

The pdfmanagement-firstaid package – temporary patches and package replacements

L^AT_EX PDF management bundle

The L^AT_EX Project*

Version 0.96t, released 2025-06-29

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}{2025-06-29}{0.96t}
3   {LaTeX PDF management bundle / firstaid-patches}
4
5 (@@=pdfmanagement)
6 \clist_map_inline:nn
7   {pgf,xmp,xcolor,color,output,colorspace,luacolor}
8   {
9     \bool_new:c          { g__pdfmanagement_firstaid_#1_bool }
10    \bool_gset_true:c { g__pdfmanagement_firstaid_#1_bool }
11  }
12 \clist_map_inline:Nn \g__pdfmanagement_firstaidoff_clist
13 {
14   \bool_if_exist:cT { g__pdfmanagement_firstaid_#1_bool }
15   {
16     \bool_gset_false:c { g__pdfmanagement_firstaid_#1_bool }
17   }
18 }
19 \msg_new:nnn { pdfmanagement } { firstaid }
20   { loading~pdfmanagement~firstaid~code~for~#1 }
21 \msg_new:nnn { pdfmanagement } { firstaid-changed }
22   { package~#1~has~changed.-Check-if-patch-is-still-valid! }
23 \msg_new:nnn { pdfmanagement } { firstaid-disabled }
24   { The~loading~of~package~#1~is~disabled.\
25     It~is~not~compatible~with~the~PDF~management. }
```

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

```

26 \msg_new:nnn { pdfmanagement } { firstaid-too-old }
27           { Package-#1-is-too-old-and-not-compatible.\\
28             Get-at-least-version-#2.}
29

```

1.1 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.

```

30 \bool_if:NT \g__pdfmanagement_firstaid_color_bool
31   {
32     \declare@file@substitution{color.sty}{color-ltx.sty}
33   }

```

1.2 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`.

```

34 \bool_if:NT \g__pdfmanagement_firstaid_xcolor_bool
35   {
36     \AddToHook
37       {
38         package/xcolor/after
39       }
40     {\RequirePackage{xcolor-patches-tmp-ltx}}

```

The patch must before color definitions are loaded, which will happen in hooks in a newer xcolor:

```

41   \DeclareHookRule{package/xcolor/after}{pdfmanagement-firstaid}{before}{xcolor}
42 }

```

1.3 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.

```

43 \bool_lazy_all:nT
44   {
45     {\sys_if_engine luatex_p:}
46     {\g__pdfmanagement_firstaid_luacolor_bool}
47     {\sys_if_output_pdf_p:}
48   }

```

```

49 {
50   \AddToHook{package/luacolor/after}
51   {
52     \cs_set_protected:Npn \__color_backend_select:nn #1#2
53     {
54       \tl_set:Nn \l__color_backend_fill_tl {#1}
55       \tl_set:Nn \l__color_backend_stroke_tl {#2}
56       \LuaCol@setattribute\LuaCol@Attribute
57       {
58         \directlua
59         {
60           oberdiek.luacolor.get("\luaescapestring{#1~#2}")
61         }
62       }
63     }
64     \cs_set_protected:Npn \__color_backend_fill:n #1
65     {
66       \tl_set:Nn \l__color_backend_fill_tl {#1}
67       \LuaCol@setattribute\LuaCol@Attribute
68       {
69         \directlua
70         {
71           oberdiek.luacolor.get("\luaescapestring{#1}")
72         }
73       }
74     }
75     \cs_set_protected:Npn \__color_backend_stroke:n #1
76     {
77       \tl_set:Nn \l__color_backend_stroke_tl {#1}
78       \LuaCol@setattribute\LuaCol@Attribute
79       {
80         \directlua
81         {
82           oberdiek.luacolor.get("\luaescapestring{#1}")
83         }
84       }
85     }
86     \cs_set_protected:Npn \__color_backend_reset: {}
87     \cs_set_eq:NN \__color_backend_fill_reset: \__color_backend_reset:
88     \cs_set_eq:NN \__color_backend_stroke_reset: \__color_backend_reset:
89   }
90 }

```

1.4 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.

```

91 \bool_if:NT \g__pdfmanagement_firstaid_pgf_bool
92   {
93     \msg_info:nnn{pdfmanagement }{firstaid}{pgf}
94     \AddToHook
95       {
96         file/pgfrcs.sty/after
97       }
98     {
99       \cs_set_eq:NN
100         \__pdfmanagement_pgfori_pgfutil@setuppdfresources
101         \pgfutil@setuppdfresources
102       \def\pgfutil@setuppdfresources
103         {
104           \__pdfmanagement_pgf_sys_setuppdfresources_plain:
105         }
106     }
107   }
108   %\def\pgfutil@addpdfresource@extgs#1{\pgf@sys@addpdfresource@extgs@plain{#1}}
109   %\def\pgfutil@addpdfresource@colorspaces#1{\pgf@sys@addpdfresource@colorspaces@plain{#1}}
110   %\def\pgfutil@addpdfresource@patterns#1{\pgf@sys@addpdfresource@patterns@plain{#1}}
111   %\def\pgfutil@setuppdfresources{\pgf@sys@setuppdfresources@plain}
112   % \pgf@sys@pdf@possible@resources %used in xform
113   %Trying to patch pgf ..
114   \cs_new_protected:Npn \__pdfmanagement_pgf_sys_setuppdfresources_plain:
115     {
116       %objects are already created ...
117       \def\pgf@sys@pdf@possible@resources
118         {
119           /ColorSpace~\pdf_object_ref:n {__pdf/Page/Resources/ColorSpace}
120           /Pattern ~\pdf_object_ref:n {__pdf/Page/Resources/Pattern}
121           /ExtGState ~\pdf_object_ref:n {__pdf/Page/Resources/ExtGState}
122         }
123       \let\pgf@sys@pdf@check@resources=\relax%
124       %not sure if needed, but perhaps the lists are used somewhere else ...
125       \let\pgf@sys@pgf@resource@list@extgs=\pgfutil@empty%
126       \let\pgf@sys@pgf@resource@list@patterns=\pgfutil@empty%
127       \let\pgf@sys@pgf@resource@list@colorspaces=\pgfutil@empty%
128       % the commands to add page resources
129       \def\pgf@sys@addpdfresource@extgs@plain##1
130         {
131           %\exp_after:wN %for transparent which passes a command
132           \__pdfmanagement_patch_pgfextgs:w ##1\q_stop
133         }

```

```

134 \def\pgf@sys@addpdfresource@patterns@plain##1
135 {
136   \_pdfmanagement_patch_pgfpatterns:w ##1\q_stop
137 }
138 \def\pgf@sys@addpdfresource@colorspaces@plain##1
139 {
140   \_pdfmanagement_patch_pgfcolorspaces:w ##1\q_stop
141 }
142 }
143
144 %\AtEndPreamble{\pgfutil@setuppdfresources}
145 % helper commands as pgf doesn't pass resources as two arguments
146 % code to add to the resources existing stuff in the format "/name value":
147 \cs_new:Npn \_pdfmanagement_split_dict_entry_aux:NNw #1 #2 /#3-#4\q_stop
148 {
149   \tl_set:Nn #1 {#3}
150   \tl_set:Nn #2 {#4}
151 }
152
153 \cs_generate_variant:Nn \tl_trim_spaces:n{V}
154 \cs_generate_variant:Nn \pdfmanagement_add:neen {nee}
155 \cs_new:Npn \_pdfmanagement_patch_pgfextgs:w #1/#2<<#3>>#4\q_stop
156 {
157   \tl_set:Ne\l_tmpa_tl{#2}
158   \pdfmanagement_add:neen
159     {Page/Resources/ExtGState}{\tl_trim_spaces:V\l_tmpa_tl}{<<#3 #4>>}
160 }
161 \cs_new:Npn \_pdfmanagement_patch_pgfpatterns:w #1/#2\space#3\q_stop
162 {
163   \pdfmanagement_add:neen
164     {Page/Resources/Pattern}{\tl_trim_spaces:n{#2}}{#3}
165 }
166 \cs_new:Npn \_pdfmanagement_patch_pgfcolorspaces:w #1/#2[#3]#4\q_stop
167 {
168   \pdfmanagement_add:neen
169     {Page/Resources/ColorSpace}{\tl_trim_spaces:n{#2}}{[#3]}
170 }
171

```

1.5 xmp

This handles the new xmp code.

```

172 \bool_if:NT \g__pdfmanagement_firstaid_xmp_bool
173 {
174   \disable@package@load{hyperxmp}{\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{hyperxm
175   \disable@package@load{pdfx}      {\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{pdfx}}
176   \AddToHook{package/doclicense/after}
177   {
178     \AtBeginDocument
179     {
180       \hypersetup
181       {
182         pdfcopyright = {\doclicenseLongTextForHyperref},

```

```

183         pdflicenseurl = {\doclicenseURL},
184     }
185 }
186 }
187 }
188 \</package>

```

1.6 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.

```

189 \<*package>
190 \bool_if:NT \g__pdfmanagement_firstaid_colorspace_bool
191 {
192     \AddToHook
193     {
194         package/colorspace/after
195     }
196     {\RequirePackage{colorspace-patches-tmp-ltx}}
197 }
198 \</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		color internal commands:
\	24, 27	_color_backend_fill:n 64
	A		_color_backend_fill_reset: 87
\AddToHook	36, 50, 94, 176, 192	\l_color_backend_fill_tl ... 54, 66
\AtBeginDocument	178	_color_backend_reset: .. 86, 87, 88
\AtEndPreamble	144	_color_backend_select:mn 52
	B		_color_backend_stroke:n 75
bool commands:			_color_backend_stroke_reset: .. 88
\bool_gset_false:N	16	\l_color_backend_stroke_tl .. 55, 77
\bool_gset_true:N	10	cs commands:
\bool_if:NTF	30, 34, 91, 172, 190	\cs_generate_variant:Nn ... 153, 154
\bool_if_exist:NTF	14	\cs_new:Npn 147, 155, 161, 166
\bool_lazy_all:nTF	43	\cs_new_protected:Npn 114
\bool_new:N	9	\cs_set_eq:NN 87, 88, 99
	C		\cs_set_protected:Npn .. 52, 64, 75, 86
clist commands:			D
\clist_map_inline:Nn	12	\DeclareHookRule 41
\clist_map_inline:nn	6	\def 102,
color commands:			108, 109, 110, 111, 117, 129, 134, 138
\color_set:nn	2	\definecolor 2
			\directlua 58, 69, 80
			\doclicenseLongTextForHyperref 182

<code>\doclicenseURL</code>	183	<code>_pdfmanagement_split_dict_-</code>	
<code>\DocumentMetadata</code>	1	<code>entry_aux:NNw</code>	147
E		<code>\ProvidesExplPackage</code>	2
exp commands:		Q	
<code>\exp_after:wN</code>	131	quark commands:	
H		<code>\q_stop</code>	132, 136, 140, 147, 155, 161, 166
<code>\hypersetup</code>	180	R	
L		<code>\relax</code>	123
<code>\let</code>	123, 125, 126, 127	<code>\RequirePackage</code>	40, 196
<code>\luaescapestring</code>	60, 71, 82	S	
M		<code>\space</code>	161
msg commands:		sys commands:	
<code>\msg_info:nnn</code>	93	<code>\sys_if_engine luatex_p:</code>	45
<code>\msg_new:nnn</code>	19, 21, 23, 26	<code>\sys_if_output_pdf_p:</code>	47
<code>\msg_warning:nnn</code>	174, 175	T	
P		T _E X and L ^A T _E X 2 _ε commands:	
pdf commands:		<code>\declare@file@substitution</code>	32
<code>\pdf_object_ref:n</code>	119, 120, 121	<code>\disable@package@load</code>	174, 175
pdfmanagement commands:		<code>\LuaCol@Attribute</code>	56, 67, 78
<code>\pdfmanagement_add:nnn</code>		<code>\LuaCol@setattribute</code>	56, 67, 78
	154, 158, 163, 168	<code>\pgf@sys@addpdfresource@colorspaces@plain</code>	109, 138
pdfmanagement internal commands:		<code>\pgf@sys@addpdfresource@extgs@plain</code>	108, 129
<code>\g__pdfmanagement_firstaid_-</code>		<code>\pgf@sys@addpdfresource@patterns@plain</code>	110, 134
<code>color_bool</code>	30	<code>\pgf@sys@pdf@check@resources</code> ...	123
<code>\g__pdfmanagement_firstaid_-</code>		<code>\pgf@sys@pdf@possible@resources</code>	112, 117
<code>colorspace_bool</code>	190	<code>\pgf@sys@pgf@resource@list@colorspaces</code>	127
<code>\g__pdfmanagement_firstaid_-</code>		<code>\pgf@sys@pgf@resource@list@extgs</code>	125
<code>luacolor_bool</code>	46	<code>\pgf@sys@pgf@resource@list@patterns</code>	126
<code>\g__pdfmanagement_firstaid_pgf_-</code>		<code>\pgf@sys@setuppdfresources@plain</code>	111
<code>bool</code>	91	<code>\pgfutil@addpdfresource@colorspaces</code>	3, 109
<code>\g__pdfmanagement_firstaid_-</code>		<code>\pgfutil@addpdfresource@extgs</code>	3, 108
<code>xcolor_bool</code>	34	<code>\pgfutil@addpdfresource@patterns</code>	3, 110
<code>\g__pdfmanagement_firstaid_xmp_-</code>		<code>\pgfutil@empty</code>	125, 126, 127
<code>bool</code>	172	<code>\pgfutil@setuppdfresources</code>	101, 102, 111, 144
<code>\g__pdfmanagement_firstaidoff_-</code>		tl commands:	
<code>clist</code>	12	<code>\tl_set:Nn</code>	54, 55, 66, 77, 149, 150, 157
<code>_pdfmanagement_patch_pgfcolorspaces:w</code>		<code>\tl_set:n</code>	153, 159, 164, 169
	140, 166	<code>\l_tmpa_tl</code>	157, 159
<code>_pdfmanagement_patch_pgfextgs:w</code>			
	132, 155		
<code>_pdfmanagement_patch_pgfpatterns:w</code>			
	136, 161		
<code>_pdfmanagement_pgf_sys_-</code>			
<code>setuppdfresources_plain:</code>	104, 114		
<code>_pdfmanagement_pgfori_pgfutil@setuppdfresources</code>			
	100		