

elpres — electronic presentations with (pdf)L^AT_EX

Volker Kiefel*

v0.6

August 19, 2020 — [update of this manual: September 03, 2020](#)

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 2 |
| 2 | Installation | 2 |
| 3 | Usage | 3 |
| 3.1 | A minimalistic example presentation | 3 |
| 3.2 | Essential elements of an elpres presentation | 4 |
| 3.3 | Alternatives to T _E X’s justified paragraphs | 5 |
| 3.4 | Vertically compressed lists | 5 |
| 3.5 | Slide layout: changes in slide title, footer, page numbers | 6 |
| 4 | Enhancements to elpres | 6 |
| 4.1 | Use of named colors of xcolor | 6 |
| 4.2 | Include graphics files | 6 |
| 4.3 | Arrange text and pictures in two (or more) columns | 7 |
| 4.4 | Incremental slides (overlays) | 7 |
| 4.5 | Navigation symbols | 9 |
| 4.6 | Run multimedia content from a presentation | 9 |
| 4.7 | Prepare a “handout” from a presentation | 10 |
| 4.7.1 | L ^A T _E X article document with a series of single-page pdf-files included | 10 |
| 4.7.2 | Convert presentation pdf-file directly into handout file with pdfjam | 11 |
| 4.8 | Create presentations with hypertext elements | 11 |
| 4.9 | Fill background of a presentation with bitmaps | 12 |
| 4.9.1 | Wallpaper package | 12 |
| 4.9.2 | Eso-pic package | 12 |
| 4.10 | Presentation schemes | 13 |
| 5 | Recipes for use of elpres | 13 |
| 5.1 | Colored text boxes | 13 |

*volker dot kiefel at freenet dot de, <http://vkiefel.de/elpres.html>

| | | |
|----------|---|-----------|
| 5.1.1 | Text boxes with <code>\colorbox</code> , <code>\fcolorbox</code> | 14 |
| 5.1.2 | Text boxes with the <code>tcolorbox</code> package | 14 |
| 5.2 | Inactivate <code>overlays</code> commands | 14 |
| 5.3 | Convert an <code>elpres</code> -presentation (PDF) into an Impress (LibreOffice) or Powerpoint presentation | 15 |
| 5.4 | The <code>pdfpc-viewer</code> : add notes | 16 |
| 6 | License | 16 |
| 7 | Version history | 16 |

A note on this updated version of the `elpres` manual

In this updated version of the manual paragraphs or sentences with changes (compared to the “official” manual for v0.6) are marked `<changed>` changed content `<changed` and new text is marked `<inserted>` new content `<inserted`.

1 Introduction

The `elpres` class is intended to be used for presentations on a computer screen, a beamer or a projector. It is derived from L^AT_EX’s `article` class and may be used with L^AT_EX, pdfL^AT_EX (and LuaL^AT_EX). The default “virtual paper size” of document pages produced by this class: width=128mm, height=96mm corresponds to a 4:3 (width:height) aspect ratio. Other aspect ratios for widescreen monitors may be selected by class options. The `elpres` class requires that the `ifthen`, `fancyhdr`, `hyperref`, `graphicx`, `xcolor` and `geometry` packages are available on the T_EX system: these packages are loaded automatically by the `elpres` class. Enhancements to `elpres` are easily made available by other packages, these include overlay support for incremental slides (package `overlays`) and slides with a background from a bitmap (`wallpaper`, `eso-pic` packages). Predefined color/layout schemes for `elpres` presentations can be acitivated with `\usepackage` (details can be found in in section 4.10)

This manual is intended to support the user with “recipes”. Use of `elpres` with its default settings should be simple, additional aspects including overlay functions, use of colors, graphics files, “handout documents” are described in section 4. Many code snippets have been included in this manual. They can be used in users’ presentation files.¹

`<inserted>` Some extensions described in this manual work only with pdf-files which should preferably be compiled with pdfL^AT_EX or LuaL^AT_EX. `<inserted`

2 Installation

If the `elpres` package has already been installed with the T_EX-system nothing needs to be done. If an updated version shall be installed, `elpres.cls`, the `.sty`, `.png` and `.eps` files should be copied

¹To make copying easier, this manual is also available as `.html` document on the `elpres` website (<http://www.vkiefel.de/elpres.html>)

into a directory, where your T_EX-system can find it: if an old `elpres` version of the existing T_EX system shall be replaced by the current version, please copy these into the appropriate position in the “local” `texmf` directory tree (e.g. `~/texmf-local/tex/latex/elpres`) and this manual (`elpres-manual.pdf`) to `~/texmf-local/doc/latex/elpres`. Then, the files database should be updated by entering (in case of a T_EX Live installation) the command:

```
mktxlsr ~/texmf-local
```

or you may enter `texhash ~/texmf-local`². Other T_EX-systems e.g. MikT_EX have their own package updating mechanisms. The `elpres` manual file should be accessible with the command “`texdoc elpres`”. If this still calls the old version of the manual, the command “`texdoc -l elpres`” will prompt you to select either the old or the new version of the manual.

3 Usage

The class is used with

```
\documentclass[options]{elpres}
```

Elpres-specific options: for font selection `tmrfont` (Times Roman), `helvetfont` (Helvetica), `cmfont` (Computer Modern), `sansfont` (Sans Serif: default) and `nofonts` (no font selection, intended for use of `elpres` with LuaL^AT_EX with its own font selection mechanisms) are used. Options for different screen aspect ratios: `4x3` (default), `16x9`, `16x10`. The option `navisymb` adds symbols for navigation to the presentation. Options of the `article` class are also available for `elpres` presentations, e.g. `10pt`, `11pt`, `12pt` for selection of font size.

Elpres-specific commands: `\distance` vertically adjusts text on a slide. The `\auvimm`³ command inserts a link to an external audio or video file. Two commands: `\fromlinktext` and `\totargettext` can be used for internal links within a presentation. With `\slidetitlecolor`, the text color of slides (`psli`) can be changed.

Elpres-specific environments: the environments `psli` and `rsli` define “slides” in `elpres`; `citemize`, `cenumerate` and `cdescription` provide vertically compressed lists.

3.1 A minimalistic example presentation

The following code may be used for your first “experiments” by adding features described in latex sections.

```
\documentclass[12pt,pdftex,4x3]{elpres}
\usepackage[latin1]{inputenc}
\usepackage[document]{ragged2e}

% \usepackage{elpreswhitebluescheme}
% \usepackage{elpreswhitetealscheme}
% \usepackage{elpresbluelightgrayscheme}
```

²The location for the “local”, manually updated class and package files may be different, depending on the T_EX system and the OS.

³for `audio video media`

```

% \usepackage{elpresgrayscheme}
% \usepackage{elpreswhiteredscheme}
\usepackage{elpreswhitetealscheme}

\begin{document}
\begin{titlepage}
\centering
\distance{1}
{
\Huge \bfseries \textcolor{eptitlecolor}{Title of the presentation} \par
}
\vspace{1.3ex} \large
Author\[\[2ex]Institution
\distance{2}
\end{titlepage}

\begin{psli}[Title of Page]
The first page: \texttt{psli} environment

\begin{itemize}
\item first line in an itemized list
\item second line in an itemized list
\end{itemize}
\end{psli}

\begin{rsli}
The second page: \texttt{rsli} environment

\begin{enumerate}
\item no title
\item not centered vertically
\end{enumerate}

\end{rsli}
\end{document}

```

You may copy the code examples from this manual more easily from the .html version of this manual available at <http://vkiefel.de/elpres/elpres-manual.html>.

3.2 Essential elements of an elpres presentation

The **title page** slide can be created with the `titlepage` environment (or `rsli`, see below), L^AT_EX's `\maketitle` command is not available. **Slides** may be created with the `psli`-environment⁴, you may add the title of the slide with the optional parameter. The contents of the slide are centered vertically.

⁴`psli`: plain slide

Another environment generating a **slide** is `rsli`⁵: slides are written without title, contents are not vertically centered.

The `\distance{number}` command allows to introduce vertical space into slides constructed with the `rsli` and `titlepage` environments. You should use pairs of `\distance{}` commands with numbers indicating the relative height of empty space, see the `titlepage` in the example above.

The use of footnotes on slides is often problematic, if they cannot be avoided, the `footmisc` package is recommended: the `perpage` option resets numbering for each new slide. For a presentation, the `symbol` option allows to use symbols instead of numbers. After inserting a new footnote, numbers or symbols are correctly inserted only after a second run of `LATEX`.

3.3 Alternatives to T_EX's justified paragraphs

By default, `LATEX` produces justified paragraphs with lines of equal length, this may often not be appropriate for the usually very short lines of text in presentations. The `LATEX` `\raggedright` command has its own deficiencies: by inhibiting hyphenation in texts with rather short lines, the right margin will often look too ragged. A solution is to use the `\RaggedRight` command of the `ragged2e` package.

3.4 Vertically compressed lists

As the spaces between lines may be too great with the `itemize` environment, the `elpres` package provides a “vertically compressed” `citemize`-environment:

```
\begin{citemize}
  \item one
  \item two
\end{citemize}
```

Similarly, a `cenumerate` and a `cdescription` environment may be used.

Another solution for the customization of `itemize` environments is given by the `enumitem` package. Therefore

```
\usepackage{enumitem}
```

should be added to the preamble, and a comma-separated list of parameters parameters can be added in the format:

```
\begin{itemize}[parameter-list]
  ...
\end{itemize}
```

The “vertically compressed” list can then be obtained with

```
\begin{itemize}[nosep]
  \item one
  \item two
\end{itemize}
```

⁵`rsli`: raw slide

Similarly, the `enumitem` package is also able to modify the `enumerate` and `description` environments.

3.5 Slide layout: changes in slide title, footer, page numbers

By default, the **text color of the titles** of `psli`-slides is black, it may be changed by redefining the “value” of the `\slidetitlecolor{}` command like

```
\slidetitlecolor{blue}
```

in the preamble. Of course, you may also use the named colors of the `xcolor` package (see section 4.1).

By default the **page number** appears at the right bottom of the slide (in the “footer” in the terminology of the `fancyhdr` package) this position can be addressed by the `\rfoot{}` command: the `\rfoot{}`, `\cfoot{}` and `\lfoot{}` commands can be used by the author of a presentation.

However, access to the header fields defined by `fancyhdr` is blocked⁶. If you wish to change the position of the page number in the footer, you can overwrite the default page number with an “empty” `\rfoot{}` command and put it to the center or the left margin of the footer (`\cfoot{\footnotesize \thepage}` or `\lfoot{\footnotesize \thepage}`). These “footer fields” are also suitable to enter a **logo** visible on all pages (in form of a graphics file using `\includegraphics[]{}`) or text with the name of the speaker’s institution.

The user may also use predefined **presentation schemes** with defined colors and layout elements (e.g. symbols used in list environments), details are described in section 4.10.

4 Enhancements to `elpres`

4.1 Use of named colors of `xcolor`

The `elpres` class automatically loads the `xcolor` package. Color related commands may therefore be used with names defined by `colorx`. They are however only accessible in groups (`dvipsnames`, `svgnames`, `x11names`). As an example, the color `Indigo` is available in `svgnames`. If you wish to use it you will have to enter `svgnames` as `elpres` class option:

```
\documentclass[11pt,16x9,svgnames]{elpres}
```

This option is then automatically “handed over” to the `xcolor` package. The complete lists of named colors are found in the `xcolor` manual.

4.2 Include graphics files

Graphics files/pictures can be included with the `includegraphics`-command of the `graphicx`-package. Please be aware that the dimensions of the pages are 128mm x 96mm and therefore included graphics are scaled appropriately. A safe way to generate a page with a picture could be (with `pict.png` as the name of the graphics file):

⁶If you enter `\rhead`, `\chead` or `\lhead`, `elpres` will issue an error message. Use of the header fields will generate unwanted effects on page layout due to `fancyhdr`.

```

\usepackage[pdftex]{graphicx} % (in preamble)
...
\begin{rslip}
  \centering
  \distance{1}
  \includegraphics[width=0.9\textwidth,%
                  height=0.9\textheight,%
                  keepaspectratio=true]{pict.png}
  \distance{1}
\end{rslip}

```

The `\includegraphics[]{}` command requires to select the correct device driver related option (e.g. `pdftex` or `dvips`) (documentclass).

4.3 Arrange text and pictures in two (or more) columns

Text and graphics may be arranged in two or more columns with `minipage` environments:

```

\begin{minipage}[b][0.8\textheight][t]{0.5\textwidth}
  \colorbox{white}{%
    \includegraphics[width=0.9\textwidth]{graphics-file.png}}
\end{minipage}
\begin{minipage}[b][0.8\textheight][t]{0.48\textwidth}
\footnotesize
\begin{citemize}
  \item ...
  \item ...
  ...
\end{citemize}
\end{minipage}

```

Details on the `minipage` environment may be found in the \LaTeX documentation.

4.4 Incremental slides (overlays)

If the contents of slides are to be made visible step by step this can be achieved by a series of output PDF or (PS) files (carrying the same page number) usually called *overlays*. It may also be of interest to change a highlighting color in a series of overlays. This is most easily done by using the excellent `overlays` package written by Andreas Nolda.

To generate a series of four overlays sequentially showing four lines of a list:

1. load the `overlays` package in the preamble
2. put a `psli` or `rslip` slide environment into an `overlays` (or `fragileoverlays`) environment
3. enter the number of overlays as the first parameter to the `overlays` environment
4. enter text contents with the `visible` command with the range of overlays showing this text content

A simple example:

```
% to be added in preamble
\usepackage{overlays}
...
\begin{overlays}{4}
\begin{psli}[Title of slide]
\begin{itemize}
  \visible{1-4}{\item first item of list}
  \visible{2-4}{\item second list item}
  \visible{3-4}{\item 3rd list item}
  \visible{4}{\item final list item}
\end{itemize}
\end{psli}
\end{overlays}
...
```

The following example uses the `alert` command to highlight lines sequentially:

```
\begin{overlays}{4}
\begin{psli}[Title of slide]
\begin{itemize}
  \alert{1}{\item first item of list}
  \alert{2}{\item second list item}
  \alert{3}{\item 3rd list item}
  \alert{4}{\item final list item}
\end{itemize}
\end{psli}
\end{overlays}
```

The last example shows short text fragments which are shown sequentially using the `only` command:

```
\begin{overlays}{4}
\begin{rsli}
  \only{1}{a short text, which will be replaced \ldots }
  \only{2}{\dots by a second \ldots}
  \only{3}{\ldots and a third \ldots}
  \only{4}{\ldots and a final text.}
\end{rsli}
\end{overlays}
```

The “hidden” text contents are written by `overlays` in the same color as the background, default is white. If you use a different background color, you have to change the color of the hidden text as well by assigning the background color to the color name `background` (understood by the `overlays` package). In the following example you define a light yellow as background:

```
% (in the preamble)
\definecolor{myyellow}{rgb}{0.96,0.98,0.72} % define color
\definecolor{background}{named}{myyellow} % color assigned to
                                         % hidden text
```



```
\pagecolor{myyellow} % color of slide background
```

If you use one of the *presentation schemes* described in section 4.10, the necessary adjustments for the background color will be made automatically. For more details on overlays, see the documentation of the package.

4.5 Navigation symbols

With the `navisymb` option of `elpres` a panel with navigation symbols appears in the right lower corner of the presentation:

```
<< < > >> ← → [n]
```

These commands (`<<`: jump to the first page, `<`: go to the previous page, `>`: go to the next page, `>>`: jump to the last page, `←`: go back in history, `→`: go forward in history, `[n]`: prompt for a page number) work with Adobe Acrobat Reader (Windows), and (with the exception of `←`, `→` and `[n]`) with `evince` (Linux). Some of these functions also work in presentation-mode of the “internal” PDF-viewer in recent versions of the Firefox-browser (Linux, Windows).

4.6 Run multimedia content from a presentation

`inserted>` In this section inclusion of video and audio files into a presentation will be described. With the `\auvimm{}` command described below you will be able to launch an external application for playing the video or audio file. This will work in many situations for video and audio files on Windows (Adobe Acrobat) and Linux (e.g. using `evince` and `Impressive`) systems in fullscreen mode. Under certain conditions it is possible to define a “poster area” on a slide and the pdf viewer starts (after clicking with the mouse pointer into this area) the presentation of the video within this frame. The advantage of this technique is that it avoids that window with the external application is started. On Linux systems the `pdfpc` viewer allows this, a detailed description is shown below. In both situations, however, the multimedia file is not incorporated into the pdf-file and must be available on the computer at the time of the presentation, ideally in the same directory as the .pdf-file. `<inserted`

The command `\auvimm{text}{media-file}` inserts `text` with a link (“poster”), which allows to start the `inserted>` external default `<inserted` application for an audio or video media file (`media-file`). This worked flawlessly on the computers of the author of this manual with Adobe Acrobat Reader (Windows) and with `evince` and `Impress!ve`⁷ (Linux).⁸ An example for using this command:

```
\auvimm{[sound-file]}{./audiofile.mp3}
```

This produces the (text-based) “button” or “poster” `[sound-file]`. You may also insert an image file as a poster with a graphics file using `\includegraphics`:

```
\auvimm{\includegraphics{/path/to/poster.png}}{./audiofile.mp3}
```

⁷Can be started from the terminal: `impressive`

⁸The `\auvimm` command uses the `\href` command of the `hyperref` package

[changed](#)> If you hit the poster area with the mouse pointer, it will launch the default application for audio files in a separate window. This should be possible even if the pdf-viewer works in fullscreen-mode at the time of presentation. If this does not work, it may be necessary to insert as “launch” command: [<changed](#)

```
\auvimm{[video-file]}{run:./videofile.mp4}
```

This approach requires that you keep the presentation file together with the multimedia files in the same folder, **also at the time of the presentation**. On an unknown computer system (where you have to show your presentation) you should test this aspect of your presentation. Sometimes a call to an external program might be blocked due security reasons in Acrobat Reader.

This `\auvimm`-command should be considered experimental. Its behavior depends on the pdf-viewer and the configuration of the OS.

[inserted](#)> On Linux systems, the pdfpc pdf-viewer⁹ allows presentation of a video embedded into a “poster area”¹⁰ on a slide without starting the external viewer in a separate window. This is supported with the `\pdfpcmovie{}{}-command` from the `pdfpc-movie`-package and with `\auvimm{}{}-command` using the “`run:./video-file`” launch-command. An example with `\auvimm{}{}-command`:

```
\auvimm{\includegraphics[width=0.4\textwidth]{poster.png}}{run:./video.mp4}
```

This pdfpc-specific extension seems to work only for video files, but not for audio files. A possible solution is to convert an audio file into a “video format”, e. g. with the `ffmpeg` program.

[<inserted](#)

4.7 Prepare a “handout” from a presentation

In advance of a lecture it is often expected that you prepare a “handout” from your presentation with more than one pages on a printed page. Therefore you have several options, two of them are described here.

4.7.1 L^AT_EX article document with a series of single-page pdf-files included

One solution to this problem is (1) to generate a series of single-page pdf-files and (2) to create a pdfL^AT_EX document with the single pages included.

(1) can be done with `gs` (the Ghostscript program):

```
gs -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=page_%03d.pdf in.pdf
```

with `in.pdf` as the initial presentation. The syntax of `gs` is described in detail on the ghostscript website¹¹. On Windows systems the name of the ghostscript command may be `gswin32c.exe` or `gswin64c.exe`. An alternative to `gs` is the `pdftk` tool:

```
pdftk in.pdf burst output page_%03d.pdf
```

The syntax of `pdftk` is explained in the documentation (`pdftk --help`).

(2) an example for a L^AT_EX document which can serve as handout:

⁹<https://pdfpc.github.io/>: pdfpc — A presenter console with multi-monitor support for PDF files

¹⁰e. g. defined by a bitmap image with the same aspect ratio as the video

¹¹<https://www.ghostscript.com/doc/current/Use.htm>

```

\documentclass[12pt]{article}
\usepackage[pdftex]{graphicx}
\usepackage[latin1]{inputenc}
\usepackage[a4paper,hmargin=2.4cm,top=24mm,bottom=28mm]{geometry}
\newcommand{\PictScaleFact}{0.45}

\begin{document}
\centering
\fbbox{\includegraphics[width=\PictScaleFact\textwidth]{page_001.pdf}}
\hspace{5mm}
\fbbox{\includegraphics[width=\PictScaleFact\textwidth]{page_002.pdf}}
\\ [2ex]
\fbbox{\includegraphics[width=\PictScaleFact\textwidth]{page_003.pdf}}
\hspace{5mm}
% ...
\fbbox{\includegraphics[width=\PictScaleFact\textwidth]{page_008.pdf}}
% ...
\end{document}

```

4.7.2 Convert presentation pdf-file directly into handout file with pdfjam

With the following command:

```
pdfjam --nup 2x4 --frame true --scale 0.9 -o new.pdf in.pdf '1-4,6-22'
```

pdfjam¹² creates a “handout” PDF document (new.pdf) fom in.pdf with the slides nr. 1-4,6-22 arranged in two columns and four rows. With the additional option “--frame true”, pdfjam draws a box around each slide. More details can be found in the pdfjam man page and the project website¹³.

4.8 Create presentations with hypertext elements

You may use two commands `\fromlinktext{}{}` and `\totargettext{}{}` for “hypertext features”. As an example: as you normally will not insert `\section{}-like` commands and therefore do not generate a “table of contents”-like page or a menu in `elpres`-presentations, you are able define links with:

```
\totargettext{text}{link-label}
```

which can be addressed by

```
\fromlinktext{text}{link-label}
```

If you use these commands to generate a **menu page** (as a hyperlinked table of contents) with items pointing to specific slides: `link-label` acts as label, `text` in the `\fromlinktext` command is converted to a link (e. g. on a menu-page). If you click on this text “button”, you jump to

¹²which regerettably is only available on Linux or other Unix-like systems

¹³<https://github.com/DavidFirth/pdfjam/blob/master/README.md>

the text labelled with `\totargettext`; `link-label` has to be identical in a `\fromlinktext` and `\totargettext` pair, text in the link and the target, of course need not to be identical.

The default for the link border color is red, it may be changed with `\hypersetup{}`, a command from the `hyperref` package¹⁴. The option for the color of the frame around the link is `linkbordercolor`, colors must be defined with three figures `[0..1]` according to the RGB color model:

```
\hypersetup{linkbordercolor={0.6 0.6 0.6}}
```

This defines a light gray color for the link border, more details on `\hypersetup{}` are described in the documentation of the `hyperref` package.

4.9 Fill background of a presentation with bitmaps

4.9.1 Wallpaper package

To create a slide background with a graphical wallpaper background using bitmap files you may use the `wallpaper` package¹⁵. Load the `wallpaper` package with

```
\usepackage{wallpaper}
```

in the preamble. In order to generate a background based on bitmap file `background.png`, enter

```
\CenterWallPaper{1}{background.png}
```

before the contents of the presentation¹⁶. This works best with bitmaps with an appropriate aspect ratio, in the case of an 4x3 screen format a bitmap picture of 640x480 pixel would fit perfectly. Moreover bitmap files may be used as tiles as described in the `wallpaper` documentation like

```
\TileSquareWallPaper{4}{background.png}
```

More details on this topic may be found in the `wallpaper` documentation.

4.9.2 Eso-pic package

Another package which allows you to paint the background with a picture is `eso-pic`¹⁷:

```
\usepackage{eso-pic}
```

```
...
```

```
\AddToShipoutPicture{
\includegraphics[height=\paperheight]{background.png}
}
```

¹⁴both, `\totargettext` and `\fromlinktext` are redefinitions of commands from the `hyperref` package

¹⁵written by Michael H.F. Wilkinson and available on CTAN

¹⁶i. e. following `\begin{document}`

¹⁷written by Rolf Niepraschk and available on CTAN

`\AddToShipoutPicture{}` puts the picture on every page, `\AddToShipoutPicture*{}` puts it on to the current page, `\ClearShipoutPicture` clears the background beginning with the current page. Details of `eso-pic`'s commands can be found in the documentation.

4.10 Presentation schemes

`Elpres` provides predefined **presentation schemes**: color and layout schemes which are applied to a presentation with `usepackage{}`. Their naming follows the convention `elpres...scheme`, an example is `elpreswhiteredscheme`.¹⁸ These `elpres`-schemes also support incremental slides with the `overlays` package. Moreover, these schemes define color names which may be used by the author of a presentation:¹⁹

`eptextcolor` “normal text color” (default values for documents without presentation scheme included in `elpres.cls`: black)

`ephighlightcolor` color, which may be used for highlighting text (default: dark red)

`eptitlecolor` color of the title of slides, may be used to highlight the title in the title slide (default: dark red)

Presentation schemes available in the current version of `elpres`:

`elpresbluelightgrayscheme` dark blue background of slides; `eptextcolor`: light gray; `eptitlecolor`: turquoise; `ephighlightcolor`: yellow; symbols in *itemize* environment: pale blue bullets; highlighted elements in *enumerate* and *description* environments: pale blue

`elpresgrayscheme` gray background of slides; `eptextcolor`: dark gray; `eptitlecolor`: dark green; `ephighlightcolor`: dark red; symbols in *itemize* environment: gray bullets; highlighted elements in *enumerate* and *description* environments: dark green

`elpreswhitebluescheme` white background of slides; `eptextcolor`: blue-black; `eptitlecolor`: blue; `ephighlightcolor`: dark red; symbols in *itemize* environment: blue bullets; highlighted elements in *enumerate* and *description* environments: blue

`elpreswhiteredscheme` white background of slides; `eptextcolor`: red/brown-black; `eptitlecolor`: dark red; `ephighlightcolor`: dark blue; symbols in *itemize* environment: red bullets; highlighted elements in *enumerate* and *description* environments: red

`elpreswhitetealscheme` white background of slides; `eptextcolor`: black-teal; `eptitlecolor`: teal; `ephighlightcolor`: dark red; symbols in *itemize* environment: blue bullets; highlighted elements in *enumerate* and *description* environments: dark teal

In all of these schemes, the colors of borders for internal and external links have been adjusted.

5 Recipes for use of `elpres`

5.1 Colored text boxes

Colored text boxes often help the presentation author to highlight small text fragments and to make slides more “readable” and more attractive.

¹⁸the name of the corresponding style or package file is `elpreswhiteredscheme.sty`

¹⁹Color names for use by the presentation author start with `ep...`

5.1.1 Text boxes with `\colorbox`, `\fcolorbox`

A simple method for colored text boxes uses the `minipage` environment and the `colorbox` or `fcolorbox` commands (`xcolor` package):

```
\begin{psli}[Slide with a textbox]
\begin{center}
\colorbox{eptitlecolor}{\begin{minipage}{0.85\textwidth}
\centering \color{white} \vspace{2ex}
{\Large
Text centered in a colorbox
} \par
\vspace{2ex}
\end{minipage}}
\end{center}
\end{psli}
```

5.1.2 Text boxes with the `tcolorbox` package

The `tcolorbox` package provides great support for colored boxes. A minimal example:

```
\begin{psli}[Slide with a textbox]
\begin{tcolorbox}[colframe=eptitlecolor!95!black]
\textcolor{eptitlecolor}{This is a textbox generated with the
\texttt{tcolorbox} package}
\tcblower This the lower part of the tcolorbox
\end{tcolorbox}
\end{psli}
```

An example with a titled colored textbox:

```
\begin{psli}[Slide with a textbox]
\begin{tcolorbox}[colback=eptitlecolor!10!white,%
colframe=eptitlecolor!95!black,%
title=Heading of a textbox]
This is another \textbf{tcolorbox}.
\tcblower
Here, the lower part of the box.
\end{tcolorbox}
\end{psli}
```

The manual of the `tcolorbox` provides perfect instruction for the use of this package.

5.2 Inactivate overlays commands

If a presentation has been prepared with overlay slides (using the `overlays`) package, the \LaTeX source code will contain `\visible`, `\only` and `\alert` commands and `overlays` and `fragileoverlays` environments.

Sometimes, it is desirable to generate a version of such a presentation without overlays effects, for example for preparation of a printed handout (see section 4.7). The effect of these overlay-specific instructions can be inactivated with this series of commands:

```
\renewenvironment{overlays}[1]{}{}%
\renewenvironment{fragileoverlays}[1]{}{}%
\renewcommand{\visible}[2]{#2}%
\renewcommand{\only}[2]{#2}%
\renewcommand{\alert}[2]{#2}%
```

which should be copied or entered in the preamble following `\usepackage{overlays}`.

5.3 Convert an elpres-presentation (PDF) into an Impress (LibreOffice) or Powerpoint presentation

Sometimes it is unclear, if organizers of a (scientific) meeting allow presentations with pdf-files but insist on files in Powerpoint format. This is an unpleasant situation, but sometimes it is better to be prepared.

There is no elegant way for a reliable solution. First, you may use one of the free or commercial online conversion tools (intended to convert .pdf into .ppt or .pptx-files). Results are often not satisfactory.

An alternative, more reliable way preferred by the author of this manual (although a little awkward) is:

1. convert the pdf-file of the presentation into a series of bitmap graphics files, (e. g. in .png format)
2. import these image .png files into pages of an empty Impress (LibreOffice or OpenOffice) presentation and save the presentation in .ppt or .pptx-Format

Details on step 1 — generate .png images:

Please copy the following command (ignoring the line-break) into a single line at the terminal command prompt or copy them into one line of a short shell script (Windows: a .bat file) and “run” or execute the shell script:

```
gs -dNOPAUSE -dBATCH -sDEVICE=png256 -r300
-sOutputFile=page_%03d.png input.pdf
```

Please do not forget to adjust the name of the input file `input.pdf`. This command generates a series of .png-files `page_001.png`, `page_002.png` ... `page_0nn.png`. On Windows systems the name of the ghostscript command is `gswin32c` or `gswin64c`.

If you wish a better resolution of the .png change `-r300` to `-r600` or even `-r1200`:

```
gs -dNOPAUSE -dBATCH -sDEVICE=png256 -r600
-dTextAlphaBits=4 -sOutputFile=page_%03d.png input.pdf
```

The additional option `-dTextAlphaBits=4` improves font antialiasing.²⁰

Details on step 2 — Import .png images:

²⁰Details can be found in the documentation of gs <https://www.ghostscript.com/doc/current/Use.htm>

Open an empty presentation with LibreOffice Impress, adjust the appropriate slide format (e. g. 4:3), import the first .png file with *Insert / Image*, then adjust the size of the included image with *Format / Object and Shape / Position and Size (F4)* in the *Position and Size* dialog box please enter: *Position X: 0.0cm, Position Y: 0.0cm, Width: 28.0cm*, activate the checkbox *Keep ratio*.²¹. The presentation can now be saved in one of the Microsoft Powerpoint formats .ppt or .pptx.

inserted>

5.4 The pdfpc-viewer: add notes

The pdf-viewer pdfpc for Linux²² (see section 4.6) is a comfortable program which allows output of presentations to two monitors or a monitor and a beamer/projector. The normal presentation window is assigned to the beamer/projector, the other output for the presenter shows the actual slide, the next slide and an area for notes. As you are viewing a certain slide, press **n** to start editing a note. The editing mode is stopped with **Esc**. The text of these notes is stored in the file `filename.pdfpc`. Further information can be found in the `man` page.

<inserted

6 License

This class is distributed under the *L^AT_EX Project Public License* (LPPL) which may be downloaded from <http://www.latex-project.org/lppl.txt>. No warranty is provided for this work (as stated in the LPPL).

7 Version history

v0.1 (19.6.2004): initial version. **v0.2** (1.9.2004): page numbers now changed to footnotesize, left and right margins slightly changed, ‘cenumerate’ and ‘cdescription’ environments added. **v0.2a** (19.9.2004): Section “License” added to the documentation. **v0.2b** (17.10.2004): Documentation completed: description of the `\distance{}` command included. **v0.2c** (28.11.2004): Documentation completed (section 4.8 added). **v0.2d** (25.12.2004): Documentation completed (section 4.9 added). **v0.2e** (15.04.2005): Documentation completed (sections 4.9.2 and 4.7 added). **v0.3** (12.08.2005): new (class) options for font selection: `tmrfont` (Times Roman), `helvetfont` (Helvetica), `cmfont` (Computer Modern), `sansfont` (Sans Serif: default). Documentation updated, sections 4.2 and 4.3 added. **v0.4** (20.01.2018): New class options for different screen aspect ratios 4x3, 16x9, 16x10; “compressed” list environments modified; documentation completed: packages for use with `elpres`: `enumitem` (alternative list environments), `overlays` (overlay support: incremental slides); section 4.7 was completely rewritten. **v0.4a** (24.01.2018): Documentation completed. **v0.5** (12.07.2020): New class options: `nofonts` (no font selection) and `navisymb` (inserts a panel of symbols for navigation), new commands: `auvimm` (generates a link to external audio, video files), `\fromlinktext` and `\totargettext` (create links within a presentation). Documentation has been partly rewritten. **v0.6** (19.08.2020): Access to the `fancyhdr`

²¹Menu structure of LibreOffice v6.3

²²<https://pdfpc.github.io/>

“header fields” (`\lhead{}`, `\chead{}` and `\rhead{}`) is now explicitly blocked. New command `\slidetitlecolor{}` for the text color of titles in `psli`-slides. New: style files for presentation schemes added. Documentation has been significantly completed: sections 3.5, 4.1, 4.10 and 5 added, section 2 on installation rewritten.