

# 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 0.95x, released 2023-03-09

## 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}{2023-03-09}{0.95x}
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/luatload 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         \@ifpackagelater{fontspec}{2023-03-10}
36         {
37           \__fontspec_keys_define_code:nnn {fontspec} {Color}
38           {
39             \tl_set:Nn \l__fontspec_hexcol_tl {#1}
40           }
41         }{}
42       }
43   }

```

## 1.2 Output routine

The patches for the LaTeX output routine commands for the tagging of the header/footer, the suppressing of links and the patch to multicol for the para tagging by default, but must be requested with the key `testphase=new-or-1`. But until latex-lab has been updated we must ensure that they are loaded with the various phases.

```

44 \bool_if:NT \g__pdfmanagement_firstaid_output_bool
45   {

```

This catches also phase-II and tagpdf.

```

46     \file_if_exist:nF {new-or-1-latex-lab-testphase.ltx}
47     {
48       \AddToHook{package/latex-lab-testphase-new-or/after}
49       {\RequirePackage{output-patches-tmp-ltx}}

```

This is for phase-I

```

50         \AddToHook{package/tagpdf/after}
51         {\RequirePackage{output-patches-tmp-ltx}}
52     }
53   }

```

## 1.3 beamer

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

## 1.4 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 patch serves also as test to check if this change can be safely added to color later.

```
54 \bool_if:NT \g__pdfmanagement_firstaid_color_bool
55   {
56     \declare@file@substitution{color.sty}{color-ltx.sty}
57   }
```

## 1.5 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 `cm` and `tHsb` are silently ignored.

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

```
58 \bool_if:NT \g__pdfmanagement_firstaid_xcolor_bool
59   {
60     \AddToHook
61       {
62         package/xcolor/after
63       }
64     {\RequirePackage{xcolor-patches-tmp-ltx}}
65   }
```

## 1.6 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.

```
66 \bool_lazy_all:nT
67   {
68     {\sys_if_engine luatex_p:}
69     {\g__pdfmanagement_firstaid_luacolor_bool}
70     {\sys_if_output_pdf_p:}
71   }
72   {
73     \AddToHook{package/luacolor/after}
74     {
75       \cs_set_protected:Npn \__color_backend_select:nn #1#2
76       {
77         \tl_set:Nn \l__color_backend_fill_tl {#1}
78         \tl_set:Nn \l__color_backend_stroke_tl {#2}
79         \LuaCol@setattribute\LuaCol@Attribute
80         {
81           \directlua
82             {
83               oberdiek.luacolor.get("\luaescapestring{#1~#2}")
```

```

84         }
85     }
86 }
87 \cs_set_protected:Npn \__color_backend_fill:n #1
88 {
89     \tl_set:Nn \l__color_backend_fill_tl {#1}
90     \LuaCol@setattribute\LuaCol@Attribute
91     {
92         \directlua
93         {
94             oberdiek.luacolor.get("\luaescapestring{#1}")
95         }
96     }
97 }
98 \cs_set_protected:Npn \__color_backend_stroke:n #1
99 {
100     \tl_set:Nn \l__color_backend_stroke_tl {#1}
101     \LuaCol@setattribute\LuaCol@Attribute
102     {
103         \directlua
104         {
105             oberdiek.luacolor.get("\luaescapestring{#1}")
106         }
107     }
108 }
109 \cs_set_protected:Npn \__color_backend_reset: {}
110 \cs_set_eq:NN \__color_backend_fill_reset: \__color_backend_reset:
111 \cs_set_eq:NN \__color_backend_stroke_reset: \__color_backend_reset:
112 }
113 }

```

## 1.7 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.

```

114
115 \bool_if:NT \g__pdfmanagement_firstaid_pgf_bool

```

```

116 {
117   \msg_info:nnn{pdfmanagement }{firstaid}{pgf}
118   \AddToHook
119     {
120       file/pgfrcs.sty/after
121     }
122     {
123       \cs_set_eq:NN
124         \__pdfmanagement_pgfori_pgfulil@setuppdfresources
125         \pgfutil@setuppdfresources
126       \def\pgfutil@setuppdfresources
127         {
128           \pdfmanagement_if_active:TF
129             {
130               \__pdfmanagement_pgf_sys_setuppdfresources_plain:
131             }
132             {
133               \__pdfmanagement_pgfori_pgfulil@setuppdfresources
134             }
135           }
136         }
137     }
138   %\def\pgfutil@addpdfresource@extgs#1{\pgf@sys@addpdfresource@extgs@plain{#1}}
139   %\def\pgfutil@addpdfresource@colorspaces#1{\pgf@sys@addpdfresource@colorspaces@plain{#1}}
140   %\def\pgfutil@addpdfresource@patterns#1{\pgf@sys@addpdfresource@patterns@plain{#1}}
141   %\def\pgfutil@setuppdfresources{\pgf@sys@setuppdfresources@plain}
142   % \pgf@sys@pdf@possible@resources %used in xform
143   %Trying to patch pgf ..
144   \cs_new_protected:Npn \__pdfmanagement_pgf_sys_setuppdfresources_plain:
145     {
146       %objects are already created ...
147       \def\pgf@sys@pdf@possible@resources
148         {
149           /ColorSpace~\pdf_object_ref:n {__pdf/Page/Resources/ColorSpace}
150           /Pattern ~\pdf_object_ref:n {__pdf/Page/Resources/Pattern}
151           /ExtGState ~\pdf_object_ref:n {__pdf/Page/Resources/ExtGState}
152         }
153       \let\pgf@sys@pdf@check@resources=\relax%
154       %not sure if needed, but perhaps the lists are used somewhere else ...
155       \let\pgf@sys@pgf@resource@list@extgs=\pgfutil@empty%
156       \let\pgf@sys@pgf@resource@list@patterns=\pgfutil@empty%
157       \let\pgf@sys@pgf@resource@list@colorspaces=\pgfutil@empty%
158       % the commands to add page resources
159       \def\pgf@sys@addpdfresource@extgs@plain##1
160         {
161           %\exp_after:wN %for transparent which passes a command
162           \__pdfmanagement_patch_pgfextgs:w ##1\q_stop
163         }
164       \def\pgf@sys@addpdfresource@patterns@plain##1
165         {
166           \__pdfmanagement_patch_pgfpatterns:w ##1\q_stop
167         }
168       \def\pgf@sys@addpdfresource@colorspaces@plain##1
169         {

```

```

170     \_pdfmanagement_patch_pgfcolorspaces:w ##1\q_stop
171   }
172 }
173
174 %\AtEndPreamble{\pgfutil@setuppdfresources}
175 % helper commands as pgf doesn't pass resources as two arguments
176 % code to add to the resources existing stuff in the format "/name value":
177 \cs_new:Npn \_pdfmanagement_split_dict_entry_aux:NNw #1 #2 /#3-#4\q_stop
178 {
179   \tl_set:Nn #1 {#3}
180   \tl_set:Nn #2 {#4}
181 }
182
183 \cs_generate_variant:Nn \tl_trim_spaces:n{V}
184 \cs_generate_variant:Nn \pdfmanagement_add:nex
185 \cs_new:Npn \_pdfmanagement_patch_pgftextgs:w #1/#2<<#3>>#4\q_stop
186 {
187   \tl_set:Nx\l_tmpa_tl{#2}
188   \pdfmanagement_add:nex
189     {Page/Resources/ExtGState}{\tl_trim_spaces:V\l_tmpa_tl}{<<#3 #4>>}
190 }
191 \cs_new:Npn \_pdfmanagement_patch_pgfpatterns:w #1/#2\space#3\q_stop
192 {
193   \pdfmanagement_add:nex
194     {Page/Resources/Pattern}{\tl_trim_spaces:n{#2}}{#3}
195 }
196 \cs_new:Npn \_pdfmanagement_patch_pgfcolorspaces:w #1/#2[#3]#4\q_stop
197 {
198   \pdfmanagement_add:nex
199     {Page/Resources/ColorSpace}{\tl_trim_spaces:n{#2}}{[#3]}
200 }
201

```

## 1.8 transparent

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

```

202 \bool_if:NT \g__pdfmanagement_firstaid_transparent_bool
203 {
204   \AddToHook{package/transparent/after}
205   {
206     \@ifpackagelater{transparent}{2022-10-27}{}
207     {
208       \msg_warning:nnn{pdfmanagement}{firstaid-too-old}
209       {transparent}
210       {1.5~from~2022-10-27}
211     }
212   }
213 }

```

## 1.9 pdfscape

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

```

214 \bool_if:NT \g__pdfmanagement_firstaid_pdfscape_bool

```

```

215 {
216   \AddToHook{package/pdfscape/after}
217   {
218     \@ifpackagelater{pdfscape}{2022-10-27}{
219       {
220         \msg_warning:nnn{pdfmanagement}{firstaid-too-old}
221         {pdfscape}
222         {0.13~from~2022-10-27}
223       }
224     }
225 }

```

## 1.10 xmp

This handles the new xmp code.

```

226 \bool_if:NT \g__pdfmanagement_firstaid_xmp_bool
227 {
228   \disable@package@load{hyperxmp}{\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{hyperxm
229   \disable@package@load{pdfx}    {\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{pdfx}}
230   \AddToHook{package/doclicense/after}
231   {
232     \AtBeginDocument
233     {
234       \hypersetup
235       {
236         pdfcopyright = {\doclicenseLongTextForHyperref},
237         pdflicenseurl = {\doclicenseURL},
238       }
239     }
240   }
241 }
242 \</package>

```

## 1.11 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.

```

243 \<package>
244 \bool_if:NT \g__pdfmanagement_firstaid_colorspace_bool
245 {
246   \AddToHook
247   {
248     package/colorspace/after
249   }
250   {\RequirePackage{colorspace-patches-tmp-ltx}}
251 }
252 \</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			
<code>\</code>	25, 28		
A		E	
<code>\AddToHook</code>	48, 50, 60, 73, 118, 204, 216, 230, 246	exp commands:	
<code>\AtBeginDocument</code>	232	<code>\exp_after:wN</code>	161
<code>\AtEndPreamble</code>	174	F	
B		file commands:	
bool commands:		<code>\file_if_exist:nTF</code>	46
<code>\bool_gset_false:N</code>	17	fontspec internal commands:	
<code>\bool_gset_true:N</code>	11	<code>\l_fontspeg_hexcol_tl</code>	39
<code>\bool_if:NTF</code>	33, 44, 54, 58, 115, 202, 214, 226, 244	<code>\_fontspec_keys_define_code:nnn</code>	37
<code>\bool_if_exist:NTF</code>	15	H	
<code>\bool_lazy_all:nTF</code>	66	<code>\hypersetup</code>	234
<code>\bool_new:N</code>	10	L	
C		<code>\let</code>	153, 155, 156, 157
clist commands:		<code>\luaescapestring</code>	83, 94, 105
<code>\clist_map_inline:Nn</code>	13	M	
<code>\clist_map_inline:nn</code>	6	msg commands:	
color commands:		<code>\msg_info:nnn</code>	117
<code>\color_set:nn</code>	3	<code>\msg_new:nnn</code>	20, 22, 24, 27
color internal commands:		<code>\msg_warning:nnn</code>	228, 229
<code>\_color_backend_fill:n</code>	87	<code>\msg_warning:nnnn</code>	208, 220
<code>\_color_backend_fill_reset:</code>	110	P	
<code>\l_color_backend_fill_tl</code>	77, 89	pdf commands:	
<code>\_color_backend_reset:</code>	109, 110, 111	<code>\pdf_object_ref:n</code>	149, 150, 151
<code>\_color_backend_select:nn</code>	75	pdfmanagement commands:	
<code>\_color_backend_stroke:n</code>	98	<code>\pdfmanagement_add:nnn</code>	184, 188, 193, 198
<code>\_color_backend_stroke_reset:</code>	111	<code>\pdfmanagement_if_active:TF</code>	128
<code>\l_color_backend_stroke_tl</code>	78, 100	pdfmanagement internal commands:	
cs commands:		<code>\g_pdfmanagement_firstaid_color_bool</code>	54
<code>\cs_generate_variant:Nn</code>	183, 184	<code>\g_pdfmanagement_firstaid_colorspace_bool</code>	244
<code>\cs_new:Npn</code>	177, 185, 191, 196	<code>\g_pdfmanagement_firstaid_fontspec_bool</code>	33
<code>\cs_new_protected:Npn</code>	144	<code>\g_pdfmanagement_firstaid_luacolor_bool</code>	69
<code>\cs_set_eq:NN</code>	110, 111, 123	<code>\g_pdfmanagement_firstaid_output_bool</code>	44
<code>\cs_set_protected:Npn</code>	75, 87, 98, 109	<code>\g_pdfmanagement_firstaid_pdflscape_bool</code>	214
D		<code>\g_pdfmanagement_firstaid_pgf_bool</code>	115
<code>\def</code>	126, 138, 139, 140, 141, 147, 159, 164, 168	<code>\g_pdfmanagement_firstaid_transparent_bool</code>	202
<code>\definecolor</code>	3		
<code>\directlua</code>	81, 92, 103		
<code>\doclicenseLongTextForHyperref</code>	236		
<code>\doclicenseURL</code>	237		
<code>\DocumentMetadata</code>	1		



\g_pdfmanagement_firstaid_- xcolor_bool . . . . .	58	
\g_pdfmanagement_firstaid_xmp_- bool . . . . .	226	
\g_pdfmanagement_firstaidoff_- clist . . . . .	13	
\_pdfmanagement_patch_pgfcolorspaces:w . . . . .	170, 196	
\_pdfmanagement_patch_pgftextgs:w . . . . .	162, 185	
\_pdfmanagement_patch_pgfpatterns:w . . . . .	166, 191	
\_pdfmanagement_pgf_sys_- setuppdfresources_plain:	130, 144	
\_pdfmanagement_pgfori_pgful@setuppdfresources . . . . .	124, 133	
\_pdfmanagement_split_dict_- entry_aux:NNw . . . . .	177	
\ProvidesExplPackage . . . . .	2	
<b>Q</b>		
quark commands:		
\q_stop . . . . .	162, 166, 170, 177, 185, 191, 196	
<b>R</b>		
\relax . . . . .	153	
\RequirePackage . . . . .	49, 51, 64, 250	
<b>S</b>		
\space . . . . .	191	
sys commands:		
\sys_if_engine luatex:TF . . . . .	31	
\sys_if_engine luatex_p: . . . . .	68	
\sys_if_output_pdf_p: . . . . .	70	
<b>T</b>		
TeX and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:		
\@ifpackagelater . . . . .	35, 206, 218	
\declare@file@substitution . . . . .	56	
\disable@package@load . . . . .	228, 229	
\LuaCol@Attribute . . . . .	79, 90, 101	
\LuaCol@setattribute . . . . .	79, 90, 101	
\pgf@sys@addpdfresource@colorspaces@plain . . . . .	139, 168	
\pgf@sys@addpdfresource@extgs@plain . . . . .	138, 159	
\pgf@sys@addpdfresource@patterns@plain . . . . .	140, 164	
\pgf@sys@pdf@check@resources . . . . .	153	
\pgf@sys@pdf@possible@resources . . . . .		
. . . . .	142, 147	
\pgf@sys@pgf@resource@list@colorspaces . . . . .	157	
\pgf@sys@pgf@resource@list@extgs . . . . .	155	
\pgf@sys@pgf@resource@list@patterns . . . . .	156	
\pgf@sys@setuppdfresources@plain . . . . .	141	
\pgfutil@addpdfresource@colorspaces . . . . .	4, 139	
\pgfutil@addpdfresource@extgs . . . . .	4, 138	
\pgfutil@addpdfresource@patterns . . . . .	4, 140	
\pgfutil@empty . . . . .	155, 156, 157	
\pgfutil@setuppdfresources . . . . .		
. . . . .	125, 126, 141, 174	
tl commands:		
\tl_set:Nn . . . . .		
. . . . .	39, 77, 78, 89, 100, 179, 180, 187	
\tl_trim_spaces:n . . . . .	183, 189, 194, 199	
\l_tmpa_tl . . . . .	187, 189	