

colorspace

Spot colors, mixed inks and more

Version 1.3 (2019-03-25)

Javier Bezos (<http://www.texnia.com>)

The aim of this package is, as its name implies, to provide tools for PDF color spaces. It requires `xcolor`, which is loaded if it has not been before. It seems to work with `tikz`.

Currently it supports what I think are the most common tools:

- Spot colors, with a clean user interface, and including tints (with the `!` notation).
- CIE LAB spot colors.
- Proper switching of color spaces.
- Mixed spot and process colors (up to 4), like shades (ie, a spot color with black). Also CIE LAB if a CMYK equivalent is provided.
- ICC based (or “tagged”) default CMYK, RGB and Gray spaces.
- Overprinting (across pages, using the color stack).

Currently only `pdftex` and `luatex` are supported. Support for `xetex` is on the ‘to do’ list, but due to the limitations of this engine this task is somewhat challenging and I’m not sure all features will be implemented.

Other functions related to the PDF color spaces (indexed, calibrated, Lab spaces) are not yet supported, but they are under study. Calibrated colors, although not directly supported, can be defined with an ICC profile created with `LPROF`¹ and then assigned to a default space as described below.

They apply to text and line art only, not external images. For the latter, `graphicx` provides a plea of (undocumented) transformations: `interpolate`, `decodearray`, `maskarray`, `intent`, `ocobjnum`, and `ocobjref`. For transparency, see `transparent`, by Heiko Oberdiek.

This package is still evolving, but the basic behaviour described here will be preserved in future versions. However, some functions from `xcolor` might not work yet (for example `\selectcolormodel` and the corresponding package options `cmk` and the like).

Declarations are global and should go in the preamble.

This package is built on the previous attempts to provide spot colors and other additional features by Jens Elstner, Stephan Lehmke and Siep Kroonenberg (with some inspiration from `ConTeXt`, too).

1 CMYK spot colors

Spot colors are defined with a single macro:

¹<http://lprof.sourceforge.net/>


```
\definespotcolor{<latex-name>}{<PDF-name>}{<CMYK-equivalent>}
```

Here $\langle latex-name \rangle$ is the L^AT_EX name, as used in `\color` and the like, $\langle PDF-name \rangle$ is the PDF name (usually taken from a swatch book; multiple spaces are collapsed into one) as shown by PDF viewers, and the four numbers are the CMYK equivalent. L^AT_EX knows nothing about the PDF name, which is just a string to be written to the generated file, while the PDF knows nothing about the L^AT_EX name.

To mix inks, see below.


EXAMPLE Write, for example:

```
\definespotcolor{foo}{BarTone 555 GN}{.8,.2,.5,.3}
```

This creates the color , which is used in L^AT_EX as `\color{foo}`. If you preflight the PDF file you will see a color named 'BarTone 555 GN' besides cyan, magenta, yellow and black.

You can use tints as usual in `xcolor`, like:

```
\color{foo!50}  
\colorlet{foo50}{foo!50}
```

which would produce , and even set tints from other tints.

NOTE The special PDF names `All` (for all plates) and `None` work as expected:

```
\definespotcolor{registration}{All}{1,1,1,1}
```

NOTE Remember as far PDF is concerned a spot color is a color space on its own.


NOTE This package does not provide a list of Pantone, TrueMatch, HKS, Folcoltone, Toyo, etc., colors. Currently you can find quite easily CMYK equivalents on the web, and after all they are intended to be used with a "physical" swatch book (and not, as often done, by picking a color from a "virtual" palette just because it looks nice on screen).

```
\definespotcolor{<latex-name>}{<PDF-name>}[<model>]{<equivalent>}
```

Internally, only CMYK is used for the equivalent color, but with this variant you can define the latter with another name space, which is then converted.

EXAMPLE The following definition is based on RGB:

```
\definespotcolor{foob}{BarTone 666 GN}[rgb]{.8, .2, .4}
```

which yields . Here `xcolor` just converts this value to CMYK, which is still the space used internally.

WARNING This conversion relies on the `xcolor` numerical methods, which do not take into account any color profile and therefore should be avoided in production if the alternate space is going to be actually used. It is provided just for convenience in drafting or in contexts where accuracy is not essential.

2 CIE LAB spot colors

Instead of targeting the CMYK space, spot colors may be defined using the CIE LAB (also known as $L^*a^*b^*$, or even just Lab) space. Unlike the default CMYK space, which is by default device dependent, the Lab space is device independent. To define a Lab spot color, you must first set the illuminant with the package option `illuminant`, which takes a value:

```
\usepackage[illuminant=d65]{colorspace}
```

Values are a, c, e, d50, d55, d65, d75.



The command to define a color is:

```
\definespotcolor{<latex-name>}{<PDF-name>}[alt=lab]{<Lab-values>}
```

Note the optional argument with the string `alt=lab` (which cannot be combined with a model). The three *<Lab-values>* have ranges $L^* = 0 \dots 100$, $a^* = -128 \dots 127$, $b^* = -128 \dots 127$, respectively.

EXAMPLE With

```
\definespotcolor{foolab}{BarTone 888 LB}[alt=lab]{50, -30, -40}
```

the result is . An example of tint is `\color{foolab!30}` (.

WARNING The package does not warn if the color falls outside the gamut of the CMYK space or if the color is not visible. To put it in other words, you should not just write some arbitrary values to see what happens.

WARNING This model is not available in `xcolor`. If you need some transformation, you must resort to an external tool like `transicc` (or `icctrans`) included in Little CMS².

WARNING There is no default illuminant – it must be set if Lab colors are defined.

NOTE The observer is currently only 2° (1931).

The Lab space allows to define colors accurately, but there is a price to pay – mixing Lab colors is not in general possible except after converting them somehow to CMYK or RGB.³ Since currently mixing colors in this package is based on CMYK, you must provide an alternate value as shown, if mixed inks are required⁴:

```
\definespotcolor{foolab}{BarTone 888 LB}[alt=lab]{50, -30, -40/1, .20, .15, .07}
```

3 Mixing spot colors

To mix spot colors you must first declare a color space (or model) including them. This is done with the following macro:

```
\definecolorspace{<latex-name>}{mixed}{<color-list>}
```

Here, *<latex-name>* is the name to be used in `\color` and the like as the color model. The second argument is the string `mixed`, and the last one is a list of up to 4 colors, either defined with `xcolor` using the CMYK model or spot.

²<http://www.littlecms.com/1/downloads.htm>

³There are no simple rules to carry out those transformations, which, I think, explains why `xcolor` does not support Lab at all

⁴Internally, and in PDF jargon, Lab is the “alternate space” and “tints transforms” are based on CMYK

EXAMPLE A simple and typical usage would be for shades:

```
\definecolorspace{fooshaded}{mixed}{foo,black}
```

Then, you can define a color in this new model named fooshaded with:

```
\definecolor{darkfoo}{fooshaded}{.6,.3}
```

Or set it with:

```
\color[fooshaded]{.6,.3}
```

In both cases, the mix is 60% foo and 30% black.

EXAMPLE If we have two spot colors named spot1 and spot2, and we want in addition yellow:

```
\definecolorspace{mix12y}{mixed}{spot1,spot2,yellow}
```

To define a new color based on this space:

```
\definecolor{mix12y}{mix12y}{.5,.4,.6}
```

And to set it:

```
\color[mix12y]{.5,.4,.6}
```

EXAMPLE Here is a duotone mixing the foob and foolab colors defined above:

```
\definecolorspace{labshaded}{mixed}{foolab,foob}
```

The colors corresponding to:

```
\color[labshaded]{1,0} \color[labshaded]{.8,.2}  
\color[labshaded]{.6,.4} \color[labshaded]{.4,.6}  
\color[labshaded]{.2,.8} \color[labshaded]{0,1}
```

are 

WARNING Due to internal limitations of xcolor, no more than four colors are allowed.

NOTE The alternate color space in the PDF file is that of the spot colors (which means currently it is CMYK).

NOTE There is an easy trick to mix colors with ! and color,num – just define an orthogonal set of colors based on the new color model:

```
\definecolor{xspot1}{mix12y}{1,0,0}  
\definecolor{xspot2}{mix12y}{0,1,0}  
\definecolor{xyellow}{mix12y}{0,0,1}
```

and then you can say:

```
\color{xspot1!30!xspot2!40!xyellow}  
\color{mix12y:xspot1,3;xspot2,2;xyellow,1}
```

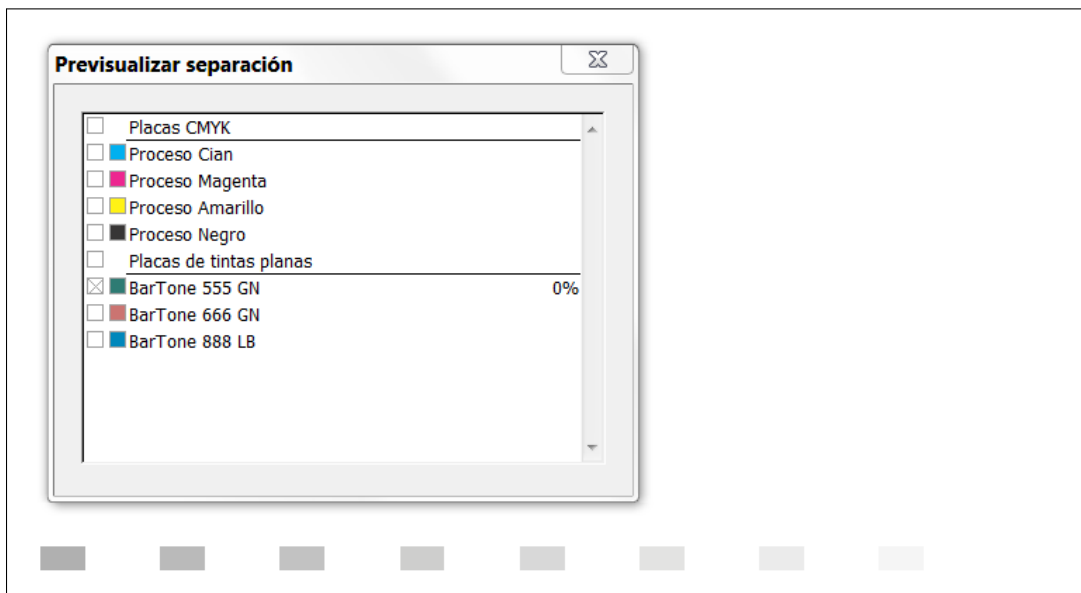


Figure 1: Plate for the foo spot color as shown by Adobe Acrobat. Note both foob and foolab, defined in this document, are listed, too.

Of course, it is just a trick and a better and direct interface is under study (none of those provided by xcolor fits well with the new mixed models).

WARNING Mixing colors in `\color`, `\definecolor` and the like from different spaces can lead to unexpected results (currently no checking is done).

Color series (see the xcolor documentation) are also partially supported. For example:

```
\definecolorseries{test}{mix12y}{grad}[mix12y]{.95,.85,.55}{3,11,17}
\definecolorseries{test}{mix12y}{last}{xyellow!50}{xspotA}
```

EXAMPLE Here is an example with some of the described techniques, based on the fooshaded space defined above (figure 1 shows the foo plate):

```
\definecolor{sfoo}{fooshaded}{1,0}
\definecolor{sblack}{fooshaded}{0,1}
\definecolorseries{shseries}{fooshaded}{last}{sfoo!40}{sblack}
\def\testclr#1{\fboxsep0pt\fbox{\colorbox{#1}{\phantom{XX}}}}
\resetcolorseries[8]{shseries}
\begin{tabular}{cccccccc}
0 \testclr{shseries!!+} & & & & & & & & 
1 \testclr{shseries!!+} & & & & & & & & 
2 \testclr{shseries!!+} & & & & & & & & 
3 \testclr{shseries!!+} & & & & & & & & 
4 \testclr{shseries!!+} & & & & & & & & 
5 \testclr{shseries!!+} & & & & & & & & 
6 \testclr{shseries!!+} & & & & & & & & 
7 \testclr{shseries!!+} & & & & & & & & 
\end{tabular}
```

0  1  2  3  4  5  6  7 

4 Page color spaces

Each PDF page must know which colors will be used (other than the predefined CMYK, RGB and Gray). By default, `colorspace` turns on for every page all newly defined colors, and that will be fine in most cases. However, you may want to set explicitly the list (for covers or plates). Use this feature with care, because (1) the asynchronous nature of \TeX (remember it affects the whole current page), and (2) each distinct color list creates a PDF resource.

```
\pagecolorspace{<color-list>}
\resetpagecolorspace
```

To change the color space for a page and the subsequent ones, you can set something like:

```
\pagecolorspace{name1,name2,name3}
```

(It can be empty.) To return to the default color space, which contains all the defined spot colors, use `\resetpagecolorspace`.

5 ICC based spaces

```
\definecolorspace*{<latex-name>}{iccbased}{<icc-file>}
```

The starred version `\definecolorspace*` does not define a new color model, but sets the behaviour of the three basic color spaces (CMYK, RGB and Gray). When belonging to the same space, the last definition for that space takes precedence and it is considered the default one. It cannot be used to define new colors or set them. Currently, only a type is supported – `iccbased`. For example,

```
\definecolorspace*{sRGB}{iccbased}{sRGB Profile.icc}
```

The space it applies to is read from the ICC profile.

The name can be used in `\pagecolorspace`. Alternatively, there are 3 reserved names: `*rgb`, `*gray`, `*cmyk`, which stand for the last-defined, default ICC based spaces. Named ICC based spaces are not set by `\resetpagecolorspace`, but the starred named are. On the other hand, the starred names are not set automatically by `\pagecolorspace`, and you must set them explicitly if you want them to be active.

NOTE Those ICC spaces do not go to the output intent dictionary (see the `pdfx` package). The latter, as the PDF reference explains, supplements rather than replaces the ICC profiles in a default color space.

EXAMPLE Given the following declarations:

```
\definecolorspace*{sRGB}{iccbased}{sRGB Profile.icc}
\definecolorspace*{colormatch}{ColorMatchRGB.icc}
```

and remembering the RGB space is always active (like the CMYK and Gray ones),

```
\pagecolorspace{}
```

leaves the RGB space unprofiled;

```
\pagecolorspace{sRGB}
```

sets the RGB space to sRGB; while the following are (in this particular example) equivalent:

```
\pagecolorspace{colormatch}  
\pagecolorspace{*rgb}  
\resetpagecolorspace
```

6 Overprinting

This is usually a pre-print task, but by setting it in the document you will get a better idea of how the colors are actually overlapped in soft proofing. However, remember the effect produced is device-dependent, and colorant overprint decisions should be made at output time (according to the PDF reference).

Very often, it is set for the whole document with the package options `knockout` (no overprint), and `overprint`. By default, the overprint mode is 1, but it can be changed with `opm=0`.

Once set the overprint state for the whole document, you can use something like:

```
{\overprintstate{1}text}  
\textoverprint[1]{text}
```

(or 0, or no; default in `\textoverprint` is 1, except with the package option `opm=0`).

Since the color stack is used, `pdfTEX` ≥ 1.40 is required.

WARNING Some PDF viewers ignore this setting.

7 Version

1.1.1. No new features. Just internal changes related to `luatex` and new manual.

1.2.0. CIE LAB spot colors and illuminant. Manual rewritten.

1.3. Fixes several incompatibilities with `pgf/tikz`.