The pdfmanagement-firstaid package – temporary patches and package replacements
LaTeX PDF management testphase bundle

The LaTeX Project*

Version 0.96d, released 2023-12-09

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,...},...}`.

```
⟨*package⟩
\ProvidesExplPackage{pdfmanagement-firstaid}{2023-12-09}{0.96d}
  {LaTeX PDF management testphase bundle / firstaid-patches}

⟨@@=pdfmanagement⟩
\clist_map_inline:nn
  {pgf,transparent,xmp,pdflscape,xcolor,color,
   beamer,output,colorspace,fontspec,luacolor}
  {
    \bool_new:c       { g__pdfmanagement_firstaid_#1_bool }
    \bool_gset_true:c { g__pdfmanagement_firstaid_#1_bool }
  }
\clist_map_inline:Nn \g__pdfmanagement_firstaidoff_clist
  {
    \bool_if_exist:cT { g__pdfmanagement_firstaid_#1_bool }
      {
        \bool_gset_false:c { g__pdfmanagement_firstaid_#1_bool }
      }
  }
\msg_new:nnn  { pdfmanagement } { firstaid }
              { loading~pdfmanagement~firstaid~code~for~#1 }
\msg_new:nnn  { pdfmanagement } { firstaid-changed }
              { package~#1~has~changed.~Check~if~patch~is~still~valid! }
\msg_new:nnn  { pdfmanagement } { firstaid-disabled }
              { The~loading~of~package~#1~is~disabled.\\
                It~is~not~compatible~with~the~PDF~management. }
```

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

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

```

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 needed. It also needs a new fontspec and a change in the color key.

```
\sys_if_engine_luatex:T
  {
    \bool_if:NT \g__pdfmanagement_firstaid_fontspec_bool
      {
        \AddToHook{package/fontspec/after}
         {
           \@ifpackagelater{fontspec}{2023-11-05}
              {\typeout{PDF~Management:~fontspec~firstaid~ignored}}
              {
                \def\c__fontspec_opacity_tl{}
                \__fontspec_keys_define_code:nnn {fontspec} {Color}
                  {
                    \color_if_exist:nTF {#1}
                      {
                        \tl_set:Nn \l__fontspec_hexcol_tl {#1}
                      }
                      {
                        \cs_if_exist:cTF { \token_to_str:N \color@ #1 }
                          {
                            \convertcolorspec{named}{#1}{HTML}\l__fontspec_hexcol_tl
                          }
                          {
                            \int_compare:nTF { \tl_count:n {#1} == 6 }
                              { \tl_set:Nn \l__fontspec_hexcol_tl {#1} }
                              {
                                \int_compare:nTF { \tl_count:n {#1} == 8 }
                                  { \fontspec_parse_colour:viii #1 }
                                  {
                                    \bool_if:NF \l__fontspec_firsttime_bool
                                      { \__fontspec_warning:nx {bad-colour} {#1} }
                                  }
                              }
                          }
                      }
                  }
```

add a comma

```
                \__fontspec_keys_define_code:nnn {fontspec} {Opacity}
                 {
                   \int_set:Nn \l__fontspec_tmp_int {255}
                   \__fontspec_int_mult_truncate:Nn \l__fontspec_tmp_int { #1 }
                   \tl_if_eq:NNF \l__fontspec_opacity_tl \c__fontspec_opacity_tl
                     {
                       \bool_if:NF \l__fontspec_firsttime_bool
```

```
                        { \__fontspec_warning:nx {opa-twice} {#1} }
                      }
                    \tl_set:Nx \l__fontspec_opacity_tl
                      {
                      ,
                        \int_compare:nT { \l__fontspec_tmp_int <= "F } {0} % zero pad
                        \int_to_hex:n { \l__fontspec_tmp_int }
                      }
                  }
```

and a brace

```
                \cs_set:Nn \__fontspec_get_features:n
                  {
                    \__fontspec_init_fontface:
                    \__fontspec_keys_set_known:nxN {fontspec-renderer} {\l__fontspec_fontfeat
                      \l__fontspec_keys_leftover_clist
                    \__fontspec_keys_set_known:nxN {fontspec} {\l__fontspec_keys_leftover_cli
                    \keys_set:nV {fontspec-opentype} \l__fontspec_keys_leftover_clist

                    \tl_if_empty:NF \l__fontspec_mapping_tl
                      { \__fontspec_update_featstr:n { mapping = \l__fontspec_mapping_tl } }

                    \str_if_eq:eeF { \l__fontspec_hexcol_tl \l__fontspec_opacity_tl }
                                   { \c__fontspec_hexcol_tl \c__fontspec_opacity_tl }
                      { \__fontspec_update_featstr:n { color = {\l__fontspec_hexcol_tl\l__fon
                    }
                }
    }
  }
}
```

1.2 beamer

beamer made use of the now unsupported syntax `linkbordercolor={.5␣.5␣.5}`. This has been changed so the patch has been removed.

1.3 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.

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

1.4 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`.

```
\bool_if:NT \g__pdfmanagement_firstaid_xcolor_bool
  {
    \AddToHook
       {
         package/xcolor/after
       }
       {\RequirePackage{xcolor-patches-tmp-ltx}}
```

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

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

1.5 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.

```
\bool_lazy_all:nT
  {
    {\sys_if_engine_luatex_p:}
    {\g__pdfmanagement_firstaid_luacolor_bool}
    {\sys_if_output_pdf_p:}
  }
  {
    \AddToHook{package/luacolor/after}
     {
       \cs_set_protected:Npn \__color_backend_select:nn #1#2
        {
          \tl_set:Nn \l__color_backend_fill_tl {#1}
          \tl_set:Nn \l__color_backend_stroke_tl {#2}
          \LuaCol@setattribute\LuaCol@Attribute
            {
              \directlua
                {
                  oberdiek.luacolor.get("\luaescapestring{#1~#2}")
                }
            }
        }
       \cs_set_protected:Npn \__color_backend_fill:n #1
         {
          \tl_set:Nn \l__color_backend_fill_tl {#1}
          \LuaCol@setattribute\LuaCol@Attribute
            {
              \directlua
                {
                  oberdiek.luacolor.get("\luaescapestring{#1}")
```

```
                }
            }
        }
      \cs_set_protected:Npn \__color_backend_stroke:n #1
        {
          \tl_set:Nn \l__color_backend_stroke_tl {#1}
          \LuaCol@setattribute\LuaCol@Attribute
            {
              \directlua
                {
                  oberdiek.luacolor.get("\luaescapestring{#1}")
                }
            }
        }
      \cs_set_protected:Npn \__color_backend_reset:  {}
      \cs_set_eq:NN \__color_backend_fill_reset:   \__color_backend_reset:
      \cs_set_eq:NN \__color_backend_stroke_reset: \__color_backend_reset:
    }
}
```

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.

```

\bool_if:NT \g__pdfmanagement_firstaid_pgf_bool
  {
    \msg_info:nnn{pdfmanagement }{firstaid}{pgf}
    \AddToHook
      {
        file/pgfrcs.sty/after
      }
      {
        \cs_set_eq:NN
          \__pdfmanagement_pgfori_pgfutil@setuppdfresources
          \pgfutil@setuppdfresources
        \def\pgfutil@setuppdfresources
```

```
          {
            \pdfmanagement_if_active:TF
              {
                \__pdfmanagement_pgf_sys_setuppdfresources_plain:
              }
              {
                \__pdfmanagement_pgfori_pgfutil@setuppdfresources
              }
          }
      }
  }
%\def\pgfutil@addpdfresource@extgs#1{\pgf@sys@addpdfresource@extgs@plain{#1}}
%\def\pgfutil@addpdfresource@colorspaces#1{\pgf@sys@addpdfresource@colorspaces@plain{#1}}
%\def\pgfutil@addpdfresource@patterns#1{\pgf@sys@addpdfresource@patterns@plain{#1}}
%\def\pgfutil@setuppdfresources{\pgf@sys@setuppdfresources@plain}
% \pgf@sys@pdf@possible@resources %used in xform
%Trying to patch pgf ..
\cs_new_protected:Npn \__pdfmanagement_pgf_sys_setuppdfresources_plain:
  {
   %objects are already created ...
    \def\pgf@sys@pdf@possible@resources
      {
        /ColorSpace~\pdf_object_ref:n {__pdf/Page/Resources/ColorSpace}
        /Pattern   ~\pdf_object_ref:n {__pdf/Page/Resources/Pattern}
        /ExtGState ~\pdf_object_ref:n {__pdf/Page/Resources/ExtGState}
      }
    \let\pgf@sys@pdf@check@resources=\relax%
    %not sure if needed, but perhaps the lists are used somewhere else ...
    \let\pgf@sys@pgf@resource@list@extgs=\pgfutil@empty%
    \let\pgf@sys@pgf@resource@list@patterns=\pgfutil@empty%
    \let\pgf@sys@pgf@resource@list@colorspaces=\pgfutil@empty%
    % the commands to add page resources
    \def\pgf@sys@addpdfresource@extgs@plain##1
      {
        %\exp_after:wN %for transparent which passes a command
          \__pdfmanagement_patch_pgfextgs:w ##1\q_stop
      }
    \def\pgf@sys@addpdfresource@patterns@plain##1
      {
        \__pdfmanagement_patch_pgfpatterns:w ##1\q_stop
      }
   \def\pgf@sys@addpdfresource@colorspaces@plain##1
      {
        \__pdfmanagement_patch_pgfcolorspaces:w ##1\q_stop
      }
  }

%\AtEndPreamble{\pgfutil@setuppdfresources}
% helper commands as pgf doesn't pass resources as two arguments
% code to add to the resources existing stuff in the format "/name value":
\cs_new:Npn \__pdfmanagement_split_dict_entry_aux:NNw #1 #2 /#3~#4\q_stop
  {
    \tl_set:Nn #1 {#3}
    \tl_set:Nn #2 {#4}
```

```
  }

\cs_generate_variant:Nn \tl_trim_spaces:n{V}
\cs_generate_variant:Nn \pdfmanagement_add:nnn {nee}
\cs_new:Npn \__pdfmanagement_patch_pgfextgs:w  #1/#2<<#3>>#4\q_stop
  {
    \tl_set:Ne\l_tmpa_tl{#2}
      \pdfmanagement_add:nee
        {Page/Resources/ExtGState}{\tl_trim_spaces:V\l_tmpa_tl}{<<#3 #4>>}
  }
\cs_new:Npn \__pdfmanagement_patch_pgfpatterns:w  #1/#2\space#3\q_stop
  {
     \pdfmanagement_add:nee
        {Page/Resources/Pattern}{\tl_trim_spaces:n{#2}}{#3}
  }
\cs_new:Npn \__pdfmanagement_patch_pgfcolorspaces:w  #1/#2[#3]#4\q_stop
  {
    \pdfmanagement_add:nee
      {Page/Resources/ColorSpace}{\tl_trim_spaces:n{#2}}{[#3]}
  }

```

1.7 transparent

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

```
\bool_if:NT \g__pdfmanagement_firstaid_transparent_bool
 {
  \AddToHook{package/transparent/after}
    {
     \@ifpackagelater{transparent}{2022-10-27}{}
       {
         \msg_warning:nnnn{pdfmanagement}{firstaid-too-old}
         {transparent}
         {1.5~from~2022-10-27}
       }
    }
 }
```

1.8 pdflscape

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

```
\bool_if:NT \g__pdfmanagement_firstaid_pdflscape_bool
 {
  \AddToHook{package/pdflscape/after}
    {
     \@ifpackagelater{pdflscape}{2022-10-27}{}
       {
         \msg_warning:nnnn{pdfmanagement}{firstaid-too-old}
         {pdflscape}
         {0.13~from~2022-10-27}
       }
    }
 }
```

1.9 xmp

This handles the new xmp code.

```
\bool_if:NT \g__pdfmanagement_firstaid_xmp_bool
 {
   \disable@package@load{hyperxmp}{\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{hyperxm
   \disable@package@load{pdfx}    {\msg_warning:nnn{pdfmanagement}{firstaid-disabled}{pdfx}}
   \AddToHook{package/doclicense/after}
    {
      \AtBeginDocument
        {
          \hypersetup
            {
              pdfcopyright  = {\doclicenseLongTextForHyperref},
              pdflicenseurl = {\doclicenseURL},
            }
        }
    }
 }
⟨/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.

```
⟨*package⟩
\bool_if:NT \g__pdfmanagement_firstaid_colorspace_bool
 {
    \AddToHook
      {
       package/colorspace/after
      }
      {\RequirePackage{colorspace-patches-tmp-ltx}}
 }
⟨/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

\\ 25, 28

A

\AddToHook 35, 107, 121, 166, 252, 264, 278, 294
\AtBeginDocument 280
\AtEndPreamble 222

B

bool commands:
\bool_gset_false:N 17
\bool_gset_true:N 11
\bool_if:NTF 33, 59, 72, 101, 105, 163, 250, 262, 274, 292
\bool_if_exist:NTF 15
\bool_lazy_all:nTF 114
\bool_new:N 10

C

clist commands:
\clist_map_inline:Nn 13
\clist_map_inline:nn 6
color commands:
\color_if_exist:nTF 43
\color_set:nn *3, 4*
color internal commands:
__color_backend_fill:n 135
__color_backend_fill_reset: . . 158
\l__color_backend_fill_tl . . 125, 137
__color_backend_reset: 157, 158, 159
__color_backend_select:nn 123
__color_backend_stroke:n 146
__color_backend_stroke_reset: . 159
\l__color_backend_stroke_tl 126, 148
\convertcolorspec 50
cs commands:
\cs_generate_variant:Nn 231, 232
\cs_if_exist:NTF 48
\cs_new:Npn 225, 233, 239, 244
\cs_new_protected:Npn 192
\cs_set:Nn 82
\cs_set_eq:NN 158, 159, 171
\cs_set_protected:Npn 123, 135, 146, 157

D

\DeclareHookRule 112
\def 40, 174, 186, 187, 188, 189, 195, 207, 212, 216
\definecolor *3, 4*
\directlua 129, 140, 151
\doclicenseLongTextForHyperref . . . 284
\doclicenseURL 285
\DocumentMetadata *1*

E

exp commands:
\exp_after:wN 209

F

fontspec commands:
\fontspec_parse_colour:niii 57
fontspec internal commands:
\l__fontspec_firsttime_bool . 59, 72
\l__fontspec_fontfeat_clist 85
__fontspec_get_features:n 82
\c__fontspec_hexcol_tl 94
\l__fontspec_hexcol_tl 45, 50, 54, 93, 95
__fontspec_init_fontface: 84
__fontspec_int_mult_truncate:Nn 69
__fontspec_keys_define_code:nnn 41, 66
\l__fontspec_keys_leftover_clist 86, 87, 88
__fontspec_keys_set_known:nnN 85, 87
\l__fontspec_mapping_tl 90, 91
\c__fontspec_opacity_tl . . . 40, 70, 94
\l__fontspec_opacity_tl 70, 75, 93, 95
\l__fontspec_tmp_int . . 68, 69, 78, 79
__fontspec_update_featstr:n 91, 95
__fontspec_warning:nn 60, 73

H

\hypersetup 282

I

int commands:
\int_compare:nTF 53, 56, 78
\int_set:Nn 68
\int_to_hex:n 79

K

keys commands:
\keys_set:nn 88

L

\let 201, 203, 204, 205
\luaescapestring 131, 142, 153

M

msg commands:
\msg_info:nnn 165
\msg_new:nnn 20, 22, 24, 27
\msg_warning:nnn 276, 277
\msg_warning:nnnn 256, 268

P

pdf commands:
\pdf_object_ref:n 197, 198, 199
pdfmanagement commands:
\pdfmanagement_add:nnn 232, 236, 241, 246
\pdfmanagement_if_active:TF . . . 176
pdfmanagement internal commands:
\g__pdfmanagement_firstaid_color_bool 101
\g__pdfmanagement_firstaid_colorspace_bool 292
\g__pdfmanagement_firstaid_fontspec_bool 33
\g__pdfmanagement_firstaid_luacolor_bool 117
\g__pdfmanagement_firstaid_pdflscape_bool 262
\g__pdfmanagement_firstaid_pgf_bool 163

\g__pdfmanagement_firstaid_transparent_bool 250
\g__pdfmanagement_firstaid_xcolor_bool 105
\g__pdfmanagement_firstaid_xmp_bool 274
\g__pdfmanagement_firstaidoff_clist 13
__pdfmanagement_patch_pgfcolorspaces:w 218, 244
__pdfmanagement_patch_pgfextgs:w 210, 233
__pdfmanagement_patch_pgfpatterns:w 214, 239
__pdfmanagement_pgf_sys_setuppdfresources_plain: 178, 192
__pdfmanagement_pgfori_pgfutil@setuppdfresources 172, 181
__pdfmanagement_split_dict_entry_aux:NNw 225
\ProvidesExplPackage 2

Q

quark commands:
\q_stop 210, 214, 218, 225, 233, 239, 244

R

\relax 201
\RequirePackage 111, 298

S

\space 239
str commands:
\str_if_eq:nnTF 93
sys commands:
\sys_if_engine_luatex:TF 31
\sys_if_engine_luatex_p: 116
\sys_if_output_pdf_p: 118

T

TEX and LATEX 2ε commands:
\@ifpackagelater 37, 254, 266
\color@ 48
\declare@file@substitution 103
\disable@package@load 276, 277
\LuaCol@Attribute 127, 138, 149
\LuaCol@setattribute . . 127, 138, 149
\pgf@sys@addpdfresource@colorspaces@plain 187, 216
\pgf@sys@addpdfresource@extgs@plain 186, 207
\pgf@sys@addpdfresource@patterns@plain 188, 212
\pgf@sys@pdf@check@resources . . 201
\pgf@sys@pdf@possible@resources 190, 195
\pgf@sys@pgf@resource@list@colorspaces 205
\pgf@sys@pgf@resource@list@extgs 203
\pgf@sys@pgf@resource@list@patterns 204
\pgf@sys@setuppdfresources@plain 189
\pgfutil@addpdfresource@colorspaces 5, 187
\pgfutil@addpdfresource@extgs 5, 186
\pgfutil@addpdfresource@patterns 5, 188
\pgfutil@empty 203, 204, 205
\pgfutil@setuppdfresources 173, 174, 189, 222
tl commands:
\tl_count:n 53, 56
\tl_if_empty:NTF 90
\tl_if_eq:NNTF 70
\tl_set:Nn 45, 54, 75, 125, 126, 137, 148, 227, 228, 235
\tl_trim_spaces:n . . 231, 237, 242, 247
\l_tmpa_tl 235, 237
token commands:
\token_to_str:N 48
\typeout 38