

The pdfmanagement-firstaid package – temporary patches and package replacements PDFmanagement bundle (testphase)

The L^AT_EX3 Project*

Version 0.95a, released 2021-02-22

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) `\DeclareDocumentMetadata` with `firstaidoff={name1,name2,...}`.

```
1 <*package>
2 \ProvidesExplPackage {pdfmanagement-firstaid} {2021-02-22} {0.95a}
3   {experimental pdf-resource management / firstaid}
4
5 <@@=pdfmanagement>
6 \clist_map_inline:nn {pgf,transparent,hyperxmp,pdflscape,xcolor}
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 }
```

1.1 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`

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

command are known to `l3color` and so `hyperref`. This only supports the color model from `l3color`. Colors defined with the models `cm`, `HSB` and `named` are silently ignored.

```

20 \ifundefined{color_set:nn}{
21 \RequirePackage{l3color}}{}
22 \bool_if:NT \g__pdfmanagement_firstaid_xcolor_bool
23 {
24   \AddToHook{package/after/xcolor}
25     {\RequirePackage{xcolor-patches-tmp-ltx}\XC@@names}
26 }

```

1.2 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.

```

27
28 \bool_if:NT \g__pdfmanagement_firstaid_pgf_bool
29 {
30   \msg_info:nnn{pdfmanagement}{firstaid}{pgf}
31   \AddToHook{file/after/pgfrcs.sty}
32   {
33     \cs_set_eq:NN
34       \__pdfmanagement_pgfori_pgfutil@setuppdfresources
35       \pgfutil@setuppdfresources
36     \def\pgfutil@setuppdfresources
37       {
38         \pdfmanagement_if_active:TF
39         {
40           \__pdfmanagement_pgf_sys_setuppdfresources_plain:
41         }
42         {
43           \__pdfmanagement_pgfori_pgfutil@setuppdfresources
44         }
45       }
46   }
47 }
48 %\def\pgfutil@addpdfresource@extgs#1{\pgf@sys@addpdfresource@extgs@plain{#1}}

```

```

49 %\def\pgfutil@addpdfresource@colorspaces#1{\pgf@sys@addpdfresource@colorspaces@plain{#1}}
50 %\def\pgfutil@addpdfresource@patterns#1{\pgf@sys@addpdfresource@patterns@plain{#1}}
51 %\def\pgfutil@setuppdfresources{\pgf@sys@setuppdfresources@plain}
52 % \pgf@sys@pdf@possible@resources %used in xform
53 %Trying to patch pgf ..
54 \cs_new_protected:Npn \__pdfmanagement_pgf_sys_setuppdfresources_plain:
55 {
56   %objects are already created ...
57   \def\pgf@sys@pdf@possible@resources
58   {
59     /ColorSpace-\pdf_object_ref:n {Page/Resources/ColorSpace}
60     /Pattern ~-\pdf_object_ref:n {Page/Resources/Pattern}
61     /ExtGState ~-\pdf_object_ref:n {Page/Resources/ExtGState}
62   }
63   \let\pgf@sys@pdf@check@resources=\relax%
64   %not sure if needed, but perhaps the lists are used somewhere else ...
65   \let\pgf@sys@pgf@resource@list@extgs=\pgfutil@empty%
66   \let\pgf@sys@pgf@resource@list@patterns=\pgfutil@empty%
67   \let\pgf@sys@pgf@resource@list@colorspaces=\pgfutil@empty%
68   % the commands to add page resources
69   \def\pgf@sys@addpdfresource@extgs@plain##1
70   {
71     \exp_after:wN %for transparent which passes a command
72     \__pdfmanagement_patch_pgfextgs:w ##1\q_stop
73   }
74   \def\pgf@sys@addpdfresource@patterns@plain##1
75   {
76     \__pdfmanagement_patch_pgfpatterns:w ##1\q_stop
77   }
78   \def\pgf@sys@addpdfresource@colorspaces@plain##1
79   {
80     \__pdfmanagement_patch_pgfcolorspaces:w ##1\q_stop
81   }
82 }
83
84 %\AtEndPreamble{\pgfutil@setuppdfresources}
85 % helper commands as pgf doesn't pass resources as two arguments
86 % code to add to the resources existing stuff in the format "/name value":
87 \cs_new:Npn \__pdfmanagement_split_dict_entry_aux:NNw #1 #2 /#3~#4\q_stop
88 {
89   \tl_set:Nn #1 {#3}
90   \tl_set:Nn #2 {#4}
91 }
92
93 \cs_new:Npn \__pdfmanagement_patch_pgfextgs:w #1/#2<<#3>>#4\q_stop
94 {
95   \exp_args:Nne
96   \__pdf_backend_PageResources_gput:nnn
97   {ExtGState}{\tl_trim_spaces:n{#2}}{<<#3>>}
98 }
99 \cs_new:Npn \__pdfmanagement_patch_pgfpatterns:w #1/#2\space#3\q_stop
100 {
101   \exp_args:Nxxx
102   \__pdf_backend_PageResources_gput:nnn

```

```

103         {Pattern}{\tl_trim_spaces:n{#2}}{#3}
104     }
105 \cs_new:Npn \__pdfmanagement_patch_pgfcOLORSPACES:w #1/#2[#3]#4\q_stop
106 {
107     \exp_args:Nne
108     \__pdf_backend_PageResources_gput:nnn
109     {ColorSpace}{\tl_trim_spaces:n{#2}}{[#3]}
110 }
111

```

1.3 transparent

We simply replace by the new version.

```

112 \bool_if:NT \g__pdfmanagement_firstaid_transparent_bool
113 {
114     \declare@file@substitution{transparent.sty}{transparent-ltx.sty}
115 }

```

1.4 pdflscape

We simply replace by the new version.

```

116 \bool_if:NT \g__pdfmanagement_firstaid_pdflscape_bool
117 {
118     \declare@file@substitution{pdflscape.sty}{pdflscape-ltx.sty}
119 }

```

1.5 hyperxmp

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

```

120 \bool_if:NT \g__pdfmanagement_firstaid_hyperxmp_bool
121 {
122     \AddToHook
123     {file/after/hyperxmp.sty}
124     {\RequirePackage{hyperxmp-patches-tmp-ltx}}
125 }
126 \</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.

```

127 \*package>
128 \<@@=pdf>
129 % this must be earlier, to avoid problems with luatex which has two pageresources
130 % lua/tex
131 \hook_gput_code:nnn {begindocument} {pdf}
132 {
133     \tl_if_exist:NT \spc@op
134     {
135         \def\spc@Pageresources#1{}
136     }

```

```

137 }
138 }
139
140 \hook_gput_code:nnn {begindocument/end} {pdf}
141 {
142   \tl_if_exist:NT \spc@op
143   {
144     \__pdf_backend_object_new:nn {__spc_extgstate_op_false}{dict}
145     \__pdf_backend_object_write:nn
146       {__spc_extgstate_op_false}
147       {/Type /ExtGState~/op~false~/OP~false}
148     \pdfmanagement_add:nnn
149       {Page/Resources/ExtGState}
150       {SPCko}
151     {\__pdf_backend_object_ref:n {__spc_extgstate_op_false}}
152     \__pdf_backend_object_new:nn {__spc_extgstate_op_true0}{dict}
153     \__pdf_backend_object_write:nn
154       {__spc_extgstate_op_true0}
155       {/Type /ExtGState~/op~true~/OP~true~/OPM~0}%
156     \pdfmanagement_add:nnn
157       {Page/Resources/ExtGState}
158       {SPCmz}
159     {\__pdf_backend_object_ref:n {__spc_extgstate_op_true0}}
160     \__pdf_backend_object_new:nn {__spc_extgstate_op_true1}{dict}
161     \__pdf_backend_object_write:nn
162       {__spc_extgstate_op_true1}
163       {/Type /ExtGState~/op~true~/OP~true~/OPM~1}%
164     \pdfmanagement_add:nnn
165       {Page/Resources/ExtGState}
166       {SPCop}
167     {\__pdf_backend_object_ref:n {__spc_extgstate_op_true1}}
168   }
169 }
170 \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.

| A | | C | |
|--------------------|-----------------------|--------------------------|---|
| \AddToHook | 24, 31, 122 | clist commands: | |
| \AtEndPreamble | 84 | \clist_map_inline:Nn | 11 |
| | | \clist_map_inline:nn | 6 |
| B | | cs commands: | |
| bool commands: | | \cs_new:Npn | 87, 93, 99, 105 |
| \bool_gset_false:N | 15 | \cs_new_protected:Npn | 54 |
| \bool_gset_true:N | 9 | \cs_set_eq:NN | 33 |
| \bool_if:NTF | 22, 28, 112, 116, 120 | D | |
| \bool_if_exist:NTF | 13 | \DeclareDocumentMetadata | 1 |
| \bool_new:N | 8 | \def | 36, 48, 49, 50, 51, 57, 69, 74, 78, 135 |

