

# 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 0.95p, released 2022-06-14

## 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}{2022-06-14}{0.95p}
3   {LaTeX PDF management testphase bundle / firstaid-patches}
4
5 <@@=pdfmanagement>
6 \clist_map_inline:nn {pgf,transparent,hyperxmp,pdflscape,xcolor,color,beamer,output,colorspac
7   {
8     \bool_new:c          { g__pdfmanagement_firstaid_#1_bool }
9     \bool_gset_true:c { g__pdfmanagement_firstaid_#1_bool }
10  }
11 \clist_map_inline:Nn \g__pdfmanagement_firstaidoff_clist
12   {
13     \bool_if_exist:cT { g__pdfmanagement_firstaid_#1_bool }
14     {
15       \bool_gset_false:c { g__pdfmanagement_firstaid_#1_bool }
16     }
17   }
18 \msg_new:nnn { pdfmanagement } { firstaid }
19             { loading~pdfmanagement~firstaid~code~for~#1 }
20 \msg_new:nnn { pdfmanagement } { firstaid-changed }
21             { package~#1~has~changed.~Check~if~patch~is~still~valid! }
```

### 1.1 Fontspec

When using luaLaTeX opacity is broken (issue #30) as fontspec/luaotfload doesn't use the pdfmanagement but write to ExtGState directly To repair this some new lua code is

---

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

needed. This also improves the color parsing and allows to use cmyk and spot colors as font colors. It is also needed to use a patch version of `fontspec-luatex.sty` to slightly change the way how the Opacity and Color key are handled. The patch is only executed if luaotfload is new enough.

```

22 \sys_if_engine_luatex:T
23 {
24   \lua_now:n
25   {
26     if~not~luaotfload.set_transparent_colorstack~
27     then~
28       tex.print("\bool_set_false:N\g__pdfmanagement_firstaid_fontspec_bool")~
29     end
30   }
31   \bool_if:NT \g__pdfmanagement_firstaid_fontspec_bool
32   {
33     \msg_info:nnn{pdfmanagement}{firstaid}{fontspec}
34     \file_if_exist:nT { luaotfload-l3color-tmp.lua }
35     {
36       \lua_load_module:n { luaotfload-l3color-tmp.lua }
37     }
38
39     \declare@file@substitution{fontspec-luatex.sty}{fontspec-luatex-tmp-ltx.sty}
40     \AddToHook{package/fontspec/after}
41     {
42       \str_if_eq:vnF{ver@fontspec.sty}{2022/01/15~v2.8a~Font~selection~for~XeLaTeX~and~
43       {\msg_warn:nnn{pdfmanagement}{firstaid-changed}{fontspec}}
44     }
45   }
46 }

```

## 1.2 Output routine

This load as few patches for the LaTeX output routine commands. They add hooks for the interruption of links and for paragraph tagging.

```

47 \bool_if:NT \g__pdfmanagement_firstaid_output_bool
48 {
49   \RequirePackage{output-patches-tmp-ltx}
50 }

```

## 1.3 beamer

beamer makes use of the now unsupported syntax `linkbordercolor={.5 .5 .5}`. To avoid an error we define this color as a name.

```

51 \bool_if:NT \g__pdfmanagement_firstaid_beamer_bool
52 {
53   \color_set:nnn{.5~.5~.5}{rgb}{0.5,0.5,0.5}
54 }

```

## 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.

```

55 \bool_if:NT \g__pdfmanagement_firstaid_color_bool
56 {
57   \declare@file@substitution{color.sty}{color-ltx.sty}
58 }

```

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

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

```

59 \bool_if:NT \g__pdfmanagement_firstaid_xcolor_bool
60 {
61   \AddToHook
62     {
63       package/xcolor/after
64     }
65     {\RequirePackage{xcolor-patches-tmp-ltx}\XC@@names}
66 }

```

## 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.

```

67
68 \bool_if:NT \g__pdfmanagement_firstaid_pgf_bool
69 {
70   \msg_info:nnn{pdfmanagement }{firstaid}{pgf}
71   \AddToHook

```

```

72     {
73         file/pgfrcs.sty/after
74     }
75     {
76         \cs_set_eq:NN
77         \__pdfmanagement_pgfori_pgfulil@setuppdfresources
78         \pgfutil@setuppdfresources
79         \def\pgfutil@setuppdfresources
80         {
81             \pdfmanagement_if_active:TF
82             {
83                 \__pdfmanagement_pgf_sys_setuppdfresources_plain:
84             }
85             {
86                 \__pdfmanagement_pgfori_pgfulil@setuppdfresources
87             }
88         }
89     }
90 }
91 %\def\pgfutil@addpdfresource@extgs#1{\pgf@sys@addpdfresource@extgs@plain{#1}}
92 %\def\pgfutil@addpdfresource@colorspaces#1{\pgf@sys@addpdfresource@colorspaces@plain{#1}}
93 %\def\pgfutil@addpdfresource@patterns#1{\pgf@sys@addpdfresource@patterns@plain{#1}}
94 %\def\pgfutil@setuppdfresources{\pgf@sys@setuppdfresources@plain}
95 % \pgf@sys@pdf@possible@resources %used in xform
96 %Trying to patch pgf ..
97 \cs_new_protected:Npn \__pdfmanagement_pgf_sys_setuppdfresources_plain:
98 {
99     %objects are already created ...
100     \def\pgf@sys@pdf@possible@resources
101     {
102         /ColorSpace~\pdf_object_ref:n {__pdf/Page/Resources/ColorSpace}
103         /Pattern ~\pdf_object_ref:n {__pdf/Page/Resources/Pattern}
104         /ExtGState ~\pdf_object_ref:n {__pdf/Page/Resources/ExtGState}
105     }
106     \let\pgf@sys@pdf@check@resources=\relax%
107     %not sure if needed, but perhaps the lists are used somewhere else ...
108     \let\pgf@sys@pgf@resource@list@extgs=\pgfutil@empty%
109     \let\pgf@sys@pgf@resource@list@patterns=\pgfutil@empty%
110     \let\pgf@sys@pgf@resource@list@colorspaces=\pgfutil@empty%
111     % the commands to add page resources
112     \def\pgf@sys@addpdfresource@extgs@plain##1
113     {
114         %\exp_after:wN %for transparent which passes a command
115         \__pdfmanagement_patch_pgfextgs:w ##1\q_stop
116     }
117     \def\pgf@sys@addpdfresource@patterns@plain##1
118     {
119         \__pdfmanagement_patch_pgfpatterns:w ##1\q_stop
120     }
121     \def\pgf@sys@addpdfresource@colorspaces@plain##1
122     {
123         \__pdfmanagement_patch_pgfcolorspaces:w ##1\q_stop
124     }
125 }

```

```

126
127 %\AtEndPreamble{\pgfutil@setuppdfresources}
128 % helper commands as pgf doesn't pass resources as two arguments
129 % code to add to the resources existing stuff in the format "/name value":
130 \cs_new:Npn \__pdfmanagement_split_dict_entry_aux:NNw #1 #2 /#3~#4\q_stop
131 {
132     \tl_set:Nn #1 {#3}
133     \tl_set:Nn #2 {#4}
134 }
135
136 \cs_generate_variant:Nn \tl_trim_spaces:n{V}
137 \cs_generate_variant:Nn \pdfmanagement_add:n{nex}
138 \cs_new:Npn \__pdfmanagement_patch_pgftextgs:w #1/#2<<#3>>#4\q_stop
139 {
140     \tl_set:Nx\l_tmpa_tl{#2}
141     \pdfmanagement_add:nex
142         {Page/Resources/ExtGState}{\tl_trim_spaces:V\l_tmpa_tl}{<<#3 #4>>}
143 }
144 \cs_new:Npn \__pdfmanagement_patch_pgfpatterns:w #1/#2\space#3\q_stop
145 {
146     \pdfmanagement_add:nex
147         {Page/Resources/Pattern}{\tl_trim_spaces:n{#2}}{#3}
148 }
149 \cs_new:Npn \__pdfmanagement_patch_pgfcolorspaces:w #1/#2[#3]#4\q_stop
150 {
151     \pdfmanagement_add:nex
152         {Page/Resources/ColorSpace}{\tl_trim_spaces:n{#2}}{[#3]}
153 }
154

```

## 1.7 transparent

We simply replace by the new version.

```

155 \bool_if:NT \g__pdfmanagement_firstaid_transparent_bool
156 {
157     \declare@file@substitution{transparent.sty}{transparent-ltx.sty}
158 }

```

## 1.8 pdfscape

We simply replace by the new version.

```

159 \bool_if:NT \g__pdfmanagement_firstaid_pdfscape_bool
160 {
161     \declare@file@substitution{pdfscape.sty}{pdfscape-ltx.sty}
162 }

```

## 1.9 hyperxmp

We add some code at the end of hyperxmp.sty.

```

163 \bool_if:NT \g__pdfmanagement_firstaid_hyperxmp_bool
164 {
165     \AddToHook
166     {

```

```

167         file/hyperxmp.sty/after
168     }
169     {\RequirePackage{hyperxmp-patches-tmp-ltx}}
170 }
171 \</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.

```

172 \<*package>
173 \bool_if:NT \g__pdfmanagement_firstaid_colorspace_bool
174 {
175     \AddToHook
176     {
177         package/colorspace/after
178     }
179     {\RequirePackage{colorspace-patches-tmp-ltx}}
180 }
181 \</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.

<b>Symbols</b>		<code>\cs_new:Npn</code> . . . . . 130, 138, 144, 149
<code>\</code> . . . . . 28		<code>\cs_new_protected:Npn</code> . . . . . 97
		<code>\cs_set_eq:NN</code> . . . . . 76
<b>A</b>		<b>D</b>
<code>\AddToHook</code> . . . . . 40, 61, 71, 165, 175		<code>\def</code> . . . 79, 91, 92, 93, 94, 100, 112, 117, 121
<code>\AtEndPreamble</code> . . . . . 127		<code>\definecolor</code> . . . . . 3
<b>B</b>		<code>\DocumentMetadata</code> . . . . . 1
bool commands:		<b>E</b>
<code>\bool_gset_false:N</code> . . . . . 15		exp commands:
<code>\bool_gset_true:N</code> . . . . . 9		<code>\exp_after:wN</code> . . . . . 114
<code>\bool_if:NTF</code> . . . . . 31,		<b>F</b>
47, 51, 55, 59, 68, 155, 159, 163, 173		file commands:
<code>\bool_if_exist:NTF</code> . . . . . 13		<code>\file_if_exist:nTF</code> . . . . . 34
<code>\bool_new:N</code> . . . . . 8		
<b>C</b>		<b>L</b>
clist commands:		<code>\let</code> . . . . . 106, 108, 109, 110
<code>\clist_map_inline:Nn</code> . . . . . 11		lua commands:
<code>\clist_map_inline:nn</code> . . . . . 6		<code>\lua_load_module:n</code> . . . . . 36
color commands:		<code>\lua_now:n</code> . . . . . 24
<code>\color_set:nn</code> . . . . . 3		<b>M</b>
<code>\color_set:nnn</code> . . . . . 53		msg commands:
cs commands:		<code>\msg_info:nnn</code> . . . . . 33, 70
<code>\cs_generate_variant:Nn</code> . . . . 136, 137		

