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

1 Usage .......................... 1
2 Identification ...................... 2
3 Declaration of Class Options 2
   3.1 Show Options to User .................. 2
   3.2 Slides and/or Notes .................. 3
   3.3 Portrait or Landscape Orientation .......... 3
   3.4 Vertical Centering of Slide Material .......... 4
   3.5 Suppress Newpage with Slide ................. 4
   3.6 Notes in Smaller Type ................. 4
   3.7 Variation in Slideframe .......... 4
   3.8 Pass Options and Load Parent Class ........ 5
4 Some Useful Macros ............... 5
   4.1 Postprocessing of Options .................. 6
5 Dimensions ......................... 6
6 Registers ......................... 7
7 Note Environment ................. 8
8 Slide Environment ................. 9
   8.1 Slide Orientation .................. 9
   8.2 Placing a Slide .................. 10
   8.3 Typeset Slide Contents ................. 11
   8.3.1 Starting a Slide .................. 11

---

*This file has version 3.27 dated 1998/04/23.*
8.3.2 Finishing a Slide ........................................ 12
8.3.3 Framing a Slide ....................................... 12
8.3.4 Check Slide Height ................................. 13
8.4 Titles on Slides ........................................... 13
8.5 Slideformat Dependent Typesetting ................. 13
8.6 Cues in the Margin ....................................... 14

9 Style Features ................................. 14
  9.1 Redefine Chapter Code .......................... 14
  9.2 Pagestyle .......................................... 15
  9.3 Fonts .............................................. 15

10 Language Adjustments .......................... 15
  10.1 Example of Adjustements in Configuration File 15

11 Directory Localization .......................... 15

12 Read Configuration File ...................... 16

13 Process the Orientation Options ............. 16
  13.1 Page Orientation ............................... 16
  13.2 Slide Dimensions and Orientation .......... 17

14 Shutoff Titles if Necessary .................. 17

15 Page Dimensions ................................. 17
  15.1 Fullsize Slides .................................. 17
  15.2 Collection of Slides ............................ 18
  15.3 Slides plus Notes ............................... 18

16 Oval Macro's ....................................... 19

1 Usage

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

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

or

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

\end{portraitslide}

\begin{landscapeslide}

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

\begin{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.

\section{Identification}

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

\NeedsTeXFormat{LaTeX2e}[1995/12/01]

Announce class names and versions:

\ProvidesClass{slidenotes}[1998/04/23 3.27 Slides and notes]

\section{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.

\begin{itemize}
\item \texttt{\MakeUppercase{help}}
\item Options (first one is default):
\begin{itemize}
\item \texttt{\MakeUppercase{notes}} \, \texttt{\MakeUppercase{slides}} \, \texttt{\MakeUppercase{minis}}: type of production;
\item \texttt{\MakeUppercase{portrait}} \, \texttt{\MakeUppercase{landscape}}: slide orientation;
\item \texttt{\MakeUppercase{mixed}}, \texttt{\MakeUppercase{unmixed}}, \texttt{\MakeUppercase{anti}}: both, chosen, anti orientation;
\item \texttt{\MakeUppercase{center}}, \texttt{\MakeUppercase{nocenter}}, \texttt{\MakeUppercase{squeeze}}: vertical slide position;
\item \texttt{\MakeUppercase{rect}}, \texttt{\MakeUppercase{oval}}, \texttt{\MakeUppercase{rules}}, \texttt{\MakeUppercase{drules}}, \texttt{\MakeUppercase{noframe}}: type of slideframe;
\item \texttt{\MakeUppercase{newpage}}, \texttt{\MakeUppercase{nonewpage}}: yes/no start slide on new page;
\end{itemize}
\end{itemize}

3.2 Slides and/or Notes

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

\begin{itemize}
\item \texttt{\MakeUppercase{notes}} \texttt{true} \texttt{minisfalse} \texttt{slidesfalse}
\item \texttt{\MakeUppercase{slides}} \texttt{false} \texttt{minisfalse} \texttt{slidestrue}
\item \texttt{\MakeUppercase{minis}} \texttt{false} \texttt{ministrue} \texttt{slidestrue}
\end{itemize}

3.3 Portrait or Landscape Orientation

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

\begin{itemize}
\item \texttt{\MakeUppercase{landscape}} \texttt{true}
\item \texttt{\MakeUppercase{portrait}} \texttt{false}
\end{itemize}

\texttt{ifmixed} \, \texttt{ifanti} The next series of flags control the production of mixed portrait/landscape slides or runs with one orientation only.

With the \texttt{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 (\texttt{graphics.sty} or \texttt{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 \texttt{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 useful 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

\ifsansnotes
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\ifsansnotes
\sansnotesfalse
\DeclareOption{smallnotes}{\sansnotestrue}

### 3.7 Variation in Slideframe

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

\newcommand*\framevariant{}\endinput
\DeclareOption{noframe}{\renewcommand*\framevariant{\n}}\endinput
\DeclareOption{rules}{\renewcommand*\framevariant{\r}}\endinput
\DeclareOption{drules}{\renewcommand*\framevariant{\d}}\endinput
\DeclareOption{rect}{\renewcommand*\framevariant{\f}}\endinput
\DeclareOption{oval}{\renewcommand*\framevariant{o}}\endinput
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 \space PRODUCTION}
4 Some Useful Macros

We define some helpful 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

5 Dimensions

\slidewidth 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}

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

\newcommand*{\slidemagnification}{\magstep4}

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 \TeX’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 interferes 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

\everyslide
\everynote

\begin{document}
\maketitle
\everynote
\end{document}

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

\newcommand*\smallersizes{%

7 Note Environment

\newenvironment{note}{%\ignorespaces}{%\endlist%}

\ifnotes
\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).
8 Slide Environment

8.1 Slide Orientation

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}\iflandscape\putsilde{90}\else\putsilde{0}\fi\newenvironment{landscapeslide}{\landscapeslidetrue\iflandscape\else\refstepcounter{slidecounter}\let\@slide=\comment\let\@endslide=\endcomment\fi\@slide}{\@endslide\iflandscape\else\putslide{0}\fi}\newenvironment{slide}{\@slide}{\@endslide\putslide{0}}\endcomment
\fi\fi
Finally the case of separate slide orientation. A skipped slide must be counted always in \texttt{slidecounter} in order to keep the numbers straight and is also recorded in \texttt{missedslidecounter}; the \texttt{comment} environment is used to get rid of a skipped slide.

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

\textbf{8.2 Placing a Slide}

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.
We also 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 pagenumeration 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}\@tempdima\slidewidth\advance\@tempdima-\tw@\slideboxrule\advance\@tempdima-\tw@\slideboxsep12

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

\ifnotes\isemptyarg{#1}\%\{\addcontentsline{toc}{section}{Slide, theslidecounter}\%\{\addcontentsline{toc}{section}{#1}\%

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

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

8.3.2 Finishing a Slide

In the second part of slide typesetting we construct the slide.

First we do vertical centering with \vfil if required.

Here we close the catching box and check its height.

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.

8.3.3 Framing a Slide

Rectangular box around slide contents.

Oval box around slide contents.
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.

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

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.

\begin{verbatim}
\newcommand\slidetitle[1]{\@slidetitle{\slidetitlefont\slidetitle#1}}
\newcommand\slidesubtitle[1]{\@slidetitle{\slidesubtitlefont\slidetitle#1}}
\ifempty[#2]{\begin{center}\slidetitle#1\end{center}\vspace{\slidetitlesep}}\fi
\end{verbatim}
8.5 Slideformat Dependent Typesetting

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.

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

Material can be selected on the basis the slide’s format, portrait or landscape orientation with respectively:
\landscapemargenonly{...}
\portraitonly{...}

For example, where a list can be placed without any difficulty on a portrait slide, it may require the use of \multicols in landscape format. If problems arise, e.g. with the fixing of catcodes, then use the basic construction \iflandscapeslide .. \else ..\fi

8.6 Cues in the Margin

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\parbox[\marginparwidth]{\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

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

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

10 Language Adjustments

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.
\newcommand*{\Collection}{COLLEGE AANTEKENINGEN}
\newcommand*{\Slidenumber}{Nr.}
\newcommand*{\Slide}{Transparant}

11 Directory Localization

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{``:\ifemptyarg{\#2}{\#1}{\#2}}
\edef\DirectorySeparator{\expandafter\DirectorySeparator\CurrentDirectory\``M}

\CurrentDirectory\DirectorySeparator
Another macro delivers the last character of a string.

\LastChar\{\LastChar{1}\}%
\@ifemptyarg{#1}\{\@expandafter\@lastchar#1\"\M}\}
def\@lastchar#1\"M\{\@emptyarg{#2}\{\@lastchar#2\"M}\}

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.

\DirectoryName\{\DirectoryName{1}\}%
{\@ifemptyarg{#1}\DirectoryNameSeparator\relax#1\else
#1\DirectoryNameSeparator\fi}

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 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}}}
\Setfolder{\mainfolder}{}
\Setfolder{\commonfolder}{}
\Setfolder{\figuresfolder}{}

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 process of typesetting they have to be reduced by the magnification applied later on.

\divide\slidewidth by\slidemagnification
\multiply\slidewidth by\@m
\dividemakeheight by\slidemagnification
\multiply\makeheight 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 Shuto 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*@\makechapterhead[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}
\setlength\oddsidemargin{0pt}
15.2 Collection of Slides

15.3 Slides plus Notes

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

Initialize the font.
16 Oval Macro’s

Macro to translate dimensions

\#1\{dimen\} or \#1\{number\} \rightarrow \#1\{number\unitlength\}

\PackageInfo{obox}{1995/09/04 vs 1.2 Oval box change}

\newcommand{\@pickonedim}[1]{\ifnextchar(}{\@@pickonedim{#1}}{#1}}
\def\@@pickonedim#1(#2){#1{#2\unitlength}}

Redefinition of oval from the picture environment to
\oval[\text{position}](\text{width}, \text{height})[\text{optional lrtb=parts}]

position parameter determines 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

\gdef\oval{\ifnextchar[{}{\@oval}{\@oval[o]}}
\gdef\@oval[#1](#2,#3){\ifnextchar[{}{\@@oval[#1](#2,#3)}{\@@oval[#1](#2,#3)[]}}
\gdef\@@oval[#1](#2,#3)[#4]{\begingroup\boxmaxdepth \maxdimen
\@ovttrue \@ovbtrue \@ovltrue \@ovrtrue
\@tfor\reserved@a :=#4\do{\csname @ov\reserved@a false\endcsname}\
\@ovxx #2\unitlength \@ovyy #3\unitlength
\ifdim\the@ovalradius=\z@ \ifdim\@ovyy >\@ovxx \@ovxx \else \@ovxx \fi
\else \the@ovalradius \fi
\advance\@ovdx -\tw@ \divide\@ovdx \tw@
\advance\@ovdy -\tw@ \divide\@ovdy \tw@
\@circlefnt \setbox\@tempboxa
\hbox{\if@ovr \@ovvert32\kern -\@tempdima \fi
\if@ovl \kern \@ovxx \@ovvert01\kern -\@tempdima \kern -\@ovxx \fi
\if@ovt \@ovhorz \kern -\@ovxx \fi
\if@ovb \raise \@ovyy \@ovhorz \fi}}
\@put{-\@ovdx}{-\@ovdy}{\box\@tempboxa}
\endgroup}

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

\def\obox{\ifnextchar[{}{\@obox}{\@@obox[\][\]}}
\def\@obox[#1]{\ifnextchar[{}{\@@obox[#1]}{\@@obox[#1][\]}}

475 \{\@oobox[#1]\{\@oobox[#1][]\}\}
476 \def\@oobox[#1][#2]#3{%
477 \begingroup
478 \@ifemptyarg{#1}{}{ovalradius{#1}%%
479 \setbox\z@\hbox{\advance\fboxsep\@wholewidth \fboxrule\z@ \fbox{#3}%%
480 \dimen0\ht\z@ \advance\dimen0\dp\z@ \#2\let\unitlength\@empty
481 \leavevmode\oval[n](\wd\z@,\dimen0)\box\z@
482 \endgroup}
483 \def\ovalradius{%\@pickonedim\@ovalradius}
484 \def\@ovalradius#1\@tempdima#1\relax
485 \edef\the@ovalradius{#1\@tempdimb}\ignorespaces
486 \def\ovalfraction#1{%
487 \edef\the@ovalradius{#1\noexpand\@tempdimb}\ignorespaces
488 \ovalradius{Opt} \% initialisation
489 \endoobox}