The pdfcol package

Heiko Oberdiek*

2026-06-20 v1.8

Abstract

Since version 1.40 pdfTeX supports color stacks. The driver file `pdftex.def` for package color defines and uses a main color stack since version v0.04b. Package pdfcol is intended for package writers. It defines macros for setting and maintaining new color stacks. It works also with luaTeX.

Contents

*Please report any issues at https://github.com/ho-tex/pdfcol/issues

1 Documentation

Version 1.40 of pdfTeX adds new primitives `\pdfcolorstackinit` and `\pdfcolorstack`. Now color stacks can be defined and used. A main color stack is maintained by the driver file `pdftex.def` similar to dvips or dvipdfm. However the number of color stacks is not limited to one in pdfTeX. Thus further color problems can now be solved, such as footnotes across pages or text that is set in parallel columns (e.g. packages parallel or parcolumn). Unlike the main color stack, the support by additional color stacks cannot be done in a transparent manner.

This package pdfcol provides an easier interface to additional color stacks without the need to use the low level primitives.

The engine luaTeX is supported, but if the package luacolor is detected (which uses attributes to set colors) the package interfaces are disabled.

1.1 Requirements

- pdfTeX 1.40 or greater or luaTeX.
- pdfTeXor luaTeX in PDF mode. (I don't know a DVI driver that support several color stacks.)
- `pdftex.def` or `luatex.def` newer than 2007.

Package pdfcol checks the requirements and sets switch `\ifpdfcolAvailable` accordingly.

1.2 Interface

`\ifpdfcolAvailable` `\ifpdfcolAvailable⟨true⟩\else⟨false⟩\fi`

If the requirements of section 1.1 are met the switch `\ifpdfcolAvailable` behaves as `\iftrue`. Otherwise the other interface macros in this section will be disabled with a message. Also the first use of such a macro will print a message. The messages are print to the `.log` file only if pdfTeX and luaTeX not used in PDF mode.

`\pdfcolErrorNoStacks` The first call of `\pdfcolErrorNoStacks` prints an error message, if color stacks are not available.

\pdfcolInitStack `\pdfcolInitStack{⟨name⟩}`

A new color stack is initialized by `\pdfcolInitStack`. The ⟨*name*⟩ is used for identifying the stack. It usually consists of letters and digits. (The name must survive a `\csname`.)

The intention of the macro is the definition of an additional color stack. Thus the stack is not page bounded like the main color stack. Black (`0 g 0 G`) is used as initial color value. And colors are written with modifier `direct` that means without setting the current transfer matrix and changing the current point (see documentation of pdfTEX for `\pdfliteral direct{...}`).

\pdfcolIfStackExists `\pdfcolIfStackExists {⟨name⟩} {⟨true⟩} {⟨false⟩}`

Macro `\pdfcolIfStackExists` checks whether color stack ⟨*name*⟩ exists. In case of success argument ⟨*then*⟩ is executed and ⟨*else*⟩ otherwise.

\pdfcolSwitchStack `\pdfcolSwitchStack{⟨name⟩}`

Macro `\pdfcolSwitchStack` switches the color stack. The color macros of package color, xcolor and the l3color module now uses the new color stack with name ⟨*name*⟩.

\pdfcolSetCurrentColor Macro `\pdfcolSetCurrentColor` replaces the topmost entry of the stack by the current color (`\current@color`).

\pdfcolSetCurrent `\pdfcolSetCurrent{⟨name⟩}`

Macro `\pdfcolSetCurrent` sets the color that is read in the top-most entry of color stack ⟨*name*⟩. If ⟨*name*⟩ is empty, the default color stack is used.

1.3 Example (for LaTeX)

In LaTeX colors can be set through the commands of the packages color/xcolor, with the in-built commands of the l3color module in l3kernel. Unless the package luacolor is used these commands do *not* naturally respect TeX groups, instead they insert literals into the PDF that start a color and the color must be reset to get back to the previous. This is typically done with a command that is inserted with `\aftergroup` *after* the enclosing group. The command `\pdfcolSwitchStack` on the other hand switches the color stack locally: At the end of the enclosing group the stack is switched back again. This can lead to the odd behaviour that a color takes it resets from the wrong stack:

```
\showoutput
 black             % main stack: black
  {
  \color{green}
   green           % main stack: black, green
   {
     green
     \pdfcolSwitchStack{test} % test stack: black
     green
     \color{yellow} % test stack: black, yellow
     yellow
                   % End of group: switch to main stack
```

```
 }               % After group: reset of 'color{yellow}'
                 % pops green from *main* stack and green ends
                 % main stack: black
 after group green? No black!!
}
```

If one checks with `\showoutput` one can see the following sequence

```
\pdfcolorstack 0 push {0 1 0 rg 0 1 0 RG} % from \color{green}
\pdfcolorstack 1 push {0 0 1 0 k 0 0 1 0 K} % from \color{yellow}
\pdfcolorstack 0 pop  % reset of \color{yellow}
\pdfcolorstack 0 pop  % reset of \color{green}
```

and pdftex complains about one `pop` too much:

```
pdfTeX warning: pdflatex-dev.exe: pop empty color page stack 0
```

To avoid such problems it is important to ensure, that the reset of colors is done from the stack on which they pushed. This means that the stack switch must outside of the group where the colors from the new stack are used:

```
black           % main stack: black
{
 \color{green}
  green           % main stack: black, green
  {
   green
   \pdfcolSwitchStack{test} % test stack: black
    {
     green
     \color{yellow} % test stack: black, yellow
     yellow         % End of group: test stack still active
    }
      % After group: color yellow popped from test stack
  }
  after group green?? No still black!!
}
```

black green green green yellow after group green?? No still black!!

Now the colors are popped from the correct stacks:

```
\pdfcolorstack 0 push {0 1 0 rg 0 1 0 RG} % from \color{green}
\pdfcolorstack 1 push {0 0 1 0 k 0 0 1 0 K} % from \color{yellow}
\pdfcolorstack 1 pop  % reset of \color{yellow}
\pdfcolorstack 0 pop  % reset of \color{green}
```

But this still gives green after the group: Yellow has been correctly ended but that doesn't mean that green takes over again: The new color stack had black as starting value and this is used once the yellow is popped. To get green again one must therefore replace the starting value of the new stack by the currently active color. For this the package offers the command `\pdfcolSetCurrentColor` which replaces the current top most color of a stack by the color stored in `\current@color`:

```
black           % main stack: black
{
 \color{green}
  green           % main stack: black, green
  {
   green
   \pdfcolSwitchStack{test} % test stack: black
   \pdfcolSetCurrentColor   % test stack: green
    {
     green
     \color{yellow} % test stack: green, yellow
     yellow         % End of group: test stack still active
    }
      % After group: color yellow popped from test stack, green used
  }
  after group green!!
}%
```

black green green green yellow after group green!!

An alternative method to the same effect is to take the current color from the main stack and to issue it again after the group:

```
black           % main stack: black
{
 \color{green}
  green           % main stack: black, green
  {
   green
   \pdfcolSwitchStack{test} % test stack: black
    {
     green
     \color{yellow} % test stack: black, yellow
     yellow         % End of group: test stack still active
    }
      % After group: color yellow popped from test stack
  }%
  \pdfcolSetCurrent{}% set current color of main stack = green
  after group green!!
}%
```

black green green green yellow after group green!!

2 Implementation

1 ⟨∗package⟩

2.1 Reload check and package identification

Reload check, especially if the package is not used with LaTeX.

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\let\expandafter\x\csname ver@pdfcol.sty\endcsname
  \ifx\x\relax % plain-TeX, first loading
  \else
    \def\empty{}%
    \ifx\x\empty % LaTeX, first loading,
      % variable is initialized, but \ProvidesPackage not yet seen
    \else
      \expandafter\ifx\csname PackageInfo\endcsname\relax
        \def\x#1#2{%
          \immediate\write-1{Package #1 Info: #2.}%
        }%
      \else
        \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
      \fi
      \x{pdfcol}{The package is already loaded}%
      \aftergroup\endinput
    \fi
  \fi
\endgroup%
```

Package identification:

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \catcode40=12 % (
  \catcode41=12 % )
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode47=12 % /
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode91=12 % [
  \catcode93=12 % ]
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
```

```
    \def\x#1#2#3[#4]{\endgroup
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \def\x#1#2[#3]{\endgroup
      #2[{#3}]%
      \ifx#1\@undefined
        \xdef#1{#3}%
      \fi
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@pdfcol.sty\endcsname
\ProvidesPackage{pdfcol}%
  [2026-06-20 v1.8 Handle new color stacks for pdfTeX (HO)]%
```

2.2 Catcodes

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname PDFCOL@AtEnd\endcsname{%
      \endlinechar=\the\endlinechar\relax
      \catcode13=\the\catcode13\relax
      \catcode32=\the\catcode32\relax
      \catcode35=\the\catcode35\relax
      \catcode61=\the\catcode61\relax
      \catcode64=\the\catcode64\relax
      \catcode123=\the\catcode123\relax
      \catcode125=\the\catcode125\relax
    }%
  }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\PDFCOL@AtEnd{%
    \PDFCOL@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{39}{12}% '
```

```
\TMP@EnsureCode{40}{12}% (
\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{43}{12}% +
\TMP@EnsureCode{44}{12}% ,
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{96}{12}% `
\edef\PDFCOL@AtEnd{\PDFCOL@AtEnd\noexpand\endinput}
```

2.3 Check requirements

\PDFCOL@RequirePackage

```
\ifcsname RequirePackage\endcsname
  \def\PDFCOL@RequirePackage#1[#2]{%
    \RequirePackage{#1}[{#2}]%
  }%
\else
  \def\PDFCOL@RequirePackage#1[#2]{\input #1.sty\relax}%
\fi
```

(*End of definition for* `\PDFCOL@RequirePackage`.)

LuaTeX Compatability

```
\ifx\pdfextension\@undefined\else
 \def\pdfcolorstackinit {\pdffeedback colorstackinit}
 \protected\def\pdfcolorstack     {\pdfextension colorstack}
\fi
```

```
\PDFCOL@RequirePackage{ltxcmds}[2010/03/01]
```

\ifpdfcolAvailable

```
\ltx@newif\ifpdfcolAvailable
\pdfcolAvailabletrue
```

(*End of definition for* `\ifpdfcolAvailable`. *This function is documented on page 2.*)

2.3.1 Check package **luacolor**

```
\ltx@newif\ifPDFCOL@luacolor
\ifcsname ver@luacolor.sty\endcsname
  \PDFCOL@luacolortrue
\else
  \PDFCOL@luacolorfalse
\fi
```

2.3.2 Check PDF mode

```
\PDFCOL@RequirePackage{infwarerr}[2007/09/09]
\PDFCOL@RequirePackage{iftex}[2019/11/07]
\ifcase\ifpdf\ifPDFCOL@luacolor 1\fi\else 1\fi0 %
  \def\PDFCOL@Message{%
    \@PackageWarningNoLine{pdfcol}%
  }%
```

```
\else
  \pdfcolAvailablefalse
  \def\PDFCOL@Message{%
    \@PackageInfoNoLine{pdfcol}%
  }%
  \PDFCOL@Message{%
    Interface disabled because of %
    \ifPDFCOL@luacolor
      package `luacolor'%
    \else
      missing PDF mode of pdfTeX%
    \fi
  }%
\fi
```

2.3.3 Check version of pdfTeX

The version check has been removed. We assume a new enough pdfTeX.

2.3.4 Check `pdftex.def`

We no longer check the version but assume that it is new enough.

```
\ifpdfcolAvailable
```

Try to load package color if it is not yet loaded (LaTeX case).

```
  \ifcsname documentclass\endcsname
    \RequirePackage{color}
    \RequirePackage{etoolbox}
```

To support l3color we need to set also the backend stack integer to the new value.

```
    \def\pdfcol@switch@kernel@colorstack#1%
     {\csname l__color_backend_stack_int\endcsname =\csname pdfcol@Stack@#1\endcsname}
  \else
    \def\pdfcol@switch@kernel@colorstack#1{}
  \fi
\fi

\let\pdfcolAvailabletrue\relax
\let\pdfcolAvailablefalse\relax
```

2.4 Enabled interface macros

```
\ifpdfcolAvailable
```

\pdfcolErrorNoStacks

```
  \let\pdfcolErrorNoStacks\relax
```

(*End of definition for* `\pdfcolErrorNoStacks`*. This function is documented on page* 2.)

\pdfcol@Value

```
  \expandafter\ifx\csname pdfcol@Value\endcsname\relax
    \def\pdfcol@Value{0 g 0 G}%
  \fi
```

(*End of definition for* `\pdfcol@Value`.)

`\pdfcol@LiteralModifier`

```
\expandafter\ifx\csname pdfcol@LiteralModifier\endcsname\relax
  \def\pdfcol@LiteralModifier{direct}%
\fi
```

(*End of definition for* `\pdfcol@LiteralModifier`.)

`\pdfcolInitStack`

```
\def\pdfcolInitStack#1{%
  \expandafter\ifx\csname pdfcol@Stack@#1\endcsname\relax
    \global\expandafter\chardef\csname pdfcol@Stack@#1\endcsname=%
        \pdfcolorstackinit\pdfcol@LiteralModifier{\pdfcol@Value}%
        \relax
    \@PackageInfo{pdfcol}{%
      New color stack `#1' = \number\csname pdfcol@Stack@#1\endcsname
    }%
  \else
    \@PackageError{pdfcol}{%
      Stack `#1' is already defined%
    }\@ehc
  \fi
}%
```

(*End of definition for* `\pdfcolInitStack`. *This function is documented on page* 3.)

`\pdfcolIfStackExists`

```
\def\pdfcolIfStackExists#1{%
  \expandafter\ifx\csname pdfcol@Stack@#1\endcsname\relax
    \expandafter\@secondoftwo
  \else
    \expandafter\@firstoftwo
  \fi
}%
```

(*End of definition for* `\pdfcolIfStackExists`. *This function is documented on page* 3.)

`\@firstoftwo`

```
\ifcsname @firstoftwo\endcsname
  \long\def\@firstoftwo#1#2{#1}%
\fi
```

(*End of definition for* `\@firstoftwo`.)

`\@secondoftwo`

```
\ifcsname @secondoftwo\endcsname
  \long\def\@secondoftwo#1#2{#2}%
\fi
```

(*End of definition for* `\@secondoftwo`.)

\pdfcolSwitchStack

```
\def\pdfcolSwitchStack#1{%
  \pdfcolIfStackExists{#1}{%
    \expandafter\let\expandafter\@pdfcolorstack
                    \csname pdfcol@Stack@#1\endcsname
    \pdfcol@switch@kernel@colorstack{#1}%
  }{%
    \pdfcol@ErrorNoStack{#1}%
  }%
}%
```

(*End of definition for* `\pdfcolSwitchStack`*. This function is documented on page 3.*)

\pdfcolSetCurrentColor

```
\def\pdfcolSetCurrentColor{%
  \pdfcolorstack\@pdfcolorstack set{\current@color}%
}%
```

(*End of definition for* `\pdfcolSetCurrentColor`*. This function is documented on page 3.*)

\pdfcolSetCurrent

```
\def\pdfcolSetCurrent#1{%
  \ifx\\#1\\%
    \pdfcolorstack\@pdfcolorstack current\relax
  \else
    \pdfcolIfStackExists{#1}{%
      \pdfcolorstack\csname pdfcol@Stack@#1\endcsname current\relax
    }{%
      \pdfcol@ErrorNoStack{#1}%
    }%
  \fi
}%
```

(*End of definition for* `\pdfcolSetCurrent`*. This function is documented on page 3.*)

\pdfcol@ErrorNoStack

```
\def\pdfcol@ErrorNoStack#1{%
  \@PackageError{pdfcol}{Stack `#1' does not exists}\@ehc
}%
```

(*End of definition for* `\pdfcol@ErrorNoStack`*.*)

2.5 Disabled interface macros

```
\else
```

\pdfcolErrorNoStacks

```
\def\pdfcolErrorNoStacks{%
  \@PackageError{pdfcol}{%
    Color stacks are not available%
  }{%
    Update pdfTeX (1.40) and `pdftex.def' (0.04b) %
        if necessary.\MessageBreak
    Ensure that `pdftex.def' is loaded %
        (package `color' or `xcolor').\MessageBreak
```

```
      Further messages can be found in TeX's %
          protocol file `\jobname.log'.\MessageBreak
      \MessageBreak
      \@ehc
    }%
    \global\let\pdfcolErrorNoStacks\relax
  }%
```

(*End of definition for* `\pdfcolErrorNoStacks`. *This function is documented on page 2.*)

`\PDFCOL@Disabled`

```
  \def\PDFCOL@Disabled{%
    \PDFCOL@Message{%
      pdfTeX's color stacks are not available%
    }%
    \global\let\PDFCOL@Disabled\relax
  }%
```

(*End of definition for* `\PDFCOL@Disabled`.)

`\pdfcolInitStack`

```
  \def\pdfcolInitStack#1{%
    \PDFCOL@Disabled
  }%
```

(*End of definition for* `\pdfcolInitStack`. *This function is documented on page 3.*)

`\pdfcolIfStackExists`

```
  \long\def\pdfcolIfStackExists#1#2#3{#3}%
```

(*End of definition for* `\pdfcolIfStackExists`. *This function is documented on page 3.*)

`\pdfcolSwitchStack`

```
  \def\pdfcolSwitchStack#1{%
    \PDFCOL@Disabled
  }%
```

(*End of definition for* `\pdfcolSwitchStack`. *This function is documented on page 3.*)

`\pdfcolSetCurrentColor`

```
  \def\pdfcolSetCurrentColor{%
    \PDFCOL@Disabled
  }%
```

(*End of definition for* `\pdfcolSetCurrentColor`. *This function is documented on page 3.*)

`\pdfcolSetCurrent`

```
  \def\pdfcolSetCurrent#1{%
    \PDFCOL@Disabled
  }%
```

(*End of definition for* `\pdfcolSetCurrent`. *This function is documented on page 3.*)

```
\fi

\PDFCOL@AtEnd%
⟨/package⟩
```

3 Installation

3.1 Download

Package. This package is available on CTAN[1]:

`CTAN:macros/latex/contrib/pdfcol/pdfcol.dtx` The source file.

`CTAN:macros/latex/contrib/pdfcol/pdfcol.pdf` Documentation.

3.2 Package installation

The easiest is to use the package manager from your TeX system. If you want to install manually:

Unpacking. The `.dtx` file is docstrip archive. The files are extracted from the `.dtx` by compiling the file with some TeX program.

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
pdfcol.sty → tex/latex/pdfcol/pdfcol.sty
pdfcol.pdf → doc/latex/pdfcol/pdfcol.pdf
pdfcol.dtx → source/latex/pdfcol/pdfcol.dtx
```

3.3 Refresh file name databases

If your TEX distribution (TEX Live, MiKTEX, …) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

Generating the documentation. You can use the `.dtx` to generate the documentation.

```
lualatex pdfcol.dtx
makeindex -s gind.ist pdfcol.idx
lualatex pdfcol.dtx
makeindex -s gind.ist pdfcol.idx
lualatex pdfcol.dtx
```

4 History

[2007/09/09 v1.0]

- First version.

[2007/12/09 v1.1]

- `\pdfcolSetCurrentColor` added.

[1] `CTAN:pkg/pdfcol`

[2007/12/12 v1.2]

- Detection for package luacolor added.

[2016/05/16 v1.3]

- Documentation updates.

[2016/05/17 v1.4]

- Use luatex85 package for new luatex compatibility

[2018/11/01 v1.5]

- Remove luatex85 dependency

[2019/12/29 v1.6]

- iftex package.

[2022-09-21 v1.7]

- Improved luatex compatibility, removed unneeded version checks.

[2026-06-20 v1.8]

- Added support for color commands of the l3color module, requires L3 programming layer 2026-06-19
- Documentation is now tagged.

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.