# elpres — electronic presentations with (pdf/Lua)LaTeX

Volker Kiefel*

v1.1

May 29, 2024

## Contents

---

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

# 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 LaTeX's `article` class and may be used with LaTeX, pdfLaTeX (and LuaLaTeX). The default "virtual paper size" of slides produced by this class: width=128mm, height=96mm which 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 TeX 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.

Some extensions described in this manual work only with `pdf`-files which should preferrably be compiled with pdfLaTeX or LuaLaTeX.

# 2  Installation

If the `elpres` package has already been installed with the TeX-system nothing needs to be done. If an updated version shall be installed (or if the preinstalled `elpres` version does not work properly[1]), `elpres.cls`, all `*.sty`, `*.png` and `*.eps` files should be copied into a directory, where the TeX-system can find it: if an old `elpres` version of the existing TeX system shall be replaced by the current version, please copy these into the appropriate position in the "local" `texmf`

---

[1]for details see the description on checking the `elpres`-installation in section 3.3

directory tree (e. g. `~/texmf/tex/latex/elpres`) and this manual (`elpres-manual.pdf`) to `~/texmf/doc/latex/elpres`, where `/texmf` may be `/texmf-local` or an analogous directory.

Then, the files database should be updated by entering (in case of a TeX Live installation) the command:

    mktexlsr ~/texmf-local

or you may enter `texhash ~/texmf-local`[2]. Other TeX-distributions e. g. MiKTeX 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

### 3.1   Class options, commands and environments

The class is used with

    \documentclass[options]{elpres}

Slides are written with the `psli` and `rsli` environments. This section lists all `elpres`-specific options, commands and environments.

**Elpres-specific options**

For font selection (LaTeX, pdfLaTeX): "`tmrfont`" (Times Roman), "`helvetfont`" (Helvetica), "`cmfont`" (Computer Modern) and "`sansfont`" (Sans Serif: default) may be used. The option "`nofonts`" (no font selection) is intended for use of `elpres` with LuaLaTeX with its own font selection mechanisms, an example is described in section 3.2.

Options for different screen aspect ratios are "`4x3`" (default),[3] "`16x9`",[4] "`16x10`".[5] The option "`bulletsymb`" selects `$\bullet$` symbols (closed circles) instead of the default `itemize`-environment symbols in all four levels, "`ballsymb`" is effective together with the presentation schemes (section 4.10) and inserts small images of colored *balls* as alternative symbols for the `itemize` environment. The option "`navisymb`" adds a small field with symbols for navigation to the right bottom area of slides (section 4.5). Options of the `article` class are also available for `elpres` presentations, e. g. `10pt`, `11pt`, `12pt` for selection of font size.

**Elpres-specific commands**

The command "`\distance`" supports the user to vertically adjust lines and/or pragaraphs on a slide. (details are described in section section 3.4). The "`\auvimm`"[6] command inserts a link to an external audio or video file (section 4.6). Two commands: "`\fromlinktext`" and "`\totargettext`" can be used for internal links within a presentation (section 4.8). With "`\slidetitlecolor`", the text color of the title of slides (`psli`) can be changed (section 3.7). With the command

---

[2]The location for the "local", manually updated class and package files may be different, depending on the TeX system and the OS.

[3]slide size 128 x 92 mm

[4]slide size 176 x 99 mm

[5]slide size 160 x 100 mm

[6]for **au**dio **vi**deo **m**edia

"\pagenrconst", the page number of the current slide can be set back to the page number of the previous slide (section 4.4, last paragraph).

**Elpres-specific environments**

The environments "psli" and "rsli" define *slides* in elpres (section 3.4); "citemize","cenumerate" and "cdescription" provide vertically compressed lists (section 3.6).

## 3.2 A minimalistic example presentation

The following code (minimal.tex) may be used for first "experiments" by adding features described in later sections.

```
\documentclass[12pt,pdftex,4x3]{elpres}
\usepackage[utf8]{inputenc}
\usepackage[document]{ragged2e}

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

The preamble of the same presentation for LuaLATEX would read:

```
% LuaLaTeX: Please use utf-8 encoding!
\documentclass[12pt,nofonts,4x3]{elpres}
\usepackage[document]{ragged2e}
\usepackage{fontspec}

\setmonofont{TeX Gyre Cursor}              %% based on 'Courier'
\setsansfont[Scale=0.92]{Tex Gyre Heros}   %% similar to 'Helvetica'
\setmainfont{TeX Gyre Termes}              %% 'Roman' style serif font
\renewcommand{\familydefault}{\sfdefault}

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

\begin{document}
% ...
\end{document}
```

The use of LuaLATEX with `elpres` is heavily recommended due to the superior font selection mechanisms.

### 3.3 Testing of the `elpres` installation

With this example you may check, if `elpres` has been installed correctly in your TEX-system: copy the listing of `minimal.tex` into a file, add the class option `ballsymb`:

```
\documentclass[12pt,pdftex,4x3,ballsymp]{elpres}
...
```

and process it with `pdflatex`. The line `\usepackage{elpreswhitetealscheme}` should not be commented out, i. e. the package `elpreswhitetealscheme` should be active. If processing of `minimal.tex` results in a message like

```
... `ep-ball-04' not found ...
```

the TEX-system does not find the file `ep-ball-04.png` in the appropriate place. In such a situation it is recommended

1. either to install the current version of `elpres` in the correct subdirectories of the "local" `texmf`-directory, details are described in section 2

2. or to copy the small elpres graphics files `ep-ball-nn.eps ep-ball-nn.png` into a directory of the `tex`-branch of the local `texmf`-directoty tree (followed by an update see of TEXs files database, for details see section 2). The files may be obtained from the `elpres` project page [http://vkiefel.de/elpres.html](http://vkiefel.de/elpres.html)[7]

3. or to copy the required image file into the current project directory.

The problem sometimes arises as soon as TEX-distributions do not install `elpres` correctly. For details see `README.md` in the archive `elpres.zip`, which may be obtained from `ctan.org`.

## 3.4 Essential elements of an `elpres` presentation

The **title page** slide can be created with the `titlepage` environment (or `rsli`, see below), LATEX's `\maketitle` command is not available. **Slides** may be created with the `psli`-environment[8], you may add the title of the slide with the optional parameter. The contents of the slide are centered vertically:

```
\begin{psli}[Slide title]
    Contents of slide
\end{psli}
```

Another environment generating a **slide** is `rsli`[9]: slides are written without title, contents are not centered vertically:

```
\begin{rsli}
    Contents of slide
\end{rsli}
```

The `\distance{number}` command allows to introduce vertical space into slides constructed with the `rsli` and `titlepage` environments. You may use of `\distance{number}` commands with numbers indicating the relative height of empty spaces between lines and/or paragraphs, see the title page in the example above (section 3.2).

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[10]. After inserting a new footnote, numbers or symbols are correctly inserted only after a second run of LATEX.

## 3.5 Alternatives to TEX's justified paragraphs

By default, LATEX produces justified paragraphs with lines of equal length, this will often not be appropriate for the usually very short lines of text in presentations. The LATEX `\raggedright` command has its own deficiencies: by inhibiting hypenation 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.

If authors of presentations wish to suppress both justification and hyphenation within the complete presentation document, the following package parameters:

---

[7][http://vkiefel.de/elpres/elpres-ball-image-files.zip](http://vkiefel.de/elpres/elpres-ball-image-files.zip)

[8]**psli**: **p**lain **sli**de

[9]**rsli**: **r**aw **sli**de

[10]The use of the `footmisc` package may produce the LATEX error message "`Counter too large`". Therefore it is preferred to use the `symbol*` and `perpage` options, details are described in the manual of the `footmisc` package.

```
\usepackage[document,originalparameters]{ragged2e}
```

can be selected. As hypenated words are not favorable for quick reading of slides by the audience this is the preferred use of `ragged2e` for the author of this package.

## 3.6 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-sperated 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}
```

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

## 3.7 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 presentaton.

However, access to the header fields defined by `fancyhdr` is blocked: if 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`. 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 envoronments), details are described in section 4.10.

# 4  Enhancements to `elpres`

Enhancements described here make use of other LATEX packages which should be available in TEX distributions like TEX Live or MiKTEX.

## 4.1  Use named colors of the `xcolor` package

The `elpres` class automatically loads the `xcolor` package. Therefore, color names defined by the `xcolor` may be used. They are however only accessible in groups (`dvipsnames`, `svgnames`, `x11names`, details can be found in the `xcolor` manual). As an example, the color `Indigo` is available in the `svgnames` goup. 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 loaded by `elpres`. 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 typical example of a page with a picture could be (`pict.png` is the name of the graphics file):

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

The `\includegraphics[]{}` comannd 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[11].

## 4.4  Add incremental slides (overlays) to a presentation

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) pages (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 `rsli` 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:

---

[11]A good reference is https://latexref.xyz/

```
\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.

Sometimes it is desirable to prepare **two or more consecutive slides with the same page number** independent of the overlays package ("manual overlays slides"). Therefore, the page number of the current slide can be set to the page number of the previous slide/page with the command `\pagenrconst`.

## 4.5  Add navigation symbols to slides

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

$\ll$ $<$ $>$ $\gg$ $\leftarrow$ $\rightarrow$ `[n]`

These commands ($\ll$: jump to the first page, $<$: go to the previous page, $>$: go to the next page, $\gg$: jump to the last page, $\leftarrow$: go back in history, $\rightarrow$: go forward in history, `[n]`: prompt for a page number) work with Adobe Acrobat Reader (Windows), and with `Okular` (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   Add multimedia content to a presentation

This section describes inclusion of (links to) video and audio files in a presentation. With the `\auvimm{}{}` command described below it is possible to lauch 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`) 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 starting a window with the external application. On Linux systems the `pdfpc` viewer allows this, a detailed description is shown below. In both situations, however, the multimedia file is not incorprated 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.

The command `\auvimm{text}{media-file}` inserts `text` with a link ("poster"), which allows to start the external default 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` (Linux).[12] 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}

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 if fullscreen-mode at the time of presentation. If this does not work, it may be necessary to insert as "launch" command:

    \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.

> The `\auvimm`-command in the current version of `elpres` should be considered experimental. Its behavior largely depends on the `pdf`-viewer and the software used for playing audio/video files.

On Linux systems, the `pdfpc` pdf-viewer[13] allows presentation of a video embedded into a "poster area"[14] 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{}{}` using the "`run:./video-file`" launch-command. An example with `\auvimm{}{}`:

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

---

[12]The `\auvimm` command uses the `\href` command of the `hyperref` package
[13]https://pdfpc.github.io/: `pdfpc` — A presenter console with multi-monitor support for PDF files
[14]e. g. defined by a bitmap image with the same aspect ratio as the video

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.

## 4.7 Prepare a "handout" from a presentation

In advance of a lecture it is often expected that you provide a "handout" of your presentation with more than one pages on a printed page. Therefore you have several options, three of them are described here. If a presentation contains overlay-slides generated with the `overlays` package as described in section 4.4, it should be recompiled with the commands described in section 5.3 added to the preamble in order to inactivate the overlays-specific commands.

### 4.7.1 LaTeX article document handout 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 pdfLaTeX 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 website[15]. On Windows systems the name of the ghostscript command may be `gswin32c.exe` or `gswin64c.exe`. An alternative to `gs` is the `pdftk` command line tool:[16]

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

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

**(2)** an example for a LaTeX document which can serve as handout:

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

---

[15]https://ghostscript.readthedocs.io/en/latest/Use.html
[16]https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/

### 4.7.2 Create handout with the `pdfpages` package

Another, more comfortable option is to use the `pdfpages` package. Here, it is not necessary to split up the presentation into single pages. An example (in "portait format") which generates pages with 2x4 slides (all slides except slide 2) using the `\includepdf[]{}` command:

```
\documentclass[12pt]{article}
\usepackage[a4paper,hmargin=2cm,bottom=3.2cm]{geometry}
\usepackage{pdfpages}
\begin{document}
\includepdf[pages={1,3-last},nup=2x4,frame=true,%
scale=0.78,%
pagecommand={\thispagestyle{plain}}]{presentation.pdf}
\end{document}
```

Details on the optional parameters of `\includepdf` can be found in the documentation of the `pdfpages` package. This code:

```
\documentclass[12pt,landscape]{article}
\usepackage[a4paper,hmargin=2cm,bottom=3.2cm]{geometry}
\usepackage{pdfpages}
\begin{document}
\includepdf[pages={1,3-last},nup=3x2,frame=true,%
scale=0.78,%
pagecommand={\thispagestyle{plain}}]{presentation.pdf}
\end{document}
```

generates pages in "landscape format" with 3x2 slides.

### 4.7.3 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'
```

`pdfjam`[17] 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 website[18].

## 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

---

[17]which regerettably is only available on Linux or other Unix-like systems, on Windows systems, the procedure described above in section 4.7.2 can be used as alternative

[18]https://github.com/DavidFirth/pdfjam/blob/master/README.md

```
\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 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` package[19]. The option for the color of the frame around the link is `linkbordercolor`, colors must be defined whith 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` package[20]. 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 presentation[21]. 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`[22]:

```
\usepackage{eso-pic}

...

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

---

[19]both, `\totargettext` and `\fromlinktext` are are redefinitions of commands from the `hyperref` package
[20]written by Michael H.F. Wilkinson and available on CTAN
[21]i. e. following \begin{document}
[22]written by Rolf Niepraschk and available on CTAN

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

## 4.10 Put color to your presentation: 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`.[23] These `elpres`-schemes also support incremental slides with the `overlays` package. With an activated presentation scheme, the elpres class option `ballsymb` changes default itemize symbols to appropriately colored "ball" symbols. Moreover, these schemes define color names which may be used by the author of a presentation:[24]

**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)

These 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.

---

[23]the name of the corresponding style or package file is `elpreswhiteredscheme.sty`

[24]Color names for use by the presentation author start with `ep...`

# 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.

### 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}
```

The advantage of these simple colored text boxes: they are compatible with incremental slides using the `overlays` package.

### 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 detailed instructions for the use of this package. However, incremental slides using the `overlays` package may not work with color boxes generated by the `tcolorbox` package. These problems are avoided with the colored text boxes described in section 5.1.1.

## 5.2 Alternative format for footnotes on slides

Footnotes on slides may be inserted with LaTeX's `\footnote{}` command. However, the layout of footnotes generated by `\footnote{}` is not adequate for small comments or additions to the main text on slides. The following definition of an alternative footnote command (`\altfnarea{}`) requires that the `eso-pic` package is loaded:

```
\usepackage{eso-pic}

\newcommand{\altfnarea}[1]{%
\AddToShipoutPictureFG*{
\setlength{\unitlength}{1mm}
    \put(1.2,1.2){\parbox[b]{0.9\paperwidth}%
    {\scriptsize \textcolor{darkgray}{#1}}}}}
```

The definition of `\altfnarea` is currently not included in `elpres.cls`, so you should copy this definition of the command into the preamble of your presentation document. Please take care to include the `eso-pic` package before the `altfnarea` definition. You may experiment with different values in this `\newcommand{}{}` definition. The command can be used within the code of a slide the following way:

```
\begin{psli}[Title of a slide demonstrating alternative
    footnotes\textsuperscript{1}]
\begin{itemize}
\item first itemized line\textsuperscript{2}
\item second itemized line
\end{itemize}
\altfnarea{\textsuperscript{1}Addition to the title,
    \textsuperscript{2}Explanation of the first line}
\end{psli}
```

Very often it will be advisable *not* to use footnotes in slides. However, speakers participating in scientific conferences usually are expected to include sources of figures, tables etc. on their slides.

## 5.3 Inactivate `overlays` commands

If a pesentation 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 instuctions can be inactivated with these commands:

```
\overlaysoff
\alertsoff
```

which should be written into the preamble following `\usepackage{overlays}`. This works with v2.12 (and probably later versions) of `overlays`. With earlier versions you may copy the commands

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

into the preamble following `\usepackage{overlays}`, however, inactivation of `\only` by this method has a different effect, so it is recommended to update the `overlays` package.

## 5.4 Convert an `elpres`-presentation (PDF) into an Impress (LibreOffice) or Powerpoint presentation

### 5.4.1 PDF ... PNG ... Impress/Powerpoint

Sometimes it is unclear, if organizers of a (scientific) meeting allow presentations with `.pdf`-files or if they insist on files in Powerpoint format. This is an unpleasant situation for a lecturer, but in such a situation 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:

1. convert the `.pdf`-file of the presentation into a series of bitmap image files, (e. g. in `.png` format)

2. import these image `.png` files into "empty" slides of an Impress (LibreOffice or OpenOffice) presentation and save the presentation in `.odp`, `.ppt` or `.pptx`-Format

**Details on step 1 — generate `.png` images:**

Please copy the following two commands (please ignore the line-breaks in the lines beginning with "`gs`")[25] into a short shell script (Windows: a `.bat` or `.cmd` script) and "run" or execute the shell script:

```
gs -dNOPAUSE  -dBATCH -sDEVICE=pdfwrite  -dDEVICEWIDTHPOINTS=793.7
   -dDEVICEHEIGHTPOINTS=595.3 -dPDFFitPage -sOutputFile=temp.pdf input.pdf

gs -dNOPAUSE -dBATCH -sDEVICE=png256 -r300 -dTextAlphaBits=4
   -sOutputFile=page_%03d.png temp.pdf
```

Please do not forget to adjust the name of the input file (`input.pdf` in the script above). The first command adjusts the page size of the `.pdf`-file to `28 x 21 cm` which is written into the temporary file `temp.pdf`: this format avoids the necessity to resize of the images in LibreOffice (for other

---

[25]i. e. "`gs ... input.pdf`" should be entered as one line and "`gs ... temp.pdf`" should be entered as a second line of a shell script.

slide formats see the footnote below[26]). The second command generates a series of `.png`-files: `page_001.png`, `page_002.png` ... `page_0nn.png`. Instead of the `png256` driver, `png16m` can be used. On Windows systems the name of the ghostscript command is `gswin32c` or `gswin64c` (instead of `gs`).

If you wish a higher resolution of the `.png` images, change `-r300` to `-r600`. The option `-dTextAlphaBits=4` improves font antialiasing.[27]

**Details on step 2 — Import `.png` images:**

Open an empty presentation with LibreOffice Impress, adjust the appropriate slide format (e. g. 4:3), import the `.png` files with *Insert / Image* into consecutive empty slides.[28] The presentation can now be saved in the format of Impresss (`.odp`) or one of the Microsoft Powerpoint formats (`.ppt` or `.pptx`).

### 5.4.2 PDF ... SVG ... Impress/Powerpoint

An alternative way to make a Powerpoint/Impress presentation from the .pdf-file is to convert it into a series of **.pdf**-pages, convert these into `.svg` format and to import the `.svg`-files into the Impress/Powerpoint presentation. The first step:[29]

```
pdftk input.pdf burst output p_%02d.pdf
```

converts the file `input.pdf` into a series of files `p_01.pdf`, `p_02.pdf`, `p_03.pdf`, …, which can then be converted with the command `pdftocairo`:[30]

```
pdftocairo -svg p_01.pdf p_01.svg
pdftocairo -svg p_02.pdf p_02.svg
pdftocairo -svg p_03.pdf p_03.svg
...
```

This series of commands is best executed from a shell script. The resulting `.svg` files are then imported into an Impress/Powerpoint presentation.

### 5.5 The `pdfpc`-viewer: add notes

The pdf-vierwer `pdfpc` for Linux[31] (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.

---

[26]The values assigned to `-dDEVICEWIDTHPOINTS` and `-dDEVICEHEIGHTPOINTS` are valid for a width:height aspect ratio of 4:3 assuming a presentation page size (LibreOffice) of 28cm x 21cm: (28/2.54)*72=793.7pt, (21/2.54)*72=595.3pt. Values for width:height aspect ratios of 16:9 (28cm x 15.75cm) are 793.7/446.46 and for a ratio of 16:10 (28cm x 17.5cm): 793.7/496.1.

[27]Details can be found in the documentation of `gs` https://ghostscript.readthedocs.io/en/latest/Use.html

[28]Menu structure of LibreOffice v6.4, v7.3

[29]The command in a Windows terminal should be: `pdftk input.pdf burst output p_%%02d.pdf`

[30]Available in the `poppler-utils` package, a Windows version may be obtained at: https://github.com/oschwartz10612/poppler-windows

[31]https://pdfpc.github.io/

# 6 License

This class is distributed under the *LATEX 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 "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 3.7, 4.1, 4.10 and 5 added, section 2 on installation rewritten. **v0.7** (20.02.2021): parts of the manual (this file) have been rewritten and an index has been included. The command \pagenrconst has been added. **v0.8** (28.02.2021): Installation instructions have been updated in section 1 and detailed instructions were added to README.md. Section 5.3 was updated. A bug in the elpres...scheme.sty files was fixed (incorrect value for \labelitemiv). **v0.9** (01.08.2021): New class option bulletsymb added, an archive elpres.tds.zip is added for ctan.org upload. **v1.0**: Class option ballsymb added, manual updated. **v1.0.1**: Corrected upload to ctan.org **v1.1** (28.05.2024): Fixes and corrections in elpres.cls; manual: parts of section 3 have been rewritten, section 5.4 has been completed: new section 5.4.2. Definition of a an alternative command for footnotes (\altfnarea) has been added to the manual (section 5.2).

# Index