA-Script, the last by hypcap.
172 % \changes{0.53}{2007/03/21}{hypcap support. (H.G.Krauth\"auser)}
173 % \changes{0.53}{2007/03/21}{topcapt support.}
174 \g@addto@macro\nag@captions{,caption,captionabove,captionbelow,hc@caption,topcaption}%
175
176 %% \S 3.2
177 \NotAnEnvironment{appendix}%
178 %% In the same vein:
179 \@for\sectioning:=frontmatter,mainmatter,backmatter\do{%
180   \expandafter\NotAnEnvironment\expandafter{\sectioning}%
181 }
182 %% \S 3.3
183 %% It's more trouble than it's worth to have another warning for
184 %% align*, since it passes through align.
185 \ObsoleteEnv{eqnarray}{amsmath's align}
186 %% \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.)

```

187 \ProvidesFile{nag-orthodox.cfg}
188     [2006/04/19 v1.8 strict rules for nag.sty (ulmi)]
189 \@for\fontcmd:=tiny,small,footnotesize,normalsize,large,Large,%
190     LARGE,huge,Huge\do{%
191   \expandafter\NotAnEnvironment\expandafter{\fontcmd}%
192 }%
193 \@for\fontcmd:=sffamily,rmfamily,ttfamily,%
194     bfseries,mdseries,scshape,%
195     itshape,upshape\do{%
196   \expandafter\NotAnEnvironment\expandafter{\fontcmd}%
197 }%
198 \@for\justsw:=centering,raggedleft,raggedright,%
199     RaggedLeft,RaggedRight\do{%
200   \expandafter\NotAnEnvironment\expandafter{\justsw}%
201 }
202 \@for\justenv:=center,flushleft,flushright\do{%
203   \expandafter\NotASwitch\expandafter{\justenv}%
204 }

```

1.6 nag-abort.cfg

Requesting this nag file will turn all complaints into errors.

```

205 \ProvidesFile{nag-abort.cfg}
206         [2007/11/10 v0.2 treat complaints as errors (ulmi)]
207 \DeclareRobustCommand\nag@warn[1]{%
208   \addtocounter{nag@sins}{1}%
209   \PackageError{nag}{#1}{#1}%
210 }
211 \DeclareRobustCommand\nag@warnNoLine[1]{%
212   \addtocounter{nag@sins}{1}%
213   \PackageError{nag}{#1}{#1}%
214 }

```

1.7 nag-experimental.cfg

Functionality that needs more testing.

```

215 \ProvidesFile{nag-experimental.cfg}
216         [2009/07/04 v0.62alpha2 experimental additions to nag (ulmi)]
217
218 Check if a float that may be positioned b is actually small enough for bottomfrac-
219 tion etc.
220
221 \let\@xa\expandafter
222 \newif\ifnag@dofloatsizecheck
223 \newif\ifnag@allfloatpositionsfailed
224 \newcommand\nag@allfloatsizechecks{}%
225 \newcommand\nag@onefloatsizecheck[2]{%
226   % #1 is size fraction of textheight,
227   % #2 is position to say in warning.
228   \ifdim \ht\@currbox>#1\textheight
229     \@tempdima -#1\textheight
230     \advance \@tempdima \ht\@currbox
231     \PackageInfo{nag}{Float too large for #2 by \the\@tempdima}%
232     % note we do not truncate.
233     % also, it's too late to add "p" now.
234   \else
235     \nag@allfloatpositionsfailedfalse
236   \fi
237 }
238
239 % \@currbox is current float box,
240 % \@fps is the current list of float specifiers.
241 \renewcommand\@largefloatcheck{%
242   \ifdim \ht\@currbox>\textheight
243     \@tempdima -\textheight
244     \advance \@tempdima \ht\@currbox
245     \@latex@warning {Float too large for page by \the\@tempdima}%
246     \ht\@currbox \textheight
247   \fi
248 }
249
250 %% the preceding is the original check.
251 \nag@dofloatsizechecktrue
252 \nag@allfloatpositionsfailedtrue
253 \def\nag@allfloatsizechecks{}%
254 \@xa\@xa\@xa\@tfor\@xa\@xa\@xa\nag@fltsz@tmp\@xa\@xa\@xa:\@xa\@xa\@xa=\cename @fps\endcsname

```



```

248 \ifx\nag@fltsz@tmp\relax
249 \nag@dofloatsizecheckfalse
250 \fi
251 \if\nag@fltsz@tmp !
252 \nag@dofloatsizecheckfalse
253 \else
254 \if\nag@fltsz@tmp t
255 \g@addto@macro\nag@allfloatsizechecks
256 {\nag@onefloatsizecheck{\topfraction}{top of page}}%
257 \else
258 \if\nag@fltsz@tmp b
259 \g@addto@macro\nag@allfloatsizechecks
260 {\nag@onefloatsizecheck{\bottomfraction}{bottom of page}}%
261 \else
262 \if\nag@fltsz@tmp p
263 \nag@allfloatpositionsfailedfalse
264 \fi
265 \fi
266 \fi
267 \fi
268 }%
269 \ifnag@dofloatsizecheck
270 \nag@allfloatsizechecks
271 \ifnag@allfloatpositionsfailed
272 \nag@warn{All float specifiers '\@fps' won't work}%
273 \fi
274 \fi
275 }%

```

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.

```

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

```

```

294 \@xa\AtEndDocument\@xa{%
295   \@xa\@latex@info@no@line\@xa{%
296     \nag@nofile
297   }%
298 }%
299 }}%
300 %

```

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.

```

301 \NeedsTeXFormat{LaTeX2e}
302 \ProvidesPackage{nag}[2010/04/05 0.621 warning about old commands (ulmi)]
303 \let\@xa\expandafter
304 \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.

```

305 \newcounter{nag@c}
306 \renewcommand\thenag@c{\roman{nag@c}}%
307 \setcounter{nag@c}{1}%
308 \begingroup
309   \let\@addtoreset\@gobbletwo
310   \newcounter{nag@sins}%
311 \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.

```

312 \newcommand\nag@ifundefined[1]{%
313   \begingroup
314   \@ifundefined{#1}{\endgroup\@firstoftwo}{\endgroup\@secondoftwo}%
315 }

```

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.

```

316 \newcommand\nag@prepend[2]{%
317 \nag@ifundefined{#1}{%
318 % if it doesn't exist, don't do anything.
319 \PackageInfo{nag}{%
320 Command \@backslashchar#1\space not defined, skipping amendment%
321 }%
322 }{%
323 \nag@ifundefined{#1 }{%
324 \let\nag@maybespace\@empty
325 }{%
326 \let\nag@maybespace\space
327 }\PackageInfo{nag}{%
328 % Command \@backslashchar#1\space appears robust\MessageBreak
329 % Modifying '@backslashchar#1\space' instead.
330 }%
331 }%
332 \@xa\let
333 \csname nag@@#1@\thenag@c\@xa\endcsname
334 \csname #1\nag@maybespace\endcsname
335 \@xa\DeclareRobustCommand\csname nag@@warning@\thenag@c\@xa\endcsname{%
336 #2%
337 }%
338 \@xa\nag@pr@p@nd\csname #1\nag@maybespace\@xa\endcsname
339 \csname nag@@#1@\thenag@c\@xa\endcsname
340 \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?

```

341 \stepcounter{nag@c}%
342 }%
343 }
344 \newcommand\nag@pr@p@nd[3]{%
345 \def#1{#3#2}%
346 }

```

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

```

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

```

```

353 \PackageWarningNoLine{nag}%
354 }

```

2.2 Obsoleteing 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.

```

355 \newcommand\ObsoleteCS[3][obsolete]{%
356 \AtBeginDocument{%
357 \nag@prepend{#2}{%
358 \nag@warn{%
359 Command \@backslashchar#2 is #1.
360 \MessageBreak
361 Use #3 instead}%
362 }%
363 }%
364 }

```

`\ObsoleteEnv`

```

365 \newcommand\ObsoleteEnv[3][obsolete]{%
366 \AtBeginDocument{%
367 \nag@prepend{#2}{%
368 \nag@warn{%
369 Environment #2 is #1.
370 \MessageBreak
371 Use #3 instead}%
372 }%
373 }%
374 }

```

2.3 Obsoleteing 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`).

```

375 \newcommand\nag@ifcsname[3]{%
376 \begingroup\@ifundefined{#1}{#3}{#2}\endgroup
377 }

```

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.

```

378 \nag@ifcsname@ifcsname}{%
379   \renewcommand*\nag@ifcsname[3]{%
380     \begingroup
381       % assume it won't be there.
382       \let\tmp@a\@secondoftwo
383       \ifcsname #1\endcsname
384       % It still might be relax from some other test. Thanks to J\'org
385       % Sommer for finding this bug.
386       \expandafter\ifx\csname #1\endcsname\relax
387       \else
388         % it's there after all
389         \let\tmp@a\@firstoftwo
390       \fi
391     \fi
392     \tmp@a{#2}{#3}%
393   \endgroup
394 }%
```

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

```

395 \global\let\nag@ifcsname\nag@ifcsname
396 }{}
```

`\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.

```

397 \newcommand\ObsoletePackage[3][obsolete]{%
398   \AtEndDocument{%
399 % |\@clsextension| is onlypreamble, for some reason.
400   \nag@ifcsname{ver@#2.sty}{%
401     \nag@warnNoLine{%
402       Package #2 is #1.\MessageBreak
403       Use #3 instead}%
404   }{}}%
405 }%
406 }
```

`\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.

```

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

```

413     Use #3 instead}%
414   }{}%
415 }%
416 }

```

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.

```

417 \newif\ifnag@haslabel
418 \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.

```

419 \newcommand\nag@hackfloat[1]{%
420   \nag@prepend{#1}{%
421     \global\nag@haslabelfalse\global\nag@hascaptionfalse
422   }%
423   \nag@prepend{end#1}{%
424     \ifnag@hascaption\relax\else
425       \nag@warn%
426     {#1 with no \protect\caption}%
427     \fi
428     % labels outside floats shouldn't complain:
429     \global\nag@hascaptiontrue
430     % (we do this always because it needs to be global)
431   }%
432 }

```

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.

```

433 \AtBeginDocument{%
434 \g@addto@macro{\@preamblecmds}{%
435   \@for\labelprovider:=\nag@labels\do{%
436     \ifx\labelprovider\@empty\else
437       \nag@prepend{\labelprovider}%
438       {\nag@captioncheck\nag@haslabeltrue}%
439     \fi
440   }%
441   \@for\captionprovider:=\nag@captions\do{%
442     \ifx\captionprovider\@empty\else
443       \nag@prepend{\captionprovider}{\global\nag@hascaptiontrue}%
444     \fi
445   }%
446 }%
447 }
448 \newcommand\nag@captioncheck{%
449   \ifnag@hascaption\else
450   \nag@warn{\protect\label\space in float, but not after
451     \protect\caption}%
452   \fi
453 }

```

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.

```

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

```

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

```

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

```

```

462 }%
463 }
    but only after all other packages get their chance to add to the list:
464 \AtBeginDocument{%
465   \nag@floatsetup
466 }
    At the very end, we will display a running total of complaints.
467 \AtBeginDocument{%
468   \AtEndDocument{%
469     \ifnum\value{nag@sins}>0%
470     \PackageWarningNoLine{nag}{\arabic{nag@sins} complaints
471       in total}%
472     \else
473     \typeout{No complaints by nag.}%
474     \fi
475   }%
476 }

```

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:

```

477 \DeclareRobustCommand\nag@ifCurrentEnvironment[3]{%
478   \bgroup
479   \def\tmp@a{#1}%
480   \ifx\@currenvir\tmp@a
481     #2%
482   \else
483     #3%
484   \fi
485 \egroup
486 }

```

And now, the two variations there are:

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

```

487 \newcommand\NotAnEnvironment[1]{%
488   \AtBeginDocument{%
489     \nag@prepend{#1}{%

```



```

490     \nag@ifCurrentEnvironment{#1}{%
491         \nag@warn{%
492             There is no environment ‘‘#1’’. \MessageBreak
493             Maybe you want a grouped \@backslashchar#1
494         }%
495     }{% OK case.
496     }%
497 }%
498 }%
499 }

```

\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.

```

500 % we need to maintain a stack of environments that are used in the
501 % \foo...\endfoo way.
502 \newcommand\nag@envstack{\relax}
503
504 \DeclareRobustCommand\nag@beginenv[1]{%
505     % push a begin-entry onto the stack. Form is
506     % |{\foo{lineno}}| for environment foo.
507     \bgroup
508     \@xa\toks@\@xa{\nag@envstack}%
509     \xdef\nag@envstack{%
510         \@nx{%
511             \@xa\@nx\csname #1\endcsname
512             \@nx{\the\inputlineno\@nx}%
513             \@nx}%
514             \the\toks@
515         }%
516     \egroup
517 }
518 \DeclareRobustCommand\nag@endenv[1]{%
519     % extract the first entry.
520     \@xa\nag@end@nv\nag@envstack\@nil #1\@nil
521 }
522
523 \def\nag@end@nv#1#2\@nil #3\@nil{%
524     \def\tmp@a{#1}%
525     \def\tmp@b{\relax}%
526     \ifx\tmp@a\tmp@b
527     % This was the end-of-stack flag.
528     \nag@warn{‘‘\@backslashchar end#3’’ without matching
529         ‘‘\@backslashchar #3’’}
530     \else
531     % We may assume this is a proper entry. See if the begin-token on
532     % the stack matches what |\nag@endenv| was passed.
533     \@xa\ifx\csname #3\@xa\endcsname\@firstoftwo #1%
534     %OK case, just pop the entry.

```

```

535     \gdef\nag@envstack{#2}%
536   \else
537     % error case
538     \nag@warn{%
539       You cannot close ‘‘\@xa\string\@firstoftwo #1’’ on line
540       \@secondoftwo #1 with ‘‘\@backslashchar end#3’’%
541     }%
542     % leave it on the stack. Some case of misnesting will always cause
543     % horrible amounts of follow-up errors. Also, scare them!
544     \fi
545   \fi
546 }

```

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

```

547 \AtEndDocument{%
548   \@xa\@tfor\@xa\looseends\@xa:\@xa=\nag@envstack\do{%
549     \@xa\ifx\looseends\relax\else
550       \nag@warnNoLine{Unmatched
551         ‘‘\@xa\@xa\@xa\string\@xa\@firstoftwo\looseends’’
552         command on line
553         \@xa\@xa\@xa\string\@xa\@secondoftwo\looseends%
554       }%
555     \fi
556   }%
557 }

```

Now, the user-side command is easy.

```

558 \newcommand\NotASwitch[1]{%
559   \AtBeginDocument{%
560     \nag@prepend{#1}{%
561       \nag@beginenv{#1}%
562     }%
563     \nag@prepend{end#1}{%
564       \nag@endenv{#1}%
565     }%
566   }%
567 }

```

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.

```
568 \AtBeginDocument{%
569   \nag@ifcsname{ver@subfig.sty}{%
570     \PackageInfo{nag}{Attempting subfig hack\@gobble}%
571     \nag@maybehacksubfig
572   }{%
573   }%
574 }
575 \def\nag@maybehacksubfig{%
576   %
577   % of course, i need to touch the single longest definition in
578   % subfig.sty, to amend one single command...
579   %
580   % The definition is taken from subfig.sty 1.3 dated 2005/07/05 by
581   % S.D. Cochran, where it is called sf@@@subfloat, and appears here
582   % under the conditions of section 6 of the LPPL 1.3. The subfig
583   % package is available on a CTAN mirror near you.
584   %
585   \long\def\nag@@original@@sf@@@subfloat##1[##2][##3]##4{%
586     \@ifundefined{FBsc@max}{%
587       }{%
588         \FB@readaux{\let\FBsuboheight\relax}%
589       }%
590       \@tempcnta=\@ne
591       \if@minipage
592         \@tempcnta=\z@
593       \else\ifdim \lastskip=\z@ \else
594         \@tempcnta=\tw@
595       \fi\fi
596       \ifmaincaptiontop
597         \sf@top=\sf@nearskip
598         \sf@bottom=\sf@farskip
599       \else
600         \sf@top=\sf@farskip
601         \sf@bottom=\sf@nearskip
602       \fi
603       \leavevmode
604       \setbox\@tempboxa \hbox{%
605         ##4}%
606       \@tempdima=\wd\@tempboxa
607       \@ifundefined{FBsc@max}{%
608         }{%
609           \global\advance\Xhsize-\wd\@tempboxa
```

```

610 \dimen@=\ht\@tempboxa
611 \advance\dimen@\dp\@tempboxa
612 \ifdim\dimen@>\FBso@max
613 \global\FBso@max\dimen@
614 \fi
615 }%
616 \vtop\bgroup
617 \vbox\bgroup
618 \ifcase\@tempcnta
619 \@minipagefalse
620 \or
621 \vskip\sf@top
622 \or
623 \ifdim \lastskip=\z@ \else
624 \@tempskipb\sf@top\relax\@xaddvskip
625 \fi
626 \fi
627 \sf@ifpositiontop{%
628 \ifx \@empty##3\relax \else
629 \sf@subcaption{##1}{##2}{##3}%
630 \vskip\sf@capskip
631 \vskip\sf@captopadj
632 \fi\egroup
633 \hrule width0pt height0pt depth0pt
634 \box\@tempboxa
635 }{%
636 \@ifundefined{FBsc@max}{%
637 \box\@tempboxa
638 }{%
639 \ifx\FBsuboheight\relax
640 \box\@tempboxa
641 \else
642 \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
643 \fi}%
644 \egroup
645 \ifx \@empty##3\relax \else
646 \vskip\sf@capskip
647 \hrule width0pt height0pt depth0pt
648 \sf@subcaption{##1}{##2}{##3}%
649 \fi
650 }%
651 \vskip\sf@bottom
652 \egroup
653 \@ifundefined{FBsc@max}{%
654 }{%
655 \addtocounter{FRobj}{-1}%
656 \ifnum\c@FRobj=0\else
657 \subfloatrowsep
658 \fi
659 }%

```

```

660     \ifmaincaptiontop\else
661         \global\advance\@nameuse{c@\@capttype}\m@ne
662     \fi
663     \endgroup\ignorespaces}%
664 %
665 \expandafter\ifx\csname sf@@@subfloat\endcsname\nag@@original@@sf@@@subfloat
666 % yup, that's it.
667 \PackageInfo{nag}{OK, equivalent to subfig 1.3, redefining
668 \backslashchar sf@@@subfloat@gobble}%
669 \global\long\def\sf@@@subfloat##1[##2][##3]##4{%
670 \ifundefined{FBsc@max}{%
671 }{%
672 \FB@readaux{\let\FBsuboheight\relax}%
673 }%
674 \@tempcnta=\@ne
675 \if@minipage
676 \@tempcnta=\z@
677 \else\ifdim \lastskip=\z@ \else
678 \@tempcnta=\tw@
679 \fi\fi
680 \ifmaincaptiontop
681 \sf@top=\sf@nearskip
682 \sf@bottom=\sf@farskip
683 \else
684 \sf@top=\sf@farskip
685 \sf@bottom=\sf@nearskip
686 \fi
687 \leavevmode
688 \setbox\@tempboxa \hbox{%
689 % ulmi: new 2007/02/25: #4 may contain label command
690 \csname nag@hascaptiontrue\endcsname
691 % and that was it.
692 ##4}%
693 \@tempdima=\wd\@tempboxa
694 \@ifundefined{FBsc@max}{%
695 }{%
696 \global\advance\Xhsize-\wd\@tempboxa
697 \dimen@=\ht\@tempboxa
698 \advance\dimen@\dp\@tempboxa
699 \ifdim\dimen@>\FBso@max
700 \global\FBso@max\dimen@
701 \fi
702 }%
703 \vtop\bgroup
704 % ulmi: new 2007/05/10: #2, #3 may contain label command
705 \csname nag@hascaptiontrue\endcsname
706 % and that was it.
707 \vbox\bgroup
708 \ifcase\@tempcnta
709 \@minipagefalse

```

```

710     \or
711         \vskip\sf@top
712     \or
713         \ifdim \lastskip=\z@ \else
714             \@tempskipb\sf@top\relax\@xaddvskip
715         \fi
716     \fi
717     \sf@ifpositiontop{%
718         \ifx \@empty##3\relax \else
719             \sf@subcaption{##1}{##2}{##3}%
720             \vskip\sf@capskip
721             \vskip\sf@captopadj
722         \fi\egroup
723         \hrule width0pt height0pt depth0pt
724         \box\@tempboxa
725     }{%
726         \@ifundefined{FBsc@max}{%
727             \box\@tempboxa
728         }{%
729             \ifx\FBsuboheight\relax
730                 \box\@tempboxa
731             \else
732                 \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
733             \fi}%
734     \egroup
735     \ifx \@empty##3\relax \else
736         \vskip\sf@capskip
737         \hrule width0pt height0pt depth0pt
738         \sf@subcaption{##1}{##2}{##3}%
739     \fi
740     }%
741     \vskip\sf@bottom
742 \egroup
743 \@ifundefined{FBsc@max}{%
744 }{%
745     \addtocounter{FRobj}{-1}%
746     \ifnum\c@FRobj=0\else
747         \subfloatrowsep
748     \fi
749 }%
750 \ifmaincaptiontop\else
751     \global\advance\@nameuse{c@\@capttype}\m@ne
752 \fi
753 \endgroup\ignorespaces}%
754 \else
755     \PackageInfo{nag}{Not redefining sf@@@subfloat, it looks odd@gobble}
756 \fi
757 }

```

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`.

```
758 \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.

```
759 \DeclareOption*{%
760   \InputIfFileExists{nag-\CurrentOption.cfg}{%
761     \PackageInfo{nag}{%
762       Loaded nag-\CurrentOption.cfg
763     }
764   }{%
765     \InputIfFileExists{\CurrentOption.nag}{%
766       \PackageWarningNoLine{nag}{%
767         Loaded old-style config file \CurrentOption.nag.\MessageBreak
768         Consider renaming the file to nag-\CurrentOption.cfg
769       }%
770     }
```

```

770   }{%
771     \PackageWarningNoLine{nag}{Required ruleset
772       \CurrentOption, and it wasn't there}
773   }%
774 }
775 }
776 \ProcessOptions*

```

Change History

0.1	General: First official version.	1	0.55	General: Some spaces crept in in	0.5	1
0.2	General: Added abort.nag, sug-		0.60	General: @preamblecmds		15
	gested by Michael Zedler	1		fixes double-dollar in conjunction		
	Rephrased umlaut.sty warning,			with hyperref; documents in-		
	suggested by Patrick Happel.	1		compatibility with rotating.		1
0.3	General: Fixed missing globals	14		Captions/Labels now done only		
	New ifdefined that won't relax			once, and not every time we en-		
	the commands	1		ter a float		15
0.4	General: bugfix	14	0.60alpha	General: changes the way la-		
	config file names changed to free			bel/caption is handled, this		
	extension	23		eliminates the current limit of		
	Handling command vs. environ-			some thousand floats you can		
	ment; bugfixes	1		have in your document. (I won-		
0.5	General: Handle the case that some-		0.60alpha2	der why nobody noticed).		1
	body else relaxes the ver@-			General: is more careful around		
	commands. Stack-based Not-			commands that aren't there.		1
	ASwitch.	1		\nag@prepend: don't even relax un-		
0.51	\nag@prepend: bugfix	11	0.60alpha4	known commands (J.Sommer)		11
0.52	General: Command NagDeclare-			General: handles eqnarray it-		
	Float added	23		self and has code in nag-		
	made eTeX-ifcsname more ro-			experimental.cfg to handle		
	bust	13		double-dollar in a more robust		
	twiddle subfig's bowels	19		way that onlyamsmath.		1
	\nag@prepend: info	11		tarballs now unpack into a sub-		
0.53	General: bugfix: more Robustness.		0.60alpha5	directory like proper citizens		
	(Jörg Sommer)	16		should.		1
0.54	\NotASwitch: bugfix: can't get			General: improves compatibility		
	around the token register. (Jörg		0.61	with subfig.		1
	Sommer)	17		General: is 0.61alpha5 with some		
				typos in the docs fixed.		1

0.61alpha1	General: fixes warnings in toc/lof/lot and unsightly uppercasing.	1	some chapters.)	1
	roman counter (external file issue)	10	sin counter should not be saved by include	10
	<code>\nag@prepend</code> : Extra indirection of warnings for robustness (uppercasing/LoF issues)	11	0.61alpha5	General: introduces compatibility hacks with version control packages which rely on dollar having constant catcode. (Workaround for svninfo and rcs, all other packages now disable double-dollar checking.)
	<code>\nag@warn</code> : Made robust.	11	0.61alpha6	General: Compatibility w/ VCS packages, pgf
0.61alpha2	General: fixes the warnings, without generating too many duplicates.	1	0.62	General: fixes a bug in the float placement code and adds more compatibility with the caption package.
	<code>\nag@prepend</code> : Creep under existing robust cover	11	0.621	General: Bugfix concerning unknown command in math mode
0.61alpha3	General: warns about inputs that fail (in particular includes that fail) and notes if a float has position t/b but is too large to ever go into such a position (log only).	1	0.62alpha1	General: Bigger warning if all float positions fail
0.61alpha4	General: exempts the complaints counter from include trickery. (Previously, nag would get confused if you includeonly only		0.62alpha2	General: Fix for marginpar etc which don't have fps

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.

Symbols	234, 237, 239, 241	<code>\@ifpackageloaded</code>
<code>\"</code>	172, 384	<code>\@currrent</code>
<code>\\$</code>	65, 74, 80, 85, 89, 93	<code>\@empty</code>
<code>\@@input</code>	277	436, 442, 459,
<code>\@addtoreset</code>	309	628, 645, 718, 735
<code>\@backslashchar</code>	320, 328,	<code>\@filef@und</code>
	329, 359, 493,	<code>\@firstoftwo</code>
	528, 529, 540, 668	19, 24, 43, 314,
<code>\@capttype</code>		389, 533, 539, 551
	166, 167, 661, 751	<code>\@fps</code>
<code>\@clsextension</code>	399, 409	235, 272
<code>\@currbox</code>	224, 226,	<code>\@gobble</code>
		59, 570, 668, 755
		<code>\@gobbletwo</code>
		309
		<code>\@ifnextchar</code>
		51
		<code>\@secondoftwo</code>
		20, 25,

<code>\nag@doubledollar</code> ..	<code>\nag@warn</code> .. 55, 167,	R
..... 12, 56, 59	207, 272, <u>347</u> ,	<code>\RCS</code> 72
<code>\nag@end@nv</code> ... 520, 523	358, 368, 425,	<code>\renewcommand</code>
<code>\nag@endenv</code> 518, 532, 564	450, 491, 528, 538	. 72, 236, 306, 379
<code>\nag@envstack</code>	<code>\nag@warnNoLine</code> 211,	<code>\rmfamily</code> 107
.... 502, 508,	351, 401, 411, 550	<code>\roman</code> 306
509, 520, 535, 548	<code>\NagDeclareFloat</code> ..	
<code>\nag@expanding@voodoo</code> 161, 758	S
..... 14, 32	<code>\NeedsTeXFormat</code> ... 301	<code>\S</code> 7, 10, 11,
<code>\nag@floats</code> 454, 458, 758	<code>\newcommand</code> ... 220,	98, 99, 101, 118,
<code>\nag@floatsetup</code> 457, 465	221, 312, 316,	122, 124, 126,
<code>\nag@fltsz@tmp</code>	344, 355, 365,	130, 132, 135,
.... 247, 248,	375, 397, 407,	140, 142, 145,
251, 254, 258, 262	419, 448, 457,	152, 156, 157,
<code>\nag@hackfloat</code> 419, 460	487, 502, 558, 758	160, 176, 182, 186
<code>\nag@hascaptionfalse</code>	<code>\newcounter</code> ... 305, 310	<code>\scshape</code> 109
..... 421	<code>\newif</code> 218, 219, 417, 418	<code>\sectioning</code> ... 179, 180
<code>\nag@hascaptiontrue</code>	<code>\noexpand</code> 304	<code>\setbox</code> 604, 688
.... 418, 429, 443	<code>\NotAnEnvironment</code> ..	<code>\setcounter</code> 307
<code>\nag@haslabelfalse</code> . 421 177, 180,	<code>\sf@@@subfloat</code> 669
<code>\nag@haslabeltrue</code> .. 438	191, 196, 200, <u>487</u>	<code>\sf@bottom</code> 598, 601,
<code>\nag@ifcfname</code>	<code>\NotASwitch</code> 203, <u>500</u> , 558	651, 682, 685, 741
. 375, 378, 379,		<code>\sf@capskip</code>
395, 400, 410, 569	O	. 630, 646, 720, 736
<code>\nag@ifCurrentEnvironment</code>	<code>\ObsoleteClass</code> 125, <u>407</u>	<code>\sf@captopadj</code> . 631, 721
..... 477, 490	<code>\ObsoleteCS</code> 102, 104,	<code>\sf@farskip</code>
<code>\nag@ifundefined</code> ..	106, 108, 110,	. 598, 600, 682, 684
.... 312, 317, 323	112, 114, 116,	<code>\sf@ifpositiontop</code> ..
<code>\nag@labels</code> 162, 435, 455	120, 121, 123, <u>355</u> 627, 717
<code>\nag@maybe@dispmath</code>	<code>\ObsoleteEnv</code> .. 185, <u>365</u>	<code>\sf@nearskip</code>
..... 33, 48	<code>\ObsoletePackage</code> ..	. 597, 601, 681, 685
<code>\nag@maybe@dispmath</code> .	. 8, 9, 127–129,	<code>\sf@subcaption</code>
..... 16, 66	131, 133, 134,	. 629, 648, 719, 738
<code>\nag@maybe@hacksubfig</code>	136, 137, 139,	<code>\sf@top</code> 597,
..... 571, 575	141, 144, 146,	600, 621, 624,
<code>\nag@maybespace</code> ...	148, 150, 153,	681, 684, 711, 714
. 324, 326, 334, 338	155, 158, 159, <u>397</u>	<code>\sffamily</code> 111
<code>\nag@nofile</code>		<code>\sloppy</code> 99
. 281, 284, 293, 296	P	<code>\slshape</code> 113
<code>\nag@onefloatsizecheck</code>	<code>\PackageError</code> . 209, 213	<code>\string</code> 57, 539, 551, 553
.... 221, 256, 260	<code>\PackageInfo</code> ... 77,	<code>\subfloatrowsep</code> 657, 747
<code>\nag@pr@p@nd</code> .. 338, 344	81, 84, 88, 92,	
<code>\nag@prepend</code> .. 165,	227, 319, 327,	T
<u>312</u> , 357, 367,	570, 667, 755, 761	<code>\texorpdfstring</code> . 17, 43
420, 423, 437,	<code>\protected@edef</code> ... 281	<code>\textbf</code> 103
443, 489, 560, 563	<code>\providecommand</code> ... 43	<code>\textheight</code> ... 224,
<code>\nag@quark</code> 45, 50	<code>\ProvidesFile</code>	225, 237, 238, 241
<code>\nag@singledollar</code> 13,	.. 1, 187, 205, 215	<code>\textit</code> 105
14, 22, 35, 40, 61	<code>\ProvidesPackage</code> .. 302	<code>\textrm</code> 107
		<code>\textsc</code> 109

<code>\textsf</code>	111	333, 335, 339, 340	X
<code>\textsl</code>	113	<code>\toks@</code>	508, 514
<code>\texttt</code>	115	<code>\topfraction</code>	256
<code>\thenag@c</code>	306,	<code>\ttfamily</code>	115
		<code>\xdef</code>	509
		<code>\Xhsize</code>	609, 696