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

The LaTeX Project*

Version 0.95h, released 2021-07-31

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,...}`.

```
⟨*package⟩
\ProvidesExplPackage{pdfmanagement-firstaid}{2021-07-31}{0.95h}
  {LaTeX PDF management testphase bundle / firstaid-patches}

⟨@@=pdfmanagement⟩
\clist_map_inline:nn {pgf,transparent,hyperxmp,pdflscape,xcolor,color,beamer}
  {
    \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 }
```

1.1 beamer

beamer makes use of the now unsupported syntax `linkbordercolor={.5 .5 .5}`. To avoid an error we define this color as a name.

```
\bool_if:NT \g__pdfmanagement_firstaid_beamer_bool
```

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

```
  {
    \color_set:nnn{.5~.5~.5}{rgb}{0.5,0.5,0.5}
  }
```

1.2 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}
  }
```

1.3 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/after/xcolor}
       {\RequirePackage{xcolor-patches-tmp-ltx}\XC@@names}
  }
```

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.

```
\bool_if:NT \g__pdfmanagement_firstaid_pgf_bool
  {
    \msg_info:nnn{pdfmanagement }{firstaid}{pgf}
    \AddToHook{file/after/pgfrcs.sty}
      {
        \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 {nex}
\cs_new:Npn \__pdfmanagement_patch_pgfextgs:w  #1/#2<<#3>>#4\q_stop
  {
    \tl_set:Nx\l_tmpa_tl{#2}
      \pdfmanagement_add:nex
        {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:nex
        {Page/Resources/Pattern}{\tl_trim_spaces:n{#2}}{#3}
  }
\cs_new:Npn \__pdfmanagement_patch_pgfcolorspaces:w  #1/#2[#3]#4\q_stop
  {
    \pdfmanagement_add:nex
      {Page/Resources/ColorSpace}{\tl_trim_spaces:n{#2}}{[#3]}
  }

```

1.5 transparent

We simply replace by the new version.

```
\bool_if:NT \g__pdfmanagement_firstaid_transparent_bool
 {
   \declare@file@substitution{transparent.sty}{transparent-ltx.sty}
 }
```

1.6 pdflscape

We simply replace by the new version.

```
\bool_if:NT \g__pdfmanagement_firstaid_pdflscape_bool
 {
   \declare@file@substitution{pdflscape.sty}{pdflscape-ltx.sty}
 }
```

1.7 hyperxmp

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

```
\bool_if:NT \g__pdfmanagement_firstaid_hyperxmp_bool
 {
```

```
    \AddToHook
      {file/after/hyperxmp.sty}
      {\RequirePackage{hyperxmp-patches-tmp-ltx}}
 }
⟨/package⟩
```

1.8 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⟩
⟨@@=pdf⟩
% this must be earlier, to avoid problems with luatex which has two pageresources
% lua/tex
\hook_gput_code:nnn {begindocument} {pdf}
  {
    \tl_if_exist:NT \spc@op
      {
        \def\spc@Pageresources#1{}
      }

  }

\hook_gput_code:nnn {begindocument/end} {pdf}
  {
    \tl_if_exist:NT \spc@op
      {
        \__pdf_backend_object_new:nn  {__spc_extgstate_op_false}{dict}
        \__pdf_backend_object_write:nn
          {__spc_extgstate_op_false}
          {/Type /ExtGState~/op~false~/OP~false}
        \pdfmanagement_add:nnn
          {Page/Resources/ExtGState}
          {SPCko}
          {\__pdf_backend_object_ref:n {__spc_extgstate_op_false}}
        \__pdf_backend_object_new:nn  {__spc_extgstate_op_true0}{dict}
        \__pdf_backend_object_write:nn
          {__spc_extgstate_op_true0}
          {/Type /ExtGState~/op~true~/OP~true~/OPM~0}%
        \pdfmanagement_add:nnn
          {Page/Resources/ExtGState}
          {SPCmz}
          {\__pdf_backend_object_ref:n {__spc_extgstate_op_true0}}
        \__pdf_backend_object_new:nn  {__spc_extgstate_op_true1}{dict}
        \__pdf_backend_object_write:nn
          {__spc_extgstate_op_true1}
          {/Type /ExtGState~/op~true~/OP~true~/OPM~1}%
         \pdfmanagement_add:nnn
          {Page/Resources/ExtGState}
          {SPCop}
          {\__pdf_backend_object_ref:n {__spc_extgstate_op_true1}}
      }
```

```
}
⟨/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

\AddToHook 30, 37, 128
\AtEndPreamble 90

B

bool commands:
\bool_gset_false:N 15
\bool_gset_true:N 9
\bool_if:NTF 20, 24, 28, 34, 118, 122, 126
\bool_if_exist:NTF 13
\bool_new:N 8

C

clist commands:
\clist_map_inline:Nn 11
\clist_map_inline:nn 6
color commands:
\color_set:nn *2*
\color_set:nnn 22
cs commands:
\cs_generate_variant:Nn 99, 100
\cs_new:Npn 93, 101, 107, 112
\cs_new_protected:Npn 60
\cs_set_eq:NN 39

D

\DeclareDocumentMetadata *1*
\def . . 42, 54, 55, 56, 57, 63, 75, 80, 84, 141
\definecolor *2*

E

exp commands:
\exp_after:wN 77

H

hook commands:
\hook_gput_code:nnn 137, 146

L

\let 69, 71, 72, 73

M

msg commands:
\msg_info:nnn 36
\msg_new:nnn 18

P

pdf commands:
\pdf_object_ref:n 65, 66, 67
pdf internal commands:
__pdf_backend_object_new:nn 150, 158, 166
__pdf_backend_object_ref:n 157, 165, 173
__pdf_backend_object_write:nn 151, 159, 167
pdfmanagement commands:
\pdfmanagement_add:nnn 100, 104, 109, 114, 154, 162, 170
\pdfmanagement_if_active:TF 44
pdfmanagement internal commands:
\g__pdfmanagement_firstaid_beamer_bool 20
\g__pdfmanagement_firstaid_color_bool 24
\g__pdfmanagement_firstaid_hyperxmp_bool 126
\g__pdfmanagement_firstaid_pdflscape_bool 122
\g__pdfmanagement_firstaid_pgf_bool 34
\g__pdfmanagement_firstaid_transparent_bool 118
\g__pdfmanagement_firstaid_xcolor_bool 28
\g__pdfmanagement_firstaidoff_clist 11
__pdfmanagement_patch_pgfcolorspaces:w 86, 112
__pdfmanagement_patch_pgfextgs:w 78, 101
__pdfmanagement_patch_pgfpatterns:w 82, 107
__pdfmanagement_pgf_sys_setuppdfresources_plain: . . 46, 60
__pdfmanagement_pgfori_pgfutil@setuppdfresources 40, 49
__pdfmanagement_split_dict_entry_aux:NNw 93
\ProvidesExplPackage 2

Q

quark commands:

\q_stop . . 78, 82, 86, 93, 101, 107, 112

R

\relax 69
\RequirePackage 31, 130

S

\space 107

T

TEX and LATEX 2ε commands:

\declare@file@substitution 26, 120, 124
\pgf@sys@addpdfresource@colorspaces@plain 55, 84
\pgf@sys@addpdfresource@extgs@plain 54, 75
\pgf@sys@addpdfresource@patterns@plain 56, 80
\pgf@sys@pdf@check@resources . . . 69
\pgf@sys@pdf@possible@resources 58, 63
\pgf@sys@pgf@resource@list@colorspaces 73
\pgf@sys@pgf@resource@list@extgs 71
\pgf@sys@pgf@resource@list@patterns 72
\pgf@sys@setuppdfresources@plain 57
\pgfutil@addpdfresource@colorspaces *2*, 55
\pgfutil@addpdfresource@extgs . *2*, 54
\pgfutil@addpdfresource@patterns *2*, 56
\pgfutil@empty 71, 72, 73
\pgfutil@setuppdfresources 41, 42, 57, 90
\spc@op 139, 148
\spc@Pageresources 141
\XC@@names 31

tl commands:

\tl_if_exist:NTF 139, 148
\tl_set:Nn 95, 96, 103
\tl_trim_spaces:n . . . 99, 105, 110, 115
\l_tmpa_tl 103, 105