# The nag package[*]

Ulrich Michael Schwarz[†]

July 7, 2005

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 All the same, there are commands, classes and packages which are outdated and superseded.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

### 1.1 Installation.

Process `nag.ins` with LaTeX to obtain four files: `nag.sty` and `l2tabu.nag` 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 implemenation docs. (This is recommended if you plan to extend nag

[*]This document corresponds to nag 0.3, dated 2005/07/07.
[†]`ulmi@users.sarovar.org`

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]{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.

## 1.3 `l2tabu.nag`

In a nutshell, `l2tabu.nag` detects the following:

- Usage of the 2.09-style font commands `\it`, `\bf`, `\rm`, `\sc`, `\sl`, `\tt` and `\cal`.
- Usage of the TeX-style commands `\over` and `\choose`.
- Usage of `\centerline`.
- Usage of the outdated packages epsfig, psfig, epsf, doublespace, fancyheadings, scrpage, umlaut, isolatin, isolatin1, t1enc, caption2, psfonts, mathptm, times, palatino, mathpple, euler and utopia, and of the outdated class scrlttr.
- 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`. 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.

## 1.4 `abort.nag`

Requesting this nag file will turn all complaints into errors.

## 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 seperate 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.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{nag}[2005/07/07 0.3 warning about nag things (ulmi)]
```

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.

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

```
 \newcommand\nag@prepend[2]{%
  \expandafter\let
    \csname nag@@#1@\the\value{nag@c}\expandafter\endcsname
    \csname #1\endcsname
  \expandafter\nag@pr@p@nd\csname #1\expandafter\endcsname
    \csname nag@@#1@\the\value{nag@c}\endcsname{#2}%
    \stepcounter{nag@c}%
}
\newcommand\nag@pr@p@nd[3]{%
  \def#1{#3#2}%
}
```

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

```
\newcommand\nag@warn{%
  \addtocounter{nag@sins}{1}%
  \PackageWarning{nag}%
}
```

```
\newcommand\nag@warnNoLine{%
  \addtocounter{nag@sins}{1}%
  \PackageWarningNoLine{nag}%
}
```

## 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.`

```
\newcommand\ObsoleteCS[3][obsolete]{%
  \AtBeginDocument{%
    \nag@prepend{#2}{%
      \nag@warn{%
Command \expandafter\protect\csname #2\endcsname\space is #1.
\MessageBreak
Use #3 instead}%
    }%
  }%
}
```

## 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`).

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

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

```
\nag@ifcsname{ifcsname}{%
  \renewcommand*\nag@ifcsname[3]{%
    \begingroup
    \ifcsname #1\endcsname #2\else #3\fi
    \endgroup
  }%
```

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

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

`\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.`

```
\newcommand\ObsoletePackage[3][obsolete]{%
  \AtEndDocument{%
    \nag@ifcsname{ver@#2.sty}{%
    \nag@warnNoLine{%
      Package #2 is #1.\MessageBreak
      Use #3 instead}%
    }{}%
  }%
}
```

`\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.`

```
\newcommand\ObsoleteClass[3][obsolete]{%
  \AtEndDocument{%
    \nag@ifcsname{ver@#2.cls}{%
    \nag@warnNoLine{%
      Class #2 is #1.\MessageBreak
      Use #3 instead}%
    }{}%
  }%
}
```

## 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.)

```
\newif\ifnag@haslabel
\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.

```
\newcommand\nag@hackfloat[1]{%
  \nag@prepend{#1}{%
    \nag@prepend{endcenter}{%
      \nag@warn%
  {\lq center\rq\space environment in #1.\MessageBreak
    Maybe you want \protect\centering\space instead}
    }%
  }
  \nag@prepend{#1}{%
```

Add checks to all macros named by `\nag@labels` and `\nag@captions`, respectively. The hascounter etc. information is now global. I don't think those should be hidden by groups. In particular, a center or minipage environment would hide the caption inside from a label outside.

*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.

```
    \@for\labelprovider:=\nag@labels\do{%
      \nag@prepend{\labelprovider}%
       {\nag@captioncheck\global\nag@haslabeltrue}
    }%
    \@for\captionprovider:=\nag@captions\do{%
      \nag@prepend{\captionprovider}{\global\nag@hascaptiontrue}%
    }%
    \global\nag@haslabelfalse\global\nag@hascaptionfalse
  }%
  \nag@prepend{end#1}{%
    \ifnag@hascaption\relax\else
    \nag@warn%
{#1 with no \protect\caption}%
    \fi
  }%
}
\newcommand\nag@captioncheck{%
  \ifnag@hascaption\else
  \nag@warn{\protect\label\space in float, but not after
    \protect\caption}%
  \fi
}
```

Define the lists of commands that are floats, generate labels, and generate captions, respectively. We don't start with defined floats (that is for l2tabu.obs to set up), but keep the list non-empty, so that we can always add to it with `\g@addto@macro{`⟨*list*⟩`}{,`⟨*things*⟩`}`.

```
\def\nag@floats{nag@dummy}
\def\nag@labels{label}
%% The latter two are used by KOMA-Script.
\def\nag@captions{caption,captionabove,captionbelow}
```

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

```
\newcommand\nag@floatsetup{%
  \@for\flo:=\nag@floats\do{%
    \expandafter\nag@hackfloat\expandafter{\flo}%
  }%
}
```

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

```
\AtBeginDocument{%
  \nag@floatsetup
}
```

At the very end, we will display a running total of complaints. This feature was more-or-less suggested by David Kastrup.

```
\AtBeginDocument{%
  \AtEndDocument{%
    \ifnum\value{nag@sins}>0%
    \PackageWarningNoLine{nag}{\arabic{nag@sins} complaints
      in total}%
    \else
    \typeout{No complaints by nag.}%
    \fi
  }%
}
```

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

```
\DeclareOption*{%
  \InputIfFileExists{\CurrentOption.nag}{%
    \typeout{package nag: Loaded \CurrentOption.nag}%
  }{%
    \PackageWarningNoLine{nag}{Required ruleset
      \CurrentOption.nag, and it wasn't there}
  }%
}
\ProcessOptions*
```

## 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.

## Change History

0.1
General: First official version. . . . . 1

0.2
General: Added abort.nag, suggested by Michael Zedler . . . . 1
Rephrased umlaut.sty warning, suggested by Patrick Happel. . 1

0.3
General: Fixed missing globals . . . 6
New ifdefined that won't relax the commands . . . . . . . . . . . . . 1