

The nag package^{*}

Ulrich Michael Schwarz[†]

February 25, 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 All the same, there are commands, classes and packages which are outdated and superseded.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	<code>nag-l2tabu.cfg</code>	2
1.4	<code>nag-orthodox.cfg</code>	4
1.5	<code>nag-abort.cfg</code>	5
2	Author-side considerations and implementation.	5
2.1	Low-level tools.	6
2.2	Obsoletifying commands.	7
2.3	Obsoletifying packages and classes.	7
2.4	Common float errors and no-nos.	8
3	Switch vs. Environment	10
4	Compatibility issues	13
4.1	The caption package	13
4.2	The subfig package	13
4.3	The float package	17
5	Loading extensions	17

^{*}This document corresponds to `nag` 0.52, dated 2007/02/25.

[†]`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- Usage of the outdated packages `epsfig`, `psfig`, `epsf`, `double-space`, `fancyheadings`, `scrpage`, `umlaut`, `isolatin`, `isolatin1`, `tlenc`, `caption2`, `psfonts`, `mathptm`,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`. Outdated maths environments are not treated because those can already be handled by Harald Harders' `onlyamsmath`.

Be warned, hence, 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      [2006/04/19 v1.8 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 is handled by H.Harders' onlyamsmath package
13 %% \S 1.7 cannot reasonably be checked programmatically
14 %% \S 1.8 \sloppy is called by parbox, among others, and would
15 %% give many spurious warnings.
16 %% \S 2.1.1
17 \ObsoleteCS[an old LaTeX 2.09 command]{bf}
18      {\protect\bfseries\space or \protect\textbf}
19 \ObsoleteCS[an old LaTeX 2.09 command]{it}
20      {\protect\itshape\space or \protect\textit}
21 \ObsoleteCS[an old LaTeX 2.09 command]{rm}
22      {\protect\rmfamily\space or \protect\textrm}
23 \ObsoleteCS[an old LaTeX 2.09 command]{sc}
24      {\protect\scshape\space or \protect\textsc}
25 \ObsoleteCS[an old LaTeX 2.09 command]{sf}
26      {\protect\sffamily\space or \protect\textsf}
27 \ObsoleteCS[an old LaTeX 2.09 command]{sl}
28      {\protect\slshape\space or \protect\textsl}
29 \ObsoleteCS[an old LaTeX 2.09 command]{tt}
30      {\protect\ttfamily\space or \protect\texttt}
31 \ObsoleteCS[an old LaTeX 2.09 command]{cal}
32      {\protect\mathcal}% Hmm, this is not in l2tabu?
33 %% \S 2.1.2
34 %% Gone with 1.8 because this never worked for the kernel \frac anyway.
35 %% \ObsoleteCS[TeX]{over}{\protect\frac}
36 %% \ObsoleteCS[TeX]{choose}{\protect\frac\space or amsmath's \protect\binom}
37 %% \S 2.1.3
38 \ObsoleteCS[TeX]{centerline}{\protect\centering\space or center environment}
39 %% \S 2.2.1
40 \ObsoleteClass{scrlettr}{the scrlettr2 package}
41 %% \S 2.2.2
42 \ObsoletePackage{epsf}{the graphicx package}
43 \ObsoletePackage{psfig}{the graphicx package}
44 \ObsoletePackage[deprecated]{epsfig}{the graphicx package directly}
45 %% \S 2.2.3
46 \ObsoletePackage{doublespace}{the setspace package}
47 %% \S 2.2.4
48 \ObsoletePackage{fancyheadings}{the fancyhdr or scrpage2 packages}
49 \ObsoletePackage{scrpage}{the scrpage2 package}
50 %% \S 2.2.5

```

```

51\ObsoletePackage{isolatin}{the inputenc package with option latin1}
52\ObsoletePackage{umlaut}{the inputenc package with suitable option
53 (latin1, utf8 ...)}
54\ObsoletePackage{isolatin1}{the inputenc package with option latin1}
55%% \S 2.2.6
56\ObsoletePackage{tlenc}{the fontenc package with option T1}
57%% \S 2.2.7 we don't check for bst yet.
58%% (This is in l2tabu 1.8)
59\ObsoletePackage{caption2}{the caption package v3.0 or later}
60%% \S 2.3.1-3
61\ObsoletePackage{times}
62 {the mathptmx, helvet (option scaled=.9), courier packages}
63\ObsoletePackage{pslatex}
64 {the mathptmx, helvet (option scaled=.9), courier packages}
65\ObsoletePackage{mathptm}
66 {the mathptmx package}
67%% \S 2.3.4-5
68\ObsoletePackage{palatino}
69 {the mathpazo, helvet (option scaled=.95), courier packages}
70\ObsoletePackage{mathpple}{the mathpazo package}
71%% \S 2.3.6 can't be checked
72%% \S 2.3.7
73\ObsoletePackage{euler}{the eulervm package}
74\ObsoletePackage{utopia}{the fourier package}
75%% \S 3.1
76\NagDeclareFloat{figure}\NagDeclareFloat{table}%
77\g@addto@macro\nag@labels{,label}%
78%% The latter two are used by KOMA-Script.
79\g@addto@macro\nag@captions{,caption,captionabove,captionbelow}%
80
81%% \S 3.2
82\NotAnEnvironment{appendix}%
83%% In the same vein:
84\@for\sectioning:=frontmatter,mainmatter,backmatter\do{%
85 \expandafter\NotAnEnvironment\expandafter{\sectioning}%
86 }
87%% \S 3.3 handled by onlyamsmath.
88%% \S 3.4 -- nothing to be done --
89\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.)

```
90 (*orthodoxnag)
```

```

91 \ProvidesFile{nag-orthodox.cfg}
92 [2006/04/19 v1.8 strict rules for nag.sty (ulmi)]
93 \@for\fontcmd:=tiny,small,footnotesize,normalsize,large,Large,%
94     LARGE,huge,Huge\do{%
95   \expandafter\NotAnEnvironment\expandafter{\fontcmd}%
96 }%
97 \@for\fontcmd:=sffamily,rmfamily,ttfamily,%
98     bfseries,mdseries,scshape,%
99     itshape,upshape\do{%
100   \expandafter\NotAnEnvironment\expandafter{\fontcmd}%
101 }%
102 \@for\justsw:=centering,raggedleft,raggedright,%
103     RaggedLeft,RaggedRight\do{%
104   \expandafter\NotAnEnvironment\expandafter{\justsw}%
105 }
106 \@for\justenv:=center,flushleft,flushright\do{%
107   \expandafter\NotASwitch\expandafter{\justenv}%
108 }
109 \</orthodoxnag>

```

1.5 nag-abort.cfg

Requesting this nag file will turn all complaints into errors.

```

110 \<*abornag>
111 \ProvidesFile{nag-abort.cfg}
112 [2005/05/08 v0.1 treat complaints as errors (ulmi)]
113 \renewcommand\nag@warn[1]{%
114   \addtocounter{nag@sins}{1}%
115   \PackageError{nag}{#1}{#1}%
116 }
117 \renewcommand\nag@warnNoLine[1]{%
118   \addtocounter{nag@sins}{1}%
119   \PackageError{nag}{#1}{#1}%
120 }
121 \</abornag>

<*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.

```
122 \NeedsTeXFormat{LaTeX2e}
123 \ProvidesPackage{nag}[2007/02/25 0.52 warning about old commands (ulmi)]
124 \let\@xa\expandafter
125 \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.

```
126 \newcounter{nag@c}
127 \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))

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.

```
128 \newcommand\nag@prepend[2]{%
129   \@ifundefined{#1}{%
130     % if it doesn't exist, don't do anything.
131     \PackageInfo{nag}{%
132       Command \@backslashchar#1\space not defined, skipping amendment
133     }
134   }{%
135     \@xa\let
136     \csname nag@@#1@thenag@c\@xa\endcsname
137     \csname #1\endcsname
138     \@xa\nag@pr@p@nd\csname #1\@xa\endcsname
139     \csname nag@@#1@thenag@c\endcsname{#2}%
140     \stepcounter{nag@c}%
141   }
142 }
143 \newcommand\nag@pr@p@nd[3]{%
144   \def#1{#3#2}%
145 }
```

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

```
146 \newcommand\nag@warn{%
147   \addtocounter{nag@sins}{1}%
148   \PackageWarning{nag}%
149 }
150 \newcommand\nag@warnNoLine{%
151   \addtocounter{nag@sins}{1}%
```

```

152 \PackageWarningNoLine{nag}%
153 }

```

2.2 Obsoletifying commands.

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

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

```

154 \newcommand\ObsoleteCS[3][obsolete]{%
155   \AtBeginDocument{%
156     \nag@prepend{#2}{%
157       \nag@warn{%
158 Command \backslashchar#2 is #1.
159 \MessageBreak
160 Use #3 instead}%
161     }%
162   }%
163 }

```

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`).

```

164 \newcommand\nag@ifcsname[3]{%
165   \begingroup\@ifundefined{#1}{#3}{#2}\endgroup
166 }

```

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

```

167 \nag@ifcsname{ifcsname}{%
168   \renewcommand*\nag@ifcsname[3]{%
169     \begingroup
170       % assume it won't be there.
171       \let\tmp@a\@secondoftwo
172       \ifcsname #1\endcsname
173       % It still might be relax from some other test. Thanks to J"org
174       % Sommer for finding this bug.
175       \expandafter\ifx\csname #1\endcsname\relax
176       \else
177         % it's there after all
178         \let\tmp@a\@firstoftwo

```

```

179     \fi
180     \fi
181     \tmp@a{#2}{#3}%
182     \endgroup
183 }%

```

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

```

184 \global\let\nag@ifcsname\nag@ifcsname
185 {}

```

`\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.

```

186 \newcommand\ObsoletePackage[3][obsolete]{%
187   \AtEndDocument{%
188     |\@clsextension| is onlypreamble, for some reason.
189     \nag@ifcsname{ver@#2.sty}{%
190       \nag@warnNoLine{%
191         Package #2 is #1.\MessageBreak
192         Use #3 instead}%
193       }{}%
194     }%
195 }

```

`\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.

```

196 \newcommand\ObsoleteClass[3][obsolete]{%
197   \AtEndDocument{%
198     |\@clsextension| is onlypreamble, for some reason.
199     \nag@ifcsname{ver@#2.cls}{%
200       \nag@warnNoLine{%
201         Class #2 is #1.\MessageBreak
202         Use #3 instead}%
203       }{}%
204     }%
205 }

```

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.)

```
206 \newif\ifnag@haslabel
207 \newif\ifnag@hascaption
```

Now, to the work proper: first, add the endcenter check, then, prepare to set up the caption/label checks locally to the floats, and finally, add the code that generates the warning.

```
208 \newcommand\nag@hackfloat[1]{%
209   \nag@prepend{#1}{%
210     \nag@prepend{endcenter}{%
211       \nag@warn%
212       {\lq center\rq\space environment in #1.\MessageBreak
213        Maybe you want \protect\centering\space instead}
214     }%
215   }
216   \nag@prepend{#1}{%
```

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.

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.

```
217   \@for\labelprovider:=\nag@labels\do{%
218     \ifx\labelprovider@empty\else
219       \nag@prepend{\labelprovider}%
220       {\nag@captioncheck\nag@haslabeltrue}
221     \fi
222   }%
223   \@for\captionprovider:=\nag@captions\do{%
224     \ifx\captionprovider@empty\else
225       \nag@prepend{\captionprovider}{\global\nag@hascaptiontrue}%
226     \fi
227   }%
228   \global\nag@haslabelfalse\global\nag@hascaptionfalse
229 }%
230 \nag@prepend{end#1}{%
231   \ifnag@hascaption\relax\else
232     \nag@warn%
233 {#1 with no \protect\caption}%
234   \fi
235 }%
236 }
237 \newcommand\nag@captioncheck{%
```

```

238 \ifnag@hascaption\else
239 \nag@warn{\protect\label\space in float, but not after
240 \protect\caption}%
241 \fi
242 }

```

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.

```

243 \def\nag@floats{}
244 \def\nag@labels{}
245 \def\nag@captions{}

```

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

```

246 \newcommand\nag@floatsetup{%
247 \@for\flo:=\nag@floats\do{%
248 \ifx\flo\@empty\else
249 \@xa\nag@hackfloat\@xa{\flo}%
250 \fi
251 }%
252 }

```

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

```

253 \AtBeginDocument{%
254 \nag@floatsetup
255 }

```

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

```

256 \AtBeginDocument{%
257 \AtEndDocument{%
258 \ifnum\value{nag@sins}>0%
259 \PackageWarningNoLine{nag}{\arabic{nag@sins} complaints
260 in total}%
261 \else
262 \typeout{No complaints by nag.}%
263 \fi
264 }%
265 }

```

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 valied 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:

```
266 \newcommand\nag@ifCurrentEnvironment[3]{%
267   \bgroup
268   \def\tmp@a{#1}%
269   \ifx\@currenenv\tmp@a
270     #2%
271   \else
272     #3%
273   \fi
274   \egroup
275 }
```

And now, the two variations there are:

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

```
276 \newcommand\NotAnEnvironment[1]{%
277   \AtBeginDocument{%
278     \nag@prepend{#1}{%
279       \nag@ifCurrentEnvironment{#1}{%
280         \nag@warn{%
281           There is no environment ‘‘#1’’. \MessageBreak
282           Maybe you want a grouped \@backslashchar#1
283         }%
284       }{% OK case.
285       }%
286     }%
287   }%
288 }
```

`\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.

```
289 % we need to maintain a stack of environments that are used in the
290 % \foo...\endfoo way.
291 \newcommand\nag@envstack{\relax}
292
293 \newcommand\nag@beginenv[1]{%
294   % push a begin-entry onto the stack. Form is
295   % |\foo{lineno}| for environment foo.
296   \xdef\nag@envstack{%
297     \@nx{%
298       \@xa\@nx\csname #1\endcsname
299       \@nx{\the\inputlineno\@nx}%

```

```

300 \@nx}%
301 \@xa\@nx\nag@envstack
302 }%
303 }
304 \newcommand\nag@endenv[1]{%
305 % extract the first entry.
306 \@xa\nag@endenv\nag@envstack\@nil #1\@nil
307 }
308
309 \def\nag@endenv#1#2\@nil #3\@nil{%
310 \def\tmp@a{#1}%
311 \def\tmp@b{\relax}%
312 \ifx\tmp@a\tmp@b
313 % This was the end-of-stack flag.
314 \nag@warn{'\@backslashchar end#3'} without matching
315 '\@backslashchar #3'}
316 \else
317 % We may assume this is a proper entry. See if the begin-token on
318 % the stack matches what |\nag@endenv| was passed.
319 \@xa\ifx\csname #3\@xa\endcsname\@firstoftwo #1%
320 %OK case, just pop the entry.
321 \gdef\nag@envstack{#2}%
322 \else
323 % error case
324 \nag@warn{%
325 You cannot close '\@xa\string\@firstoftwo #1' on line
326 \@secondoftwo #1 with '\@backslashchar end#3'%
327 }%
328 % leave it on the stack. Some case of misnesting will always cause
329 % horrible amounts of follow-up errors. Also, scare them!
330 \fi
331 \fi
332 }

```

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

```

333 \AtEndDocument{%
334 \@xa\@tfor\@xa\looseends\@xa:\@xa=\nag@envstack\do{%
335 \@xa\ifx\looseends\relax\else
336 \nag@warnNoLine{Unmatched
337 '\@xa\@xa\@xa\string\@xa\@firstoftwo\looseends'
338 command on line
339 \@xa\@xa\@xa\string\@xa\@secondoftwo\looseends%
340 }%
341 \fi
342 }%
343 }

```

Now, the user-side command is easy.

```

344 \newcommand\NotASwitch[1]{%
345 \AtBeginDocument{%

```

```

346 \nag@prepend{#1}{%
347 \nag@beginenv{#1}%
348 }%
349 \nag@prepend{end#1}{%
350 \nag@endenv{#1}%
351 }%
352 }%
353 }

```

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.

```

354 \AtBeginDocument{%
355 \nag@ifcsname{ver@subfig.sty}{%
356 \PackageInfo{nag}{Attempting subfig hack\@gobble}%
357 \nag@maybehacksubfig
358 }{%
359 }%
360 }
361 \def\nag@maybehacksubfig{%
362 %
363 % of course, i need to touch the single longest definition in
364 % subfig.sty, to amend one single command...
365 %
366 % The definition is taken from subfig.sty 1.3 dated 2005/07/05 by
367 % S.D. Cochran, where it is called sf@@@subfloat, and appears here
368 % under the conditions of section 6 of the LPPL 1.3. The subfig
369 % package is available on a CTAN mirror near you.
370 %
371 \long\def\nag@@original@@sf@@@subfloat##1[##2][##3]##4{%
372 \@ifundefined{FBsc@max}{%
373 }{%

```

```

374 \FB@readaux{\let\FBsuboheight\relax}%
375 }%
376 \@tempcnta=\@ne
377 \if@minipage
378 \@tempcnta=\z@
379 \else\ifdim \lastskip=\z@ \else
380 \@tempcnta=\tw@
381 \fi\fi
382 \ifmaincaptiontop
383 \sf@top=\sf@nearskip
384 \sf@bottom=\sf@farskip
385 \else
386 \sf@top=\sf@farskip
387 \sf@bottom=\sf@nearskip
388 \fi
389 \leavevmode
390 \setbox\@tempboxa \hbox{%
391 ##4}%
392 \@tempdima=\wd\@tempboxa
393 \ifundefined{FBsc@max}{%
394 }{%
395 \global\advance\Xhsize-\wd\@tempboxa
396 \dimen@=\ht\@tempboxa
397 \advance\dimen@\dp\@tempboxa
398 \ifdim\dimen@>\FBso@max
399 \global\FBso@max\dimen@
400 \fi
401 }%
402 \vtop\bgroup
403 \vbox\bgroup
404 \ifcase\@tempcnta
405 \@minipagefalse
406 \or
407 \vskip\sf@top
408 \or
409 \ifdim \lastskip=\z@ \else
410 \@tempskipb\sf@top\relax\@xaddvskip
411 \fi
412 \fi
413 \sf@ifpositiontop{%
414 \ifx \@empty##3\relax \else
415 \sf@subcaption{##1}{##2}{##3}%
416 \vskip\sf@capskip
417 \vskip\sf@captopadj
418 \fi\egroup
419 \hrule width0pt height0pt depth0pt
420 \box\@tempboxa
421 }{%
422 \ifundefined{FBsc@max}{%
423 \box\@tempboxa

```

```

424     }{%
425         \ifx\FBsuboheight\relax
426             \box\@tempboxa
427         \else
428             \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
429         \fi}%
430     \egroup
431     \ifx \@empty##3\relax \else
432         \vskip\sf@capskip
433         \hrule width0pt height0pt depth0pt
434         \sf@subcaption{##1}{##2}{##3}%
435     \fi
436 }%
437 \vskip\sf@bottom
438 \egroup
439 \@ifundefined{FBsc@max}{%
440 }{%
441     \addtocounter{FRobj}{-1}%
442     \ifnum\c@FRobj=0\else
443         \subfloatrowsep
444     \fi
445 }%
446 \ifmaincaptiontop\else
447     \global\advance\@nameuse{c@\@captype}\m@ne
448 \fi
449 \endgroup\ignorespaces}%
450 %
451 \expandafter\ifx\csname sf@@@subfloat\endcsname\nag@@original@@sf@@@subfloat
452 % yup, that's it.
453 \PackageInfo{nag}{OK, equivalent to subfig 1.3, redefining
454 \backslashchar sf@@@subfloat\@gobble}%
455 \global\long\def\sf@@@subfloat##1[##2][##3]##4{%
456 \@ifundefined{FBsc@max}{%
457 }{%
458     \FB@readaux{\let\FBsuboheight\relax}%
459 }%
460 \@tempcnta=\@ne
461 \if@minipage
462     \@tempcnta=\z@
463 \else\ifdim \lastskip=\z@ \else
464     \@tempcnta=\tw@
465 \fi\fi
466 \ifmaincaptiontop
467     \sf@top=\sf@narskip
468     \sf@bottom=\sf@farskip
469 \else
470     \sf@top=\sf@farskip
471     \sf@bottom=\sf@narskip
472 \fi
473 \leavevmode

```

```

474 \setbox\@tempboxa \hbox{%
475   %% ulmi: new 2007/02/25: #4 may contain label command
476   \csname nag@hascaptiontrue\endcsname
477   %% and that was it.
478   ##4}%
479 \@tempdima=\wd\@tempboxa
480 \@ifundefined{FBsc@max}{%
481 }{%
482   \global\advance\Xhsize-\wd\@tempboxa
483   \dimen@=\ht\@tempboxa
484   \advance\dimen@\dp\@tempboxa
485   \ifdim\dimen@>\FBso@max
486   \global\FBso@max\dimen@
487   \fi
488 }%
489 \vtop\bgroup
490   \vbox\bgroup
491     \ifcase\@tempcnta
492     \@minipagefalse
493     \or
494     \vskip\sf@top
495     \or
496     \ifdim \lastskip=\z@ \else
497       \@tempskipb\sf@top\relax\@xaddvskip
498     \fi
499     \fi
500     \sf@ifpositiontop{%
501       \ifx \@empty##3\relax \else
502         \sf@subcaption{##1}{##2}{##3}%
503         \vskip\sf@capskip
504         \vskip\sf@captopadj
505       \fi\egroup
506       \hrule width0pt height0pt depth0pt
507       \box\@tempboxa
508     }{%
509       \@ifundefined{FBsc@max}{%
510         \box\@tempboxa
511       }{%
512         \ifx\FBsuboheight\relax
513           \box\@tempboxa
514         \else
515           \vbox to \FBsuboheight{\FBafil\box\@tempboxa\FBbfil}%
516         \fi}%
517     \egroup
518     \ifx \@empty##3\relax \else
519       \vskip\sf@capskip
520       \hrule width0pt height0pt depth0pt
521       \sf@subcaption{##1}{##2}{##3}%
522     \fi
523   }%

```



```

524     \vskip\sf@bottom
525   \egroup
526   \@ifundefined{FBsc@max}{%
527   }{%
528     \addtocounter{FRobj}{-1}%
529     \ifnum\c@FRobj=0\else
530       \subfloatrowsep
531     \fi
532   }%
533   \ifmaincaptiontop\else
534     \global\advance\@nameuse{c@\@captype}\m@ne
535   \fi
536   \endgroup\ignorespaces}%
537 \else
538   \PackageInfo{nag}{Not redefining sf@@subfloat, it looks odd\@gobble}
539 \fi
540 }

```

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`.

```

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

```

5 Loading extensions

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

```

542 \DeclareOption*{%
543   \InputIfFileExists{nag-\CurrentOption.cfg}{%
544     \PackageInfo{nag}{%
545       Loaded nag-\CurrentOption.cfg
546     }
547   }{%
548     \InputIfFileExists{\CurrentOption.nag}{%
549       \PackageWarningNoLine{nag}{%
550         Loaded old-style config file \CurrentOption.nag.\MessageBreak
551         Consider renaming the file to nag-\CurrentOption.cfg
552       }%
553     }{%
554       \PackageWarningNoLine{nag}{Required ruleset
555         \CurrentOption, and it wasn't there}
556     }%
557   }
558 }
559 \ProcessOptions*

```

</nag>

Change History

0.1			Handling command vs. environ- ment; bugfixes	1	
	General: First official version.	1			
0.2			0.5	General: Handle the case that some- body else relaxes the ver@- commands. Stack-based No- tASwitch.	1
	General: Added abort.nag, suggested by Michael Zedler	1			
	Rephrased umlaut.sty warning, suggested by Patrick Happel.	1			
0.3			0.51	\nag@prepend: bugfix	6
	General: Fixed missing globals	9			
	New ifdefined that won't relax the commands	1	0.52	General: Command NagDeclareFloat added	17
0.4				made eTeX-ifcsname more robust . .	7
	General: bugfix	9		twiddle subfig's bowels	13
	config file names changed to free extension	17		\nag@prepend: info	6

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	392, 395–397, 420,	253, 256, 277, 345, 354
\ "	173	423, 426, 428,
\@backslashchar	474, 479, 482–	\AtEndDocument
. 132, 158,	484, 507, 510, 513, 515	. . . 187, 197, 257, 333
282, 314, 315, 326, 454	\@tempcnta	B
\@capttype 447, 534	. . . 376, 378, 380,	\bfseries 18
\@clsextension . 188, 198	404, 460, 462, 464, 491	\bgroup 267,
\@currenvir 269	\@tempdima 392, 479	402, 403, 489, 490
\@empty 218, 224,	\@tempskipb 410, 497	\binom 36
248, 414, 431, 501, 518	\@tfor 334	\box 420, 423, 426,
\@firstoftwo	\@xa 124,	428, 507, 510, 513, 515
. . . 178, 319, 325, 337	135, 136, 138, 249,	C
\@gobble 356, 454, 538	298, 301, 306, 319,	\c@FRobj 442, 529
\@minipagefalse 405, 492	325, 334, 335, 337, 339	\caption 233, 240
\@nameuse 447, 534	\@xaddvskip 410, 497	\captionprovider 223–225
\@ne 376, 460	A	
\@nil 306, 309	\advance 395,	D
\@nx 125, 297–301	397, 447, 482, 484, 534	\dimen@ . 396–399, 483–486
\@secondoftwo 171, 326, 339	\AtBeginDocument 155,	\dp 397, 484
\@tempboxa 390,		

E		
\egroup 274, 418,	
	430, 438, 505, 517, 525	
\endfoo 290	
F		
\FB@readaux 374, 458	
\FBafil 428, 515	
\FBbfil 428, 515	
\FBso@max	398, 399, 485, 486	
\FBsuboheight	.. 374,	
	425, 428, 458, 512, 515	
\fontcmd	... 93, 95, 97, 100	
\foo 290, 295	
\frac 34–36	
G		
\g@addto@macro	77, 79, 541	
\gdef 321	
\global 184,	
	225, 228, 395, 399,	
	447, 455, 482, 486, 534	
H		
\hbox 390, 474	
\hrule	.. 419, 433, 506, 520	
\ht 396, 483	
I		
\if@minipage	... 377, 461	
\ifcase 404, 491	
\ifdim 379,	
	398, 409, 463, 485, 496	
\ifmaincaptiontop	..	
	... 382, 446, 466, 533	
\ifnag@hascaption	..	
 207, 231, 238	
\ifnag@haslabel 206	
\ignorespaces	.. 449, 536	
\inputlineno 299	
\itshape 20	
J		
\justenv 106, 107	
\justsw 102, 104	
L		
\label 239	
\labelprovider	. 217–219	
\lastskip	379, 409, 463, 496	
\leavevmode 389, 473	
\long 371, 455	
\looseends	
	... 334, 335, 337, 339	
M		
\m@ne 447, 534	
\mathcal 32	
N		
\nag@@original@sf@@subfloat	19, 21, 23, 25, 27,	
 371, 451	
\nag@beginenv	.. 293, 347	
\nag@captioncheck	..	
 220, 237	
\nag@captions	79, 223, 245	
\nag@end@nv 306, 309	
\nag@endenv	. 304, 318, 350	
\nag@envstack	.. 291,	
	296, 301, 306, 321, 334	
\nag@floats	. 243, 247, 541	
\nag@floatsetup	246, 254	
\nag@hackfloat	. 208, 249	
\nag@hascaptionfalse	228	
\nag@hascaptiontrue	225	
\nag@haslabelfalse	. 228	
\nag@haslabeltrue	.. 220	
\nag@ifcsname	164, 167,	
	168, 184, 189, 199, 355	
\nag@ifCurrentEnvironment 266, 279	
\nag@labels	.. 77, 217, 244	
\nag@maybehacksubfig 357, 361	
\nag@pr@p@nd	... 138, 143	
\nag@prepend	<u>128</u> , 156,	
	209, 210, 216, 219,	
	225, 230, 278, 346, 349	
\nag@warn 113,	
	<u>146</u> , 157, 211, 232,	
	239, 280, 314, 324	
\nag@warnNoLine	117,	
	150, 190, 200, 336	
\NagDeclareFloat	76, 541	
\NeedsTeXFormat 122	
\newcommand 128,	
	143, 146, 150, 154,	
	164, 186, 196, 208,	
	237, 246, 266, 276,	
	291, 293, 304, 344, 541	
\newcounter 126, 127	
\newif 206, 207	
\noexpand 125	
\NotAnEnvironment	82,	
	85, 95, 100, 104, <u>276</u>	
\NotASwitch	. 107, <u>289</u> , 344	
O		
\ObsoleteClass	.. 40, <u>196</u>	
\ObsoleteCS 17,	
	19, 21, 23, 25, 27,	
	29, 31, 35, 36, 38, <u>154</u>	
\ObsoletePackage	...	
 9, 10, 42–44,	
	46, 48, 49, 51, 52,	
	54, 56, 59, 61, 63,	
	65, 68, 70, 73, 74, <u>186</u>	
\or 406, 408, 493, 495	
P		
\PackageError	.. 115, 119	
\PackageInfo	... 131,	
	356, 453, 538, 544	
\ProvidesFile	. 2, 91, 111	
\ProvidesPackage	... 123	
R		
\renewcommand	113, 117, 168	
\rmfamily 22	
S		
\S 8, 11–14, 16, 33,	
	37, 39, 41, 45, 47,	
	50, 55, 57, 60, 67,	
	71, 72, 75, 81, 87, 88	
\scshape 24	
\sectioning 84, 85	
\setbox 390, 474	
\sf@@@subfloat 455	
\sf@bottom 384,	
	387, 437, 468, 471, 524	
\sf@capskip	
	... 416, 432, 503, 519	
\sf@captopadj	.. 417, 504	
\sf@farskip	
	... 384, 386, 468, 470	
\sf@ifpositiontop	..	
 413, 500	
\sf@nearskip	
	... 383, 387, 467, 471	

\sf@subcaption	\textsc	24	416, 417, 432, 437,
... 415, 434, 502, 521	\textsf	26	494, 503, 504, 519, 524
\sf@top . 383, 386, 407,	\textsl	28	\vtop
410, 467, 470, 494, 497	\texttt	30	402, 489
\sffamily	\thenag@c	136, 139	W
\sloppy	\tmp@a	171, 178,	\wd
\slshape	181, 268, 269, 310, 312		392, 395, 479, 482
\string	\tmp@b	311, 312	X
\subfloatrowsep 443, 530	\ttfamily	30	\xdef
	\tw@	380, 464	\Xhsize
			395, 482
T			
\textbf	V		Z
\textit	\vbox ...	403, 428, 490, 515	\z@
\textrm	\vskip	407,	378,
			379, 409, 462, 463, 496