

The `nag` package*

Dr. Ulrich Michael Schwarz[†]

November 27, 2011

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 All the same, there are commands, classes and packages which are outdated and superseded.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>	8
1.7	<code>nag-experimental.cfg</code>	8
2	Author-side considerations and implementation.	11
2.1	Low-level tools.	11
2.2	Obsoletifying commands.	13
2.3	Obsoletifying packages and classes.	14
2.4	Common float errors and no-nos.	16
3	Switch vs. Environment	18
4	Compatibility issues	21
4.1	The <code>caption</code> package	21
4.2	The <code>subfig</code> package	21
4.3	The <code>float</code> package	25
4.4	The <code>topcapt</code> package and the <code>subfig</code> package	25
4.5	The <code>rotating</code> package	25
4.6	Version control packages	26

*This document corresponds to `nag` 0.7, dated 2011/11/19. 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           [2010/05/17 v2.11 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}%$
45 \AtEndDocument{\catcode'\$=3 \relax}

```

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.

```

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57           \nag@doubledollar...\nag@doubledollar\space is obsolete.\MessageBreak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

```

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

```

69 \AtBeginDocument{%
70   \@ifpackageloaded{rcs}{%

```

```

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

```

```

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

```

```

171 }%
172 %% The latter two are used by KOMA-Script, the last by hypcap.
173 % \changes{0.53}{2007/03/21}{hypcap support. (H.G.Krauth\ "auser)}
174 % \changes{0.53}{2007/03/21}{topcapt support.}
175 \g@addto@macro\nag@captions{,caption,captionabove,captionbelow,hc@caption,topcaption}%
176
177 %% \S 3.2
178 \NotAnEnvironment{appendix}%
179 %% In the same vein:
180 \@for\sectioning:=frontmatter,mainmatter,backmatter\do{%
181   \expandafter\NotAnEnvironment\expandafter{\sectioning}%
182 }
183 %% \S 3.3
184 %% It's more trouble than it's worth to have another warning for
185 %% align*, since it passes through align.
186 \ObsoleteEnv{eqnarray}{amsmath's align}
187 %% \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.)

```

188 \ProvidesFile{nag-orthodox.cfg}
189   [2006/04/19 v1.8 strict rules for nag.sty (ulmi)]
190 \@for\fontcmd:=tiny,small,footnotesize,normalsize,large,Large,%
191   LARGE,huge,Huge\do{%
192   \expandafter\NotAnEnvironment\expandafter{\fontcmd}%
193 }%
194 \@for\fontcmd:=sffamily,rmfamily,ttfamily,%
195   bfseries,mdseries,scshape,%
196   itshape,upshape\do{%
197   \expandafter\NotAnEnvironment\expandafter{\fontcmd}%
198 }%
199 \@for\justsw:=centering,raggedleft,raggedright,%
200   RaggedLeft,RaggedRight\do{%
201   \expandafter\NotAnEnvironment\expandafter{\justsw}%
202 }
203 \@for\justenv:=center,flushleft,flushright\do{%
204   \expandafter\NotASwitch\expandafter{\justenv}%
205 }

```

1.6 nag-abort.cfg

Requesting this nag file will turn all complaints into errors.

```
206 \ProvidesFile{nag-abort.cfg}
207           [2007/11/10 v0.2 treat complaints as errors (ulmi)]
208 \DeclareRobustCommand\nag@warn[1]{%
209   \addtocounter{nag@sins}{1}%
210   \PackageError{nag}{#1}{#1}%
211 }
212 \DeclareRobustCommand\nag@warnNoLine[1]{%
213   \addtocounter{nag@sins}{1}%
214   \PackageError{nag}{#1}{#1}%
215 }
```

1.7 nag-experimental.cfg

Functionality that needs more testing.

```
216 \ProvidesFile{nag-experimental.cfg}
217           [2009/07/04 v0.62alpha2 experimental additions to nag (ulmi)]
    Patch handling of nofiles: suppressed lines give an Info-level message in the
    logfile now. The message doesn't quite give the original line, but a sanitized
    version. Reason: otherwise, we might need to execute the setup code #2.
218 \long\def\nag@protected@dontwrite#1#2#3%
219   {\write\m@ne}%
220   \def\nag@line{#3}%
221   \@onelevel@sanitize\nag@line
222   \PackageInfo{nag}{%
223     \string\nofiles\space in effect.
224     Did not write line \MessageBreak
225     '\nag@line'
226   }%
227   \if@nobreak\ifvmode\nobreak\fi\fi}%
228
229 \if@filesw
230   \def\nofiles{%
231     \@fileswfalse
232     \typeout{No auxiliary output files.^^J}%
233     \global\let\protected@write=\nag@protected@dontwrite
234     \let\makeindex\relax
235     \let\makeglossary\relax}
236 \else
237   % already \nofiles.
238   \global\let\protected@write=\nag@protected@dontwrite
239 \fi
    Amend "no space for a new foo" message to point out eTeX alleviates some
    problems in that area.
240 \gdef\ch@ck#1#2#3{%
```



```

241 \ifnum\count1#1<#2\else
242   \errhelp{%
243     eTeX has more counters, dimens, etc., maybe
244     that will help.
245   }
246   \errmessage{No room for a new #3}%
247 \fi}

248 \def\@testdef #1#2#3{%
249   \def\reserved@a{#3}%
250   \expandafter \ifx \csname #1@#2\endcsname\reserved@a
251   \else
252     \@tempswattrue
253     \begingroup
254     \@onelevel@sanitize\reserved@a
255     \expandafter\let\expandafter\nag@tmpb\csname #1@#2\endcsname
256     \ifx\nag@tmpb\relax
257       \let\nag@tmpb\@empty
258     \else
259       \@onelevel@sanitize\nag@tmpb
260     \fi
261     \PackageInfo{nag}{%
262       Label '#2' appears to have changed from\MessageBreak
263       '\nag@tmpb'\MessageBreak
264       to '\reserved@a'
265     }%
266     \endgroup
267 \fi}

```

Check if a float that may be positioned b is actually small enough for bottom-fraction etc.

```

268 \let\@xa\expandafter
269 \newif\ifnag@dofloatsizecheck
270 \newif\ifnag@allfloatpositionsfailed
271 \newcommand\nag@allfloatsizechecks{}%
272 \newcommand\nag@onefloatsizecheck[2]{%
273   % #1 is size fraction of textheight,
274   % #2 is position to say in warning.
275   \ifdim \ht\@currbox>#1\textheight
276     \@tempdima -#1\textheight
277     \advance \@tempdima \ht\@currbox
278     \PackageInfo{nag}{Float too large for #2 by \the\@tempdima}%
279     % note we do not truncate.
280     % also, it's too late to add "p" now.
281   \else
282     \nag@allfloatpositionsfailedfalse
283   \fi
284 }
285 % \@currbox is current float box,
286 % \@fps is the current list of float specifiers.
287 \renewcommand\@largefloatcheck{%

```

```

288 \ifdim \ht\@currbox>\textheight
289   \@tempdima -\textheight
290   \advance \@tempdima \ht\@currbox
291   \@latex@warning {Float too large for page by \the\@tempdima}%
292   \ht\@currbox \textheight
293 \fi
294 %% the preceding is the original check.
295 \nag@dofloatsizechecktrue
296 \nag@allfloatpositionsfailedtrue
297 \def\nag@allfloatsizechecks{}%
298 \@xa\@xa\@xa\@tfor\@xa\@xa\@xa\nag@fltsz@tmp\@xa\@xa\@xa:\@xa\@xa\@xa=\csname @fps\endcsname
299   \ifx\nag@fltsz@tmp\relax
300     \nag@dofloatsizecheckfalse
301   \fi
302   \if\nag@fltsz@tmp !
303     \nag@dofloatsizecheckfalse
304   \else
305     \if\nag@fltsz@tmp t
306       \g@addto@macro\nag@allfloatsizechecks
307         {\nag@onefloatsizecheck{\topfraction}{top of page}}%
308     \else
309       \if\nag@fltsz@tmp b
310         \g@addto@macro\nag@allfloatsizechecks
311           {\nag@onefloatsizecheck{\bottomfraction}{bottom of page}}%
312       \else
313         \if\nag@fltsz@tmp p
314           \nag@allfloatpositionsfailedfalse
315         \fi
316       \fi
317     \fi
318   \fi
319 }%
320 \ifnag@dofloatsizecheck
321   \nag@allfloatsizechecks
322   \ifnag@allfloatpositionsfailed
323     \nag@warn{All float specifiers '@fps' won't work}%
324   \fi
325 \fi
326 }%

```

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.

```

327 \def\@input#1{%
328   \IfFileExists{#1}{\@@input\@filef@und}{%
329     \typeout{No file #1.}
330     \@latex@warning{File '#1' not found}
331     %{The file '#1' was requested but not found }
332     \protected@edef\nag@nofile{File '#1' requested, but not found}%
333     \@xa\AtEndDocument\@xa}%

```

```

334 \@xa\@latex@info@no@line\@xa{%
335 \nag@nofile
336 }%
337 }%
338 }}%
339 %
340 \def\@input@#1{\InputIfFileExists{#1}{}}{%
341 \typeout{No file #1.}
342 \@latex@warning{File ‘#1’ not found}
343 {The file ‘#1’ was requested but not found }
344 \edef\nag@nofile{File ‘#1’ requested, but not found}%
345 \@xa\@AtEndDocument\@xa{%
346 \@xa\@latex@info@no@line\@xa{%
347 \nag@nofile
348 }%
349 }%
350 }}%
351 %

```

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.

```

352 \NeedsTeXFormat{LaTeX2e}
353 \ProvidesPackage{nag}[2011/11/25 0.7 warning about old commands (ulmi)]
354 \let\@xa\expandafter
355 \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.

```

356 \newcounter{nag@c}
357 \renewcommand\thenag@c{\roman{nag@c}}%
358 \setcounter{nag@c}{1}%
359 \begingroup
360 \let\@addtoreset@gobbletwo
361 \newcounter{nag@sins}%
362 \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.

```
363 \newcommand\nag@ifundefined[1]{%
364   \begingroup
365   \@ifundefined{#1}{\endgroup\@firstoftwo}{\endgroup\@secondoftwo}%
366 }
```

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.

```
367 \newcommand\nag@prepend[2]{%
368   \nag@ifundefined{#1}{%
369     % if it doesn't exist, don't do anything.
370     \PackageInfo{nag}{%
371       Command \@backslashchar#1\space not defined, skipping amendment%
372     }%
373   }{%
374     \nag@ifundefined{#1 }{%
375       \let\nag@maybespace\@empty
376     }{%
377       \let\nag@maybespace\space
378       \% \PackageInfo{nag}{%
379         % Command \@backslashchar#1\space appears robust\MessageBreak
380         % Modifying '@@backslashchar#1\space' instead.
381         }%
382     }%
383     \@xa\let
384     \csname nag@@#1@\thenag@c\@xa\endcsname
385     \csname #1\nag@maybespace\endcsname
386     \@xa\DeclareRobustCommand\csname nag@@warning@\thenag@c\@xa\endcsname{%
387       #2%
388     }%
389     \@xa\nag@pr@p@nd\csname #1\nag@maybespace\@xa\endcsname
390     \csname nag@@#1@\thenag@c\@xa\endcsname
391     \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?

```
392   \stepcounter{nag@c}%
393 }%
394 }
```

```

395 \newcommand\nag@pr@p@end[3]{%
396   \def#1{#3#2}%
397 }

```

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

```

398 \DeclareRobustCommand\nag@warn{%
399   \addtocounter{nag@sins}{1}%
400   \PackageWarning{nag}%
401 }
402 \DeclareRobustCommand\nag@warnNoLine{%
403   \addtocounter{nag@sins}{1}%
404   \PackageWarningNoLine{nag}%
405 }
406 \providecommand\PackageInfoNoLine[2]{%
407   \PackageInfo{#1}{#2@gobble}%
408 }
409 \DeclareRobustCommand\nag@suggestNoLine[1]{%
410   \PackageInfoNoLine{nag}{#1}%
411 }

```

2.2 Obsoleteing commands.

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

`\ObsoleteCS` Usage: `\ObsoleteCS[reason][CS][suggestions]` Mark `\langle CS \rangle` as obsolete. `\langle reason \rangle` defaults to obsolete. When the macro is used anyway, the following warning is logged:
Command `\langle CS \rangle` is `\langle reason \rangle`. Use `\langle suggestions \rangle` instead.

```

412 \newcommand\ObsoleteCS[3][obsolete]{%
413   \AtBeginDocument{%
414     \nag@prepend{#2}{%
415       \nag@warn{%
416 Command \@backslashchar#2 is #1.
417 \MessageBreak
418 Use #3 instead}%
419     }%
420   }%
421 }

```

`\ObsoleteEnv`

```

422 \newcommand\ObsoleteEnv[3][obsolete]{%
423   \AtBeginDocument{%
424     \nag@prepend{#2}{%
425       \nag@warn{%
426 Environment #2 is #1.
427 \MessageBreak
428 Use #3 instead}%
429     }%

```

```

430 }%
431 }

```

2.3 Obsoletifying packages and classes.

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

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

```

432 \newcommand\nag@ifcsname[3]{%
433   \begingroup\@ifundefined{#1}{#3}{#2}\endgroup
434 }

```

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.

```

435 \nag@ifcsname{ifcsname}{%
436   \renewcommand*\nag@ifcsname[3]{%
437     \begingroup
438       % assume it won't be there.
439       \let\tmp@a\@secondoftwo
440       \ifcsname #1\endcsname
441       % It still might be relax from some other test. Thanks to J\"org
442       % Sommer for finding this bug.
443       \expandafter\ifx\csname #1\endcsname\relax
444       \else
445         % it's there after all
446         \let\tmp@a\@firstoftwo
447       \fi
448     \fi
449     \tmp@a{#2}{#3}%
450   \endgroup
451 }%

```

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

```

452 \global\let\nag@ifcsname\nag@ifcsname
453 }{}

```

`\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.

```

454 \newcommand\ObsoletePackage[3][obsolete]{%
455   \AtEndDocument{%
456     |\@clsextension| is onlypreamble, for some reason.
457     \nag@ifcsname{ver@#2.sty}{%
458       \nag@warnNoLine{%

```

```

459     Package #2 is #1.\MessageBreak
460     Use #3 instead}%
461   }{}%
462 }%
463 }

```

```

\SuggestedPackage Usage: \SuggestedPackage[<reason>]{<package>}
464 \newcommand\SuggestedPackage[2][might be useful to you]{%
465   \AtEndDocument{%
466     \nag@ifcsname{ver@#2.sty}{%
467       % Attaboy!
468     }{}%
469     \nag@suggestNoLine{%
470       Not loaded: Package #2 #1}%
471   }%
472 }%
473 }%

```

```

\IncompatiblePackages Usage: \IncompatiblePackages[<reason>]{<package>}{<package>}{<hint>}
474 \newcommand\IncompatiblePackages[4][are incompatible]{%
475   \AtEndDocument{%
476     \nag@ifcsname{ver@#2.sty}{%
477       \nag@ifcsname{ver@#3.sty}{%
478         \nag@warnNoLine{%
479           Packages #2 and #3 #1.\MessageBreak
480           #4}%
481       }{}%
482     }{}
483   }%
484 }%

```

```

\ObsoleteClass Usage: \ObsoleteClass[<reason>]{<class>}{<alternative>}. Mark <class> as ob-
solete. <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.
485 \newcommand\ObsoleteClass[3][obsolete]{%
486   \AtEndDocument{%
487     % |\@clsextension| is onlypreamble, for some reason.
488     \nag@ifcsname{ver@#2.cls}{%
489       \nag@warnNoLine{%
490         Class #2 is #1.\MessageBreak
491         Use #3 instead}%
492     }{}%
493   }%
494 }

```

```

\BadFileLoadOrder
495 \def\nag@quark{\nag@quark}
496 \ifx\@listfiles\undefined

```

```

497 % emulate a silent listfiles
498 \def\@listfiles#1\@{%
499 \fi
500 \newcommand\BadFileLoadOrder[3][This might cause problems]{%
501 \AtEndDocument{%
502   \nag@ifLoadOrder{#2}{#3}{%
503     \nag@warnNoLine{%
504       '#3' loaded after '#2'.\MessageBreak
505       #1}%
506   }%
507 }%
508 }
509 \def\nag@ifLoadOrder#1#2{%
510 \def\nag@tmporder@a ##1#1##2\relax{%
511   \ifx\nag@quark##2\nag@quark
512     \noexpand@gobble
513   \else
514     \nag@tmporder@b ##2,#2\relax
515   \fi
516 }%
517 \def\nag@tmporder@b ##1#2##2\relax{%
518   \ifx\nag@quark##2\nag@quark
519     \noexpand@gobble
520   \else
521     \noexpand@firstofone
522   \fi
523 }%
524 \@xa\protected@edef\@xa\nag@tmporder\@xa{\@xa\nag@tmporder@a\@filelist,,#1\relax}%
525 \nag@tmporder
526 }

```

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.

```
527 \newif\ifnag@haslabel
```



```
528 \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 `\@captype`). 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.

```
529 \newcommand\nag@hackfloat[1]{%
530   \nag@prepend{#1}{%
531     \global\nag@haslabelfalse\global\nag@hascaptionfalse
532   }%
533   \nag@prepend{end#1}{%
534     \ifnag@hascaption\relax\else
535       \nag@warn%
536     {#1 with no \protect\caption}%
537     \fi
538     % labels outside floats shouldn't complain:
539     \global\nag@hascaptiontrue
540     % (we do this always because it needs to be global)
541   }%
542 }
```

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.

```
543 \AtBeginDocument{%
544   \g@addto@macro{\@preamblecmds}{%
545     \@for\labelprovider:=\nag@labels\do{%
546       \ifx\labelprovider\@empty\else
547         \nag@prepend{\labelprovider}%
548         {\nag@captioncheck\nag@haslabeltrue}%
549       \fi
550     }%
551     \@for\captionprovider:=\nag@captions\do{%
552       \ifx\captionprovider\@empty\else
553         \nag@prepend{\captionprovider}{\global\nag@hascaptiontrue}%
554       \fi
555     }%
556   }
```

```

556 }%
557 }
558 \newcommand\nag@captioncheck{%
559 \ifnag@hascaption\else
560 \nag@warn{\protect\label\space in float, but not after
561 \protect\caption}%
562 \fi
563 }

```

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.

```

564 \def\nag@floats{}
565 \def\nag@labels{}
566 \def\nag@captions{}

```

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

```

567 \newcommand\nag@floatsetup{%
568 \@for\flo:=\nag@floats\do{%
569 \ifx\flo\@empty\else
570 \@xa\nag@hackfloat\@xa{\flo}%
571 \fi
572 }%
573 }

```

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

```

574 \AtBeginDocument{%
575 \nag@floatsetup
576 }

```

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

```

577 \AtBeginDocument{%
578 \AtEndDocument{%
579 \ifnum\value{nag@sins}>0%
580 \PackageWarningNoLine{nag}{\arabic{nag@sins} complaints
581 in total}%
582 \else
583 \typeout{No complaints by nag.}%
584 \fi
585 }%
586 }

```

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:

```
587 \DeclareRobustCommand\nag@ifCurrentEnvironment[3]{%
588   \bgroup
589   \def\tmp@a{#1}%
590   \ifx\@currenenv\tmp@a
591     #2%
592   \else
593     #3%
594   \fi
595 \egroup
596 }
```

And now, the two variations there are:

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

```
597 \newcommand\NotAnEnvironment[1]{%
598   \AtBeginDocument{%
599     \nag@prepend{#1}{%
600       \nag@ifCurrentEnvironment{#1}{%
601         \nag@warn{%
602           There is no environment ‘#1’.\MessageBreak
603           Maybe you want a grouped \@backslashchar#1
604         }%
605       }{% OK case.
606       }%
607     }%
608   }%
609 }
```

`\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.

```
610 % we need to maintain a stack of environments that are used in the
611 % \foo...\endfoo way.
612 \newcommand\nag@envstack{\relax}
613
614 \DeclareRobustCommand\nag@beginenv[1]{%
615   % push a begin-entry onto the stack. Form is
616   % |{\foo{lineno}}| for environment foo.
617   \bgroup
618   \@xa\toks@\@xa{\nag@envstack}%
619   \xdef\nag@envstack{%
```

```

620 \@nx{%
621 \@xa\@nx\csname #1\endcsname
622 \@nx{\the\inputlineno\@nx}%
623 \@nx}%
624 \the\toks@
625 }%
626 \egroup
627 }
628 \DeclareRobustCommand\@nag@endenv[1]{%
629 % extract the first entry.
630 \@xa\@nag@end@nv\@nag@envstack\@nil #1\@nil
631 }
632
633 \def\@nag@end@nv#1#2\@nil #3\@nil{%
634 \def\tmp@a{#1}%
635 \def\tmp@b{\relax}%
636 \ifx\tmp@a\tmp@b
637 % This was the end-of-stack flag.
638 \@nag@warn{'\@backslashchar end#3'} without matching
639 '\@backslashchar #3'}
640 \else
641 % We may assume this is a proper entry. See if the begin-token on
642 % the stack matches what |\@nag@endenv| was passed.
643 \@xa\ifx\csname #3\@xa\endcsname\@firstoftwo #1%
644 %OK case, just pop the entry.
645 \gdef\@nag@envstack{#2}%
646 \else
647 % error case
648 \@nag@warn{%
649 You cannot close '\@xa\string\@firstoftwo #1' on line
650 \@secondoftwo #1 with '\@backslashchar end#3'%
651 }%
652 % leave it on the stack. Some case of misnesting will always cause
653 % horrible amounts of follow-up errors. Also, scare them!
654 \fi
655 \fi
656 }

```

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

```

657 \AtEndDocument{%
658 \@xa\@tfor\@xa\looseends\@xa:\@xa=\@nag@envstack\do{%
659 \@xa\ifx\looseends\relax\else
660 \@nag@warnNoLine{Unmatched
661 '\@xa\@xa\@xa\string\@xa\@firstoftwo\looseends'
662 command on line
663 \@xa\@xa\@xa\string\@xa\@secondoftwo\looseends%
664 }%
665 \fi
666 }%
667 }

```

Now, the user-side command is easy.

```
668 \newcommand\NotASwitch[1]{%
669   \AtBeginDocument{%
670     \nag@prepend{#1}{%
671       \nag@beginenv{#1}%
672     }%
673     \nag@prepend{end#1}{%
674       \nag@endenv{#1}%
675     }%
676   }%
677 }
```

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`.

```
678 \AtBeginDocument{%
679   \nag@ifcsname{ver@subfig.sty}{%
680     \PackageInfo{nag}{Attempting subfig hack\@gobble}%
681     \nag@maybehacksubfig
682   }{%
683   }%
684 }
685 \def\nag@maybehacksubfig{%
686   %
687   % of course, i need to touch the single longest definition in
688   % subfig.sty, to amend one single command...
689   %
690   % The definition is taken from subfig.sty 1.3 dated 2005/07/05 by
691   % S.D. Cochran, where it is called sf@@@subfloat, and appears here
692   % under the conditions of section 6 of the LPPL 1.3. The subfig
693   % package is available on a CTAN mirror near you.
```

```

694 %
695 \long\def\nag@@original@@sf@@@subfloat##1[##2][##3]##4{%
696   \ifundefined{FBsc@max}{%
697     }{%
698     \FB@readaux{\let\FBsuboheight\relax}%
699     }%
700     \@tempcnta=\@ne
701     \if@minipage
702       \@tempcnta=\z@
703     \else\ifdim \lastskip=\z@ \else
704       \@tempcnta=\tw@
705     \fi\fi
706     \ifmaincaptiontop
707       \sf@top=\sf@nearskip
708       \sf@bottom=\sf@farskip
709     \else
710       \sf@top=\sf@farskip
711       \sf@bottom=\sf@nearskip
712     \fi
713     \leavevmode
714     \setbox\@tempboxa \hbox{%
715       ##4}%
716     \@tempdima=\wd\@tempboxa
717     \ifundefined{FBsc@max}{%
718       }{%
719       \global\advance\Xhsize-\wd\@tempboxa
720       \dimen@=\ht\@tempboxa
721       \advance\dimen@\dp\@tempboxa
722       \ifdim\dimen@>\FBso@max
723         \global\FBso@max\dimen@
724       \fi
725     }%
726     \vtop\bgroup
727     \vbox\bgroup
728       \ifcase\@tempcnta
729         \@minipagefalse
730       \or
731         \vskip\sf@top
732       \or
733         \ifdim \lastskip=\z@ \else
734           \@tempskipb\sf@top\relax\@xadvskip
735         \fi
736       \fi
737       \sf@ifpositiontop{%
738         \ifx \@empty##3\relax \else
739           \sf@subcaption{##1}{##2}{##3}%
740           \vskip\sf@capskip
741           \vskip\sf@captopadj
742         \fi\egroup
743         \hrule width0pt height0pt depth0pt

```

```

744     \box\@tempboxa
745   }{%
746     \ifundefined{FBsc@max}{%
747       \box\@tempboxa
748     }{%
749       \ifx\FBsuboheight\relax
750         \box\@tempboxa
751       \else
752         \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
753       \fi}%
754     \egroup
755   \ifx \@empty##3\relax \else
756     \vskip\sf@capskip
757     \hrule width0pt height0pt depth0pt
758     \sf@subcaption{##1}{##2}{##3}%
759   \fi
760 }%
761 \vskip\sf@bottom
762 \egroup
763 \ifundefined{FBsc@max}{%
764 }{%
765   \addtocounter{FRobj}{-1}%
766   \ifnum\c@FRobj=0\else
767     \subfloatrowsep
768   \fi
769 }%
770 \ifmaincaptiontop\else
771   \global\advance\@nameuse{c@\@captype}\m@ne
772 \fi
773 \endgroup\ignorespaces}%
774 %
775 \expandafter\ifx\cscname sf@@@subfloat\endcscname\nag@@original@@sf@@@subfloat
776 % yup, that's it.
777 \PackageInfo{nag}{OK, equivalent to subfig 1.3, redefining
778   \@backslashchar sf@@@subfloat@gobble}%
779 \global\long\def\sf@@@subfloat##1[##2][##3]##4{%
780   \ifundefined{FBsc@max}{%
781     }{%
782       \FB@readaux{\let\FBsuboheight\relax}%
783     }%
784     \@tempcnta=\@ne
785     \if@minipage
786       \@tempcnta=\z@
787     \else\ifdim \lastskip=\z@ \else
788       \@tempcnta=\tw@
789     \fi\fi
790     \ifmaincaptiontop
791       \sf@top=\sf@nearskip
792       \sf@bottom=\sf@farskip
793     \else

```

```

794     \sf@top=\sf@farskip
795     \sf@bottom=\sf@nearskip
796 \fi
797 \leavevmode
798 \setbox\@tempboxa \hbox{%
799     %% ulmi: new 2007/02/25: #4 may contain label command
800     \csname nag@hascaptiontrue\endcsname
801     %% and that was it.
802     ##4}%
803 \@tempdima=\wd\@tempboxa
804 \@ifundefined{FBsc@max}{%
805 }{%
806     \global\advance\Xhsize-\wd\@tempboxa
807     \dimen@=\ht\@tempboxa
808     \advance\dimen@\dp\@tempboxa
809     \ifdim\dimen@>\FBso@max
810     \global\FBso@max\dimen@
811     \fi
812 }%
813 \vtop\bgroup
814     %% ulmi: new 2007/05/10: #2, #3 may contain label command
815     \csname nag@hascaptiontrue\endcsname
816     %% and that was it.
817     \vbox\bgroup
818         \ifcase\@tempcnta
819         \@minipagefalse
820         \or
821             \vskip\sf@top
822         \or
823             \ifdim \lastskip=\z@ \else
824                 \@tempskipb\sf@top\relax\@xaddvskip
825             \fi
826         \fi
827     \sf@ifpositiontop{%
828         \ifx \@empty##3\relax \else
829             \sf@subcaption{##1}{##2}{##3}%
830             \vskip\sf@capskip
831             \vskip\sf@captopadj
832         \fi\egroup
833         \hrule width0pt height0pt depth0pt
834         \box\@tempboxa
835     }{%
836         \@ifundefined{FBsc@max}{%
837             \box\@tempboxa
838         }{%
839             \ifx\FBsuboheight\relax
840                 \box\@tempboxa
841             \else
842                 \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
843             \fi}%

```



```

844     \egroup
845     \ifx \@empty##3\relax \else
846     \vskip\sf@capskip
847     \hrule width0pt height0pt depth0pt
848     \sf@subcaption{##1}{##2}{##3}%
849     \fi
850     }%
851     \vskip\sf@bottom
852     \egroup
853     \@ifundefined{FBsc@max}{%
854     }{%
855     \addtocounter{FRObj}{-1}%
856     \ifnum\c@FRObj=0\else
857     \subfloatrowsep
858     \fi
859     }%
860     \ifmaincaptiontop\else
861     \global\advance\@nameuse{c@\@captype}\m@ne
862     \fi
863     \endgroup\ignorespaces}%
864 \else
865     \PackageInfo{nag}{Not redefining sf@@@subfloat, it looks odd@gobble}
866 \fi
867 }

```

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`.

```
868 \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 `\pgfuserlibrary` 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.

```

869 \DeclareOption*{%
870   \InputIfFileExists{nag-\CurrentOption.cfg}{%
871     \PackageInfo{nag}{%
872       Loaded nag-\CurrentOption.cfg
873     }
874   }{%
875     \InputIfFileExists{\CurrentOption.nag}{%
876       \PackageWarningNoLine{nag}{%
877         Loaded old-style config file \CurrentOption.nag.\MessageBreak
878         Consider renaming the file to nag-\CurrentOption.cfg
879       }%
880     }{%
881       \PackageWarningNoLine{nag}{Required ruleset
882         \CurrentOption, and it wasn't there}
883     }%
884   }
885 }
886 \ProcessOptions*

```

Change History

0.1	General: First official version. 1	0.4	General: bugfix 15
0.2	General: Added abort.nag, suggested by Michael Zedler 1		config file names changed to free extension 23
	Rephrased umlaut.sty warning, suggested by Patrick Happel. . . 1		Handling command vs. environment; bugfixes 1
0.3	General: Fixed missing globals . . 15	0.5	General: Handle the case that somebody else relaxes the ver@-commands. Stack-based NotASwitch. 1
	New ifdefined that won't relax the commands 1		

0.51		0.60alpha5	
	<code>\nag@prepend</code> : bugfix	General: improves compatibility with subfig.	11 1
0.52		0.61	
	General: Command NagDeclare-Float added	General: is 0.61alpha5 with some typos in the docs fixed.	23 1
	made eTeX-ifcsname more robust	0.61alpha1	13
	twiddle subfig's bowels	General: fixes warnings in toc/lof/lot and unsightly uppercasing.	19 1
	<code>\nag@prepend</code> : info	roman counter (external file issue)	11 10
0.53		<code>\nag@prepend</code> : Extra indirection of warnings for robustness (uppercasing/LoF issues)	16 11
	General: bugfix: more Robustness. (Jörg Sommer)	<code>\nag@warn</code> : Made robust.	17 11
0.54		0.61alpha2	
	<code>\NotASwitch</code> : bugfix: can't get around the token register. (Jörg Sommer)	General: fixes the warnings, without generating too many duplicates.	17 1
0.55		<code>\nag@prepend</code> : Creep under existing robust cover	1 11
	General: Some spaces crept in in 0.5	0.61alpha3	1 1
0.60		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).	15 1
	General: @preamblecmds	0.61alpha4	
	fixes double-dollar in conjunction with hyperref; documents incompatibility with rotating.	General: exempts the complaints counter from include trickery. (Previously, nag would get confused if you includeonly only some chapters.)	1 1
	Captions/Labels now done only once, and not every time we enter a float	sin counter should not be saved by include	15 10
0.60alpha		0.61alpha5	
	General: changes the way label/caption is handled, this eliminates the current limit of some thousand floats you can have in your document. (I wonder why nobody noticed).	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.)	1 1
0.60alpha2		0.61alpha6	
	General: is more careful around commands that aren't there.	General: Compatibility w/ VCS packages, pgf	1 4
	<code>\nag@prepend</code> : don't even relax unknown commands (J.Sommer)	0.62	
0.60alpha4		General: fixes a bug in the float	
	General: handles eqnarray itself and has code in nag-experimental.cfg to handle double-dollar in a more robust way that onlyamsmath.		1
	tarballs now unpack into a subdirectory like proper citizens should.		1

placement code and adds more compatibility with the caption package.	1	catcode issue	1
0.621		0.62alpha1	
General: Bugfix concerning unknown command in math mode	1	General: Bigger warning if all float positions fail	8
0.622		0.62alpha2	
General: Bugfix: math in captions		General: Fix for marginpar etc which don't have fps	8

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	<code>\@latex@warning</code> . . .	<code>\advance</code>	227,
<code>\"</code> 241, 280, 292	240, 610, 612,
<code>\\$</code>	66, 75, 81, 86, 90, 94	<code>\@minipagefalse</code>	620, 710
<code>\@@input</code>	278	<code>\@nameuse</code>	662, 752
<code>\@addtoreset</code>	310	<code>\@preamblecmds</code>	435
<code>\@backslashchar</code> 321, 329,	<code>\@secondoftwo</code>	20, 25,
	330, 360, 494,	315, 383, 541, 554
	529, 530, 541, 669	<code>\@sptoken</code>	51
<code>\@capttype</code> 167, 168, 662, 752	<code>\@svnBeginRead</code>	81
<code>\@clsextension</code>	400, 410	<code>\@tempboxa</code> 605, 607, 610–
<code>\@currbox</code>	225, 227,	612, 635, 638,
	235, 238, 240, 242		641, 643, 689,
<code>\@currenvr</code>	481		694, 697–699,
<code>\@empty</code>	325,	<code>\@tfor</code>	248, 549
	437, 443, 460,	<code>\@typeset@protect</code>	24
	629, 646, 719, 736	<code>\@undefined</code>	167
<code>\@filef@und</code>	278	<code>\@xa</code>	218, 248, 283,
<code>\@firstoftwo</code> 19, 24, 43, 315,	284, 295, 296,
	390, 534, 540, 552		304, 333, 334,
<code>\@fps</code>	236, 273		336, 339–341,
<code>\@gobble</code>	60, 571, 669, 756		461, 509, 512,
<code>\@gobbletwo</code>	310		521, 534, 540,
<code>\@ifnextchar</code>	52		549, 550, 552, 554
<code>\@ifpackageloaded</code> 70, 80, 84, 88, 92	<code>\@xaddvskip</code>	625, 715
<code>\@input</code>	277	<code>\[</code>	58
<code>\@input@</code>	290	<code>\]</code>	58
<code>\@largefloatcheck</code> . .	237	<code>_</code>	74
<code>\@latex@info@no@line</code> 284, 296	A	
		<code>\active</code>	44, 48, 74
		B	
		<code>\bfseries</code>	104
		<code>\bgroup</code>	47,
		50, 73, 479, 508,
			617, 618, 704, 708
		<code>\binom</code>	122
		<code>\bottomfraction</code>	261
		<code>\box</code>	635, 638, 641, 643,
		725, 728, 731, 733
		C	
		<code>\c@FObj</code>	657, 747
		<code>\caption</code>	427, 452
		<code>\captionprovider</code> 442–444
		<code>\catcode</code>	44, 45, 48, 74,
		75, 81, 86, 90, 94
		<code>\changes</code>	164, 173, 174
		D	
		<code>\DeclareRobustCommand</code> 208,

212, 336, 348, 352, 478, 505, 519	\ignorespaces . 664, 754	\nag@expanding@voodoo 14, 32
\dimen@ 611–614, 698–701	\inputlineno 513	\nag@floats 455, 459, 759
	\itshape 106	\nag@floatsetup 458, 466
E	J	\nag@fltsz@tmp
\edef 294	\justenv 203, 204 248, 249,
\egroup 60, 62, 68, 486, 517, 633, 645, 653, 723, 735, 743	\justsw 199, 201	252, 255, 259, 263
	L	\nag@hackfloat 420, 461
F	\label 451	\nag@hascaptionfalse 422
\FB@readaux ... 589, 673	\labelprovider 436–438	\nag@hascaptiontrue 419, 430, 444
\FBafil 643, 733	\lastskip 594, 624, 678, 714	\nag@haslabelfalse .. 422
\FBbfil 643, 733	\leavevmode ... 604, 688	\nag@haslabeltrue .. 439
\FBso@max 613, 614, 700, 701	\long 586, 670	\nag@ifcsname 376, 379, 380,
\FBsuboheight 589, 640, 643, 673, 730, 733	\looseends 549, 550, 552, 554	396, 401, 411, 570
\fontcmd 190, 192, 194, 197	M	\nag@ifCurrentEnvironment 478, 491
\frac 120–122	\m@ne 662, 752	\nag@ifundefined 313, 318, 324
	\mathcal 118	\nag@labels 163, 436, 456
	\message 28	\nag@maybe@dispmath 33, 49
G	N	\nag@maybedispmath 16, 67
\g@addto@macro 81, 163, 175, 256, 260, 435, 759	\nag@@original@@sf@@@subfloat 586, 666	\nag@maybehacksubfig 572, 576
\gdef 49, 536	\nag@allfloatpositionsfailedfalse 232, 264	\nag@maybespace 325, 327, 335, 339
\global 67, 396, 422, 430, 444, 610, 614, 662, 670, 697, 701, 752	\nag@allfloatpositionsfailedtrue 246	\nag@nofile 282, 285, 294, 297
	\nag@allfloatsizechecks 221, 247, 256, 260, 271	\nag@onefloatsizecheck 222, 257, 261
H	\nag@beginenv . 505, 562	\nag@pr@p@nd .. 339, 345
\hbox 605, 689	\nag@captioncheck 439, 449	\nag@prepend .. 166, 313, 358, 368, 421, 424, 438, 444, 490, 561, 564
	\nag@captions 175, 442, 457	\nag@quark 46, 51
I	\nag@dofloatsizecheckfalse 250, 253	\nag@singledollar 13, 14, 22, 35, 40, 62
\if@minipage .. 592, 676	\nag@dofloatsizechecktrue 245	\nag@warn .. 56, 168, 208, 273, 348, 359, 369, 426, 451, 492, 529, 539
\IfFileExists 278	\nag@doubledollar 12, 57, 60	\nag@warnNoLine 212, 352, 402, 412, 551
\ifinner 19	\nag@end@nv ... 521, 524	\NagDeclareFloat 162, 759
\ifmaincaptiontop .. . 597, 661, 681, 751	\nag@endenv 519, 533, 565	
\ifmmode 53	\nag@envstack 503, 509, 510, 521, 536, 549	
\ifnag@allfloatpositionsfailedfalse 220, 272		
\ifnag@dofloatsizecheck 219, 270		
\ifnag@hascaption 419, 425, 450		
\ifnag@haslabel ... 418		

<code>\NeedsTeXFormat</code> ... 302	228, 320, 328,	<code>\sf@ifpositiontop</code> ..
<code>\newcommand</code> ... 221,	571, 668, 756, 762 628, 718
222, 313, 317,	<code>\protected@edef</code> ... 282	<code>\sf@nearskip</code>
345, 356, 366,	<code>\providecommand</code> ... 43	. 598, 602, 682, 686
376, 398, 408,	<code>\ProvidesFile</code>	<code>\sf@subcaption</code>
420, 449, 458,	.. 1, 188, 206, 216	. 630, 649, 720, 739
488, 503, 559, 759	<code>\ProvidesPackage</code> .. 303	<code>\sf@top</code>
<code>\newcounter</code> ... 306, 311		598,
<code>\newif</code> 219, 220, 418, 419		601, 622, 625,
<code>\noexpand</code> 305		682, 685, 712, 715
<code>\NotAnEnvironment</code> ..		<code>\sffamily</code> 112
.... 178, 181,		<code>\sloppy</code> 100
192, 197, 201, 488		<code>\slshape</code> 114
<code>\NotASwitch</code> 204, 501 , 559		<code>\string</code> 58, 540, 552, 554
		<code>\subfloatrowsep</code> 658, 748
	R	
	<code>\RCS</code> 73	
	<code>\renewcommand</code>	
	. 73, 237, 307, 380	
	<code>\rmfamily</code> 108	
	<code>\roman</code> 307	
	S	
	<code>\S</code> 7, 10, 11, 99,	
	100, 102, 119,	
	123, 125, 127,	
	131, 133, 136,	
	141, 143, 146,	
	153, 157, 158,	
	161, 177, 183, 187	
	<code>\scshape</code> 110	
	<code>\sectioning</code> ... 180, 181	
	<code>\setbox</code> 605, 689	
	<code>\setcounter</code> 308	
	<code>\sf@@@subfloat</code> 670	
	<code>\sf@bottom</code> 599, 602,	
	652, 683, 686, 742	
	<code>\sf@capskip</code>	
	. 631, 647, 721, 737	
	<code>\sf@captopadj</code> . 632, 722	
	<code>\sf@farskip</code>	
	. 599, 601, 683, 685	
	T	
	<code>\texorpdfstring</code> . 17, 43	
	<code>\textbf</code> 104	
	<code>\textheight</code> ... 225,	
	226, 238, 239, 242	
	<code>\textit</code> 106	
	<code>\textrm</code> 108	
	<code>\textsc</code> 110	
	<code>\textsf</code> 112	
	<code>\textsl</code> 114	
	<code>\texttt</code> 116	
	<code>\thenag@c</code> 307,	
	334, 336, 340, 341	
	<code>\toks@</code> 509, 515	
	<code>\topfraction</code> 257	
	<code>\ttfamily</code> 116	
	X	
	<code>\xdef</code> 510	
	<code>\Xhsize</code> 610, 697	
	O	
<code>\ObsoleteClass</code> 126, 408		
<code>\ObsoleteCS</code> 103, 105,		
107, 109, 111,		
113, 115, 117,		
121, 122, 124, 356		
<code>\ObsoleteEnv</code> .. 186, 366		
<code>\ObsoletePackage</code> ..		
. 8, 9, 128–130,		
132, 134, 135,		
137, 138, 140,		
142, 145, 147,		
149, 151, 154,		
156, 159, 160, 398		
	P	
<code>\PackageError</code> . 210, 214		
<code>\PackageInfo</code> ... 78,		
82, 85, 89, 93,		