

# The pdfmanagement-firstaid package – temporary patches and package replacements L<sup>A</sup>T<sub>E</sub>X PDF management testphase bundle

The L<sup>A</sup>T<sub>E</sub>X Project\*

Version 0.96f, released 2024-03-01

## 1 pdfmanagement-firstaid documentation

This code is temporary! It tries to patch commands of other packages or even replace package which are incompatible with the pdfmanagement, to remove clashes and test if everything works as expected. This code should disappear when packages adapt to the central interfaces.

The package contains an number of sections for various packages. Every section can be disabled in (the first) `\DocumentMetadata` with `debug={firstaidoff={name1,name2,...},...}`.

```
1 <*package>
2 \ProvidesExplPackage{pdfmanagement-firstaid}{2024-03-01}{0.96f}
3   {LaTeX PDF management testphase bundle / firstaid-patches}
4
5 <@=pdfmanagement>
6 \clist_map_inline:nn
7   {pgf,transparent,xmp,pdflscape,xcolor,color,
8     beamer,output,colorspace,fontspec,luacolor}
9   {
10    \bool_new:c      { g__pdfmanagement_firstaid_#1_bool }
11    \bool_gset_true:c { g__pdfmanagement_firstaid_#1_bool }
12  }
13 \clist_map_inline:Nn \g__pdfmanagement_firstaidoff_clist
14  {
15    \bool_if_exist:cT { g__pdfmanagement_firstaid_#1_bool }
16    {
17      \bool_gset_false:c { g__pdfmanagement_firstaid_#1_bool }
18    }
19  }
20 \msg_new:nnn { pdfmanagement } { firstaid }
21             { loading~pdfmanagement~firstaid~code~for~#1 }
22 \msg_new:nnn { pdfmanagement } { firstaid-changed }
23             { package~#1~has~changed.~Check~if~patch~is~still~valid! }
24 \msg_new:nnn { pdfmanagement } { firstaid-disabled }
25             { The~loading~of~package~#1~is~disabled.\\
26             It~is~not~compatible~with~the~PDF~management. }
```

---

\*E-mail: [latex-team@latex-project.org](mailto:latex-team@latex-project.org)

```

27 \msg_new:nnn { pdfmanagement } { firstaid-too-old }
28           { Package-#1-is-too-old-and-not-compatible.\\
29             Get-at-least-version-#2.}
30

```

## 1.1 Fontspec

When using luaLaTeX opacity is broken (issue #30) as fontspec/luatex doesn't use the pdfmanagement but write to ExtGState directly To repair this some new lua code is needed. It also needs a new fontspec and a change in the color key.

```

31 \sys_if_engine luatex:T
32   {
33     \bool_if:NT \g__pdfmanagement_firstaid_fontspec_bool
34       {
35         \AddToHook{package/fontspect/after}
36           {
37             \@ifpackagelater{fontspec}{2023-11-05}
38               {\typeout{PDF-Management:~fontspec-firstaid-ignored}}
39             {
40               \def\c__fontspec_opacity_tl{}
41               \__fontspec_keys_define_code:nnn {fontspec} {Color}
42                 {
43                   \color_if_exist:nTF {#1}
44                     {
45                       \tl_set:Nn \l__fontspec_hexcol_tl {#1}
46                     }
47                   {
48                     \cs_if_exist:cTF { \token_to_str:N \color@ #1 }
49                       {
50                         \convertcolorspec{named}{#1}{HTML}\l__fontspec_hexcol_tl
51                       }
52                       {
53                         \int_compare:nTF { \tl_count:n {#1} == 6 }
54                           { \tl_set:Nn \l__fontspec_hexcol_tl {#1} }
55                           {
56                             \int_compare:nTF { \tl_count:n {#1} == 8 }
57                               { \fontspec_parse_colour:viii #1 }
58                               {
59                                 \bool_if:NF \l__fontspec_firsttime_bool
60                                   { \__fontspec_warning:nx {bad-colour} {#1} }
61                               }
62                             }
63                           }
64                         }
65                       }
66

```

add a comma

```

66     \__fontspec_keys_define_code:nnn {fontspec} {Opacity}
67       {
68         \int_set:Nn \l__fontspec_tmp_int {255}
69         \__fontspec_int_mult_truncate:Nn \l__fontspec_tmp_int { #1 }
70         \tl_if_eq:NMF \l__fontspec_opacity_tl \c__fontspec_opacity_tl
71           {
72             \bool_if:NF \l__fontspec_firsttime_bool

```

```

73         { \_fontspec_warning:nx {opa-twice} {#1} }
74     }
75     \tl_set:Nx \l__fontspec_opacity_tl
76     {
77         ,
78         \int_compare:nT { \l__fontspec_tmp_int <= "F } {0} % zero pad
79         \int_to_hex:n { \l__fontspec_tmp_int }
80     }
81 }

```

and a brace

```

82     \cs_set:Nn \_fontspec_get_features:n
83     {
84         \_fontspec_init_fontface:
85         \_fontspec_keys_set_known:nxN {fontspec-renderer} {\l__fontspec_fontfeat
86         \l__fontspec_keys_leftover_clist
87         \_fontspec_keys_set_known:nxN {fontspec} {\l__fontspec_keys_leftover_clist
88         \keys_set:nV {fontspec-opentype} \l__fontspec_keys_leftover_clist
89
90         \tl_if_empty:NF \l__fontspec_mapping_tl
91         { \_fontspec_update_featstr:n { mapping = \l__fontspec_mapping_tl } }
92
93         \str_if_eq:eeF { \l__fontspec_hexcol_tl \l__fontspec_opacity_tl }
94         { \c__fontspec_hexcol_tl \c__fontspec_opacity_tl }
95         { \_fontspec_update_featstr:n { color = {\l__fontspec_hexcol_tl\l__font
96     }
97 }
98 }
99 }
100 }

```

## 1.2 beamer

beamer made use of the now unsupported syntax `linkbordercolor={.5 .5 .5}`. This has been changed so the patch has been removed.

## 1.3 color

`color` is not incompatible, but the new `hyperref` driver makes use of `l3color` to set the colors. It is therefore necessary to patch some internal `color` commands, so that colors defined with its `\definecolor` command are known to `l3color` and so `hyperref`. This only supports the color models from `l3color` (which covers all standard model of the `color` package). The named model is mapped to `\color_set:nn`.

This change serves also as test to check if this change can be safely added to `color` later.

```

101 \bool_if:NT \g__pdfmanagement_firstaid_color_bool
102 {
103     \declare@file@substitution{color.sty}{color-ltx.sty}
104 }

```

## 1.4 xcolor

xcolor is not incompatible, but the new hyperref driver makes use of l3color to set the colors. It is therefore necessary to patch xcolor, so that colors defined with its `\definecolor` command are known to l3color and so hyperref. This only supports the color model from l3color. Colors defined with the models `cmY` and `tHsb` are silently ignored.

The named model is mapped to `\color_set:nn`.

```
105 \bool_if:NT \g__pdfmanagement_firstaid_xcolor_bool
106   {
107     \AddToHook
108       {
109         package/xcolor/after
110       }
111     {\RequirePackage{xcolor-patches-tmp-ltx}}
```

The patch must be before color definitions are loaded, which will happen in hooks in a newer xcolor:

```
112   \DeclareHookRule{package/xcolor/after}{pdfmanagement-firstaid}{before}{xcolor}
113 }
```

## 1.5 luacolor

The luacolor package doesn't take colors from l3color into account. We add a fix, but only for pdf mode. luacolor can disable the code by clearing the hook if needed.

```
114 \bool_lazy_all:nT
115   {
116     {\sys_if_engine luatex_p:}
117     {\g__pdfmanagement_firstaid_luacolor_bool}
118     {\sys_if_output_pdf_p:}
119   }
120   {
121     \AddToHook{package/luacolor/after}
122     {
123       \cs_set_protected:Npn \__color_backend_select:nn #1#2
124       {
125         \tl_set:Nn \l__color_backend_fill_tl {#1}
126         \tl_set:Nn \l__color_backend_stroke_tl {#2}
127         \LuaCol@setattribute\LuaCol@Attribute
128           {
129             \directlua
130               {
131                 oberdiek.luacolor.get("\luaescapestring{#1~#2}")
132               }
133           }
134       }
135       \cs_set_protected:Npn \__color_backend_fill:n #1
136       {
137         \tl_set:Nn \l__color_backend_fill_tl {#1}
138         \LuaCol@setattribute\LuaCol@Attribute
139           {
140             \directlua
141               {
142                 oberdiek.luacolor.get("\luaescapestring{#1}")

```

```

143         }
144     }
145 }
146 \cs_set_protected:Npn \__color_backend_stroke:n #1
147 {
148     \tl_set:Nn \l__color_backend_stroke_tl {#1}
149     \LuaCol@setattribute\LuaCol@Attribute
150     {
151         \directlua
152         {
153             oberdiek.luacolor.get("\luaescapestring{#1}")
154         }
155     }
156 }
157 \cs_set_protected:Npn \__color_backend_reset: {}
158 \cs_set_eq:NN \__color_backend_fill_reset: \__color_backend_reset:
159 \cs_set_eq:NN \__color_backend_stroke_reset: \__color_backend_reset:
160 }
161 }

```

## 1.6 pgf

In `pgf`, resource management is set up in the file `pgfutil-common.tex`. This then provides three functions for adding to the resources, all of which are objects:

- `\pgfutil@addpdfresource@extgs`: Extended graphics state
- `\pgfutil@addpdfresource@colorspaces`: Color spaces
- `\pgfutil@addpdfresource@patterns`: Patterns

These resource dictionaries are used by adding entries in a cumulative sense; the macro layer deals with ensuring that each entry is only given once. Note that the objects themselves must be given only once for each page.

To support these functions, there are a series of set-up macros which install these resources. That has to take place for every page: the exact route therefore depends on the driver.

For the `pdfmanagement` project we need to avoid that `pgf` interferes in `ExtGState`, `ColorSpace` and `Pattern` (Shadings are added to the `xform` resources and so probably unproblematic for now). The actual patch is in a file hook guarded by the boolean, the rest of the code is always defined.

```

162
163 \bool_if:NT \g__pdfmanagement_firstaid_pgf_bool
164 {
165     \msg_info:nnn{pdfmanagement }{firstaid}{pgf}
166     \AddToHook
167     {
168         file/pgfrcs.sty/after
169     }
170     {
171         \cs_set_eq:NN
172         \__pdfmanagement_pgfori_pgfutil@setuppdfresources
173         \pgfutil@setuppdfresources
174         \def\pgfutil@setuppdfresources

```

```

175     {
176         \pdfmanagement_if_active:TF
177         {
178             \__pdfmanagement_pgf_sys_setuppdfresources_plain:
179         }
180         {
181             \__pdfmanagement_pgfori_pgfulil@setuppdfresources
182         }
183     }
184 }
185 }
186 %\def\pgfutil@addpdfresource@extgs#1{\pgf@sys@addpdfresource@extgs@plain{#1}}
187 %\def\pgfutil@addpdfresource@colorspaces#1{\pgf@sys@addpdfresource@colorspaces@plain{#1}}
188 %\def\pgfutil@addpdfresource@patterns#1{\pgf@sys@addpdfresource@patterns@plain{#1}}
189 %\def\pgfutil@setuppdfresources{\pgf@sys@setuppdfresources@plain}
190 % \pgf@sys@pdf@possible@resources %used in xform
191 %Trying to patch pgf ..
192 \cs_new_protected:Npn \__pdfmanagement_pgf_sys_setuppdfresources_plain:
193 {
194     %objects are already created ...
195     \def\pgf@sys@pdf@possible@resources
196     {
197         /ColorSpace~\pdf_object_ref:n {__pdf/Page/Resources/ColorSpace}
198         /Pattern ~\pdf_object_ref:n {__pdf/Page/Resources/Pattern}
199         /ExtGState ~\pdf_object_ref:n {__pdf/Page/Resources/ExtGState}
200     }
201     \let\pgf@sys@pdf@check@resources=\relax%
202     %not sure if needed, but perhaps the lists are used somewhere else ...
203     \let\pgf@sys@pgf@resource@list@extgs=\pgfutil@empty%
204     \let\pgf@sys@pgf@resource@list@patterns=\pgfutil@empty%
205     \let\pgf@sys@pgf@resource@list@colorspaces=\pgfutil@empty%
206     % the commands to add page resources
207     \def\pgf@sys@addpdfresource@extgs@plain##1
208     {
209         %\exp_after:wN %for transparent which passes a command
210         \__pdfmanagement_patch_pgfextgs:w ##1\q_stop
211     }
212     \def\pgf@sys@addpdfresource@patterns@plain##1
213     {
214         \__pdfmanagement_patch_pgfpatterns:w ##1\q_stop
215     }
216     \def\pgf@sys@addpdfresource@colorspaces@plain##1
217     {
218         \__pdfmanagement_patch_pgfcolorspaces:w ##1\q_stop
219     }
220 }
221
222 %\AtEndPreamble{\pgfutil@setuppdfresources}
223 % helper commands as pgf doesn't pass resources as two arguments
224 % code to add to the resources existing stuff in the format "/name value":
225 \cs_new:Npn \__pdfmanagement_split_dict_entry_aux:NNw #1 #2 /#3-#4\q_stop
226 {
227     \tl_set:Nn #1 {#3}
228     \tl_set:Nn #2 {#4}

```

```

229   }
230
231 \cs_generate_variant:Nn \tl_trim_spaces:n{V}
232 \cs_generate_variant:Nn \pdfmanagement_add:nnn {nee}
233 \cs_new:Npn \__pdfmanagement_patch_pgftextgs:w #1/#2<<#3>>#4\q_stop
234   {
235     \tl_set:Ne\l_tmpa_tl{#2}
236     \pdfmanagement_add:nee
237       {Page/Resources/ExtGState}{\tl_trim_spaces:V\l_tmpa_tl}{<<#3 #4>>}
238   }
239 \cs_new:Npn \__pdfmanagement_patch_pgfpatterns:w #1/#2\space#3\q_stop
240   {
241     \pdfmanagement_add:nee
242       {Page/Resources/Pattern}{\tl_trim_spaces:n{#2}}{#3}
243   }
244 \cs_new:Npn \__pdfmanagement_patch_pgfcolorspaces:w #1/#2[#3]#4\q_stop
245   {
246     \pdfmanagement_add:nee
247       {Page/Resources/ColorSpace}{\tl_trim_spaces:n{#2}}{[#3]}
248   }
249

```

## 1.7 transparent

We check if the new version is used and issue a warning otherwise

```

250 \bool_if:NT \g__pdfmanagement_firstaid_transparent_bool
251   {
252     \AddToHook{package/transparent/after}
253       {
254         \@ifpackagelater{transparent}{2022-10-27}{}
255         {
256           \msg_warning:nnnn{pdfmanagement}{firstaid-too-old}
257             {transparent}
258             {1.5~from~2022-10-27}
259         }
260       }
261   }

```

## 1.8 pdflscape

We check if the new version is used and issue a warning otherwise

```

262 \bool_if:NT \g__pdfmanagement_firstaid_pdflscape_bool
263   {
264     \AddToHook{package/pdflscape/after}
265       {
266         \@ifpackagelater{pdflscape}{2022-10-27}{}
267         {
268           \msg_warning:nnnn{pdfmanagement}{firstaid-too-old}
269             {pdflscape}
270             {0.13~from~2022-10-27}
271         }
272       }
273   }

```

## 1.9 xmp

This handles the new xmp code.

```

274 \bool_if:NT \g_pdfmanagement_firstaid_xmp_bool
275 {
276   \disable@package@load{hyperxmp}{\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{hyperxm
277   \disable@package@load{pdfx}    {\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{pdfx}}
278   \AddToHook{package/doclicense/after}
279   {
280     \AtBeginDocument
281     {
282       \hypersetup
283       {
284         pdfcopyright = {\doclicenseLongTextForHyperref},
285         pdflicenseurl = {\doclicenseURL},
286       }
287     }
288   }
289 }
290 \</package>

```

## 1.10 colorspace

This is rather difficult as no real places to inject patches at first a try to avoid that its ExtGState is missing: it can not be avoided to recreate the objects (and so to get duplicates) as colorspace uses temporary macros whose contents is lost.

```

291 \<package>
292 \bool_if:NT \g_pdfmanagement_firstaid_colorspace_bool
293 {
294   \AddToHook
295   {
296     package/colorspace/after
297   }
298   {\RequirePackage{colorspace-patches-tmp-ltx}}
299 }
300 \</package>

```

# 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	B
\\ .....	<b>bool</b> commands:
	\bool_gset_false:N ..... 17
	\bool_gset_true:N ..... 11
	\bool_if:NTF ..... 33,
	59, 72, 101, 105, 163, 250, 262, 274, 292
	\bool_if_exist:NTF ..... 15
	\bool_lazy_all:nTF ..... 114
	\bool_new:N ..... 10
<b>A</b>	
\AddToHook .....	
. 35, 107, 121, 166, 252, 264, 278, 294	
\AtBeginDocument .....	280
\AtEndPreamble .....	222



<b>C</b>	
clist commands:	
\clist_map_inline:Nn	13
\clist_map_inline:nn	6
color commands:	
\color_if_exist:nTF	43
\color_set:nn	3, 4
color internal commands:	
\__color_backend_fill:n	135
\__color_backend_fill_reset:	158
\l__color_backend_fill_tl	125, 137
\__color_backend_reset:	157, 158, 159
\__color_backend_select:nn	123
\__color_backend_stroke:n	146
\__color_backend_stroke_reset:	159
\l__color_backend_stroke_tl	126, 148
\convertcolorspec	50
cs commands:	
\cs_generate_variant:Nn	231, 232
\cs_if_exist:NTF	48
\cs_new:Npn	225, 233, 239, 244
\cs_new_protected:Npn	192
\cs_set:Nn	82
\cs_set_eq:NN	158, 159, 171
\cs_set_protected:Npn	123, 135, 146, 157
<b>D</b>	
\DeclareHookRule	112
\def	40, 174, 186, 187, 188, 189, 195, 207, 212, 216
\definecolor	3, 4
\directlua	129, 140, 151
\doclicenseLongTextForHyperref	284
\doclicenseURL	285
\DocumentMetadata	1
<b>E</b>	
exp commands:	
\exp_after:wN	209
<b>F</b>	
fontspec commands:	
\fontspec_parse_colour:niii	57
fontspec internal commands:	
\l__fontspec_firsttime_bool	59, 72
\l__fontspec_fontfeat_clist	85
\__fontspec_get_features:n	82
\c__fontspec_hexcol_tl	94
\l__fontspec_hexcol_tl	45, 50, 54, 93, 95
\__fontspec_init_fontface:	84
\__fontspec_int_mult_truncate:Nn	69
\__fontspec_keys_define_code:nnn	41, 66
\l__fontspec_keys_leftover_clist	86, 87, 88
\__fontspec_keys_set_known:nnN	85, 87
\l__fontspec_mapping_tl	90, 91
\c__fontspec_opacity_tl	40, 70, 94
\l__fontspec_opacity_tl	70, 75, 93, 95
\l__fontspec_tmp_int	68, 69, 78, 79
\__fontspec_update_featstr:n	91, 95
\__fontspec_warning:nn	60, 73
<b>H</b>	
\hypersetup	282
<b>I</b>	
int commands:	
\int_compare:nTF	53, 56, 78
\int_set:Nn	68
\int_to_hex:n	79
<b>K</b>	
keys commands:	
\keys_set:nn	88
<b>L</b>	
\let	201, 203, 204, 205
\luaescapestring	131, 142, 153
<b>M</b>	
msg commands:	
\msg_info:nnn	165
\msg_new:nnn	20, 22, 24, 27
\msg_warning:nnn	276, 277
\msg_warning:mnnn	256, 268
<b>P</b>	
pdf commands:	
\pdf_object_ref:n	197, 198, 199
pdfmanagement commands:	
\pdfmanagement_add:nnn	232, 236, 241, 246
\pdfmanagement_if_active:TF	176
pdfmanagement internal commands:	
\g__pdfmanagement_firstaid_color_bool	101
\g__pdfmanagement_firstaid_colorspace_bool	292
\g__pdfmanagement_firstaid_fontspec_bool	33
\g__pdfmanagement_firstaid_luacolor_bool	117
\g__pdfmanagement_firstaid_pdflscape_bool	262
\g__pdfmanagement_firstaid_pgf_bool	163

<code>\g_pdfmanagement_firstaid_</code>		<code>\color@</code>	48
<code>transparent_bool</code>	250	<code>\declare@file@substitution</code>	103
<code>\g_pdfmanagement_firstaid_</code>		<code>\disable@package@load</code>	276, 277
<code>xcolor_bool</code>	105	<code>\LuaCol@Attribute</code>	127, 138, 149
<code>\g_pdfmanagement_firstaid_xmp_</code>		<code>\LuaCol@setattribute</code>	127, 138, 149
<code>bool</code>	274	<code>\pgf@sys@addpdfresource@colorspaces@plain</code>	187, 216
<code>\g_pdfmanagement_firstaidoff_</code>		<code>\pgf@sys@addpdfresource@extgs@plain</code>	186, 207
<code>clist</code>	13	<code>\pgf@sys@addpdfresource@patterns@plain</code>	188, 212
<code>\_pdfmanagement_patch_pgfcolorspaces:w</code>	218, 244	<code>\pgf@sys@pdf@check@resources</code>	201
<code>\_pdfmanagement_patch_pgfextgs:w</code>	210, 233	<code>\pgf@sys@pdf@possible@resources</code>	190, 195
<code>\_pdfmanagement_patch_pgfpatterns:w</code>	214, 239	<code>\pgf@sys@pgf@resource@list@colorspaces</code>	205
<code>\_pdfmanagement_pgf_sys_</code>		<code>\pgf@sys@pgf@resource@list@extgs</code>	203
<code>setuppdfresources_plain:</code>	178, 192	<code>\pgf@sys@pgf@resource@list@patterns</code>	204
<code>\_pdfmanagement_pgfori_pgful@setuppdfresources:</code>	172, 181	<code>\pgf@sys@setuppdfresources@plain</code>	189
<code>\_pdfmanagement_split_dict_</code>		<code>\pgfutil@addpdfresource@colorspaces</code>	5, 187
<code>entry_aux:NNw</code>	225	<code>\pgfutil@addpdfresource@extgs</code>	5, 186
<code>\ProvidesExplPackage</code>	2	<code>\pgfutil@addpdfresource@patterns</code>	5, 188
<b>Q</b>			
quark commands:		<code>\pgfutil@empty</code>	203, 204, 205
<code>\q_stop</code>	210, 214, 218, 225, 233, 239, 244	<code>\pgfutil@setuppdfresources</code>	173, 174, 189, 222
<b>R</b>			
<code>\relax</code>	201	tl commands:	
<code>\RequirePackage</code>	111, 298	<code>\tl_count:n</code>	53, 56
<b>S</b>			
<code>\space</code>	239	<code>\tl_if_empty:NTF</code>	90
str commands:		<code>\tl_if_eq:NNTF</code>	70
<code>\str_if_eq:nnTF</code>	93	<code>\tl_set:Nn</code>	45, 54, 75, 125, 126, 137, 148, 227, 228, 235
sys commands:		<code>\tl_trim_spaces:n</code>	231, 237, 242, 247
<code>\sys_if_engine luatex:TF</code>	31	<code>\l_tmpa_tl</code>	235, 237
<code>\sys_if_engine luatex:p:</code>	116	token commands:	
<code>\sys_if_output_pdf:p:</code>	118	<code>\token_to_str:N</code>	48
<b>T</b>			
T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:		<code>\typeout</code>	38
<code>\@ifpackagelater</code>	37, 254, 266		