The slidenotes package*

Hans van der Meer
hansm@wins.uva.nl

Printed January 16, 2004

Abstract

This article describes the use and the implementation of the *slidenotes class*. Its purpose is the typesetting of slides and accompanying notes. Slides can be in portrait or landscape orientation. Options are the typesetting of the slides or slides plus notes, or a collection of slides in small format.

Contents

*This file has version 3.27 dated 1998/04/23.

1 Usage

An input file may consist of a series of slides accompanied by notes. Below the format of a slide in a nutshell:

`slide`

```
\begin{slide}[SLIDE TITLE (optional)]
   ...
   \slidesubtitle{SUBTITLE}
   ...
   <more slide text>
   ...
\end{slide}
```

or

`portraitslide`

```
\begin{portraitslide}[SLIDE TITLE (optional)]
   ...
```

```
\end{portraitslide}
```

landscapeslide

```
\begin{landscapeslide}[SLIDE TITLE (optional)]
   ...
\end{landscapeslide}
```

Do not put anything text outside notes/slide, because it interferes destructively with the layout.

note

```
\begin{note}
   ...
   \cue{MARGIN TEXT}
   ...
   <more note text>
   ...
\end{note}
```

Use `\chapter` for the main structure, each note will be converted to a `\section`. Several parameters and font macro's are defined for various style features; they can be redefined, either in the document or in the *slidenotes.cfg* file. These parameters exist for the dimensions of the slide, the font in the various text parts, several standard texts.

Options exist for:

- choosing between slides, slides + notes, a collection of slides;
- landscape or portrait format;
- notes in a type one notch smaller than in the slides;
- variation in frame around slide.
- variation in vertical centering of the contents.

See the implementation sections for further details.

2 Identification

This document class can only be used with LaTeX 2_ε, so we make sure that an appropriate message is displayed when another TeX format is used.

```
1 ⟨cls | obox⟩\NeedsTeXFormat{LaTeX2e}[1995/12/01]
```

Announce class names and versions:

```
2 ⟨*cls⟩
3 \ProvidesClass{slidenotes}[1998/04/23 3.27  Slides and notes]
```

3 Declaration of Class Options

In this part we define the options for this class that are additional to those of its parent class.

3.1 Show Options to User

Show options to the user with option `help`.

```
\DeclareOption{help}{\ClassWarningNoLine{slidenotes}{%
    Options (first one is default):
      \MessageBreak
    notes,slides,minis:\space type of production;
      \MessageBreak
    portrait,landscape:\space slide orientation;
      \MessageBreak
    mixed,unmixed,anti:\space both, chosen, anti orientation;
      \MessageBreak
    center,nocenter,squeeze:\space vertical slide position;
      \MessageBreak
    rect,oval,rules,drules,noframe:\space type of slideframe;
      \MessageBreak
    newpage,nonewpage:\space yes/no start slide on new page;
      \MessageBreak
    smallnotes:\space notes one notch smaller}}
```

3.2 Slides and/or Notes

`\ifnotes` `\ifslides` `\ifminis`

Flag `\ifnotes` governs the production of notes in the typesetting of slides and notes, flag `\ifslides` signals the typesetting of slides only and `\ifminis` is for the production of a smallsized slide collection. Options `notes`, `slides` and `minis` perform the selection of these modes.

```
\newif\ifnotes
\newif\ifslides
\newif\ifminis
\DeclareOption{notes}{\notestrue\minisfalse\slidesfalse}
\DeclareOption{slides}{\notesfalse\minisfalse\slidestrue}
\DeclareOption{minis}{\notesfalse\ministrue\slidesfalse}
```

3.3 Portrait or Landscape Orientation

`\iflandscape`

Typesetting of slides can come in portrait or landscape format; even mixing of these within one document is possible.

```
\newif\iflandscape
\DeclareOption{landscape}{\landscapetrue}
\DeclareOption{portrait}{\landscapefalse}
```

`\ifmixed` `\ifanti`

The next series of flags control the production of mixed portrait/landscape slides or runs with one orientation only.

With the `mixed` option one may mix portrait and landscape oriented slides. If a slide has an orientation differing from the one currently in effect (from the landscape and portrait class option), it will be rotated over 90 degrees. This however, requires the presence of one of the graphics styles (*graphics.sty* or *graphicx.sty*) and a conforming printer driver. If one of these styles is not already present, one will be loaded at the start of the document.

The `unmixed` option is for the case where rotation is not possible (as for ex. the native OzTeX printer driver) or not wanted. Only slides conforming to the current orientation are typeset. E.g. if landscape format is chosen, then all slides produced

from the environments `landscapeslide` and the neutral `slide` are done, those of `portraitslide` are skipped. If there are any slides skipped, then a warning will be issued and the job must be rerun with the `anti` option; this will produce the missing slides in the complementary orientation. The `anti` option can coexist with the `unmixed` option.

```
\newif\ifmixed
\newif\ifanti
\DeclareOption{mixed}{\mixedtrue\antifalse}
\DeclareOption{unmixed}{\mixedfalse}
\DeclareOption{anti}{\mixedfalse\antitrue}
```

3.4 Vertical Centering of Slide Material

`\ifcenter` `\ifsqueeze` Everything below the title at the head of the slide can be centered vertically as well as kept tightly below the head, the options `center` and `nocenter` accomplish this. Filling will be done with `\vfil`, thereby allowing it to be overridden simply by the inclusion of a `\vfill`. Furthermore one can `squeeze` the slidebox to minimal height for the slides + notes combination.

```
\newif\ifcenter
\newif\ifsqueeze
\DeclareOption{center}{\squeezefalse\centertrue}
\DeclareOption{nocenter}{\squeezefalse\centerfalse}
\DeclareOption{squeeze}{\squeezetrue\centerfalse}
```

3.5 Suppress Newpage with Slide

`\ifnewpage` In typesetting notes it may be usefull to suppress the newpage at the beginning of a slide. This option has effect in note production only.

```
\newif\ifnewpage
\DeclareOption{newpage}{\newpagetrue}
\DeclareOption{nonewpage}{\newpagefalse}
```

3.6 Notes in Smaller Type

`\ifsmallnotes` The `smallnotes` option switches the font set in the notes one notch smaller than the standard sizes chosen in the style. The sizes in the slides are always those specified by the style.

```
\newif\ifsmallnotes
\smallnotesfalse
\DeclareOption{smallnotes}{\smallnotestrue}
```

3.7 Variation in Slideframe

`\framevariant` The frame of a slide can be rectangular, oval, delineated by rules or empty.

```
\newcommand*\framevariant{}
\DeclareOption{noframe}{\renewcommand*\framevariant{n}}
\DeclareOption{rules}{\renewcommand*\framevariant{r}}
\DeclareOption{drules}{\renewcommand*\framevariant{d}}
\DeclareOption{rect}{\renewcommand*\framevariant{f}}
\DeclareOption{oval}{\renewcommand*\framevariant{o}}
```

The \obox macro is in file obox.sty; if necessary it is read when document processing starts.

```
\AtBeginDocument{\@ifundefined{obox}%
  {\IfFileExists{obox.sty}{\RequirePackage{obox}}%
    {\if\framevariant o
    \ClassWarningNoLine{slidenotes}{oval option disabled}%
    \renewcommand*\framevariant{f}\fi}}%
  {}}
```

\ovalslideframerule An oval slideframe is drawn using LaTeX's picture command. Therefore the range of linethicknesses is restricted to those provided by the circle fonts. The following macro is defined to one of the standard linethickness commands.

```
\newcommand*\ovalslideframerule{\thicklines}
```

3.8 Pass Options and Load Parent Class

\parentclass Since the *slidenotes class* is implemented as a modification of an existing document class, we must load the parent class. However in order to ease changes in parent class the name of this class is parametrized in macro \parentclass. An obvious candidate is *report* or friends.

Customize with a definition of \parentclass before calling \documentclass.

```
\providecommand\parentclass{report}
```

Set the class defaults.

```
\ExecuteOptions{notes,portrait,mixed,rect,center,newpage}
```

The options of the \documentclass call which are not specific for the *slidenotes class* must be passed to the parent class. We take the opportunity to select some defaults, e.g. production of a titlepage in case the parent class is *article*.

```
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{\parentclass}}
\PassOptionsToClass{titlepage,a4paper}{\parentclass}
\ProcessOptions
```

After this we load the parent class. Note that the landscape option is not passed if notes is also specified. For the production of skipped slides we may also need landscaped printing.

```
\iflandscape\ifnotes\else
  \PassOptionsToClass{landscape}{\parentclass}%
\fi\fi
\LoadClass{\parentclass}
```

Show some of the options chosen.

```
\ClassWarningNoLine{slidenotes}{%
  \iflandscape LANDSCAPE\else PORTRAIT\fi
  \space SLIDES\space
  \ifminis COLLECTION \fi \ifnotes + NOTES \fi
  PRODUCTION}
```

4 Some Useful Macros

We define some helpfull macros first, as we need them later on. Defining is with `\providecommand` just in case the parent class has already loaded them.

`\@swapdimens` This macro swaps two dimens with the help of a third.

```
\providecommand*\@swapdimens[3]{#3=#1 #1=#2 #2=#3\relax}
```

`\@ifemptyarg` Testing for the presence or absence of a parameter.

```
\providecommand{\@ifemptyarg}[1]{% {absence}{presence}
  \ifx\@empty#1\@empty
  \expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi}
```

4.1 Postprocessing of Options

Reset the squeeze option to false when slides or minis are produced. This is done to prevent the slide numbering jumping up and down.

```
\ifminis\squeezefalse\fi
\ifslides\squeezefalse\fi
```

Production of notes turns `mixed` on and `anti` off. When typesetting notes, no rotation on the part of the dvi driver is required; both can be displayed.

```
\ifnotes\mixedtrue\antifalse\fi
```

When the skipped slides are in production we must effectively change the orientation from landscape to portrait or vice versa.

```
\ifanti
  \iflandscape\landscapefalse\else\landscapetrue\fi
  \ClassWarningNoLine{slidenotes}{ANTI OPTION => ORIENTATION CHANGED}
\fi
```

`verbatim` We need the *verbatim.sty* for skipping portions of the text.

```
\RequirePackage{verbatim}
```

`\rotatebox` We need the `\rotatebox` macro in case of mixed slides or minis; if necessary load *graphics.sty*. We are even prepared for the case where this file cannot be found.

```
\ifmixed\ifnotes\else
  \AtBeginDocument{%
    \@ifundefined{rotatebox}%
      {\IfFileExists{graphics.sty}%
        {\RequirePackage{graphics}}%
        {\mixedfalse}%
      }{}}
\fi\fi
```

5 Dimensions

`\slidewidth` `\slideheight` We give default values for height and width in portrait format, `\slideheight` and `\slidewidth` respectively. In the initialization section the landscape format is set by an exchange of height and width. Please note that these values are set for slides

production: magnified values. They will be automatically diminished when slides & notes are made. All other dimensions (frame separation, linethickness, etc.) must be given unmagnified values, just as one would do in normal typesetting.

```
\newlength\slidewidth
\newlength\slideheight
\setlength\slidewidth{176mm}
\setlength\slideheight{248mm}
```

`\slidemagnification` For the production of the slides the magnification is set to `\slidemagnification`.

```
\newcommand*\slidemagnification{\magstep4}
```

`\slideboxrule` `\slideboxsep` `\slidenotesep` `\slidetitlesep` Furthermore we define special parameters for the separation between frame and contents of the slide, `\slideboxsep` and for the thickness of the frame, `\slideboxrule`. Length `\slidenotesep` is the distance between the slide and the note, `\slidetitlesep` the skip below a call of a (sub)title macro.

```
\newlength\slideboxrule
\newlength\slideboxsep
\newlength\slidenotesep
\newlength\slidetitlesep
\setlength\slideboxrule{.2mm}
\setlength\slideboxsep{3mm}
\setlength\slidenotesep{8mm}
\setlength\slidetitlesep{\z@}
```

6 Registers

`\slidecounter` The slides are counted and the counter is reset whenever a new chapter is begun, if there is a chapter structure, of course.

```
\@ifundefined{c@chapter}%
  {\newcounter{slidecounter}}%
  {\newcounter{slidecounter}[chapter]}
```

`\missedslidecounter` If we have both the `mixed` and `anti` options off we must keep count of the number of slides passed over. Always define this counter, it may be present in the auxiliary files from a previous run. We cannot use the standard LaTeX's `\newcounter` command, because the the value reported will be the value produced by the last included file; the reason being the recording of the counter in the auxiliary files. Therefore we use plain TeX's `\newcount`.

```
\newcount\missedslidecounter
```

`\minicounter` Minis come just four to a page, therefore we count them.

```
\newcounter{minicounter}
```

`\slidebox` Slides are typeset in a box for later placement and height adjustement.

```
\newsavebox\slidebox
```

`\slidemarker` Slides are marked by a text and a number. The text by default comes from the `\chapter` macro and is put into a token register named `\slidemarker`. In the article style, however there is no chapter structure. There we intialize this token

register from the `\title`; note that we need to do this at `\begin{document}`, because `\maketitle` interfers by setting `\@title` empty.

```
\newtoks\slidemarker
\AtBeginDocument{%
  \@ifundefined{@title}{\def\@title{\Collection}}{}%
  \slidemarker={\@title}}
```

`\everyslide` `\everynote` At the start of each slide a token register `\everyslide` is executed, where the user can place settings that should be local to each slide. The same for the start of each note where `\everynote` is executed.

```
\newtoks\everyslide
\newtoks\everynote
```

7 Note Environment

`note` In case of productions of slides the notes are simply left out; we effectuate this with the comment mechanism of the *verbatim.sty*. Otherwise the material in the `note` environment is typeset. The environment for notes is started with font selections; the fontsize is initialized to normalsize, but all sizes are set one notch down for the `smallnotes` option. Execution starts the note with the code in token register `\everynote`.

```
\ifnotes
\newenvironment{note}[1][]{%
  \notesfont
  \ifsmallnotes\smallersizes\fi
  \normalsize
  \the\everynote
```

Separate slide and note with some whitespace.

```
  \addvspace{\slidenotesep}%
```

We use the `list` environment for offsetting the margin inside the notes, a variation on the `quote` environment. However, the leftmargin is reduced to zero, because we do not want extra indentation for the text of the notes.

```
  \list{}{\leftmargin=\z@
   \rightmargin=\marginwidth
   \parsep=\parskip}\item[]
```

Finish the `note` environment, close page with a filler when newpage option is in effect.

```
  \ignorespaces}{\endlist\ifnewpage\vfill\fi}
```

If notes are not wanted, `\let` the `note` environment to `comment`.

```
\else
  \let\note=\comment
  \let\endnote=\endcomment
\fi
```

`\smallersizes` This macro switches all of LaTeX's sizes one notch down (except for `\tiny`, the smallest size).

```
\newcommand*\smallersizes{%
```

```
  \let\Huge\huge
  \let\huge\LARGE
  \let\LARGE\Large
  \let\Large\large
  \let\large\normalsize
  \let\normalsize\small
  \let\small\footnootsize
  \let\footnotesize\scriptsize
  \let\scriptsize\tiny
  }
```

8 Slide Environment

8.1 Slide Orientation

portraitslide
landscapeslide
slide

Here we differentiate between the various possibilities for the slide orientation. Note that we set a flag for the current orientation in order to allow orientation dependent code.

First the case of mixed orientations.

```
\ifmixed
\newenvironment{portraitslide}{%
  \landscapeslidefalse
  \iflandscape\@swapdimens\slideheight\slidewidth\@tempdima
  \fi
  \@slide}{\@endslide
  \iflandscape\putslide{90}\else\putslide{0}\fi}
\newenvironment{landscapeslide}{%
  \landscapeslidetrue
  \iflandscape\else\@swapdimens\slideheight\slidewidth\@tempdima
  \fi
  \@slide}{\@endslide
  \iflandscape\putslide{0}\else\putslide{90}\fi}
\newenvironment{slide}{\@slide}{\@endslide\putslide{0}}
```

Secondly the case where missed slides are produced. A tricky point here is that we have already complemented the value of `\iflandscape`; therefore no dimensions have to be interchanged and the tests all seem to point the wrong way.

```
\else\ifanti
\newenvironment{portraitslide}{%
  \landscapeslidefalse
  \iflandscape
    \refstepcounter{slidecounter}%
    \let\@slide=\comment\let\@endslide=\endcomment
  \fi
  \@slide}{\@endslide
  \iflandscape\else\putslide{0}\fi}
\newenvironment{landscapeslide}{%
  \landscapeslidetrue
  \iflandscape\else
    \refstepcounter{slidecounter}%
    \let\@slide=\comment\let\@endslide=\endcomment
  \fi
```

```
  \@slide}{\@endslide
  \iflandscape\putslide{0}\fi}
\newenvironment{slide}{%
  \refstepcounter{slidecounter}%
  \let\@slide=\comment\let\@endslide=\endcomment
  \@slide}{\@endslide}
```

Finally the case of separate slide orientation. A skipped slide must be counted always in `slidecounter` in order to keep the numbers straight and is also recorded in `missedslidecounter`; the `comment` environment is used to get rid of a skipped slide.

```
\else
\newenvironment{portraitslide}{%
  \landscapeslidefalse
  \iflandscape
    \refstepcounter{slidecounter}%
    \global\advance\missedslidecounter\@ne
    \let\@slide=\comment\let\@endslide=\endcomment
  \fi
  \@slide}{\@endslide
  \iflandscape\else\putslide{0}\fi}
\newenvironment{landscapeslide}{%
  \landscapeslidetrue
  \iflandscape\else
    \refstepcounter{slidecounter}%
    \global\advance\missedslidecounter\@ne
    \let\@slide=\comment\let\@endslide=\endcomment
  \fi
  \@slide}{\@endslide
  \iflandscape\putslide{}\fi}
\newenvironment{slide}{\@slide}{\@endslide\putslide{0}}
\fi\fi
```

And then we have to report if there are any skipped slides.

```
\ifmixed\else\ifanti\else
\AtEndDocument{\ifnum\missedslidecounter>\z@
  \ClassWarningNoLine{slidenotes}{Rerun with 'anti'
  for \the\missedslidecounter\space slide(s) skipped}%
  \fi}
\fi\fi
```

8.2 Placing a Slide

`\putslide` The positioning of a slide is split off from its production in the following macro. We define this positioning differently for the various production schemes.

```
\ifnotes
  \newcommand*\putslide[1]{\noindent\rputslide{#1}\par}
\fi
\ifslides
  \newcommand*\putslide[1]{\begin{center}\rputslide{#1}\end{center}}
\fi
\ifminis
  \newcommand*\putslide[1]{\noindent
```

```
    \rputslide{#1}%
    \stepcounter{minicounter}%
      \ifnum\value{minicounter}=2
        \par
      \else\ifnum\value{minicounter}=4
        \newpage\setcounter{minicounter}{0}%
      \else\hfil\fi\fi}
\fi
```

`\rputslide` We alse need variants in accordance with the need of slide rotation.

```
\newcommand*\rputslide[1]{\usebox{\slidebox}}
\ifmixed\ifnotes\else
  \renewcommand*\rputslide[1]{%
    \ifnum#1=\z@\usebox{\slidebox}%
    \else\rotatebox{#1}{\kern-.5\slidewidth\usebox{\slidebox}}%
    \fi}
\fi\fi
```

8.3 Typeset Slide Contents

`\@slide` Slides are collected between `\@slide` and `\@endslide`. Each slide starts a new page, except when a collection is made or possibly in note production. We program the `\newpage` here in order to get the pagenumbering in the table of contents right. The font is set to the special chosen `\slidefont`. The fontsize is initialized to the normal size. From the optional argument the title is put into the *toc* file.

8.3.1 Starting a Slide

```
\newcommand*\@slide[1][]{%
\ifnotes
\ifnewpage\newpage\else\addvspace{\bigskipamount}\fi
\else
\ifminis\else\newpage\fi
\fi
  \refstepcounter{slidecounter}%
  \slidefont
  \normalsize
```

Write a line to the *.toc* file when notes are in production.

```
\ifnotes
  \@ifemptyarg{#1}%
    {\addcontentsline{toc}{section}{\Slide\,\theslidecounter}}%
    {\addcontentsline{toc}{section}{#1}}%
\fi
```

The contents of the slide is typeset in a `minipage`, which is captured in a box. Execution starts the box with the code in token register `\everyslide`. When the optional argument is not empty, this furnishes the slide title at its top. The width of the `minipage`, in which the contents of the slide are typeset, is the `\slidewidth` adjusted for the thickness of the surrounding frame and the separation between frame and contents.

```
  \@tempdima\slidewidth
  \advance\@tempdima-\tw@\slideboxrule
  \advance\@tempdima-\tw@\slideboxsep
```

```
  \@tempdimb\slideheight
  \advance\@tempdimb-\tw@\slideboxrule
  \advance\@tempdimb-\tw@\slideboxsep
  \begin{lrbox}{\slidebox}%
    \the\everyslide
    \ifsqueeze
        \begin{minipage}{\@tempdima}%
    \else
        \begin{minipage}[][\@tempdimb][t]{\@tempdima}%
    \fi
```

We want to find out the height of the slide, therefore we catch the slide contents in a `\vbox`.

```
     \setbox\@tempboxa=\vbox\bgroup\leavevmode
     \@ifemptyarg{#1}{}{\slidetitle{#1}}\ifcenter\vfil\fi
     \ignorespaces}
```

8.3.2 Finishing a Slide

`\@endslide` In the second part of slide typesetting we construct the slide.

```
\newcommand*\@endslide{%
```

First we do vertical centering with `\vfil` if required.

```
  \ifcenter\vfil\fi
```

Here we close the catching box and check its height.

```
    \egroup
    \chk@slideheight\@tempboxa
    \unvbox\@tempboxa
    \end{minipage}\end{lrbox}%
```

The slide contents are enclosed in a framed box and then together with the marker placed into a `\parbox`. The height of the box is adjusted to slideheight minus frame. The complete slide is again captured in `\slidebox` because we may want to subject it to a rotation.

```
  \sbox\slidebox{\parbox{\slidewidth}{%
    \fboxsep\slideboxsep
    \fboxrule\slideboxrule
    \csname frame@\framevariant\endcsname
    \\*[\smallskipamount]
    \makebox[\slidewidth]{%
      \small\strut\the\slidemarker
      \hfil\Slidenumber~\theslidecounter}%
  }}}
```

8.3.3 Framing a Slide

`\frame@f` Rectangular box around slide contents.

```
\newcommand*\frame@f{%
  \fbox{\usebox\slidebox}%
  }
```

`\frame@o` Oval box around slide contents.

```
\newcommand*\frame@o{%
```

```
  \@tempdimb\dp\slidebox
  \advance\@tempdimb-\fboxsep
  \advance\@tempdimb-\fboxrule
  \obox[][\ovalslideframerule]{%
    \raisebox{\@tempdimb}{\usebox\slidebox}}%
  }
```

`\frame@n` Nothing around slide contents.

```
\newcommand*\frame@n{%
  \advance\fboxsep\fboxrule \fboxrule\z@
  \fbox{\usebox\slidebox}%
  }
```

`\frame@r` Rule above and below slide contents.

```
\newcommand*\frame@r{%
  \rule{\slidewidth}{\fboxrule}\\*[\fboxsep]%
  \usebox{\slidebox}\\*[\fboxsep]%
  \rule{\slidewidth}{\fboxrule}%
  }
```

`\frame@d` Double rules above and below slide contents.

```
\newcommand*\frame@d{%
  \rule{\slidewidth}{\fboxrule}\\*[\tw@\fboxrule]%
  \rule[\baselineskip]{\slidewidth}{\fboxrule}%
    \vspace{-\baselineskip}\\*[\fboxsep]%
  \usebox{\slidebox}\\*[\fboxsep]%
  \rule{\slidewidth}{\fboxrule}\\*[\tw@\fboxrule]%
  \rule[\baselineskip]{\slidewidth}{\fboxrule}%
    \vspace{-\baselineskip}%
  }
```

8.3.4 Check Slide Height

`\chk@slideheight` Here we define the macro that checks the height of slides and gives a warning when overfilling occurs. Keep the changes to `\dimen@` local by grouping.

```
\newcommand*\chk@slideheight[1]{{%
  \dimen@\ht#1
  \advance\dimen@\dp#1
  \ifnum\dimen@>\slideheight
    \ClassWarning{slidenotes}{slide too high by \the\dimen@}%
  \fi}}
```

8.4 Titles on Slides

`\slidetitle` `\slidesubtitle` Put centered title or subtitle on the slide. The first (and optional) argument controls the font setting. These titles are followed by a `\medskip`; whitespace before the title is the responsibility of the slidewriter. These titles are suppressed if their argument somehow turns out to be empty.

```
\newcommand*\slidetitle{\@slidetitle{\slidetitlefont}}
\newcommand*\slidesubtitle{\@slidetitle{\slidesubtitlefont}}
\newcommand*\@slidetitle[2]{%
    \@ifemptyarg{#2}{}%
        {\begin{center}#1#2\end{center}\vspace{\slidetitlesep}}}
```

8.5 Slideformat Dependent Typesetting

\iflandscapeslide Here we define a flag to be used for the implementation of format dependent code selection. This flag is initialized to the current default orientation.

```
\newif\iflandscapeslide
\iflandscape\landscapeslidetrue\else\landscapeslidefalse\fi
```

\portraitonly \landscapeonly Material can be selected on the basis the slide's format, portrait or landscape orientation with respectively:
`\landscapeonly{...}`
`\portraitonly{...}`
For example, where a list can be placed without any difficulty on a portrait slide, it may require the use of `multicols` in landscapeformat. If problems arise, e.g. with the fixing of catcodes, then use the basic construction `\iflandscapeslide .. \else ..\fi`

```
\newcommand\landscapeonly[1]{\iflandscapeslide#1\fi}
\newcommand\portraitonly[1]{\iflandscapeslide\else#1\fi}
```

8.6 Cues in the Margin

\marginwidth The length variable `\marginwidth` holds the margin space that will be reserved for short remarks.

```
\newlength\marginwidth
\setlength\marginwidth{3cm}
```

\cue Giving short remarks in the margin of the notes text. To be used for finding specific text parts with a glance.

```
\newcommand*\cue[2][\cuefont]{%
  \marginpar{\hspace*{-\marginwidth}%
  \parbox{\marginparwidth}{\raggedright\sloppy#1#2}}}
```

`\cue` cannot allow marginpar switching from left to right margin when typesetting twosided material; therefore force all marginpars into the same margin.

```
\@mparswitchfalse
```

9 Style Features

9.1 Redefine Chapter Code

In order to process the slide marking two macro's in the production of chapter heads are redefined. Also a new chapter starts a new page and the pagestyle for this page is set to `empty`.

```
\@ifundefined{c@chapter}{}{%
  \let\old@makechapterhead=\@makechapterhead
  \let\old@makeschapterhead=\@makeschapterhead
  \renewcommand*\@makechapterhead[1]{%
    \global\slidemarker={#1}\old@makechapterhead{#1}%
    \thispagestyle{empty}\newpage}%
  \renewcommand*\@makeschapterhead[1]{%
    \global\slidemarker={#1}\old@makeschapterhead{#1}%
    \thispagestyle{empty}\newpage}%
  }
```

9.2 Pagestyle

Slides and minis want the `empty` style.

```
\ifnotes\else\pagestyle{empty}\fi
```

9.3 Fonts

\headerfont
\slidefont
\notesfont
\footnotefont
\slidetitlefont
\slidesubtitlefont
\cuefont

Choose fonts for the various elements of slides and notes. Their names speak for themselves.

```
\newcommand*\headerfont{\rmfamily}
\newcommand*\slidefont{\sffamily}
\newcommand*\notesfont{\rmfamily}
\newcommand*\footnotefont{\sffamily\mdseries\upshape}
\newcommand*\slidetitlefont{\Large\bfseries\boldmath}
\newcommand*\slidesubtitlefont{\normalsize\bfseries\boldmath}
\newcommand*\cuefont{\sffamily\bfseries\boldmath}
```

10 Language Adjustments

\Collection
\Slidenumber
\Slide

We define some language specific verbs, such as `\Collection` and `\Slidenumber`. To be customized by redefinition in the configuration file.

```
\newcommand*\Collection{LECTURE NOTES}
\newcommand*\Slidenumber{\#}
\newcommand*\Slide{Slide}
```

10.1 Example of Adjustements in Configuration File

Example of a configuration file with dutch equivalents for language dependent items.

```
⟨/cls⟩
⟨*cfg⟩
\renewcommand*\Collection{COLLEGE AANTEKENINGEN}
\renewcommand*\Slidenumber{Nr.}
\renewcommand*\Slide{Transparant}
⟨/cfg⟩
⟨*cls⟩
```

11 Directory Localization

\CurrentDirectory
\DirectorySeparator

We can determine from `\@currdir` which character separates directories in a path name. E.g. in UNIX this is `/` from the string `./`, but in the MacOS the current directory and the separator are both `:`. Therefore we extract from `\@currdir` the last character (of two at most).
Make `\CurrentDirectory` a synonyme for `\@currdir`.

```
\let\CurrentDirectory=\@currdir
\def\DirectorySeparator#1#2‘\^^M{\@ifemptyarg{#2}{#1}{#2}}
\edef\DirectorySeparator{%
\expandafter\DirectorySeparator\CurrentDirectory‘\^^M}
```

\LastChar Another macro delivers the last character of a string.

```
\providecommand*{\LastChar}[1]{%
  \@ifemptyarg{#1}{}{\expandafter\@lastchar#1‘\^^M}}
\def\@lastchar#1#2‘\^^M{\@ifemptyarg{#2}{#1}{\@lastchar#2‘\^^M}}
```

\DirectoryName The next macro ensures that a path name ends correctly, when a filename is concatenated with it. If the directory separator character isn't the last character, it is added.

```
\providecommand*{\DirectoryName}[1]{\@ifemptyarg{#1}{}%
  {\if\LastChar{#1}\DirectorySeparator\relax#1\else
    #1\DirectorySeparator\fi}}
```

\Setfolder Macro `\Setfolder` can be used to install a standard folder (directory) name. E.g. a name `\figuresfolder` can be defined as the standard place for figures. Supply as first argument to `\Setfolder` the macro name for the folder. e.g. `\figuresfolder` and as second parameter its location on disk. Below three of these folders (here initialized with empty names) are defined in the example configuration file.

```
\newcommand*\Setfolder[2]{\edef#1{\DirectoryName{#2}}}
⟨/cls⟩
⟨*cfg⟩
\Setfolder{\mainfolder}{}
\Setfolder{\commonfolder}{}
\Setfolder{\figuresfolder}{}
⟨/cfg⟩
⟨*cls⟩
```

12 Read Configuration File

Last, but not least, see if there is a configuration file `slidenotes.cfg` and read it for the final adjustments. In this file one can change things like the font selection, size of various length variables, etc.

```
\InputIfFileExists{slidenotes.cfg}{}{}
```

13 Process the Orientation Options

Final settings take place afterwards, because the variables involved may have been changed value in the configuration file.

13.1 Page Orientation

In case of landscape orientation the height and width of the page need to be exchanged for slides + notes, for slides in the anti orientation also. For notes we always use a portrait paperformat with both portrait and landscape oriented slides on it.

```
\ifnotes
  \iflandscape\@swapdimens\paperheight\paperwidth\@tempdima\fi
\else
  \ifanti\@swapdimens\paperheight\paperwidth\@tempdima\fi
\fi
```

13.2 Slide Dimensions and Orientation

At this point height and width of the slides are known in their final sizes. For the proces of typesetting they have to be reduced by the magnification applied later on.

```
\divide\slidewidth by\slidemagnification
\multiply\slidewidth by\@m
\divide\slideheight by\slidemagnification
\multiply\slideheight by\@m
```

In case of landscape format the height and width of slides need to be exchanged, because they are defined in portrait format.

```
\iflandscape\@swapdimens\slideheight\slidewidth\@tempdima\fi
```

14 Shutoff Titles if Necessary

`\maketitle` Kill the title and the chapter pages for slide production, but not when a collection is prepared. Also allow for the fact that `\chapter` effects a change to pagestyle `plain`.

```
\ifslides
  \renewcommand\maketitle{}
  \@ifundefined{c@chapter}{}{%
    \renewcommand*\@makechapterhead[1]{\global\slidemarker={#1}%
      \thispagestyle{empty}}%
    \renewcommand*\@makeschapterhead[1]{\global\slidemarker={#1}%
      \thispagestyle{empty}}%
  }
\fi
```

Disable typesetting of the table of contents unless notes are in production.

```
\ifnotes\else\AtBeginDocument{\let\tableofcontents=\relax}\fi
```

15 Page Dimensions

Here we do some calculations on sizes and set the variables to their proper values.

15.1 Fullsize Slides

First we typeset with an enlarged magnification.

```
\ifslides
  \mag\slidemagnification\relax
```

Horizontal dimensions:

```
  \setlength\hoffset{-1in}
    \divide\hoffset by\slidemagnification
    \multiply\hoffset by\@m
  \setlength\textwidth{\paperwidth}
    \divide\textwidth by\slidemagnification
    \multiply\textwidth by\@m
  \setlength\oddsidemargin{0pt}
  \setlength\evensidemargin{0pt}
```

Vertical dimensions:

```
  \setlength\voffset{-1in}
    \divide\voffset by\slidemagnification
    \multiply\voffset by\@m
  \setlength\headheight{0pt}
  \setlength\headsep{0pt}
  \setlength\textheight{\paperheight}
    \addtolength\textheight{-2\topmargin}
    \divide\textheight by\slidemagnification
    \multiply\textheight by\@m
\fi
```

15.2 Collection of Slides

```
\ifminis
  \setlength\headheight{0pt}
  \setlength\headsep{0pt}
  \setlength\footskip{0pt}
  \setbox\@tempboxa\vbox{\hbox{\slidefont\small\strut}\smallskip\null}
  \@tempdima=2\slidewidth
  \ifmixed
    \advance\@tempdima by 2\ht\@tempboxa
  \else
    \advance\@tempdima by 8mm
  \fi
  \ifdim\@tempdima>\textwidth \setlength\textwidth{\@tempdima}\fi
  \@tempdimb=2\slideheight
    \advance\@tempdimb by 2\ht\@tempboxa
  \ifdim\@tempdimb>\textheight
    \setlength\textheight{\@tempdimb}
    \@tempdimb\paperheight
    \advance\@tempdimb-\textheight
    \addtolength\topmargin{-\@tempdimb}
  \fi
  \setlength\oddsidemargin{0pt}
  \setlength\evensidemargin{0pt}
\fi
```

15.3 Slides plus Notes

Change margin settings and enlarge `\marginparwidth` by the extra margin.

```
\ifnotes
  \setlength\marginparsep{5mm}
  \addtolength\marginparwidth{\marginwidth}
\fi
```

Initialize the font.

```
\notesfont
```

⟨/cls⟩

16 Oval Macro's

Macro to translate dimensions
`#1{dimen}` or `#1(number)` → `#1{number\unitlength}`

438 ⟨∗obox⟩
```
439 \ProvidesClass{obox}[1995/09/04 vs 1.2  Oval box change]
440 \def\@pickonedim#1{\@ifnextchar(%)
441     {\@@pickonedim{#1}}{#1}}
442 \def\@@pickonedim#1(#2){#1{#2\unitlength}}
```

Redefinition of oval from the picture environment to
`\oval[position](width,height)[optional lrtb=parts]`
position parameter determis how oval will be placed:
`o` midpoint (default as in LaTeX picture environment)
`t` top of oval at baseline
`b` bottom of oval at baseline
`c` halfheight of oval at baseline
`n` neutral placement

```
443 \gdef\oval{\@ifnextchar[%]
444     {\@oval}{\@oval[o]}}
445 \gdef\@oval[#1](#2,#3){\@ifnextchar[%]
446     {\@@oval[#1](#2,#3)}{\@@oval[#1](#2,#3)[]}}
447 \gdef\@@oval[#1](#2,#3)[#4]{\begingroup\boxmaxdepth \maxdimen
448   \@ovttrue \@ovbtrue \@ovltrue \@ovrtrue
449   \@tfor\reserved@a :=#4\do{\csname @ov\reserved@a false\endcsname}%
450   \@ovxx #2\unitlength \@ovyy #3\unitlength
451   \@tempdimb \ifdim\the@ovalradius=\z@
452     \ifdim \@ovyy >\@ovxx \@ovxx \else \@ovyy \fi
453     \else \the@ovalradius \fi
454   \advance \@tempdimb -\tw@\p@
455   \@getcirc \@tempdimb
456   \@ovro \ht\@tempboxa \@ovri \dp\@tempboxa
457   \@ovdx\@ovxx \advance\@ovdx -\@tempdima \divide\@ovdx \tw@
458   \@ovdy\@ovyy \advance\@ovdy -\@tempdima \divide\@ovdy \tw@
459   \@circlefnt \setbox\@tempboxa
460   \hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
461   \if@ovl \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx \fi
462   \if@ovt \@ovhorz \kern -\@ovxx \fi
463   \if@ovb \raise \@ovyy \@ovhorz \fi}%
464   \advance\@ovdx\@ovro \advance\@ovdy\@ovro
465   \if#1t\@ovdx\z@\@ovdy\tw@\@ovdy\fi
466   \if#1b\@ovdx\z@\@ovdy\z@\fi
467   \if#1c\@ovdx\z@\fi
468   \if#1n\@ovdx\z@\@ovdy\dp\@tempboxa\fi
469   \ht\@tempboxa\z@ \dp\@tempboxa\z@
470   \@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}%
471   \endgroup}
```

Oval analogon of `\fbox` is added:
`\obox[optional ovalradius(length)][linethickness-cmd]{box-contents}`

```
472 \def\obox{\@ifnextchar[%]
473     {\@obox}{\@@obox[][]}}
474 \def\@obox[#1]{\@ifnextchar[%]
```

```
    {\@@obox[#1]}{\@@obox[#1][]}}
\def\@@obox[#1][#2]#3{%
  \begingroup
    \@ifemptyarg{#1}{}{\ovalradius{#1}}%
    \setbox\z@\hbox{\advance\fboxsep\@wholewidth \fboxrule\z@ \fbox{#3}}%
    \dimen@\ht\z@ \advance\dimen@\dp\z@ #2\let\unitlength\@empty
    \leavevmode\oval[n](\wd\z@,\dimen@)\box\z@
  \endgroup}
\def\ovalradius{\@pickonedim\@ovalradius}
\def\@ovalradius#1{\@tempdima#1\relax
    \edef\the@ovalradius{\number\@tempdima sp}\ignorespaces}
\def\ovalfraction#1{%
    \edef\the@ovalradius{#1\noexpand\@tempdimb}\ignorespaces}
\ovalradius{0pt}                % initialisation
```
⟨/obox⟩