

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

The L^AT_EX Project*

Version 0.95s, released 2022-09-26

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}{2022-09-26}{0.95s}
3   {LaTeX PDF management testphase bundle / firstaid-patches}
4
5 <@@=pdfmanagement>
6 \clist_map_inline:nn {pgf,transparent,xmp,pdfjscape,xcolor,color,beamer,output,colorspace,fontenc}
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! }
22 \msg_new:nnn { pdfmanagement } { firstaid-disabled }
23   { The~loading~of~package~#1~is~disabled.\
24     It~is~not~compatible~with~the~PDF~management. }
25
```

*E-mail: latex-team@latex-project.org

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. 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 luatex is new enough.

```
26 \sys_if_engine luatex:T
27 {
28   \lua_now:n
29   {
30     if-not~luatex.set_transparent_colorstack~
31     then~
32       tex.print("\bool_gset_false:N\g__pdfmanagement_firstaid_fontspec_bool")~
33     end
34   }
35   \bool_if:NT \g__pdfmanagement_firstaid_fontspec_bool
36   {
37     \msg_info:nnn{pdfmanagement}{firstaid}{fontspec}
38     \file_if_exist:nT { luatex-l3color-tmp.lua }
39     {
40       \lua_load_module:n { luatex-l3color-tmp.lua }
41     }
42
43     \declare@file@substitution{fontspec-luatex.sty}{fontspec-luatex-tmp-ltx.sty}
44     \AddToHook{package/fontspec/after}
45     {
46       \str_if_eq:vnF{ver@fontspec.sty}{2022/01/15-v2.8a-Font-selection-for-XeLaTeX-and-}
47       {\msg_warn:nnn{pdfmanagement}{firstaid-changed}{fontspec}}
48     }
49   }
50 }
```

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.

```
51 \bool_if:NT \g__pdfmanagement_firstaid_output_bool
52 {
53   \RequirePackage{output-patches-tmp-ltx}
54 }
```

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.

```
55 \bool_if:NT \g__pdfmanagement_firstaid_beamer_bool
56 {
57   \color_set:nnn{.5~.5~.5}{rgb}{0.5,0.5,0.5}
58 }
```

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.

```
59 \bool_if:NT \g__pdfmanagement_firstaid_color_bool
60 {
61     \declare@file@substitution{color.sty}{color-ltx.sty}
62 }
```

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`.

```
63 \bool_if:NT \g__pdfmanagement_firstaid_xcolor_bool
64 {
65     \AddToHook
66     {
67         package/xcolor/after
68     }
69     {\RequirePackage{xcolor-patches-tmp-ltx}}
70 }
```

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.

```

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

```

```

126     {
127         \__pdfmanagement_patch_pgfcOLORSPACES:w ##1\q_stop
128     }
129 }
130
131 %\AtEndPreamble{\pgfutil@setuppdfresources}
132 % helper commands as pgf doesn't pass resources as two arguments
133 % code to add to the resources existing stuff in the format "/name value":
134 \cs_new:Npn \__pdfmanagement_split_dict_entry_aux:NNw #1 #2 /#3~#4\q_stop
135 {
136     \tl_set:Nn #1 {#3}
137     \tl_set:Nn #2 {#4}
138 }
139
140 \cs_generate_variant:Nn \tl_trim_spaces:n{V}
141 \cs_generate_variant:Nn \pdfmanagement_add:nnn {nex}
142 \cs_new:Npn \__pdfmanagement_patch_pgftexts:w #1/#2<<#3>>#4\q_stop
143 {
144     \tl_set:Nx\l_tmpa_tl{#2}
145     \pdfmanagement_add:nex
146         {Page/Resources/ExtGState}{\tl_trim_spaces:V\l_tmpa_tl}{<<#3 #4>>}
147 }
148 \cs_new:Npn \__pdfmanagement_patch_pgfpatterns:w #1/#2\space#3\q_stop
149 {
150     \pdfmanagement_add:nex
151         {Page/Resources/Pattern}{\tl_trim_spaces:n{#2}}{#3}
152 }
153 \cs_new:Npn \__pdfmanagement_patch_pgfcOLORSPACES:w #1/#2[#3]#4\q_stop
154 {
155     \pdfmanagement_add:nex
156         {Page/Resources/ColorSpace}{\tl_trim_spaces:n{#2}}{[#3]}
157 }
158

```

1.7 transparent

We simply replace by the new version.

```

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

```

1.8 pdfscape

We simply replace by the new version.

```

163 \bool_if:NT \g__pdfmanagement_firstaid_pdfscape_bool
164 {
165     \declare@file@substitution{pdfscape.sty}{pdfscape-ltx.sty}
166 }

```

1.9 xmp

This handles the new xmp code.

```

167 \bool_if:NT \g__pdfmanagement_firstaid_xmp_bool
168 {
169   \disable@package@load{hyperxmp}{\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{hyperxm
170   \disable@package@load{pdfx}      {\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{pdfx}}
171   \AddToHook
172     {
173       file/doclicense.sty/after
174     }
175     {
176       \hypersetup
177       {
178         pdfcopyright = {\doclicenseLongTextForHyperref},
179         pdflicenseurl = {\doclicenseURL},
180       }
181     }
182 }
183 \</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.

```

184 \<package>
185 \bool_if:NT \g__pdfmanagement_firstaid_colorspace_bool
186 {
187   \AddToHook
188     {
189       package/colorspace/after
190     }
191     {\RequirePackage{colorspace-patches-tmp-ltx}}
192 }
193 \</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>\bool_if_exist:NTF</code> 13
<code>\</code> 23, 32		<code>\bool_new:N</code> 8
A		C
<code>\AddToHook</code> 44, 65, 75, 171, 187		clist commands:
<code>\AtEndPreamble</code> 131		<code>\clist_map_inline:Nn</code> 11
		<code>\clist_map_inline:nn</code> 6
B		color commands:
bool commands:		<code>\color_set:nn</code> 3
<code>\bool_gset_false:N</code> 15		<code>\color_set:nnn</code> 57
<code>\bool_gset_true:N</code> 9		cs commands:
<code>\bool_if:NTF</code> 35,		<code>\cs_generate_variant:Nn</code> 140, 141
51, 55, 59, 63, 72, 159, 163, 167, 185		

<code>\cs_new:Npn</code>	134, 142, 148, 153	<code>\g__pdfmanagement_firstaid_pgf_-</code> <code>bool</code>	72
<code>\cs_new_protected:Npn</code>	101	<code>\g__pdfmanagement_firstaid_-</code> <code>transparent_bool</code>	159
<code>\cs_set_eq:NN</code>	80	<code>\g__pdfmanagement_firstaid_-</code> <code>xcolor_bool</code>	63
D		<code>\g__pdfmanagement_firstaid_xmp_-</code> <code>bool</code>	167
<code>\def</code>	83, 95, 96, 97, 98, 104, 116, 121, 125	<code>\g__pdfmanagement_firstaidoff_-</code> <code>clist</code>	11
<code>\definecolor</code>	3	<code>__pdfmanagement_patch_pgfcolorspaces:w</code>	127, 153
<code>\doclicenseLongTextForHyperref</code>	178	<code>__pdfmanagement_patch_pgftextgs:w</code>	119, 142
<code>\doclicenseURL</code>	179	<code>__pdfmanagement_patch_pgfpatterns:w</code>	123, 148
<code>\DocumentMetadata</code>	1	<code>__pdfmanagement_pgf_sys_-</code> <code>setuppdfresources_plain:</code>	87, 101
E		<code>__pdfmanagement_pgfiori_pgfutil@setuppdfresources</code>	81, 90
exp commands:		<code>__pdfmanagement_split_dict_-</code> <code>entry_aux:NNw</code>	134
<code>\exp_after:wN</code>	118	<code>\ProvidesExplPackage</code>	2
F		Q	
file commands:		quark commands:	
<code>\file_if_exist:nTF</code>	38	<code>\q_stop</code>	119, 123, 127, 134, 142, 148, 153
H		R	
<code>\hypersetup</code>	176	<code>\relax</code>	110
L		<code>\RequirePackage</code>	53, 69, 191
<code>\let</code>	110, 112, 113, 114	S	
lua commands:		<code>\space</code>	148
<code>\lua_load_module:n</code>	40	str commands:	
<code>\lua_now:n</code>	28	<code>\str_if_eq:nnTF</code>	46
M		sys commands:	
msg commands:		<code>\sys_if_engine luatex:TF</code>	26
<code>\msg_info:nnn</code>	37, 74	T	
<code>\msg_new:nnn</code>	18, 20, 22	T _E X and L ^A T _E X 2 _ε commands:	
<code>\msg_warn:nnn</code>	47	<code>\declare@file@substitution</code>	43, 61, 161, 165
<code>\msg_warning:nnn</code>	169, 170	<code>\disable@package@load</code>	169, 170
P		<code>\pgf@sys@addpdfresource@colorspaces@plain</code>	96, 125
pdf commands:		<code>\pgf@sys@addpdfresource@extgs@plain</code>	95, 116
<code>\pdf_object_ref:n</code>	106, 107, 108	<code>\pgf@sys@addpdfresource@patterns@plain</code>	97, 121
pdfmanagement commands:		<code>\pgf@sys@pdf@check@resources</code>	110
<code>\pdfmanagement_add:nnn</code>	141, 145, 150, 155	<code>\pgf@sys@pdf@possible@resources</code>	99, 104
<code>\pdfmanagement_if_active:TF</code>	85	<code>\pgf@sys@pgf@resource@list@colorspaces</code>	114
pdfmanagement internal commands:			
<code>\g__pdfmanagement_firstaid_-</code> <code>beamer_bool</code>	55		
<code>\g__pdfmanagement_firstaid_-</code> <code>color_bool</code>	59		
<code>\g__pdfmanagement_firstaid_-</code> <code>colorspace_bool</code>	185		
<code>\g__pdfmanagement_firstaid_-</code> <code>fontspec_bool</code>	35		
<code>\g__pdfmanagement_firstaid_-</code> <code>output_bool</code>	51		
<code>\g__pdfmanagement_firstaid_-</code> <code>pdflscape_bool</code>	163		

