

The pdfpc-movie package

<https://gitlab.com/SFr682k/pdfpc-movie>

Sebastian Friedl
sfr682k@t-online.de

2018/07/21 (v1.0)

Abstract

This package provides a `\pdfpcmovie` command for hyperlinking movies in a way compatible to the PDF Presenter Console (`pdfpc`), a GPL2 licensed multi-monitor PDF presentation viewer application available on GitHub¹.

Contents

Dependencies and other requirements	2
Installation	2
License	2
I The documentation	3
Loading pdfpc-movie	3
Including movie files	3
II The package code	5
Index	8

¹<http://pdfpc.github.io/>

Dependencies and other requirements

pdfpc-movie requires the use of \TeX 2 ϵ .
It depends on the following packages:

etoolbox

hyperref

pgfkeys

Installation

Extract the *package* file first:

1. Run \TeX over the file `pdfpc-movie.ins`
2. Move the resulting `.sty` file to `TEXMF/tex/latex/pdfpc-movie/`

Then, you can compile the *documentation* yourself by executing

```
lualatex pdfpc-movie-doc.dtx
makeindex -s gind.ist pdfpc-movie-doc.idx
makeindex -s gglo.ist -o pdfpc-movie-doc.gls pdfpc-movie-doc.glo
lualatex pdfpc-movie-doc.dtx
lualatex pdfpc-movie-doc.dtx
```

or just use the precompiled documentation shipped with the source files.

In both cases, copy the files `pdfpc-movie-doc.pdf` and `README.md` to `TEXMF/doc/latex/pdfpc-movie/`

License

© 2018 Sebastian Friedl

This work may be distributed and/or modified under the conditions of the \TeX Project Public License, either version 1.3c of this license or (at your option) any later version.

The latest version of this license is available at <http://www.latex-project.org/lppl.txt> and version 1.3c or later is part of all distributions of \TeX version 2008-05-04 or later.

This work has the LPPL maintenace status ‘maintained’.

Current maintainer of this work is Sebastian Friedl.

This work consists of the following files:

- `pdfpc-movie.dtx`,
- `pdfpc-movie.ins`,
- `pdfpc-movie-doc.dtx` and
- the derived file `pdfpc-movie.sty`.

Part I

The documentation

Loading pdfpc-movie

Load pdfpc-movie as any other package by adding `\usepackage{pdfpc-movie}` to your preamble. There are no package options available.

Including movie files

The package provides a `\pdfpcmovie` command using the syntax

```
\pdfpcmovie            \pdfpcmovie[<options>]{<poster material>}{<path to movie file>}
```

It creates a hyperlink to a movie with *<path to movie file>* being the path of the movie file relative to the PDF it is linked from. These hyperlinks are evaluated by pdfpc, so that the movie (usually) starts playing when clicking onto *<poster material>*.

It will *not* embed the movie into the PDF file in the sense that it is part of the `foo.pdf` file. Hence, the file must be copied and passed along with the PDF file in a manner allowing the viewer application to find the file at *<path to movie file>*.

As a recommendation, *<path to movie file>* should *not* contain any “special” characters or spaces.

The *<poster material>* is placed inside a `\hbox` for determining and changing the space occupied by the poster material; therefore, it is not possible to insert any line breaks or similar \TeX code not allowed inside `\hboxes`.

The movie will playback in the area taken by the *<poster material>* or set via the width/height keys. Using a frame of the movie as *<poster material>* will ensure the correct aspect ratio.

The following *<options>* may be given as comma-separated list:

width=*<TeX dimension>* Overrides the width of *<poster material>* and sets it to the given *<TeX dimension>*

height=*<TeX dimension>* Overrides the height of *<poster material>* and sets it to the given *<TeX dimension>*

depth=*<TeX dimension>* Overrides the depth of *<poster material>* and sets it to the given *<TeX dimension>*

autostart Causes the movie to start playing immediately when the page is shown.

A test with version 4.1.2 of pdfpc lead to the result that it is possible to autostart multiple movies placed on the same page; however, it turned out that they didn't play synchronously, but slightly “shifted” (by approximately 0.25 s).

loop Let the movie start again when the end has been reached.
Normally, the movie just stops there.

noprogress Instructs pdfpc to avoid showing the progress bar below the movie.

As of pdfpc version 4.1.2, the progress bar will still be shown when the cursor is moved across the movie's bottom margin.

start=*<time>* Makes the movie start at *<time>* seconds.
<time> has to be an integer value without suffixes. For example, start=2 is valid, while start=4.2 and start=1s aren't.

stop=*<time>* Makes the movie stop (and eventually restart) when the playback position has reached *<time>* seconds.
As above, *<time>* has to be an integer value without suffixes.

If you don't need any options, you may omit the optional [*<options>*] parameter.

Examples

```
\pdfpcmovie{\includegraphics[width=\textwidth]{foo}}{foo.mp4}
```

Creates a link to the movie foo.mp4, using an image as poster material.
The poster image will be replaced by the movie when clicking it.

```
\pdfpcmovie[autostart, width=8cm, height=4.5cm]{Restart}{bar.avi}
```

Creates a link to the movie bar.avi.
The movie is started as soon as the page is displayed and takes an area of 8 cm width and 4.5 cm height.
The movie stops after playing.

```
\pdfpcmovie[autostart, start=2, stop=5, loop]%  
{\includegraphics[height=3cm]{res/baz}}{res/baz.mov}
```

Creates a link to the movie baz.mov residing in the sub-directory res/.
As soon as the page is displayed, the movie starts at 2 seconds and restarts when the playback position has reached 5 seconds.

Part II

The package code

Identify the package and force $\text{\LaTeX}2_{\epsilon}$:

```
1 \ProvidesPackage{pdfpc-movie}%  
2 [2018/07/21 v1.0 pdfpc compatible hyperlinking of movies]  
3 \NeedsTeXFormat{LaTeX2e}
```

Require additional packages needed by pdfpc-movie:

```
4 \RequirePackage{etoolbox}  
5 \AtEndPreamble{\RequirePackage{hyperref}}  
6 \RequirePackage{pgfkeys}
```

Variables and options

Define some internal boolean variables required to assemble the URL parameters:

```
7 \newif\if@pdfpcmovie@autostart  
8 \newif\if@pdfpcmovie@loop  
9 \newif\if@pdfpcmovie@noprogess
```

Define a counter for storing the number of non-processed options:

```
10 \newcount\pdfpcmovie@remopts
```

Define a box and some dimensions to override the poster material's size:

```
11 \newbox\pdfpcmovie@posterbox  
12 \newdimen\pdfpcmovie@posterwd  
13 \newdimen\pdfpcmovie@posterht  
14 \newdimen\pdfpcmovie@posterdp
```

Define the keys available as optional arguments of the `\pdfpcmovie` command. Each key requiring additional URL parameters advances `\pdfpcmovie@remopts` by 1:

```
15 \pgfkeys{%  
16 /pdfpc-movie/width/.cd, .value required,  
17 .code={\pdfpcmovie@posterwd=#1\relax},  
18 /pdfpc-movie/height/.cd, .value required,  
19 .code={\pdfpcmovie@posterht=#1\relax},  
20 /pdfpc-movie/depth/.cd, .value required,  
21 .code={\pdfpcmovie@posterdp=#1\relax},  
22 /pdfpc-movie/autostart/.cd, .value forbidden,  
23 .code={%  
24 \@pdfpcmovie@autostarttrue  
25 \advance\pdfpcmovie@remopts 1\relax},  
26 /pdfpc-movie/loop/.cd, .value forbidden,  
27 .code={%  
28 \@pdfpcmovie@looptrue  
29 \advance\pdfpcmovie@remopts 1\relax},  
30 /pdfpc-movie/noprogess/.cd, .value forbidden,  
31 .code={%  
32 \@pdfpcmovie@noprogesstrue  
33 \advance\pdfpcmovie@remopts 1\relax},  
34 /pdfpc-movie/start/.cd, .value required,  
35 .code={%  
36 \def\pdfpcmovie@start{#1}  
37 \advance\pdfpcmovie@remopts 1\relax},  
38 /pdfpc-movie/stop/.cd, .value required,
```

```

39     .code={%
40     \def\pdfpcmovie@stop{#1}
41     \advance\pdfpcmovie@remopts 1\relax},
42 }

```

The `\pdfpcmovie` macro

`\pdfpcmovie` First, `\def` is used to ensure that there is a `\pdfpcmovie` command to (re-)define:

```

43 \def\pdfpcmovie{}
44 \renewcommand{\pdfpcmovie}[3][[]]{%

```

Leave vertical mode to ensure correct placement of the movie box if `\pdfpcmovie` is used at the beginning of a paragraph:

```

45 \leavevmode%

```

Check whether the requested movie file is present. Emit a warning if it's not:

```

46 \IfFileExists{\currdir #3}{\PackageWarning{pdfpc-movie}{%
47     The movie file "#3" could not be found in the current
48     directory, where it must reside for viewing.}%
49 }%

```

Create a box with the poster content and determine its size:

```

50 \setbox\pdfpcmovie@posterbox=\hbox{#2}%

```

Reset all variables to their default values.

After that, the optional argument containing all options gets processed:

```

51 \@pdfpcmovie@autostartfalse%
52 \@pdfpcmovie@loopfalse%
53 \@pdfpcmovie@noprogessfalse%
54 \let\pdfpcmovie@start\@empty%
55 \let\pdfpcmovie@stop\@empty%
56 \pdfpcmovie@posterwd=\wd\pdfpcmovie@posterbox%
57 \pdfpcmovie@posterht=\ht\pdfpcmovie@posterbox%
58 \pdfpcmovie@posterdp=\dp\pdfpcmovie@posterbox%
59 %
60 \pdfpcmovie@remopts=0%
61 \pgfkeys{/pdfpc-movie/.cd, #1}%

```

All hyperlink arguments appended to the file name are stored in the `\pdfpcmovie@params` variable. For every specified option, the respective hyperlink code is appended.

If there are non-processed arguments remaining, an `&` separator is appended:

```

62 \ifnum\pdfpcmovie@remopts>0\def\pdfpcmovie@params{?}%
63 \else\def\pdfpcmovie@params{}%
64 \fi%
65 %
66 \if@pdfpcmovie@autostart%
67 \edef\pdfpcmovie@params{\pdfpcmovie@params autostart}%
68 \advance\pdfpcmovie@remopts -1\relax%
69 \ifnum\pdfpcmovie@remopts>0%
70 \edef\pdfpcmovie@params{\pdfpcmovie@params\&}%
71 \fi%
72 \fi%
73 %
74 \if@pdfpcmovie@loop%
75 \edef\pdfpcmovie@params{\pdfpcmovie@params loop}%
76 \advance\pdfpcmovie@remopts -1\relax%
77 \ifnum\pdfpcmovie@remopts>0%

```

```

78     \edef\pdfpcmovie@params{\pdfpcmovie@params\&}%
79     \fi%
80     \fi%
81     %
82     \if@pdfpcmovie@noprogess%
83         \edef\pdfpcmovie@params{\pdfpcmovie@params noprogess}%
84         \advance\pdfpcmovie@remopts -1\relax%
85         \ifnum\pdfpcmovie@remopts>0%
86             \edef\pdfpcmovie@params{\pdfpcmovie@params\&}%
87             \fi%
88         \fi%
89         %
90         \ifx\pdfpcmovie@start\@empty\else%
91             \edef\pdfpcmovie@params{\pdfpcmovie@params start=\pdfpcmovie@start}%
92             \advance\pdfpcmovie@remopts -1\relax%
93             \ifnum\pdfpcmovie@remopts>0%
94                 \edef\pdfpcmovie@params{\pdfpcmovie@params\&}%
95                 \fi%
96             \fi%
97             %
98             \ifx\pdfpcmovie@stop\@empty\else%
99                 \edef\pdfpcmovie@params{\pdfpcmovie@params stop=\pdfpcmovie@stop}%
100                \advance\pdfpcmovie@remopts -1\relax%
101                \ifnum\pdfpcmovie@remopts>0%
102                    \edef\pdfpcmovie@params{\pdfpcmovie@params\&}%
103                    \fi%
104                \fi%
105                %
106                %
107                %

```

Set the size of the poster material box ...

```

108     \wd\pdfpcmovie@posterbox=\pdfpcmovie@posterwd%
109     \ht\pdfpcmovie@posterbox=\pdfpcmovie@posterht%
110     \dp\pdfpcmovie@posterbox=\pdfpcmovie@posterdp%

```

... and – last but not least – create the hyperlink:

```

111     \href{run:#3\pdfpcmovie@params}{\box\pdfpcmovie@posterbox}%
112 }

```

Famous last words

Enough code for this package. \endinput.

```

113 \endinput

```