

The `pdfoverlay` package

David Purton*

2019/11/03 v1.1

Abstract

This `pdfoverlay` package provides a simple interface to overlay text on to an existing PDF. The text to be overlaid is typeset and positioned normally as you would any other L^AT_EX document. Some or all of the pages of the PDF can be included and not all pages of the PDF need have overlayed text. It's also possible to include text between pages of the PDF.

Contents

1	Introduction	1
2	Acknowledgements	2
3	Bug Reports and Feature Requests	2
4	Documentation	2
4.1	Basic Usage	2
4.2	Main Interface	2
5	Implementation	3
5.1	Messages	3
5.2	Private Variables and Helper Functions	4
5.3	Public Expl3 Functions	6
5.4	Public L ^A T _E X Interface	9
Change History		10
Index		10

*Email: dcpurton@marshwiggle.net

1 Introduction

It's often desirable to take an existing PDF and easily add annotations or text overlaying the PDF. This might arise if you wish to add comments to a PDF, fill in a PDF form, or add text to a PDF where space has been left for notes.

This package provides a simple interface to do this without having to resort to inserting one page at a time. Some or all of the pages of the PDF can be included and not all pages of the PDF need have overlayed text. It's also possible to include text between pages of the PDF.

Another advantage of this package is that the overlayed text can be set as normal flowing from one page to another or with manual page breaks if you wish. It's also possible to use any standard method to position text at arbitrary places on a given page.

2 Acknowledgements

This package is particularly indebted to the `pdfpages` package to count the number of pages in the PDF file, the `eso-pic` package to output pages from the PDF file in the document background, and the `everypage` package to ensure that PDF pages are inserted at every page in the document.

3 Bug Reports and Feature Requests

Bug reports and feature requests can be made at the `pdfoverlay` package GitHub repository. See <https://github.com/dcpurton/pdfoverlay>.

4 Documentation

4.1 Basic Usage

At it's simplest, the only thing required is to specify the PDF to overlay with text and then begin typing your document. The pages of the PDF will be inserted as you go automatically. If your text takes up more pages than the PDF then any remaining text is inserted on blank pages.

```
\documentclass{article}
\usepackage{pdfoverlay}
\pdfoverlaySetPDF{filename.pdf}
\begin{document}
...
\end{document}
```

The PDF will be centered on and scaled to fit the document page size while keeping a fixed aspect ratio.

4.2 Main Interface

<u>\pdfoverlaySetPDF \pdfoverlay_set_pdf:n</u>	<code>\pdfoverlaySetPDF {\langle PDF filename \rangle}</code> Specify the <i>\langle PDF filename \rangle</i> to overlay text on to. This function must be called first. It can be used either in the preamble or at any point in the document body. It is possible to call this more than once to include multiple PDF files. If the specified <i>\langle PDF filename \rangle</i> cannot be found an error is generated.
<u>\pdfoverlaySetGraphicsOptions \pdfoverlay_set_graphics_options:n</u>	<code>\pdfoverlaySetGraphicsOptions {\langle Options \rangle}</code> Set the <i>\langle Options \rangle</i> to be passed to <code>\includegraphics</code> when each page of the PDF is output. See the <code>graphicx</code> package documentation for valid options. Note that it is not necessary to specify the <code>page</code> option as this is appended automatically in the right format. The default options are: <code>keepaspectratio</code> , <code>width=\paperwidth</code> , <code>height=\paperheight</code> .
<u>\pdfoverlayIncludeToPage \pdfoverlay_include_to_page:n</u>	<code>\pdfoverlayIncludeToPage {\langle page number \rangle}</code> Output all pages in the PDF file up until the specified <i>\langle page number \rangle</i> . If the specified <i>\langle page number \rangle</i> does not evaluate to an integer or does not exist in the PDF an error is generated.
<u>\pdfoverlayIncludeToLastPage \pdfoverlay_include_to_last_page:</u>	<code>\pdfoverlayIncludeToLastPage</code> Output all remaining pages in the PDF file.
<u>\pdfoverlaySkipToPage \pdfoverlay_skip_to_page:n</u>	<code>\pdfoverlaySkipToPage {\langle page number \rangle}</code> Skip to the specified <i>\langle page number \rangle</i> . This can be before or after the current page in the PDF file. If the specified <i>\langle page number \rangle</i> does not evaluate to an integer or does not exist in the PDF an error is generated.
<u>\pdfoverlayPauseOutput \pdfoverlay_pause_output:</u>	<code>\pdfoverlayPauseOutput</code> Pause outputting pages from the PDF file. Any subsequent text will be output on blank pages.
<u>\pdfoverlayResumeOutput \pdfoverlay_resume_output:</u>	<code>\pdfoverlayResumeOutput</code> Resume outputting pages from the PDF file.

5 Implementation

```
1  {*package}
2  (@@=pdfoverlay)
3  \RequirePackage{xparse}
4  \ProvidesExplPackage{pdfoverlay}{2019/11/03}{1.1}
5    {Overlay text on an existing PDF document (DCP)}
```

The `everypage` and `pdfpages` packages are required.

```
6 \RequirePackage{everypage, pdfpages}
```

Call `__pdfoverlay_output_pdf_page`: on every page using the `everypage` package.

```
7 \AddEverypageHook { \__pdfoverlay_output_pdf_page: }
```

Add `\null` to the end of the document if an action is pending. This is required to ensure that the last requested page from the PDF file is output even if there isn't any other content on the page.

```
8 \AtEndDocument {
9   \bool_if:NT \g__pdfoverlay_action_pending_bool
10  {
11    \null
12  }
13 }
```

5.1 Messages

Error message when specified PDF file cannot be found.

#1: PDF file name

```
14 \msg_new:nnnn { pdfoverlay } { file-not-found }
15   { PDF~file~'#1'~not~found. }
16   { Unable~to~find~the~file~'#1'. \\
17     Check~that~the~file~exists~and~you~have~spelt~it~correctly. }
```

Error message when no PDF file has been set.

```
18 \msg_new:nnnn { pdfoverlay } { file-not-set }
19   { PDF~file~not~set. }
20   { You~have~not~specified~a~PDF~file. \\
21     Set~a~PDF~file~using~\pdfOverlaySetPDF. }
```

Error message when the requested page number cannot be found in the PDF file.

#1: PDF file name

#2: Requested page number

#3: Total number of pages in PDF

```
22 \msg_new:nnnn { pdfoverlay } { page-not-found }
23   { Page~not~found~in~PDF. }
24   { PDF~file~'#1'~does~not~contain~page~#2. \\
25     Specify~a~page~between~1~and~#3. }
```

Error message when the requested page number to include to is less than the current page number being output from the PDF file.

#1: Requested page number

#2: Current page number in PDF

#3: Total number of pages in PDF

```
26 \msg_new:nnnn { pdfoverlay } { page-too-low }
27   { Requested~page~less~than~current~page~in~PDF. }
28   { You~have~requested~to~include~to~page~#1, \\
29     but~the~current~page~is~already~at~page~#2. \\
30     Specify~a~page~between~#2~and~#3. }
```

5.2 Private Variables and Helper Functions

\g_pdfoverlay_pdf_file_name_str Store the PDF file name.
 31 \str_new:N \g_pdfoverlay_pdf_file_name_str
(End definition for \g_pdfoverlay_pdf_file_name_str.)

\g_pdfoverlay_page_count_int Store the total number of pages in the PDF file.
 32 \int_new:N \g_pdfoverlay_page_count_int
(End definition for \g_pdfoverlay_page_count_int.)

\g_pdfoverlay_page_int Keep track of the current page in the PDF file.
 33 \int_new:N \g_pdfoverlay_page_int
(End definition for \g_pdfoverlay_page_int.)

\g_pdfoverlay_output_active_bool Flag to store whether to output pages from the PDF file or not.
 34 \bool_new:N \g_pdfoverlay_output_active_bool
(End definition for \g_pdfoverlay_output_active_bool.)

\g_pdfoverlay_action_pending_bool Flag to store whether an action (\pdfoverlay_include_to_page:n or \pdfoverlay_set_page:n) is pending. If an action is pending, these functions will start a new page.
 35 \bool_new:N \g_pdfoverlay_action_pending_bool
(End definition for \g_pdfoverlay_action_pending_bool.)

\g_pdfoverlay_graphics_options_clist Hold the options passed to \includegraphics when including a page from the PDF. The default options scale the PDF page to fit the document page while keeping the original PDF page aspect ratio.
 36 \clist_new:N \g_pdfoverlay_graphics_options_clist
 37 \clist_set:Nn \g_pdfoverlay_graphics_options_clist
 38 {
 39 keepaspectratio ,
 40 width = \paperwidth ,
 41 height = \paperheight ,
 42 page = \int_use:N \g_pdfoverlay_page_int
 43 }

(End definition for \g_pdfoverlay_graphics_options_clist.)

_pdfoutput_pdf_page: Main function to output the current page of the PDF file. This is called before every page is shipped using the everypage package.
 44 \cs_new:Nn _pdfoutput_pdf_page:
 45 {
 Check if we are currently outputting pages, a PDF file has been set, and the requested page exists in the PDF file.
 46 \bool_lazy_all:nT
 47 {
 48 { \bool_if_p:N \g_pdfoverlay_output_active_bool }
 49 { \bool_not_p:n
 50 { \str_if_empty_p:N \g_pdfoverlay_pdf_file_name_str } }
 51 { \int_compare_p:n

```

52         { \c_zero_int <= \g__pdfoverlay_page_int
53             < \g__pdfoverlay_page_count_int } }
54     }
55 {

```

Place requested page from the PDF file in the background of the document page with the specified format.

```

56         \AddToShipoutPictureBG*
57     {
58         \_\_pdfoverlay_format_pdf_page:
59     }

```

Increment the current page counter.

```
60         \int_gincr:N \g__pdfoverlay_page_int
```

Set action pending flag to false.

```

61         \bool_gset_false:N \g__pdfoverlay_action_pending_bool
62     }
63 }

```

(End definition for __pdfoverlay_output_pdf_page:.)

__pdfoverlay_format_pdf_page: Format the PDF page in the centre of the document page with options from \g__pdfoverlay_graphics_options_clist. Although marked as private, this function could be redefined to position the PDF page at a different location on the document page. See the `eso-pic` package documentation for hints on how to position the PDF page.

```

64 \cs_new:Nn \_\_pdfoverlay_format_pdf_page:
65 {
66     \AtPageCenter
67     {
68         \makebox (0, 0)
69         {
70             \use:x
71             {
72                 \exp_not:N \includegraphics
73                 [ \clist_use:Nn \g__pdfoverlay_graphics_options_clist { , } ]
74                 { \g__pdfoverlay_pdf_file_name_str }
75             }
76         }
77     }
78 }

```

(End definition for __pdfoverlay_format_pdf_page:.)

__pdfoverlay_check_page_p:n __pdfoverlay_check_page:nTF Check if a PDF file has been set and the requested page is valid. If not, generate suitable error messages.

```

79 \prg_new_if:nTF \_\_pdfoverlay_if_page_exists:n #1 { p, T, F, TF }
80 {
81     \str_if_empty:NTF \g__pdfoverlay_pdf_file_name_str
82     {
83         \msg_error:nn { pdfoverlay } { file-not-set }
84         \prg_return_false:
85     }
86     {
87         \int_compare:nTF

```

```

88     {
89         \c_one_int <= #1 <= \g__pdfoverlay_page_count_int
90     }
91     {
92         \prg_return_true:
93     }
94     {
95         \msg_error:nxxxx { pdfoverlay } { page-not-found }
96         { \str_use:N \g__pdfoverlay_pdf_file_name_str }
97         { \int_eval:n { #1 } }
98         { \int_use:N \g__pdfoverlay_page_count_int }
99         \prg_return_false:
100    }
101 }
102 }
```

(End definition for `_pdfoverlay_check_page:nTF`.)

5.3 Public Expl3 Functions

`\pdfoverlay_set_pdf:n` Specify a PDF file to overlay text on to.

```

103 \cs_new:Npn \pdfoverlay_set_pdf:n #1
104 {
```

Test if the file exists and generate a suitable error if it doesn't.

```

105     \file_if_exist:nTF { #1 }
106     {
107         \str_gset:Nn \g__pdfoverlay_pdf_file_name_str { #1 }
```

Use the `pdfpages` package to find the number of pages in the PDF file.

```

108     \edef \AM@currentdocname { #1 }
109     \AM@getpagecount
```

Initialise variables.

```

110     \int_gset_eq:NN \g__pdfoverlay_page_count_int \AM@pagecount
111     \int_gset_eq:NN \g__pdfoverlay_page_int \c_zero_int
112     \bool_gset_true:N \g__pdfoverlay_output_active_bool
113     \bool_gset_false:N \g__pdfoverlay_action_pending_bool
114 }
115 {
116     \msg_error:nnn { pdfoverlay } { file-not-found } { #1 }
117 }
118 }
```

(End definition for `\pdfoverlay_set_pdf:n`. This function is documented on page 2.)

`\pdfoverlay_set_graphics_options:n` Set the options as a comma separated list to be passed to `\includegraphics` when outputting each page from the PDF file. The `page` option is appended automatically in the correct format.

```

119 \cs_new:Nn \pdfoverlay_set_graphics_options:n
120 {
121     \clist_gset:Nn \g__pdfoverlay_graphics_options_clist { #1 }
122     \clist_gput_right:Nn \g__pdfoverlay_graphics_options_clist
123     {
124         page = \int_use:N \g__pdfoverlay_page_int
```

```
125     }
126 }
```

(End definition for `\pdfoverlay_set_graphics_options:n`. This function is documented on page 2.)

`\pdfoverlay_include_to_page:n` Output all pages in the PDF file up until the specified page. Since the actual PDF page is inserted by `__pdfoverlay_output_pdf_page:` using the `everypage` package before the page is shipped, we just insert the required number of blank pages into the document.

```
127 \cs_new:Npn \pdfoverlay_include_to_page:n #1
128 {
```

Check if a PDF file is set and the requested page exists.

```
129     \__pdfoverlay_if_page_exists:nT { #1 }
130 {
```

Check if the requested page is greater than the current page.

```
131     \int_compare:nTF
132     {
133         #1 >= \g__pdfoverlay_page_int + 1
134     }
135 {
```

Check if an action is pending and start a new page if so.

```
136     \bool_lazy_all:nT
137     {
138         { \bool_if_p:n { \g__pdfoverlay_action_pending_bool } }
139         { \int_compare_p:n { \g__pdfoverlay_page_int <
140             \g__pdfoverlay_page_count_int - 1 } }
141         { \int_compare_p:n { #1 != \g__pdfoverlay_page_int + 1 } }
142     }
143     {
144         \null
145         \clearpage
146     }
```

Loop until the specified page, inserting blank pages.

```
147     \int_while_do:nNnn { \g__pdfoverlay_page_int } < { #1 - 1 }
148     {
149         \null
150         \clearpage
151     }
```

Set the action pending flag, so final page is output even if no text is overlayed on it.

```
152     \bool_gset_true:N \g__pdfoverlay_action_pending_bool
153     }
154     {
155         \msg_error:nnxxx { pdfoverlay } { page-too-low }
156         { \int_eval:n { #1 } }
157         { \int_eval:n { \g__pdfoverlay_page_int + 1 } }
158         { \int_use:N \g__pdfoverlay_page_count_int }
159     }
160 }
161 }
```

(End definition for `\pdfoverlay_include_to_page:n`. This function is documented on page 3.)

\pdfoverlay_include_to_last_page: Output all remaining pages in the PDF file.

```

162 \cs_new:Nn \pdfoverlay_include_to_last_page:
163 {
164     \pdfoverlay_include_to_page:n { \g__pdfoverlay_page_count_int }
165 }
```

(End definition for \pdfoverlay_include_to_last_page:. This function is documented on page 3.)

\pdfoverlay_skip_to_page:n Skip to the specified page number in the PDF file.

```

166 \cs_new:Npn \pdfoverlay_skip_to_page:n #1
167 {
168     \__pdfoverlay_if_page_exists:nT { #1 }
169 }
```

Check if an action is pending and start a new page if so.

```

170     \bool_if:nT { \g__pdfoverlay_action_pending_bool }
171     {
172         \null
173         \clearpage
174     }
```

Set the page numer for the next page to be output.

```
175     \int_gset:Nn \g__pdfoverlay_page_int { #1 - 1 }
```

Set the action pending flag, so final page is output even if no text is overlayed on it.

```

176     \bool_gset_true:N \g__pdfoverlay_action_pending_bool
177 }
178 }
```

(End definition for \pdfoverlay_skip_to_page:n. This function is documented on page 3.)

\pdfoverlay_pause_output: Pause outputting pages from the PDF file.

```

179 \cs_new:Nn \pdfoverlay_pause_output:
180 {
181     \bool_gset_false:N \g__pdfoverlay_output_active_bool
182 }
```

(End definition for \pdfoverlay_pause_output:. This function is documented on page 3.)

\pdfoverlay_resume_output: Resume outputting pages from the PDF file.

```

183 \cs_new:Nn \pdfoverlay_resume_output:
184 {
185     \bool_gset_true:N \g__pdfoverlay_output_active_bool
186 }
```

(End definition for \pdfoverlay_resume_output:. This function is documented on page 3.)

5.4 Public L^AT_EX Interface

\pdfoverlaySetPDF Specify a PDF file to overlay text on to.

```
187 \NewDocumentCommand \pdfoverlaySetPDF { m }
188   {
189     \pdfoverlay_set_pdf:n { #1 }
190   }
```

(End definition for `\pdfoverlaySetPDF`. This function is documented on page 2.)

\pdfoverlaySetGraphicsOptions Set the options as a comma separated list to be passed to `\includegraphics` when outputting each page from the PDF file. The `page` option is appended automatically in the right format.

```
191 \NewDocumentCommand \pdfoverlaySetGraphicsOptions { m }
192   {
193     \pdfoverlay_set_graphics_options:n { #1 }
194   }
```

(End definition for `\pdfoverlaySetGraphicsOptions`. This function is documented on page 2.)

\pdfoverlayIncludeToPage Output all pages in the PDF file up until the specified page.

```
195 \NewDocumentCommand \pdfoverlayIncludeToPage { m }
196   {
197     \pdfoverlay_include_to_page:n { #1 }
198   }
```

(End definition for `\pdfoverlayIncludeToPage`. This function is documented on page 3.)

\pdfoverlayIncludeToLastPage Output all remaining pages in the PDF file.

```
199 \NewDocumentCommand \pdfoverlayIncludeToLastPage { }
200   {
201     \pdfoverlay_include_to_last_page:
202   }
```

(End definition for `\pdfoverlayIncludeToLastPage`. This function is documented on page 3.)

\pdfoverlaySkipToPage Skip ahead to the specified page number in the PDF file.

```
203 \NewDocumentCommand \pdfoverlaySkipToPage { m }
204   {
205     \pdfoverlay_skip_to_page:n { #1 }
206   }
```

(End definition for `\pdfoverlaySkipToPage`. This function is documented on page 3.)

\pdfoverlayPauseOutput Pause outputting pages from the PDF file.

```
207 \NewDocumentCommand \pdfoverlayPauseOutput { }
208   {
209     \pdfoverlay_pause_output:
210   }
```

(End definition for `\pdfoverlayPauseOutput`. This function is documented on page 3.)

```
\pdfoverlayResumeOutput Resume outputting pages from the PDF file.
211 \NewDocumentCommand \pdfoverlayResumeOutput { }
212 {
213     \pdfoverlay_resume_output:
214 }
```

(End definition for `\pdfoverlayResumeOutput`. This function is documented on page 3.)

215 ⟨/package⟩

Change History

v1.0	v1.1
General: First public release 1	General: Fix depracated macros 1

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols	
\\	16, 20, 24, 28, 29
A	
\AddEverypageHook	7
\AddToShipoutPictureBG	56
\AtEndDocument	8
\AtPageCenter	66
B	
bool commands:	
\bool_gset_false:N	61, 113, 181
\bool_gset_true:N	112, 152, 176, 185
\bool_if:NTF	9
\bool_if:nTF	170
\bool_if_p:N	48
\bool_if_p:n	138
\bool_lazy_all:nTF	46, 136
\bool_new:N	34, 35
\bool_not_p:n	49
C	
\clearpage	145, 150, 173
clist commands:	
\clist_gput_right:Nn	122
\clist_gset:Nn	121
\clist_new:N	36
\clist_set:Nn	37
E	
\edef	108
exp commands:	
\exp_not:N	72
F	
file commands:	
\file_if_exist:nTF	105
I	
\includegraphics	2, 5, 7, 9, 72
int commands:	
\int_compare:nTF	87, 131
\int_compare_p:n	51, 139, 141
\int_eval:n	97, 156, 157
\int_gincr:N	60
\int_gset:Nn	175
\int_gset_eq:NN	110, 111
\int_new:N	32, 33
\int_use:N	42, 98, 124, 158
\int_while_do:nNnn	147
\c_one_int	89
\c_zero_int	52, 111

M	
\makebox	68
msg commands:	
\msg_error:nn	83
\msg_error:nnn	116
\msg_error:nnnn	95, 155
\msg_new:nnnn	14, 18, 22, 26
N	
\NewDocumentCommand	
... 187, 191, 195, 199, 203, 207, 211	
\null	3, 11, 144, 149, 172
P	
\paperheight	2, 41
\paperwidth	2, 40
pdfoverlay commands:	
\pdfoverlay_include_to_last_- page: ... 3, 162, 201	
\pdfoverlay_include_to_page:n 3, 4, 127, 164, 197	
\pdfoverlay_pause_output: 3, 179, 209	
\pdfoverlay_resume_output: 3, 183, 213	
\pdfoverlay_set_graphics_- options:n ... 2, 119, 193	
\pdfoverlay_set_page:n ... 4	
\pdfoverlay_set_pdf:n ... 2, 103, 189	
\pdfoverlay_skip_to_page:n 3, 166, 205	
pdfoverlay internal commands:	
\g__pdfoverlay_action_pending_- bool 9, 35, 61, 113, 138, 152, 170, 176	
__pdfoverlay_check_page:nTF ... 79	
__pdfoverlay_check_page_p:n ... 79	
__pdfoverlay_format_pdf_page: ... 58, 64	
\g__pdfoverlay_graphics_options_- clist ... 5, 36, 73, 121, 122	
__pdfoverlay_if_page_exists:n ... 79	
__pdfoverlay_if_page_exists:nTF ... 129, 168	
\g__pdfoverlay_output_active_- bool ... 34, 48, 112, 181, 185	
 	
__pdfoverlay_output_pdf_page: ...	3, 7, 7, 44
\g__pdfoverlay_page_count_int ...	32, 53, 89, 98, 110, 140, 158, 164
\g__pdfoverlay_page_int ...	33, 42, 52, 60, 111, 124, 133, 139, 141, 147, 157, 175
\g__pdfoverlay_pdf_file_name_str	31, 50, 74, 81, 96, 107
\pdfoverlayIncludeToLastPage ... 3, 199	
\pdfoverlayIncludeToPage ... 3, 195	
\pdfoverlayPauseOutput ... 3, 207	
\pdfoverlayResumeOutput ... 3, 211	
\pdfoverlaySetGraphicsOptions ... 2, 191	
\pdfoverlaySetPDF ... 2, 21, 187	
\pdfoverlaySkipToPage ... 3, 203	
prg commands:	
\prg_new_conditional:Npnn ... 79	
\prg_return_false: ... 84, 99	
\prg_return_true: ... 92	
\ProvidesExplPackage	4
R	
\RequirePackage	3, 6
S	
str commands:	
\str_gset:Nn ... 107	
\str_if_empty:NTF ... 81	
\str_if_empty_p:N ... 50	
\str_new:N ... 31	
\str_use:N ... 96	
T	
TeX and L ^A T _E X 2 ϵ commands:	
\AM@currentdocname ... 108	
\AM@getpagecount ... 109	
\AM@pagecount ... 110	
U	
use commands:	
\use:n ... 70	