

elpres — electronic presentations with (pdf)L^AT_EX

Volker Kiefel*

v0.5

July 12, 2020

Contents

1	Introduction	2
2	Installation	2
3	Usage	2
3.1	Essential elements of an <code>elpres</code> presentation	3
3.2	Alternatives to T _E X’s justified paragraphs	3
3.3	Vertically compressed lists	4
4	Enhancements to <code>elpres</code>	4
4.1	Include graphics files	4
4.2	Arrange text and pictures in two (or more) columns	5
4.3	Incremental slides (overlays)	5
4.4	Navigation symbols	6
4.5	Call multimedia content from a presentation	7
4.6	Prepare a “handout” from a presentation	7
4.6.1	L ^A T _E X article document with a series of single-page pdf-files included	7
4.6.2	Convert presentation pdf-file directly into handout file with <code>pdfjam</code>	8
4.7	Create presentations with hypertext elements	8
4.8	Fill background of a presentation with bitmaps	9
4.8.1	<code>Wallpaper</code> package	9
4.8.2	<code>Eso-pic</code> package	9
5	License	10
6	Version history	10

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

1 Introduction

The `elpres` class can be used for presentations on a screen, a beamer or 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 documents 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 system. 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).

2 Installation

Copy `elpres.cls` into a directory, where your L^AT_EX-system can find it and update the files database¹.

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`² 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.

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

A simple example document:

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

\begin{document}
\begin{titlepage}
\centering
```

¹e.g. by using the `mktexlsr` or `texhash` command (TeXlive)

²for `audio video media`

```

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

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

\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
\end{rsli}
\end{document}

```

3.1 Essential elements of an elpres presentation

The **title page** can be created within the `titlepage` environment, the `\maketitle` command is not available. **Slides** may be created with the `psli`-environment³, you may add the title of the slide with the optional parameter. The contents of the slide are centered vertically.

Another environment generating a **slide** is `rsli`⁴: 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`.

3.2 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`psli`: plain slide

⁴`rsli`: raw slide

3.3 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}
```

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

4 Enhancements to `elpres`

4.1 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 × 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):

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

4.2 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.

4.3 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 `overlays` package written by Andreas Nolda together with `elpres`.

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:

```

\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 “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

```

For more details on `overlays`, see the documentation of the package.

4.4 Navigation symbols

With the `navisymb` option a panel with navigation symbols (colored in light gray) appears in the right lower corner of the presentation:

```
<< < > >> ← → [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 the exception of \leftarrow , \rightarrow 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).

4.5 Call multimedia content from a presentation

The command `\auvimm{text}{media-file}` inserts `text` with a link, which calls the default application for audio or video media files `media-file`. This seems to work with Adobe Acrobat Reader (Windows) and with `evince` (Linux). An example for using this command:

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

This produces the (text-based) “button” `[sound-file]`. If you hit it with the mouse pointer it will call the default application for audio files. 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 `elpres`-command should be considered experimental. Its behavior depends on the pdf-reader and the configuration of the OS.

4.6 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.

4.6.1 \LaTeX 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 pdf \LaTeX 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 website⁵. An alternative to `gs` is `pdftk` tool:

```
pdftk in.pdf burst output page_%02d.pdf
```

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

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

⁵<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}

```

4.6.2 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'
```

pdfjam⁶ 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 website⁷.

4.7 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

⁶which regerettably is only available on Linux or other Unix-like systems

⁷<https://github.com/DavidFirth/pdfjam/blob/master/README.md>

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` package⁸. 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.

4.8 Fill background of a presentation with bitmaps

4.8.1 Wallpaper package

To create a slide background with a graphical wallpaper background using bitmap files you may use the `wallpaper` package⁹. 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 presentation¹⁰. 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.

4.8.2 Eso-pic package

Another package which allows you to paint the background with a picture is `eso-pic`¹¹:

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

```
...
```

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

⁸both, `\totargettext` and `\fromlinktext` are redefinitions of commands from the `hyperref` package

⁹written by Michael H.F. Wilkinson and available on CTAN

¹⁰i. e. following `\begin{document}`

¹¹written 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.

5 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).

6 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 4.7 added). **v0.2d** (25.12.2004): Documentation completed (section 4.8 added). **v0.2e** (15.04.2005): Documentation completed (sections 4.8.2 and 4.6 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 4.1 and 4.2 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 4.6 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.