

The pdfmanagement-firstaid package – temporary patches and package replacements L^AT_EX PDF management testphase bundle

The L^AT_EX Project*

Version 0.95w, released 2023-02-20

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}{2023-02-20}{0.95w}
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

```

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. 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 luatload is new enough.

```

31 \sys_if_engine luatex:T
32   {
33     \lua_now:n
34     {
35       if-not-luatload.set_transparent_colorstack~
36       then~
37         tex.print("\bool_gset_false:N\g__pdfmanagement_firstaid_fontspec_bool")~
38         end
39     }
40   \bool_if:NT \g__pdfmanagement_firstaid_fontspec_bool
41     {
42       \msg_info:nnn{pdfmanagement}{firstaid}{fontspec}
43       \file_if_exist:nT { luatload-l3color-tmp.lua }
44       {
45         \lua_load_module:n { luatload-l3color-tmp.lua }
46       }
47
48       \declare@file@substitution{fontspec-luatex.sty}{fontspec-luatex-tmp-ltx.sty}
49       \AddToHook{package/fontspec/after}
50       {
51         \str_if_eq:vnF{ver@fontspec.sty}{2022/01/15-v2.8a-Font-selection-for-XeLaTeX-and-
52           {\msg_warn:nnn{pdfmanagement}{firstaid-changed}{fontspec}}
53       }
54     }
55   }

```

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.

```

56 \bool_if:NT \g__pdfmanagement_firstaid_output_bool
57   {

```

This catches also phase-II and tagpdf.

```

58   \file_if_exist:nF {new-or-1-latex-lab-testphase.ltx}
59   {
60     \AddToHook{package/latex-lab-testphase-new-or/after}
61     {\RequirePackage{output-patches-tmp-ltx}}

```

This is for phase-I

```
62     \AddToHook{package/tagpdf/after}
63     {\RequirePackage{output-patches-tmp-ltx}}
64   }
65 }
```

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.

```
66 \bool_if:NT \g__pdfmanagement_firstaid_color_bool
67 {
68   \declare@file@substitution{color.sty}{color-ltx.sty}
69 }
```

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`.

```
70 \bool_if:NT \g__pdfmanagement_firstaid_xcolor_bool
71 {
72   \AddToHook
73   {
74     package/xcolor/after
75   }
76   {\RequirePackage{xcolor-patches-tmp-ltx}}
77 }
```

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.

```
78 \bool_lazy_all:nT
79 {
80   {\sys_if_engine luatex_p:}
81   {\g__pdfmanagement_firstaid_luacolor_bool}
82   {\sys_if_output_pdf_p:}
83 }
```

```

84 {
85   \AddToHook{package/luacolor/after}
86   {
87     \cs_set_protected:Npn \__color_backend_select:nn #1#2
88     {
89       \tl_set:Nn \l__color_backend_fill_tl {#1}
90       \tl_set:Nn \l__color_backend_stroke_tl {#2}
91       \LuaCol@setattribute\LuaCol@Attribute
92       {
93         \directlua
94         {
95           oberdiek.luacolor.get("\luaescapestring{#1~#2}")
96         }
97       }
98     }
99     \cs_set_protected:Npn \__color_backend_fill:n #1
100    {
101      \tl_set:Nn \l__color_backend_fill_tl {#1}
102      \LuaCol@setattribute\LuaCol@Attribute
103      {
104        \directlua
105        {
106          oberdiek.luacolor.get("\luaescapestring{#1}")
107        }
108      }
109    }
110    \cs_set_protected:Npn \__color_backend_stroke:n #1
111    {
112      \tl_set:Nn \l__color_backend_stroke_tl {#1}
113      \LuaCol@setattribute\LuaCol@Attribute
114      {
115        \directlua
116        {
117          oberdiek.luacolor.get("\luaescapestring{#1}")
118        }
119      }
120    }
121    \cs_set_protected:Npn \__color_backend_reset: {}
122    \cs_set_eq:NN \__color_backend_fill_reset: \__color_backend_reset:
123    \cs_set_eq:NN \__color_backend_stroke_reset: \__color_backend_reset:
124  }
125 }

```

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.

```

126
127 \bool_if:NT \g__pdfmanagement_firstaid_pgf_bool
128 {
129   \msg_info:nnn{pdfmanagement }{firstaid}{pgf}
130   \AddToHook
131     {
132       file/pgfrcs.sty/after
133     }
134     {
135       \cs_set_eq:NN
136         \__pdfmanagement_pgfori_pgfutil@setuppdfresources
137         \pgfutil@setuppdfresources
138       \def\pgfutil@setuppdfresources
139         {
140           \pdfmanagement_if_active:TF
141             {
142               \__pdfmanagement_pgf_sys_setuppdfresources_plain:
143             }
144             {
145               \__pdfmanagement_pgfori_pgfutil@setuppdfresources
146             }
147           }
148         }
149     }
150   %\def\pgfutil@addpdfresource@extgs#1{\pgf@sys@addpdfresource@extgs@plain{#1}}
151   %\def\pgfutil@addpdfresource@colorspaces#1{\pgf@sys@addpdfresource@colorspaces@plain{#1}}
152   %\def\pgfutil@addpdfresource@patterns#1{\pgf@sys@addpdfresource@patterns@plain{#1}}
153   %\def\pgfutil@setuppdfresources{\pgf@sys@setuppdfresources@plain}
154   % \pgf@sys@pdf@possible@resources %used in xform
155   %Trying to patch pgf ..
156   \cs_new_protected:Npn \__pdfmanagement_pgf_sys_setuppdfresources_plain:
157     {
158       %objects are already created ...
159       \def\pgf@sys@pdf@possible@resources
160         {
161           /ColorSpace~\pdf_object_ref:n {__pdf/Page/Resources/ColorSpace}
162           /Pattern ~\pdf_object_ref:n {__pdf/Page/Resources/Pattern}
163           /ExtGState ~\pdf_object_ref:n {__pdf/Page/Resources/ExtGState}
164         }
165       \let\pgf@sys@pdf@check@resources=\relax%
166       %not sure if needed, but perhaps the lists are used somewhere else ...
167       \let\pgf@sys@pgf@resource@list@extgs=\pgfutil@empty%
168       \let\pgf@sys@pgf@resource@list@patterns=\pgfutil@empty%

```

```

169 \let\pgf@sys@pgf@resource@list@colorspaces=\pgfutil@empty%
170 % the commands to add page resources
171 \def\pgf@sys@addpdfresource@extgs@plain##1
172 {
173     %\exp_after:wN %for transparent which passes a command
174     \__pdfmanagement_patch_pgftxtgs:w ##1\q_stop
175 }
176 \def\pgf@sys@addpdfresource@patterns@plain##1
177 {
178     \__pdfmanagement_patch_pgfpatterns:w ##1\q_stop
179 }
180 \def\pgf@sys@addpdfresource@colorspaces@plain##1
181 {
182     \__pdfmanagement_patch_pgfcolorspaces:w ##1\q_stop
183 }
184 }
185
186 %\AtEndPreamble{\pgfutil@setuppdfresources}
187 % helper commands as pgf doesn't pass resources as two arguments
188 % code to add to the resources existing stuff in the format "/name value":
189 \cs_new:Npn \__pdfmanagement_split_dict_entry_aux:NNw #1 #2 /#3-#4\q_stop
190 {
191     \tl_set:Nn #1 {#3}
192     \tl_set:Nn #2 {#4}
193 }
194
195 \cs_generate_variant:Nn \tl_trim_spaces:n{V}
196 \cs_generate_variant:Nn \pdfmanagement_add:nex{nex}
197 \cs_new:Npn \__pdfmanagement_patch_pgftxtgs:w #1/#2<<#3>>#4\q_stop
198 {
199     \tl_set:Nx\l_tmpa_tl{#2}
200     \pdfmanagement_add:nex
201     {Page/Resources/ExtGState}{\tl_trim_spaces:V\l_tmpa_tl}{<<#3 #4>>}
202 }
203 \cs_new:Npn \__pdfmanagement_patch_pgfpatterns:w #1/#2\space#3\q_stop
204 {
205     \pdfmanagement_add:nex
206     {Page/Resources/Pattern}{\tl_trim_spaces:n{#2}}{#3}
207 }
208 \cs_new:Npn \__pdfmanagement_patch_pgfcolorspaces:w #1/#2[#3]#4\q_stop
209 {
210     \pdfmanagement_add:nex
211     {Page/Resources/ColorSpace}{\tl_trim_spaces:n{#2}}{[#3]}
212 }
213

```

1.8 transparent

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

```

214 \bool_if:NT \g__pdfmanagement_firstaid_transparent_bool
215 {
216     \AddToHook{package/transparent/after}
217     {
218         \@ifpackagelater{transparent}{2022-10-27}{}

```

```

219     {
220       \msg_warning:nnnn{pdfmanagement}{firstaid-too-old}
221       {transparent}
222       {1.5~from~2022-10-27}
223     }
224   }
225 }

```

1.9 pdfscape

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

```

226 \bool_if:NT \g__pdfmanagement_firstaid_pdfscape_bool
227 {
228   \AddToHook{package/pdfscape/after}
229   {
230     \@ifpackagelater{pdfscape}{2022-10-27}{
231       {
232         \msg_warning:nnnn{pdfmanagement}{firstaid-too-old}
233         {pdfscape}
234         {0.13~from~2022-10-27}
235       }
236     }
237 }

```

1.10 xmp

This handles the new xmp code.

```

238 \bool_if:NT \g__pdfmanagement_firstaid_xmp_bool
239 {
240   \disable@package@load{hyperxmp}{\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{hyperxmp}}
241   \disable@package@load{pdfx} {\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{pdfx}}
242   \AddToHook{package/doclicense/after}
243   {
244     \AtBeginDocument
245     {
246       \hypersetup
247       {
248         pdfcopyright = {\doclicenseLongTextForHyperref},
249         pdflicenseurl = {\doclicenseURL},
250       }
251     }
252   }
253 }
254 \end{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.

```

255 \end{package}
256 \bool_if:NT \g__pdfmanagement_firstaid_colorspace_bool

```

```

257 {
258   \AddToHook
259     {
260       package/colorspace/after
261     }
262     {\RequirePackage{colorspace-patches-tmp-ltx}}
263 }
264 \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>\cs_set_eq:NN</code> 122, 123, 135
<code>\</code> 25, 28, 37	<code>\cs_set_protected:Npn</code> 87, 99, 110, 121
A	D
<code>\AddToHook</code> 49, 60, 62, 72, 85, 130, 216, 228, 242, 258	<code>\def</code> 138, 150, 151, 152, 153, 159, 171, 176, 180
<code>\AtBeginDocument</code> 244	<code>\definecolor</code> 3
<code>\AtEndPreamble</code> 186	<code>\directlua</code> 93, 104, 115
B	<code>\doclicenseLongTextForHyperref</code> . . . 248
bool commands:	<code>\doclicenseURL</code> 249
<code>\bool_gset_false:N</code> 17	<code>\DocumentMetadata</code> 1
<code>\bool_gset_true:N</code> 11	E
<code>\bool_if:NTF</code> 40, 56, 66, 70, 127, 214, 226, 238, 256	exp commands:
<code>\bool_if_exist:NTF</code> 15	<code>\exp_after:wN</code> 173
<code>\bool_lazy_all:nTF</code> 78	F
<code>\bool_new:N</code> 10	file commands:
C	<code>\file_if_exist:nTF</code> 43, 58
clist commands:	H
<code>\clist_map_inline:Nn</code> 13	<code>\hypersetup</code> 246
<code>\clist_map_inline:nn</code> 6	L
color commands:	<code>\let</code> 165, 167, 168, 169
<code>\color_set:nn</code> 3	lua commands:
color internal commands:	<code>\lua_load_module:n</code> 45
<code>__color_backend_fill:n</code> 99	<code>\lua_now:n</code> 33
<code>__color_backend_fill_reset:</code> . . 122	<code>\luaescapestring</code> 95, 106, 117
<code>\l__color_backend_fill_tl</code> . . . 89, 101	M
<code>__color_backend_reset:</code> 121, 122, 123	msg commands:
<code>__color_backend_select:mn</code> 87	<code>\msg_info:nnn</code> 42, 129
<code>__color_backend_stroke:n</code> 110	<code>\msg_new:nnn</code> 20, 22, 24, 27
<code>__color_backend_stroke_reset:</code> . 123	<code>\msg_warn:nnn</code> 52
<code>\l__color_backend_stroke_tl</code> . 90, 112	<code>\msg_warning:nnn</code> 240, 241
cs commands:	<code>\msg_warning:nnnn</code> 220, 232
<code>\cs_generate_variant:Nn</code> 195, 196	
<code>\cs_new:Npn</code> 189, 197, 203, 208	
<code>\cs_new_protected:Npn</code> 156	

P	R
pdf commands:	<code>\relax</code> 165
<code>\pdf_object_ref:n</code> 161, 162, 163	<code>\RequirePackage</code> 61, 63, 76, 262
pdfmanagement commands:	S
<code>\pdfmanagement_add:nnn</code> 196, 200, 205, 210	<code>\space</code> 203
<code>\pdfmanagement_if_active:TF</code> ... 140	str commands:
pdfmanagement internal commands:	<code>\str_if_eq:nnTF</code> 51
<code>\g__pdfmanagement_firstaid_-</code> <code>color_bool</code> 66	sys commands:
<code>\g__pdfmanagement_firstaid_-</code> <code>colorspace_bool</code> 256	<code>\sys_if_engine luatex:TF</code> 31
<code>\g__pdfmanagement_firstaid_-</code> <code>fontspec_bool</code> 40	<code>\sys_if_engine luatex_p:</code> 80
<code>\g__pdfmanagement_firstaid_-</code> <code>luacolor_bool</code> 81	<code>\sys_if_output_pdf_p:</code> 82
<code>\g__pdfmanagement_firstaid_-</code> <code>output_bool</code> 56	T
<code>\g__pdfmanagement_firstaid_-</code> <code>pdflscape_bool</code> 226	TeX and L ^A T _E X 2 _ε commands:
<code>\g__pdfmanagement_firstaid_pgf_-</code> <code>bool</code> 127	<code>\@ifpackagelater</code> 218, 230
<code>\g__pdfmanagement_firstaid_-</code> <code>transparent_bool</code> 214	<code>\declare@file@substitution</code> .. 48, 68
<code>\g__pdfmanagement_firstaid_-</code> <code>xcolor_bool</code> 70	<code>\disable@package@load</code> 240, 241
<code>\g__pdfmanagement_firstaid_xmp_-</code> <code>bool</code> 238	<code>\LuaCol@Attribute</code> 91, 102, 113
<code>\g__pdfmanagement_firstaidoff_-</code> <code>clist</code> 13	<code>\LuaCol@setattribute</code> ... 91, 102, 113
<code>__pdfmanagement_patch_pgfcolorspaces:w</code> 182, 208	<code>\pgf@sys@addpdfresource@colorspaces@plain</code> 151, 180
<code>__pdfmanagement_patch_pgfextgs:w</code> 174, 197	<code>\pgf@sys@addpdfresource@extgs@plain</code> 150, 171
<code>__pdfmanagement_patch_pgfpatterns:w</code> 178, 203	<code>\pgf@sys@addpdfresource@patterns@plain</code> 152, 176
<code>__pdfmanagement_pgf_sys_-</code> <code>setuppdfresources_plain:</code> 142, 156	<code>\pgf@sys@pdf@check@resources</code> .. 165
<code>__pdfmanagement_pgfori_pgfutil@setuppdfresources</code> 136, 145	<code>\pgf@sys@pdf@possible@resources</code> 154, 159
<code>__pdfmanagement_split_dict_-</code> <code>entry_aux:NNw</code> 189	<code>\pgf@sys@pgf@resource@list@colorspaces</code> 169
<code>\ProvidesExplPackage</code> 2	<code>\pgf@sys@pgf@resource@list@extgs</code> 167
Q	<code>\pgf@sys@pgf@resource@list@patterns</code> 168
quark commands:	<code>\pgf@sys@setuppdfresources@plain</code> 153
<code>\q_stop</code> 174, 178, 182, 189, 197, 203, 208	<code>\pgfutil@addpdfresource@colorspaces</code> 4, 151
	<code>\pgfutil@addpdfresource@extgs</code> 4, 150
	<code>\pgfutil@addpdfresource@patterns</code> 4, 152
	<code>\pgfutil@empty</code> 167, 168, 169
	<code>\pgfutil@setuppdfresources</code> 137, 138, 153, 186
	tl commands:
	<code>\tl_set:Nn</code> 89, 90, 101, 112, 191, 192, 199
	<code>\tl_trim_spaces:n</code> .. 195, 201, 206, 211
	<code>\l_tmpa_tl</code> 199, 201