

# 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 0.95z, released 2023-08-29

## 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}{2023-08-29}{0.95z}
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 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 patch serves also as test to check if this change can be safely added to color later.

```

44 \bool_if:NT \g__pdfmanagement_firstaid_color_bool
45   {
46     \declare@file@substitution{color.sty}{color-ltx.sty}
47   }

```

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

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

```

48 \bool_if:NT \g__pdfmanagement_firstaid_xcolor_bool

```

```

49 {
50   \AddToHook
51     {
52       package/xcolor/after
53     }
54     {\RequirePackage{xcolor-patches-tmp-ltx}}
55 }

```

## 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.

```

56 \bool_lazy_all:nT
57 {
58   {\sys_if_engine luatex_p:}
59   {\g_pdfmanagement_firstaid_luacolor_bool}
60   {\sys_if_output_pdf_p:}
61 }
62 {
63   \AddToHook{package/luacolor/after}
64   {
65     \cs_set_protected:Npn \__color_backend_select:nn #1#2
66     {
67       \tl_set:Nn \l__color_backend_fill_tl {#1}
68       \tl_set:Nn \l__color_backend_stroke_tl {#2}
69       \LuaCol@setattribute\LuaCol@Attribute
70       {
71         \directlua
72         {
73           oberdiek.luacolor.get("\luaescapestring{#1~#2}")
74         }
75       }
76     }
77   \cs_set_protected:Npn \__color_backend_fill:n #1
78   {
79     \tl_set:Nn \l__color_backend_fill_tl {#1}
80     \LuaCol@setattribute\LuaCol@Attribute
81     {
82       \directlua
83       {
84         oberdiek.luacolor.get("\luaescapestring{#1}")
85       }
86     }
87   }
88   \cs_set_protected:Npn \__color_backend_stroke:n #1
89   {
90     \tl_set:Nn \l__color_backend_stroke_tl {#1}
91     \LuaCol@setattribute\LuaCol@Attribute
92     {
93       \directlua
94       {
95         oberdiek.luacolor.get("\luaescapestring{#1}")
96       }
97     }
98   }
99 }

```

```

97         }
98     }
99     \cs_set_protected:Npn \__color_backend_reset: {}
100     \cs_set_eq:NN \__color_backend_fill_reset: \__color_backend_reset:
101     \cs_set_eq:NN \__color_backend_stroke_reset: \__color_backend_reset:
102 }
103 }

```

## 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.

```

104
105 \bool_if:NT \g__pdfmanagement_firstaid_pgf_bool
106 {
107     \msg_info:nnn{pdfmanagement}{firstaid}{pgf}
108     \AddToHook
109     {
110         file/pgfrcs.sty/after
111     }
112     {
113         \cs_set_eq:NN
114         \__pdfmanagement_pgfori_pgfutil@setuppdfresources
115         \pgfutil@setuppdfresources
116         \def\pgfutil@setuppdfresources
117         {
118             \pdfmanagement_if_active:TF
119             {
120                 \__pdfmanagement_pgf_sys_setuppdfresources_plain:
121             }
122             {
123                 \__pdfmanagement_pgfori_pgfutil@setuppdfresources
124             }
125         }
126     }
127 }
128 %\def\pgfutil@addpdfresource@extgs#1{\pgf@sys@addpdfresource@extgs@plain{#1}}

```

```

129 %\def\pgfutil@addpdfresource@colorspaces#1{\pgf@sys@addpdfresource@colorspaces@plain{#1}}
130 %\def\pgfutil@addpdfresource@patterns#1{\pgf@sys@addpdfresource@patterns@plain{#1}}
131 %\def\pgfutil@setuppdfresources{\pgf@sys@setuppdfresources@plain}
132 % \pgf@sys@pdf@possible@resources %used in xform
133 %Trying to patch pgf ..
134 \cs_new_protected:Npn \__pdfmanagement_pgf_sys_setuppdfresources_plain:
135 {
136   %objects are already created ...
137   \def\pgf@sys@pdf@possible@resources
138   {
139     /ColorSpace-\pdf_object_ref:n {__pdf/Page/Resources/ColorSpace}
140     /Pattern -\pdf_object_ref:n {__pdf/Page/Resources/Pattern}
141     /ExtGState -\pdf_object_ref:n {__pdf/Page/Resources/ExtGState}
142   }
143   \let\pgf@sys@pdf@check@resources=\relax%
144   %not sure if needed, but perhaps the lists are used somewhere else ...
145   \let\pgf@sys@pgf@resource@list@extgs=\pgfutil@empty%
146   \let\pgf@sys@pgf@resource@list@patterns=\pgfutil@empty%
147   \let\pgf@sys@pgf@resource@list@colorspaces=\pgfutil@empty%
148   % the commands to add page resources
149   \def\pgf@sys@addpdfresource@extgs@plain##1
150   {
151     %\exp_after:wN %for transparent which passes a command
152     \__pdfmanagement_patch_pgfextgs:w ##1\q_stop
153   }
154   \def\pgf@sys@addpdfresource@patterns@plain##1
155   {
156     \__pdfmanagement_patch_pgfpatterns:w ##1\q_stop
157   }
158   \def\pgf@sys@addpdfresource@colorspaces@plain##1
159   {
160     \__pdfmanagement_patch_pgfcolorspaces:w ##1\q_stop
161   }
162 }
163
164 %\AtEndPreamble{\pgfutil@setuppdfresources}
165 % helper commands as pgf doesn't pass resources as two arguments
166 % code to add to the resources existing stuff in the format "/name value":
167 \cs_new:Npn \__pdfmanagement_split_dict_entry_aux:NNw #1 #2 /#3~#4\q_stop
168 {
169   \tl_set:Nn #1 {#3}
170   \tl_set:Nn #2 {#4}
171 }
172
173 \cs_generate_variant:Nn \tl_trim_spaces:n{V}
174 \cs_generate_variant:Nn \pdfmanagement_add:nnn {nex}
175 \cs_new:Npn \__pdfmanagement_patch_pgfextgs:w #1/#2<<#3>>#4\q_stop
176 {
177   \tl_set:Nx\l_tmpa_tl{#2}
178   \pdfmanagement_add:nex
179     {Page/Resources/ExtGState}{\tl_trim_spaces:V\l_tmpa_tl}{<<#3 #4>>}
180 }
181 \cs_new:Npn \__pdfmanagement_patch_pgfpatterns:w #1/#2\space#3\q_stop
182 {

```

```

183     \pdfmanagement_add:nex
184     {Page/Resources/Pattern}{\tl_trim_spaces:n{#2}}{#3}
185   }
186 \cs_new:Npn \__pdfmanagement_patch_pgfcOLORSPACES:w #1/#2[#3]#4\q_stop
187   {
188     \pdfmanagement_add:nex
189     {Page/Resources/ColorSpace}{\tl_trim_spaces:n{#2}}{[#3]}
190   }
191

```

## 1.7 transparent

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

```

192 \bool_if:NT \g__pdfmanagement_firstaid_transparent_bool
193   {
194     \AddToHook{package/transparent/after}
195     {
196       \@ifpackagelater{transparent}{2022-10-27}{}
197       {
198         \msg_warning:nnnn{pdfmanagement}{firstaid-too-old}
199         {transparent}
200         {1.5~from~2022-10-27}
201       }
202     }
203   }

```

## 1.8 pdfscape

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

```

204 \bool_if:NT \g__pdfmanagement_firstaid_pdfscape_bool
205   {
206     \AddToHook{package/pdfscape/after}
207     {
208       \@ifpackagelater{pdfscape}{2022-10-27}{}
209       {
210         \msg_warning:nnnn{pdfmanagement}{firstaid-too-old}
211         {pdfscape}
212         {0.13~from~2022-10-27}
213       }
214     }
215   }

```

## 1.9 xmp

This handles the new xmp code.

```

216 \bool_if:NT \g__pdfmanagement_firstaid_xmp_bool
217   {
218     \disable@package@load{hyperxmp}{\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{hyperxmp}}
219     \disable@package@load{pdfx}{\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{pdfx}}
220     \AddToHook{package/doclicense/after}
221     {
222       \AtBeginDocument
223       {

```

```

224     \hypersetup
225     {
226         pdfcopyright = {\doclicenseLongTextForHyperref},
227         pdflicenseurl = {\doclicenseURL},
228     }
229 }
230 }
231 }
232 \end{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.

```

233 \begin{package}
234 \bool_if:NT \g__pdfmanagement_firstaid_colorspace_bool
235 {
236     \AddToHook
237     {
238         package/colorspace/after
239     }
240     {\RequirePackage{colorspace-patches-tmp-ltx}}
241 }
242 \end{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
<b>A</b>	
<code>\AddToHook</code>	50, 63, 108, 194, 206, 220, 236
<code>\AtBeginDocument</code>	222
<code>\AtEndPreamble</code>	164
<b>B</b>	
bool commands:	
<code>\bool_gset_false:N</code>	17
<code>\bool_gset_true:N</code>	11
<code>\bool_if:NTF</code>	33, 44, 48, 105, 192, 204, 216, 234
<code>\bool_if_exist:NTF</code>	15
<code>\bool_lazy_all:nTF</code>	56
<code>\bool_new:N</code>	10
<b>C</b>	
clist commands:	
<code>\clist_map_inline:Nn</code>	13
<code>\clist_map_inline:nn</code>	6
color commands:	
<code>\color_set:nn</code>	2
color internal commands:	
<code>\_color_backend_fill:n</code>	77
<code>\_color_backend_fill_reset:</code>	100
<code>\l_color_backend_fill_tl</code>	67, 79
<code>\_color_backend_reset:</code>	99, 100, 101
<code>\_color_backend_select:nn</code>	65
<code>\_color_backend_stroke:n</code>	88
<code>\_color_backend_stroke_reset:</code>	101
<code>\l_color_backend_stroke_tl</code>	68, 90
cs commands:	
<code>\cs_generate_variant:Nn</code>	173, 174
<code>\cs_new:Npn</code>	167, 175, 181, 186
<code>\cs_new_protected:Npn</code>	134
<code>\cs_set_eq:NN</code>	100, 101, 113
<code>\cs_set_protected:Npn</code>	65, 77, 88, 99

<b>D</b>	
<code>\def</code> .....	116, 128, 129, 130, 131, 137, 149, 154, 158
<code>\definecolor</code> .....	2
<code>\directlua</code> .....	71, 82, 93
<code>\doclicenseLongTextForHyperref</code> ...	226
<code>\doclicenseURL</code> .....	227
<code>\DocumentMetadata</code> .....	1
<b>E</b>	
exp commands:	
<code>\exp_after:wN</code> .....	151
<b>F</b>	
fontspec internal commands:	
<code>\l_fontspeg_hexcol_tl</code> .....	39
<code>\__fontspec_keys_define_code:nnn</code> ...	37
<b>H</b>	
<code>\hypersetup</code> .....	224
<b>L</b>	
<code>\let</code> .....	143, 145, 146, 147
<code>\luaescapestring</code> .....	73, 84, 95
<b>M</b>	
msg commands:	
<code>\msg_info:nnn</code> .....	107
<code>\msg_new:nnn</code> .....	20, 22, 24, 27
<code>\msg_warning:nnn</code> .....	218, 219
<code>\msg_warning:nnnn</code> .....	198, 210
<b>P</b>	
pdf commands:	
<code>\pdf_object_ref:n</code> .....	139, 140, 141
pdfmanagement commands:	
<code>\pdfmanagement_add:nnn</code> .....	174, 178, 183, 188
<code>\pdfmanagement_if_active:TF</code> ...	118
pdfmanagement internal commands:	
<code>\g__pdfmanagement_firstaid_-color_bool</code> .....	44
<code>\g__pdfmanagement_firstaid_-colorspace_bool</code> .....	234
<code>\g__pdfmanagement_firstaid_-fontspec_bool</code> .....	33
<code>\g__pdfmanagement_firstaid_-luacolor_bool</code> .....	59
<code>\g__pdfmanagement_firstaid_-pdflscape_bool</code> .....	204
<code>\g__pdfmanagement_firstaid_pgf_-bool</code> .....	105
<code>\g__pdfmanagement_firstaid_-transparent_bool</code> .....	192
<code>\g__pdfmanagement_firstaid_-xcolor_bool</code> .....	48
<code>\g__pdfmanagement_firstaid_xmp_-bool</code> .....	216
<code>\g__pdfmanagement_firstaidoff_-clist</code> .....	13
<code>\__pdfmanagement_patch_pgfcolorspaces:w</code> .....	160, 186
<code>\__pdfmanagement_patch_pgfextgs:w</code> .....	152, 175
<code>\__pdfmanagement_patch_pgfpatterns:w</code> .....	156, 181
<code>\__pdfmanagement_pgf_sys_-setuppdfresources_plain:</code> ...	120, 134
<code>\__pdfmanagement_pgforgi_pgfutil@setuppdfresources</code> .....	114, 123
<code>\__pdfmanagement_split_dict_-entry_aux:NNw</code> .....	167
<code>\ProvidesExplPackage</code> .....	2
<b>Q</b>	
quark commands:	
<code>\q_stop</code> .....	152, 156, 160, 167, 175, 181, 186
<b>R</b>	
<code>\relax</code> .....	143
<code>\RequirePackage</code> .....	54, 240
<b>S</b>	
<code>\space</code> .....	181
sys commands:	
<code>\sys_if_engine luatex:TF</code> .....	31
<code>\sys_if_engine luatex_p:</code> .....	58
<code>\sys_if_output_pdf_p:</code> .....	60
<b>T</b>	
$\TeX$ and $\LaTeX 2_{\epsilon}$ commands:	
<code>\@ifpackagelater</code> .....	35, 196, 208
<code>\declare@file@substitution</code> .....	46
<code>\disable@package@load</code> .....	218, 219
<code>\LuaCol@Attribute</code> .....	69, 80, 91
<code>\LuaCol@setattribute</code> .....	69, 80, 91
<code>\pgf@sys@addpdfresource@colorspaces@plain</code> .....	129, 158
<code>\pgf@sys@addpdfresource@extgs@plain</code> .....	128, 149
<code>\pgf@sys@addpdfresource@patterns@plain</code> .....	130, 154
<code>\pgf@sys@pdf@check@resources</code> ..	143
<code>\pgf@sys@pdf@possible@resources</code> ..	132, 137
<code>\pgf@sys@pgf@resource@list@colorspaces</code> .....	147
<code>\pgf@sys@pgf@resource@list@extgs</code> ..	145

