

The nag package^{*}

Ulrich Michael Schwarz[†]

June 10, 2007

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 All the same, there are commands, classes and packages which are outdated and superseded.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	nag-l2tabu.cfg	2
1.4	nag-orthodox.cfg	6
1.5	nag-abort.cfg	7
1.6	nag-experimental.cfg	7
2	Author-side considerations and implementation.	7
2.1	Low-level tools.	7
2.2	Obsoletifying commands.	9
2.3	Obsoletifying packages and classes.	9
2.4	Common float errors and no-nos.	11
3	Switch vs. Environment	13
4	Compatibility issues	15
4.1	The caption package	15
4.2	The subfig package	16
4.3	The float package	20
4.4	The topcapt package and the subfig package	20
4.5	The rotating package	20
5	Loading extensions	20

^{*}This document corresponds to nag 0.60, dated 2007/06/10. Other versions can be found at <http://absatzen.de/>

[†]ulmi@absatzen.de

1 User-side considerations.

1.1 Installation.

Process `nag.ins` with \TeX to obtain some files: `nag.sty` and `nag-l2tabu.cfg` et al. must go to a place where \TeX 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 \TeX file you want to use them on may work under many circumstances.) You can, as usual, run \TeX 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 `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 \TeX 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 < *l2tabunag>
2 \ProvidesFile{nag-l2tabu.cfg}
3 [2007/06/10 v2.0 l2tabu rules for nag.sty (ulmi)]
4 %%
5 %% The sins.
6 %%
7 %% Section numbers refer to l2tabuen 1.7 revised/enlarged dated 2004OCT24
8 %% \S 1.1
9 \ObsoletePackage{a4wide}{the \lq a4paper\rq\space class option}
10 \ObsoletePackage{a4}{the \lq a4paper\rq\space class option}
11 %% \S 1.2--1.5 cannot reasonably be checked programmatically
12 %% \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.

```

13 \def\nag@doubledollar{$$}%$
14 \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.

```

15 \def\nag@expanding@voodoo#1#2#3{\relax\relax\nag@singledollar}
16
17 \def\nag@maybedispmath{%
18   \texorpdfstring{%
19     %% in TeX context, do tricky stuff.
20     \ifinner\expandafter\@firstoftwo
21     \else\expandafter\@secondoftwo\fi
22     {% in inner mode, $$ is an empty formula, so no testing wanted.
23       \nag@singledollar}%
24     {%
25       \ifx\protect\@typeset@protect\expandafter\@firstoftwo
26       \else\expandafter\@secondoftwo\fi
27       {% normal case: looks like typesetting
28         %% protect against strictly expanding context
29         %% like TeX' \message: the first expanding voodoo will expand,
30         %% removing the rest, inserting \relax\relax$ instead. This is
31         %% not totally transparent, but \let\relax\relax is as close
32         %% to a no-op as we can get.
33         \let\nag@expanding@voodoo\nag@expanding@voodoo
34         \protect\nag@maybe@dispmath}%
35       {% some other case, hide ourselves
36         \nag@singledollar}%
37     }%
38   }{%
39     %% in pdf context, just be a math shift. This creates the "math
40     %% shift not allowed" warnings we all love.
41     \nag@singledollar

```

```

42 }%
43 }

```

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`.

```

44 \AtBeginDocument{\providecommand\texorpdfstring{\@firstoftwo}}
45 \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.

```

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$\nag@maybedispmath%$
67 \egroup
68
69
70 %% \S 1.7 cannot reasonably be checked programmatically
71 %% \S 1.8 \sloppy is called by parbox, among others, and would
72 %% give many spurious warnings.
73 %% \S 2.1.1
74 \ObsoleteCS[an old LaTeX 2.09 command]{bf}
75   {\protect\bfseries\space or \protect\textbf}
76 \ObsoleteCS[an old LaTeX 2.09 command]{it}
77   {\protect\itshape\space or \protect\textit}
78 \ObsoleteCS[an old LaTeX 2.09 command]{rm}
79   {\protect\rmfamily\space or \protect\textrm}
80 \ObsoleteCS[an old LaTeX 2.09 command]{sc}
81   {\protect\scshape\space or \protect\textsc}

```

```

82 \ObsoleteCS[an old LaTeX 2.09 command]{sf}
83     {\protect\sffamily\space or \protect\textsf}
84 \ObsoleteCS[an old LaTeX 2.09 command]{sl}
85     {\protect\slshape\space or \protect\textsl}
86 \ObsoleteCS[an old LaTeX 2.09 command]{tt}
87     {\protect\ttfamily\space or \protect\texttt}
88 \ObsoleteCS[an old LaTeX 2.09 command]{cal}
89     {\protect\mathcal}% Hmm, this is not in l2tabu?
90 %% \S 2.1.2
91 %% Gone with 1.8 because this never worked for the kernel \frac anyway.
92 %% \ObsoleteCS[TeX]{over}{\protect\frac}
93 %% \ObsoleteCS[TeX]{choose}{\protect\frac\space or amsmath's \protect\binom}
94 %% \S 2.1.3
95 \ObsoleteCS[TeX]{centerline}{\protect\centering\space or center environment}
96 %% \S 2.2.1
97 \ObsoleteClass{scrlettr}{the scrlettr2 package}
98 %% \S 2.2.2
99 \ObsoletePackage{epsf}{the graphicx package}
100 \ObsoletePackage{psfig}{the graphicx package}
101 \ObsoletePackage[deprecated]{epsfig}{the graphicx package directly}
102 %% \S 2.2.3
103 \ObsoletePackage{doublespace}{the setspace package}
104 %% \S 2.2.4
105 \ObsoletePackage{fancyheadings}{the fancyhdr or scrpage2 packages}
106 \ObsoletePackage{scrpage}{the scrpage2 package}
107 %% \S 2.2.5
108 \ObsoletePackage{isolatin}{the inputenc package with option latin1}
109 \ObsoletePackage{umlaut}{the inputenc package with suitable option
110 (latin1, utf8 ...)}
111 \ObsoletePackage{isolatin1}{the inputenc package with option latin1}
112 %% \S 2.2.6
113 \ObsoletePackage{tlenc}{the fontenc package with option T1}
114 %% \S 2.2.7 we don't check for bst yet.
115 %% (This is in l2tabu 1.8)
116 \ObsoletePackage{caption2}{the caption package v3.0 or later}
117 %% \S 2.3.1-3
118 \ObsoletePackage{times}
119     {the mathptmx, helvet (option scaled=.9), courier packages}
120 \ObsoletePackage{pslatex}
121     {the mathptmx, helvet (option scaled=.9), courier packages}
122 \ObsoletePackage{mathptm}
123     {the mathptmx package}
124 %% \S 2.3.4-5
125 \ObsoletePackage{palatino}
126     {the mathpazo, helvet (option scaled=.95), courier packages}
127 \ObsoletePackage{mathpple}{the mathpazo package}
128 %% \S 2.3.6 can't be checked
129 %% \S 2.3.7
130 \ObsoletePackage{euler}{the eulervm package}
131 \ObsoletePackage{utopia}{the fourier package}

```

```

132%% \S 3.1
133 \NagDeclareFloat{figure}\NagDeclareFloat{table}%
134 \g@addto@macro\nag@labels{,label}%
135% \changes{0.60}{2007/03/31}{alternate center-in-float check, doesn't
136% take up as many macro names}
137 \nag@prepend{endcenter}{}%
138 \ifx\@capytype\@undefined\else
139 \nag@warn{\lq center\rq\space environment in \@capytype.\MessageBreak
140 Maybe you want \protect\centering\space instead}%
141 \fi
142}%
143%% The latter two are used by KOMA-Script, the last by hypcap.
144% \changes{0.53}{2007/03/21}{hypcap support. (H.G.Krauth\'auser)}
145% \changes{0.53}{2007/03/21}{topcapt support.}
146 \g@addto@macro\nag@captions{,caption,captionabove,captionbelow,hc@caption,topcaption}%
147
148%% \S 3.2
149 \NotAnEnvironment{appendix}%
150%% In the same vein:
151 \@for\sectioning:=frontmatter,mainmatter,backmatter\do{%
152 \expandafter\NotAnEnvironment\expandafter{\sectioning}%
153 }
154%% \S 3.3
155%% It's more trouble than it's worth to have another warning for
156%% align*, since it passes through align.
157 \ObsoleteEnv{eqnarray}{amsmath's align}
158%% \S 3.4 -- nothing to be done --
159 </l2tabunag>

```

1.4 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.)

```

160 <*orthodoxnag>
161 \ProvidesFile{nag-orthodox.cfg}
162 [2006/04/19 v1.8 strict rules for nag.sty (ulmi)]
163 \@for\fontcmd:=tiny,small,footnotesize,normalsize,large,Large,%
164 LARGE,huge,Huge\do{%
165 \expandafter\NotAnEnvironment\expandafter{\fontcmd}%
166}%
167 \@for\fontcmd:=sffamily,rmfamily,ttfamily,%
168 bfseries,mdseries,scshape,%
169 itshape,upshape\do{%
170 \expandafter\NotAnEnvironment\expandafter{\fontcmd}%
171}%

```

```

172 \@for\justsw:=centering,raggedleft,raggedright,%
173           RaggedLeft,RaggedRight\do{%
174   \expandafter\NotAnEnvironment\expandafter{\justsw}%
175 }
176 \@for\justenv:=center,flushleft,flushright\do{%
177   \expandafter\NotASwitch\expandafter{\justenv}%
178 }
179 \</orthodoxnag>

```

1.5 nag-abort.cfg

Requesting this nag file will turn all complaints into errors.

```

180 \<abortnag>
181 \ProvidesFile{nag-abort.cfg}
182           [2005/05/08 v0.1 treat complaints as errors (ulmi)]
183 \renewcommand\nag@warn[1]{%
184   \addtocounter{nag@sins}{1}%
185   \PackageError{nag}{#1}{#1}%
186 }
187 \renewcommand\nag@warnNoLine[1]{%
188   \addtocounter{nag@sins}{1}%
189   \PackageError{nag}{#1}{#1}%
190 }
191 \</abortnag>

<*experimentalnag>

```

1.6 nag-experimental.cfg

Functionality that needs more testing.

```

192 \ProvidesFile{nag-experimental.cfg}
193           [2007/06/10 v0.1b experimental additions to nag (ulmi)]
</experimentalnag>

<*nag>

```

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.

```

194 \NeedsTeXFormat{LaTeX2e}

```

```

195 \ProvidesPackage{nag}[2007/06/10 0.60 warning about old commands (ulmi)]
196 \let\@xa\expandafter
197 \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.

```

198 \newcounter{nag@c}
199 \newcounter{nag@sins}

```

`\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.

```

200 \newcommand\nag@ifundefined[1]{%
201   \begingroup
202   \@ifundefined{#1}{\endgroup\@firstoftwo}{\endgroup\@secondoftwo}%
203 }

```

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.

```

204 \newcommand\nag@prepend[2]{%
205   \nag@ifundefined{#1}{%
206     % if it doesn't exist, don't do anything.
207     \PackageInfo{nag}{%
208       Command \@backslashchar#1\space not defined, skipping amendment%
209     }%
210   }{%
211     \@xa\let
212     \csname nag@@#1@\thenag@c\@xa\endcsname
213     \csname #1\endcsname
214     \@xa\nag@pr@p@nd\csname #1\@xa\endcsname
215     \csname nag@@#1@\thenag@c\endcsname{#2}%

```

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?

```

216   \stepcounter{nag@c}%
217 }%
218 }
219 \newcommand\nag@pr@p@nd[3]{%
220   \def#1{#3#2}%
221 }

```

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

```
222 \newcommand\nag@warn{%
223   \addtocounter{nag@sins}{1}%
224   \PackageWarning{nag}%
225 }
226 \newcommand\nag@warnNoLine{%
227   \addtocounter{nag@sins}{1}%
228   \PackageWarningNoLine{nag}%
229 }
```

2.2 Obsoletefying commands.

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

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

```
230 \newcommand\ObsoleteCS[3][obsolete]{%
231   \AtBeginDocument{%
232     \nag@prepend{#2}{%
233       \nag@warn{%
234 Command \@backslashchar#2 is #1.
235 \MessageBreak
236 Use #3 instead}%
237     }%
238   }%
239 }
```

`\ObsoleteEnv`

```
240 \newcommand\ObsoleteEnv[3][obsolete]{%
241   \AtBeginDocument{%
242     \nag@prepend{#2}{%
243       \nag@warn{%
244 Environment #2 is #1.
245 \MessageBreak
246 Use #3 instead}%
247     }%
248   }%
249 }
```

2.3 Obsoletefying 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`).

```
250 \newcommand\nag@ifcsname[3]{%
251   \begingroup\@ifundefined{#1}{#3}{#2}\endgroup
252 }
```

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.

```
253 \nag@ifcsname{ifcsname}{%
254   \renewcommand*\nag@ifcsname[3]{%
255     \begingroup
256     % assume it won't be there.
257     \let\tmp@a\@secondoftwo
258     \ifcsname #1\endcsname
259     % It still might be relax from some other test. Thanks to J\'org
260     % Sommer for finding this bug.
261     \expandafter\ifx\csname #1\endcsname\relax
262     \else
263     % it's there after all
264     \let\tmp@a\@firstoftwo
265     \fi
266   \fi
267   \tmp@a{#2}{#3}%
268   \endgroup
269 }
```

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

```
270 \global\let\nag@ifcsname\nag@ifcsname
271 {}{}
```

`\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.

```
272 \newcommand\ObsoletePackage[3][obsolete]{%
273   \AtEndDocument{%
274     |\@clsextension| is onlypreamble, for some reason.
275     \nag@ifcsname{ver@#2.sty}{%
276       \nag@warnNoLine{%
277         Package #2 is #1.\MessageBreak
278         Use #3 instead}%
279     }{}%
280   }%
281 }
```

`\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.

```

282 \newcommand\ObsoleteClass[3][obsolete]{%
283   \AtEndDocument{%
284     |\@clsextension| is onlypreamble, for some reason.
285     \nag@ifcsname{ver@#2.cls}{%
286       \nag@warnNoLine{%
287         Class #2 is #1.\MessageBreak
288         Use #3 instead}%
289       }{}%
290     }%
291 }

```

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.

```

292 \newif\ifnag@haslabel
293 \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.

```

294 \newcommand\nag@hackfloat[1]{%
295   \nag@prepend{#1}{%
296     \global\nag@haslabelfalse\global\nag@hascaptionfalse
297   }%
298   \nag@prepend{end#1}{%
299     \ifnag@hascaption\relax\else
300       \nag@warn%
301       {#1 with no \protect\caption}%
302     \fi
303     % labels outside floats shouldn't complain:
304     \global\nag@hascaptiontrue

```

```

305 % (we do this always because it needs to be global)
306 }%
307 }

```

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.

```

308 \AtBeginDocument{%
309   \g@addto@macro{\@preamblecmds}{%
310     \@for\labelprovider:=\nag@labels\do{%
311       \ifx\labelprovider\empty\else
312         \nag@prepend{\labelprovider}%
313         {\nag@captioncheck\nag@haslabeltrue}%
314       \fi
315     }%
316     \@for\captionprovider:=\nag@captions\do{%
317       \ifx\captionprovider\empty\else
318         \nag@prepend{\captionprovider}{\global\nag@hascaptiontrue}%
319       \fi
320     }%
321   }%
322 }
323 \newcommand\nag@captioncheck{%
324   \ifnag@hascaption\else
325     \nag@warn{\protect\label\space in float, but not after
326       \protect\caption}%
327   \fi
328 }

```

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.

```

329 \def\nag@floats{}
330 \def\nag@labels{}
331 \def\nag@captions{}

```

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

```

332 \newcommand\nag@floatsetup{%
333   \@for\flo:=\nag@floats\do{%
334     \ifx\flo\@empty\else
335       \xa\nag@hackfloat\x@\flo}%
336     \fi
337   }%
338 }

```

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

```

339 \AtBeginDocument{%
340   \nag@floatsetup
341 }

```

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

```

342 \AtBeginDocument{%
343   \AtEndDocument{%
344     \ifnum\value{nag@sins}>0%
345       \PackageWarningNoLine{nag}{\arabic{nag@sins} complaints
346         in total}%
347     \else
348       \typeout{No complaints by nag.}%
349     \fi
350   }%
351 }

```

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:

```

352 \DeclareRobustCommand\nag@ifCurrentEnvironment[3]{%
353   \bgroup
354   \def\tmp@a{#1}%
355   \ifx\@currenvir\tmp@a
356     #2%
357   \else
358     #3%
359   \fi
360   \egroup
361 }

```

And now, the two variations there are:

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

```

362 \newcommand\NotAnEnvironment[1]{%
363   \AtBeginDocument{%
364     \nag@prepend{#1}{%
365       \nag@ifCurrentEnvironment{#1}{%
366         \nag@warn{%
367           There is no environment ‘‘#1’’. \MessageBreak
368           Maybe you want a grouped \@backslashchar#1
369         }%
370       }{% OK case.
371       }%
372     }%
373   }%
374 }
```

`\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.

```

375 % we need to maintain a stack of environments that are used in the
376 % \foo...\endfoo way.
377 \newcommand\nag@envstack{\relax}
378
379 \DeclareRobustCommand\nag@beginenv[1]{%
380   % push a begin-entry onto the stack. Form is
381   % |{\foo{lineno}}| for environment foo.
382   \bgroup
383   \@xa\toks@\@xa{\nag@envstack}%
384   \xdef\nag@envstack{%
385     \@nx{%
386       \@xa\@nx\csize #1\endcsize
387       \@nx{\the\inputlineno\@nx}%
388     \@nx}%
389     \the\toks@
390   }%
391   \egroup
392 }
393 \DeclareRobustCommand\nag@endenv[1]{%
394   % extract the first entry.
395   \@xa\nag@endenv\nag@envstack\@nil #1\@nil
396 }
397
398 \def\nag@endenv#1#2\@nil #3\@nil{%
399   \def\tmp@a{#1}%
400   \def\tmp@b{\relax}%
401   \ifx\tmp@a\tmp@b
402     % This was the end-of-stack flag.
403     \nag@warn{‘‘\@backslashchar end#3’’ without matching
404       ‘‘\@backslashchar #3’’}
```

```

405 \else
406   % We may assume this is a proper entry. See if the begin-token on
407   % the stack matches what |\nag@endenv| was passed.
408   \@xa@ifx\csname #3\@xa\endcsname\@firstoftwo #1%
409   %OK case, just pop the entry.
410   \gdef\nag@envstack{#2}%
411 \else
412 % error case
413 \nag@warn{%
414   You cannot close ‘‘\@xa\string\@firstoftwo #1’’ on line
415   \@secondoftwo #1 with ‘‘\@backslashchar end#3’’%
416 }%
417 % leave it on the stack. Some case of misnesting will always cause
418 % horrible amounts of follow-up errors. Also, scare them!
419 \fi
420 \fi
421 }

```

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

```

422 \AtEndDocument{%
423   \@xa\@tfor\@xa\looseends\@xa:\@xa=\nag@envstack\do{%
424     \@xa@ifx\looseends\relax\else
425       \nag@warnNoLine{Unmatched
426         ‘‘\@xa\@xa\@xa\string\@xa\@firstoftwo\looseends’’
427         command on line
428         \@xa\@xa\@xa\string\@xa\@secondoftwo\looseends%
429       }%
430     \fi
431   }%
432 }

```

Now, the user-side command is easy.

```

433 \newcommand\NotASwitch[1]{%
434   \AtBeginDocument{%
435     \nag@prepend{#1}%
436     \nag@beginenv{#1}%
437   }%
438   \nag@prepend{end#1}%
439   \nag@endenv{#1}%
440 }%
441 }%
442 }

```

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`.

```

443 \AtBeginDocument{%
444   \nag@ifcsname{ver@subfig.sty}{%
445     \PackageInfo{nag}{Attempting subfig hack \@gobble}%
446     \nag@maybehacksubfig
447   }{%
448   }%
449 }
450 \def\nag@maybehacksubfig{%
451   %
452   % of course, i need to touch the single longest definition in
453   % subfig.sty, to amend one single command...
454   %
455   % The definition is taken from subfig.sty 1.3 dated 2005/07/05 by
456   % S.D. Cochran, where it is called sf@@@subfloat, and appears here
457   % under the conditions of section 6 of the LPPL 1.3. The subfig
458   % package is available on a CTAN mirror near you.
459   %
460   \long\def\nag@@original@sff@@@subfloat##1[##2][##3]##4{%
461     \@ifundefined{FBsc@max}{%
462       {%
463         \FB@readaux{\let\FBsuboheight\relax}%
464       }%
465       \@tempcnta=\@ne
466       \if@minipage
467         \@tempcnta=\z@
468       \else\ifdim \lastskip=\z@ \else
469         \@tempcnta=\tw@
470       \fi\fi
471       \ifmaincaptiontop
472         \sf@top=\sf@nearskip
473         \sf@bottom=\sf@farskip
474       \else
475         \sf@top=\sf@farskip
476         \sf@bottom=\sf@nearskip
477       \fi
478       \leavevmode
479       \setbox\@tempboxa \hbox{%

```

```

480     ##4}%
481 \@tempdima=\wd\@tempboxa
482 \@ifundefined{FBsc@max}{%
483 }{%
484   \global\advance\Xhsize-\wd\@tempboxa
485   \dimen@=\ht\@tempboxa
486   \advance\dimen@\dp\@tempboxa
487   \ifdim\dimen@>\FBso@max
488     \global\FBso@max\dimen@
489   \fi
490 }%
491 \vtop\bgroup
492   \vbox\bgroup
493     \ifcase\@tempcnta
494     \@minipagefalse
495     \or
496     \vskip\sf@top
497     \or
498     \ifdim \lastskip=\z@ \else
499       \@tempskipb\sf@top\relax\@xaddvskip
500     \fi
501   \fi
502   \sf@ifpositiontop{%
503     \ifx \@empty##3\relax \else
504       \sf@subcaption{##1}{##2}{##3}%
505       \vskip\sf@capskip
506       \vskip\sf@captopadj
507     \fi\egroup
508     \hrule width0pt height0pt depth0pt
509     \box\@tempboxa
510   }{%
511     \@ifundefined{FBsc@max}{%
512       \box\@tempboxa
513     }{%
514       \ifx\FBsuboheight\relax
515         \box\@tempboxa
516       \else
517         \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
518       \fi}%
519   \egroup
520   \ifx \@empty##3\relax \else
521     \vskip\sf@capskip
522     \hrule width0pt height0pt depth0pt
523     \sf@subcaption{##1}{##2}{##3}%
524   \fi
525 }%
526 \vskip\sf@bottom
527 \egroup
528 \@ifundefined{FBsc@max}{%
529 }{%

```

```

530     \addtocounter{FRobj}{-1}%
531     \ifnum\c@FRobj=0\else
532         \subfloatrowsep
533     \fi
534 }%
535 \ifmaincaptiontop\else
536     \global\advance\@nameuse{c@\@capttype}\m@ne
537 \fi
538 \endgroup\ignorespaces}%
539 %
540 \expandafter\ifx\csname sf@@@subfloat\endcsname\nag@@original@@sf@@@subfloat
541 % yup, that's it.
542 \PackageInfo{nag}{OK, equivalent to subfig 1.3, redefining
543     \@backslashchar sf@@@subfloat\@gobble}%
544 \global\long\def\sف@@@subfloat##1[##2][##3]##4{%
545     \@ifundefined{FBsc@max}{%
546     }{%
547         \FB@readaux{\let\FBsuboheight\relax}%
548     }%
549     \@tempcnta=\@ne
550     \if@minipage
551         \@tempcnta=\z@
552     \else\ifdim \lastskip=\z@ \else
553         \@tempcnta=\tw@
554     \fi\fi
555     \ifmaincaptiontop
556         \sf@top=\sf@narskip
557         \sf@bottom=\sf@farskip
558     \else
559         \sf@top=\sf@farskip
560         \sf@bottom=\sf@narskip
561     \fi
562     \leavevmode
563     \setbox\@tempboxa \hbox{%
564         %% ulmi: new 2007/02/25: #4 may contain label command
565         \csname nag@hascaptiontrue\endcsname
566         %% and that was it.
567         ##4}%
568     \@tempdima=\wd\@tempboxa
569     \@ifundefined{FBsc@max}{%
570     }{%
571         \global\advance\Xhsize-\wd\@tempboxa
572         \dimen@=\ht\@tempboxa
573         \advance\dimen@\dp\@tempboxa
574         \ifdim\dimen@>\FBso@max
575             \global\FBso@max\dimen@
576         \fi
577     }%
578     \vtop\bgroup
579     %% ulmi: new 2007/05/10: #2, #3 may contain label command

```

```

580 \csname nag@hascaptiontrue\endcsname
581 %% and that was it.
582 \vbox\bgroup
583 \ifcase\@tempcnta
584 \@minipagefalse
585 \or
586 \vskip\sf@top
587 \or
588 \ifdim \lastskip=\z@ \else
589 \@tempskipb\sf@top\relax\@xaddvskip
590 \fi
591 \fi
592 \sf@ifpositiontop{%
593 \ifx \@empty##3\relax \else
594 \sf@subcaption{##1}{##2}{##3}%
595 \vskip\sf@capskip
596 \vskip\sf@captopadj
597 \fi\egroup
598 \hrule width0pt height0pt depth0pt
599 \box\@tempboxa
600 }{%
601 \ifundefined{FBsc@max}{%
602 \box\@tempboxa
603 }{%
604 \ifx\FBsuboheight\relax
605 \box\@tempboxa
606 \else
607 \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
608 \fi}%
609 \egroup
610 \ifx \@empty##3\relax \else
611 \vskip\sf@capskip
612 \hrule width0pt height0pt depth0pt
613 \sf@subcaption{##1}{##2}{##3}%
614 \fi
615 }%
616 \vskip\sf@bottom
617 \egroup
618 \@ifundefined{FBsc@max}{%
619 }{%
620 \addtocounter{FRobj}{-1}%
621 \ifnum\c@FRobj=0\else
622 \subfloatrowsep
623 \fi
624 }%
625 \ifmaincaptiontop\else
626 \global\advance\@nameuse{c@\@captype}\m@ne
627 \fi
628 \endgroup\ignorespaces}%
629 \else

```

```

630      \PackageInfo{nag}{Not redefining sf@@@subfloat, it looks odd\@gobble}
631    \fi
632 }

```

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`.

```

633 \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. 2 again.

5 Loading extensions

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

```

634 \DeclareOption*{%
635   \InputIfFileExists{nag-\CurrentOption.cfg}{%
636     \PackageInfo{nag}{%
637       Loaded nag-\CurrentOption.cfg
638     }
639   }{%
640     \InputIfFileExists{\CurrentOption.nag}{%
641       \PackageWarningNoLine{nag}{%
642         Loaded old-style config file \CurrentOption.nag.\MessageBreak
643         Consider renaming the file to nag-\CurrentOption.cfg
644       }%
645     }{%
646       \PackageWarningNoLine{nag}{Required ruleset
647         \CurrentOption, and it wasn't there}
648     }%
649   }
650 }
651 \ProcessOptions*

```

</nag>

Change History

0.1	General: First official version. 1	0.52	General: Command NagDeclareFloat added 20
0.2	General: Added abort.nag, suggested by Michael Zedler 1		made eTeX-ifcsname more robust . 10
	Rephrased umlaut.sty warning, suggested by Patrick Happel. 1		twiddle subfig's bowels 16
0.3	General: Fixed missing globals 12	0.53	\nag@prepend: info 8
	New ifdefined that won't relax the commands 1		General: bugfix: more Robustness. (Jörg Sommer) 13
0.4	General: bugfix 12	0.54	\NotASwitch: bugfix: can't get around the token register. (Jörg Sommer) 14
	config file names changed to free extension 20	0.55	General: Some spaces crept in in 0.5 . . 1
	Handling command vs. environment; bugfixes 1	0.60	General: @preamblecmds 12
0.5	General: Handle the case that somebody else relaxes the ver@-commands. Stack-based NotASwitch. 1		Captions/Labels now done only once, and not every time we enter a float 12
0.51	\nag@prepend: bugfix 8	0.60alpha2	\nag@prepend: don't even relax unknown commands (J.Sommer) . . . 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		
\@firstoftwo 144, 259	\@firstoftwo 20, 25, 44, 202, 264, 408, 414, 426	26, 202, 257, 415, 428
\\$ 65	\@gobble . 59, 445, 543, 630	\@sptoken 50
\@backslashchar 208, 234, 368, 403, 404, 415, 543	\@ifnextchar 51	\@tempboxa 479,
\@capytype 138, 139, 536, 626	\@minipagefalse 494, 584	481, 484–486, 509,
\@clsextension . 274, 284	\@nameuse 536, 626	512, 515, 517,
\@currentvir 355	\@ne 465, 549	563, 568, 571–
\@empty 311, 317, 334, 503, 520, 593, 610	\@nil 395, 398	573, 599, 602, 605, 607
	\@nx 197, 385–388	\@tempcnta
	\@preamblecmds 309	. . . 465, 467, 469,
	\@secondoftwo . . . 21,	493, 549, 551, 553, 583
		\@tempdima 481, 568

\@tempskipb 499, 589	\FBafil 517, 607	M
\@tfor 423	\FBbfil 517, 607	\m@ne 536, 626
\@typeset@protect ... 25	\FBso@max 487, 488, 574, 575	\mathcal 89
\@undefined 138	\FBsuboheight .. 463,	\message 29
\@xa 196,	514, 517, 547, 604, 607	
211, 212, 214, 335,	\fontcmd 163, 165, 167, 170	N
383, 386, 395, 408,	\foo 376, 381	\nag@@original@sff@@@subfloat
414, 423, 424, 426, 428	\frac 91–93 460, 540
\xaddvskip 499, 589		\nag@beginenv .. 379, 436
\[..... 57	G	\nag@captioncheck ..
\] 57	\g@addto@macro 313, 323
	... 134, 146, 309, 633	\nag@captions 146, 316, 331
A	\gdef 48, 410	\nag@doubledollar ..
\active 45, 47	\global .. 66, 270, 296, 13, 56, 59
\advance 484,	304, 318, 484, 488,	\nag@end@nv 395, 398
486, 536, 571, 573, 626	536, 544, 571, 575, 626	\nag@endenv . 393, 407, 439
\AtBeginDocument 44,		\nag@envstack .. 377,
45, 231, 241, 308,	H	383, 384, 395, 410, 423
339, 342, 363, 434, 443	\hbox 479, 563	\nag@expanding@voodoo
\AtEndDocument	\hrule .. 508, 522, 598, 612 15, 33
... 273, 283, 343, 422	\ht 485, 572	\nag@floats . 329, 333, 633
		\nag@floatsetup 332, 340
B		\nag@hackfloat . 294, 335
\bfseries 75	I	\nag@hascaptionfalse 296
\bgroup ... 46, 49, 353,	\if@minipage ... 466, 550	\nag@hascaptiontrue
382, 491, 492, 578, 582	\ifcase 493, 583 293, 304, 318
\binom 93	\ifdim 468,	\nag@haslabelfalse . 296
\box 509, 512, 515,	487, 498, 552, 574, 588	\nag@haslabeltrue .. 313
517, 599, 602, 605, 607	\ifinner 20	\nag@ifcsname 250, 253,
	\ifmaincaptiontop ..	254, 270, 275, 285, 444
C	... 471, 535, 555, 625	\nag@ifCurrentEnvironment
\c@FRobj 531, 621	\ifmmode 52 352, 365
\caption 301, 326	\ifnag@hascaption ..	\nag@ifundefined 200, 205
\captionprovider 316–318 293, 299, 324	\nag@labels . 134, 310, 330
\catcode 45, 47	\ifnag@haslabel 292	\nag@maybe@dispmath
\changes 135, 144, 145	\ignorespaces .. 538, 628 34, 48
	\inputlineno 387	\nag@maybe@dispmath .
D	\itshape 77 17, 66
\DeclareRobustCommand	J	\nag@maybe@hacksubfig
..... 352, 379, 393	\justenv 176, 177 446, 450
\dimen@ . 485–488, 572–575	\justsw 172, 174	\nag@pr@p@nd ... 214, 219
\dp 486, 573		\nag@prepend 137, 200,
	L	232, 242, 295, 298,
E	\label 325	312, 318, 364, 435, 438
\egroup 59, 61,	\labelprovider . 310–312	\nag@quark 50
67, 360, 391, 507,	\lastskip 468, 498, 552, 588	\nag@singledollar ..
519, 527, 597, 609, 617	\leavevmode 478, 562	. 14, 15, 23, 36, 41, 61
\endfoo 376	\long 460, 544	\nag@warn 55, 139,
	\looseends	183, 222, 233, 243,
F	... 423, 424, 426, 428	300, 325, 366, 403, 413
\FB@readaux 463, 547		

<code>\nag@warnNoLine</code> .. 187, 226, 276, 286, 425	<code>\ProvidesFile</code> 2, 161, 181, 192	<code>\string</code> .. 57, 414, 426, 428
<code>\NagDeclareFloat</code> 133, 633	<code>\ProvidesPackage</code> ... 195	<code>\subfloatrowsep</code> 532, 622
<code>\NeedsTeXFormat</code> 194		T
<code>\newcommand</code> 200, 204, 219, 222, 226, 230, 240, 250, 272, 282, 294, 323, 332, 362, 377, 433, 633	R <code>\renewcommand</code> 183, 187, 254 <code>\rmfamily</code> 79	<code>\texorpdfstring</code> .. 18, 44 <code>\textbf</code> 75 <code>\textit</code> 77 <code>\textrm</code> 79 <code>\textsc</code> 81 <code>\textsf</code> 83 <code>\textsl</code> 85 <code>\texttt</code> 87 <code>\thenag@c</code> 212, 215 <code>\tmp@a</code> 257, 264, 267, 354, 355, 399, 401 <code>\tmp@b</code> 400, 401 <code>\toks@</code> 383, 389 <code>\ttfamily</code> 87 <code>\tw@</code> 469, 553
<code>\newcounter</code> 198, 199	S <code>\S</code> 8, 11, 12, 70, 71, 73, 90, 94, 96, 98, 102, 104, 107, 112, 114, 117, 124, 128, 129, 132, 148, 154, 158	
<code>\newif</code> 292, 293	<code>\scshape</code> 81	
<code>\noexpand</code> 197	<code>\sectioning</code> 151, 152	
<code>\NotAnEnvironment</code> 149, 152, 165, 170, 174, <u>362</u>	<code>\setbox</code> 479, 563 <code>\sf@@@subfloat</code> 544 <code>\sf@bottom</code> 473, 476, 526, 557, 560, 616 <code>\sf@capskip</code> 505, 521, 595, 611 <code>\sf@captopadj</code> .. 506, 596 <code>\sf@farskip</code> 473, 475, 557, 559 <code>\sf@ifpositiontop</code> 502, 592 <code>\sf@nearskip</code> 472, 476, 556, 560 <code>\sf@subcaption</code> 504, 523, 594, 613 <code>\sf@top</code> . 472, 475, 496, 499, 556, 559, 586, 589 <code>\sffamily</code> 83 <code>\sloppy</code> 71 <code>\slshape</code> 85	
<code>\NotASwitch</code> . 177, <u>375</u> , 433		V <code>\vbox</code> ... 492, 517, 582, 607 <code>\vskip</code> 496, 505, 506, 521, 526, 586, 595, 596, 611, 616 <code>\vtop</code> 491, 578
O		W
<code>\ObsoleteClass</code> .. 97, <u>282</u>		<code>\wd</code> 481, 484, 568, 571
<code>\ObsoleteCS</code> 74, 76, 78, 80, 82, 84, 86, 88, 92, 93, 95, <u>230</u>		X <code>\xdef</code> 384 <code>\Xhsize</code> 484, 571
<code>\ObsoleteEnv</code> ... 157, <u>240</u>		Z
<code>\ObsoletePackage</code> 9, 10, 99– 101, 103, 105, 106, 108, 109, 111, 113, 116, 118, 120, 122, 125, 127, 130, 131, <u>272</u>		<code>\z@</code> 467, 468, 498, 551, 552, 588
<code>\or</code> 495, 497, 585, 587		
P		
<code>\PackageError</code> .. 185, 189		
<code>\PackageInfo</code> ... 207, 445, 542, 630, 636		
<code>\providecommand</code> 44		