

The pgf-cmykshadings package

David Purton*

2018/10/17 v1.0

Abstract

The `pgf-cmykshadings` package provides support for CMYK and grayscale shadings for the `pgf` package. By default `pgf` only supports RGB shadings. Horizontal, vertical, radial, and functional shadings are supported for `pdfLATEX`, `LuaLATEX`, and `XYLATEX`. `pgf-cmykshadings` attempts to produce shadings consistent with the currently selected `xcolor` colour model. The `rgb`, `cmyk`, and `gray` colour models from the `xcolor` package are supported.

Contents

1	Introduction	2
2	Acknowledgements	2
3	Bug Reports and Feature Requests	2
4	Documentation	2
4.1	Basic Usage	2
4.1.1	Package options	3
4.1.2	Load order	3
4.1.3	Colour models	3
4.1.4	Functional shadings	4
4.2	Main Interface	4
4.2.1	Declaring shadings	4
4.2.2	Using shadings	5
4.2.3	Utility functions	5
5	Implementation	7
5.1	Main Package	7
5.2	Drivers	16
5.2.1	pdftex driver	16
5.2.2	xetex driver	19
5.2.3	luatex driver	19
5.2.4	dvipdfmx driver	21
	Change History	24

*Email: dcpurton@marshwiggles.net

1 Introduction

The `pgf` package, and other packages built on top of it, only support RGB shadings (colour gradients). This means that printing applications requiring CMYK shadings can not easily be produced. It also can lead to unexpected colour mismatches in documents when attempting to define a shading from colours defined in CMYK. This can occur when the `natural` colour model of the `xcolor` package is in use and colours like cyan and magenta are defined as CMYK. An attempt to produce a shading using these colours will be silently converted to RGB, but RGB cyan and RGB magenta look significantly different from CMYK cyan and magenta. This is a significant cause of confusion for end users.

The following example illustrates this problem and the corresponding solution provided by the `pgf-cmykshadings` package.

```
\begin{tikzpicture}
  \fill[cyan] (0,0) rectangle (1,1);
  \shade[left color=cyan, right color=magenta]
    (1.25,0) rectangle (3.75,1);
  \fill[magenta] (4,0) rectangle (5,1);
\end{tikzpicture}
```

`pgf` behaviour:



`pgf-cmykshadings` behaviour:



2 Acknowledgements

Substantial parts of the code for the `pgf-cmykshadings` package are taken from the `pgf` package files `pgfcoresshade.code.tex`, `pgfsys-pdf.tex.def`, `pgfsys-dvipdfmx.def`, and `pgfsys-luatex.def` copyright © 2006 Till Tantau and then slightly modified to support CMYK and grayscale shadings.

3 Bug Reports and Feature Requests

Bug reports and feature requests can be made at the `pgf-cmykshadings` package GitHub repository. See <https://github.com/dcpurton/pgf-cmykshadings>.

4 Documentation

4.1 Basic Usage

All that is required to use CMYK shadings instead of RGB shadings is to include the package in your document preamble:

```
\usepackage{pgf-cmykshadings}
```

However, there are some caveats in using the package, which are outlined below.

4.1.1 Package options

`pgd-cmykshadings` supports the following package options:

`cmyk` (default) to use CMYK shadings when the `xcolor` package `natural` colour model is in use.

`rgb` to use the RGB shadings when the `xcolor` package `natural` colour model is in use.

4.1.2 Load order

`pgf-cmykshadings` should be loaded *before* any shadings are defined otherwise these will be defined as RGB. This means you should load `pgf-cmykshadings` before (for example) `tikz` and `beamer`.

If you want to pass custom options to `xcolor` (e.g., a colour model or set of named colours), you should load `pgf-cmykshadings` *after* `xcolor` or use `\PassOptionsToPackage` *before* loading `pgf-cmykshadings`.

4.1.3 Colour models

`pgf-cmykshadings` attempts to produce shadings consistent with the currently selected `xcolor` package colour model. The `rgb`, `cmyk`, and `gray` colour models from the `xcolor` package are supported.

Note: The colour model chosen for a shading is based on the `xcolor` colour model *at the time the shading is created*. This is either when `\pgfdeclare*shading` is called with no optional argument or when `\pgfusesshading` is called if `\pgfdeclare*shading` was called with an optional argument.

If the `xcolor` package `natural` colour model is in use then the shading colour model will be CMYK by default (equivalent to passing the `cmyk` option to the `pgf-cmykshadings` package). RGB shadings can be output by default instead by passing the `rgb` option to the `pgf-cmykshadings` package.

In practice this means that if you are using the `natural` colour model of the `xcolor` package you can still get mismatched colours if you, for example, create a shading from green (which is defined as RGB) to magenta (which is defined as CMYK). The shading has to pick one colour model and will look different to one of the solid colours.

For this reason it is recommended to always load the `xcolor` package before the `pgf-cmykshadings` package with either the `rgb`, `cmyk`, or `gray` options to avoid colour surprises.

```
\begin{tikzpicture}
  \fill[green] (0,0) rectangle (1,1);
  \shade[left color=green, right color=magenta]
    (1.25,0) rectangle (3.75,1);
  \fill[magenta] (4,0) rectangle (5,1);
\end{tikzpicture}
```

`xcolor natural` colour model:



`xcolor cmyk` colour model:



`xcolor rgb` colour model:



`xcolor gray` colour model:



4.1.4 Functional shadings

By nature, the PostScript® code used to generate functional shadings must output either RGB or CMYK data. For this reason, `\pgfdeclarefunctionalshading` is *not* portable across colour models.

Take particular care that the same colour model is in use at declaration time and use time for functional shadings declared with an optional argument as otherwise the PostScript® data will not match the declared colour space and you will end up with a malformed PDF.

This also means that you should *not* use the functional shadings from the `tikz` shading library (`bilinear interpolation`, `color wheel`, `color wheel black center`, `color wheel white center`, and `Mandelbrot set`) except when the `xcolor rgb` colour model is in use, otherwise you will end up with a malformed PDF.

Having said this, it *is* possible to create portable functional shadings by providing conditional code to append colour transformations to the PostScript® data. A variety of `\pgffuncshading*to*` (e.g., `\pgffuncshadingrgbtocmyk`) macros along with `\ifpgfshadingmodel*` (e.g., `\ifpgfshadingmodelcmyk`) conditionals are provided to assist with these transformations.

```
\pgfdeclarefunctionalshading[black]{twospots}
  {\pgfpointorigin}{\pgfpoint{3.5cm}{3.5cm}}{}{
    2 copy
    45 sub dup mul exch
    40 sub dup mul 0.5 mul add sqrt
    dup mul neg 1.0005 exch exp 1.0 exch sub
    3 1 roll
    70 sub dup mul .5 mul exch
    70 sub dup mul add sqrt
    dup mul neg 1.002 exch exp 1.0 exch sub
    1.0 3 1 roll
    \ifpgfshadingmodelcmyk
      \pgffuncshadingrgbtocmyk
    \fi
    \ifpgfshadingmodelgray
      \pgffuncshadingrgbtogray
    \fi
  }
```

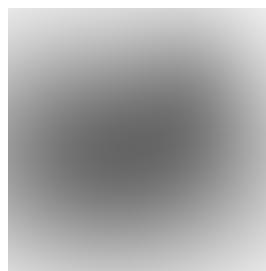
xcolor cmyk model:



xcolor rgb model:



xcolor gray model:



4.2 Main Interface

4.2.1 Declaring shadings

The four standard `pgf` functions for declaring shadings are supported as documented in the `pgf` manual.

There is one extension provided by the `pgf-cmykshadings` package. It is possible to specify CMYK colours directly in the colour specification argument using a syntax analogous to the RGB, Gray, and named colours already supported by the `pgf` package. i.e., `cmk(position)=(C),(M),(Y),(K)`.

Shadings declared *without* an optional argument are created immediately in the currently active `xcolor` colour model. Shadings declared *with* an optional argument are created at the time they are actually used (using `\pgfuseshading`).

<code>\pgfdeclarehorizontalshading</code>	<code>\pgfdeclarehorizontalshading[<i>color list</i>]{<i>shading name</i>}{<i>shading height</i>}{<i>color specification</i>}</code>
---	--

Declare a horizontal shading.

<code>\pgfdeclareverticalshading</code>	<code>\pgfdeclareverticalshading[<i>color list</i>]{<i>shading name</i>}{<i>shading width</i>}{<i>color specification</i>}</code>
---	---

Declare a vertical shading.

<code>\pgfdeclareradialshading</code>	<code>\pgfdeclareradialshading[<i>color list</i>]{<i>shading name</i>}{<i>center point</i>}{<i>color specification</i>}</code>
---------------------------------------	--

Declare a radial shading.

<code>\pgfdeclarefunctionalshading</code>	<code>\pgfdeclarefunctionalshading[<i>color list</i>]{<i>shading name</i>}{<i>lower left corner</i>}{<i>upper right corner</i>}{<i>init code</i>}{<i>type 4 function</i>}</code>
---	--

Declare a functional shading.

4.2.2 Using shadings

Shadings are used as documented in the `pgf` manual.

<code>\pgfuseshading</code>	<code>\pgfuseshading{<i>shading name</i>}</code>
-----------------------------	--

Use a previously declared shading. If the specified shading was declared with an optional argument, then the shading will be created at this point in the currently active `xcolor` colour space.

<code>\pgfshadepath</code>	<code>\pgfshadepath{<i>shading name</i>}{<i>angle</i>}</code>
----------------------------	---

Shade the currently active `pgf` path using the specified shading at the specified angle.

<code>\pgfadditionalshadetransform</code>	<code>\pgfadditionalshadetransform{<i>transformation</i>}</code>
---	--

This command is used to specify an additional transformation that should be applied to shadings when `\pgfshadepath` used.

4.2.3 Utility functions

The following functions are mainly useful for in declaring functional shadings.

<hr/> <hr/> <code>\pgfshadecolorortorgb</code>	<code>\pgfshadecolorortorgb{<color name>}{<macro>}</code> This command takes <i><color name></i> as input and stores the colour's red/green/blue components as real numbers between 0.0 and 1.0 separated by spaces (which is exactly what you need if you want to push it on a stack) in <i><macro></i> . This macro can then be used inside the <i><type 4 function></i> argument for <code>\pgfdeclarefunctionalshading</code> . In addition, three macros suffixed with red , green and blue are defined, which store the individual components of <i><color name></i> . These can also be used in the <i><type 4 function></i> argument.
<hr/> <hr/> <code>\pgfshadecolorortocmyk</code>	<code>\pgfshadecolorortocmyk{<color name>}{<macro>}</code> This command is analogous to <code>\pgfshadecolorortorgb</code> , but stores the colour's cyan/magenta/yellow/black components. Four macros suffixed with cyan , magenta , yellow , and black are also defined.
<hr/> <hr/> <code>\pgfshadecolorotogray</code>	<code>\pgfshadecolorotogray{<color name>}{<macro>}</code> This command is analogous to <code>\pgfshadecolorortorgb</code> , but stores the colour's gray component. Although it's not needed, for consistency a second macro suffixed with gray is also defined.
<hr/> <hr/> <code>\pgffuncshadingrgbtocmyk</code>	<code>\pgffuncshadingrgbtocmyk</code> Within the <i><type 4 function></i> argument of <code>\pgfdeclarefunctionalshading</code> , this command can be used to convert the top 3 elements on the stack from RGB to CMYK. In combination with the <code>\ifpgfshadingmodelcmyk</code> conditional this macro can be used to make functional shading declarations more portable across colour models.
<hr/> <hr/> <code>\pgffuncshadingrgbtogray</code>	<code>\pgffuncshadingrgbtogray</code> Within the <i><type 4 function></i> argument of <code>\pgfdeclarefunctionalshading</code> , this command can be used to convert the top 3 elements on the stack from RGB to grayscale. In combination with the <code>\ifpgfshadingmodelgray</code> conditional this macro can be used to make functional shading declarations more portable across colour models.
<hr/> <hr/> <code>\pgffuncshadingcmyktorgb</code>	<code>\pgffuncshadingcmyktorgb</code> Within the <i><type 4 function></i> argument of <code>\pgfdeclarefunctionalshading</code> , this command can be used to convert the top 4 elements on the stack from CMYK to RGB. In combination with the <code>\ifpgfshadingmodelrgb</code> conditional this macro can be used to make functional shading declarations more portable across colour models.
<hr/> <hr/> <code>\pgffuncshadingcmyktogray</code>	<code>\pgffuncshadingcmyktogray</code> Within the <i><type 4 function></i> argument of <code>\pgfdeclarefunctionalshading</code> , this command can be used to convert the top 4 elements on the stack from CMYK to grayscale. In combination with the <code>\ifpgfshadingmodelgray</code> conditional this macro can be used to make functional shading declarations more portable across colour models.
<hr/> <hr/> <code>\pgffuncshadinggraytorgb</code>	<code>\pgffuncshadinggraytorgb</code> Within the <i><type 4 function></i> argument of <code>\pgfdeclarefunctionalshading</code> , this command can be used to convert the top element on the stack from grayscale to RGB. In combination with the <code>\ifpgfshadingmodelrgb</code> conditional this macro can be used to make functional shading declarations more portable across colour models.

\pgffuncshadinggraytocmyk**\pgffuncshadinggraytocmyk**

Within the *<type 4 function>* argument of `\pgfdeclarefunctionalshading`, this command can be used to convert the top element on the stack from grayscale to CMYK. In combination with the `\ifpgfshadingmodelcmyk` conditional this macro can be used to make functional shading declarations more portable across colour models.

\ifpgfshadingmodelrgb**\ifpgfshadingmodelrgb**

Within the *<type 4 function>* argument of `\pgfdeclarefunctionalshading`, this command can be used to test if the `xcolor` colour model is **rgb** *at the time the shading is created*. This can be used to ensure that the data output in the *<type 4 function>* correctly matches the active colour model.

\ifpgfshadingmodelcmyk**\ifpgfshadingmodelcmyk**

Within the *<type 4 function>* argument of `\pgfdeclarefunctionalshading`, this command can be used to test if the `xcolor` colour model is **cmyk** *at the time the shading is created*. This can be used to ensure that the data output in the *<type 4 function>* correctly matches the active colour model.

\ifpgfshadingmodelgray**\ifpgfshadingmodelgray**

Within the *<type 4 function>* argument of `\pgfdeclarefunctionalshading`, this command can be used to test if the `xcolor` colour model is **gray** *at the time the shading is created*. This can be used to ensure that the data output in the *<type 4 function>* correctly matches the active colour model.

5 Implementation

5.1 Main Package

```
1 <*package>
2 \ProvidesPackage{pgf-cmykshadings}%
3   [2018/10/17 CMYK shadings support for PGF (DCP)]
4 \RequirePackage{pgf}
```

Replace dependence on `\pgf@converttrgbstring` and `\pgf@rgbconv` with generic macros `\pgf@convertstring` and `\pgf@conv`.

```
5 \def\pgf@parsefunc#1{%
6   \edef\temp{{#1}}%
7   \expandafter\pgf@convertstring\temp%
8   \edef\temp{{\pgf@conv}}%
9   \expandafter\pgf@@parsefunc\temp}
```

Replace RGB parsing macros with new macros selected on the basis of the current colour space (`\pgf@shading@mode`).

```
10 \def\pgf@@parsefunc#1{%
11   \let\pgf@bounds=\pgfutil@empty%
12   \let\pgf@funcs=\pgfutil@empty%
13   \let\pgf@psfuncs=\pgfutil@empty%
14   \let\pgf@encode=\pgfutil@empty%
15   \let\pgf@sys@shading@ranges=\pgfutil@empty%
16   \pgf@sys@shading@range@num=0\relax%
```

```

17 \csname pgf@parsefirst\pgf@shading@model\endcsname[#1; ]%
18 \csname pgf@parselastdom\pgf@shading@model\endcsname[#1; ]%
19 \csname pgf@parsemid\pgf@shading@model\endcsname[#1; ]%
20 \ifx\pgf@bounds\pgfutil@empty%
21 \edef\pgf@pdfparsedomain{0 1}%
22 \edef\pgf@pdfparsedfunction{\pgf@singlefunc\space}%
23 \else%
24 \edef\pgf@pdfparsedomain{\pgf@doma\space\pgf@domb}%
25 \edef\pgf@pdfparsedfunction{%
26 << /FunctionType 3 /Domain [\pgf@doma\space\pgf@domb] /Functions
27 [\pgf@funcs\space] /Bounds [\pgf@bounds] /Encode [0 1 \pgf@encode]
28 >> }% <<
29 \fi%
30 \xdef\pgf@psfuncs{\pgf@psfuncs}%
31 }

```

Define RGB parsing macros.

```

32 \let\pgf@parsefirstrgb\pgf@parsefirst
33 \let\pgf@parselastdomrgb\pgf@parselastdom
34 \let\pgf@parsemidrgb\pgf@parsemid
35 \let\pgf@parserestrgb\pgf@parserest

```

Define new CMYK parsing macros.

```

36 \def\pgf@parsefirstcmyk[cmyk(#1)=(#2,#3,#4,#5)#6]{%
37 \pgfmathsetlength\pgf@x{#1}%
38 \edef\pgf@sys@shading@start@pos{\the\pgf@x}%
39 \pgf@sys@bp@correct\pgf@x%
40 \edef\pgf@doma{\pgf@sys@tonumber{\pgf@x}}%
41 \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
42 \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
43 \edef\pgf@sys@shading@start@cmyk{\pgf@sys@cmyk}%
44 \let\pgf@sys@prevcolor=\pgf@sys@shading@start@cmyk%
45 \let\pgf@sys@prevpos=\pgf@sys@shading@start@pos%
46 \edef\pgf@prevcolor{\pgf@cmyk}%
47 \edef\pgf@firstcolor{\pgf@cmyk}}
48 \def\pgf@parselastdomcmyk[cmyk(#1)=(#2,#3,#4,#5); {%
49 \pgfutil@ifnextchar]{%
50 \pgfmathsetlength\pgf@x{#1}%
51 \edef\pgf@sys@shading@end@pos{\the\pgf@x}%
52 \pgf@max=\pgf@x\relax%
53 \pgf@sys@bp@correct\pgf@x%
54 \edef\pgf@domb{\pgf@sys@tonumber{\pgf@x}}%
55 \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
56 \edef\pgf@sys@shading@end@cmyk{\pgf@sys@cmyk}%
57 \pgfutil@gobble}{\pgf@parselastdomcmyk[]}}
58 \def\pgf@parsemidcmyk[cmyk(#1)=(#2,#3,#4,#5); {\pgf@parserestcmyk[]
59 \def\pgf@parserestcmyk[cmyk(#1)=(#2,#3,#4,#5); {%
60 \advance\pgf@sys@shading@range@num by1\relax%
61 \pgfutil@ifnextchar]{%
62 \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
63 \edef\pgf@singlefunc{\space%
64 << /FunctionType 2 /Domain [0 1] /C0
65 [\pgf@prevcolor] /C1 [\pgf@cmyk] /N 1 >> }% <<
66 \edef\pgf@funcs{\pgf@funcs\space%
67 << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /C0

```



```

68     [\pgf@prevcolor] /C1 [\pgf@cmk] /N 1 >> }% <<
69 \edef\pgf@psfuncs{\pgf@prevx\space
70     \pgf@cmk\space \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
71 \pgfmathsetlength\pgf@x{#1}%
72 \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
73     {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@cmk}}}%
74 \edef\pgf@sys@prevpos{\the\pgf@x}%
75 \let\pgf@sys@prevcolor=\pgf@sys@cmk%
76 \pgfutil@gobble}{%
77 \pgfmathsetlength\pgf@x{#1}%
78 \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
79 \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
80     {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@cmk}}}%
81 \edef\pgf@sys@prevpos{\the\pgf@x}%
82 \let\pgf@sys@prevcolor=\pgf@sys@cmk%
83 \edef\pgf@psfuncs{\pgf@prevx\space \pgf@cmk\space
84     \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
85 \pgf@sys@bp@correct\pgf@x%
86 \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
87 \edef\pgf@bounds{\pgf@bounds\space\pgf@sys@tonumber{\pgf@x}}%
88 \edef\pgf@encode{\pgf@encode\space0 1}%
89 \edef\pgf@singlefunc{\space%
90     << /FunctionType 2 /Domain [0 1] /CO
91     [\pgf@prevcolor] /C1 [\pgf@cmk] /N 1 >> }% <<
92 \edef\pgf@funcs{\pgf@funcs\space%
93     << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /CO
94     [\pgf@prevcolor] /C1 [\pgf@cmk] /N 1 >> }% <<
95 \edef\pgf@prevcolor{\pgf@cmk}%
96 \pgf@parserestcmyk[]}%
97 \def\pgf@getcmyktuplewithmixin#1#2#3#4{%
98     \pgfutil@definecolor{pgfshadetemp}{cmyk}{#1,#2,#3,#4}%
99     \pgfutil@ifundefined{applycolormixins}{\pgfshadetemp}%
100 \pgfutil@extractcolorspec{pgfshadetemp}{\pgf@tempcolor}%
101 \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{cmyk}{\pgf@cmkcolor}%
102 \expandafter\pgf@getcmyk@@\pgf@cmkcolor!}%
103 \def\pgf@getcmyk@@#1,#2,#3,#4!{%
104     \def\pgf@cmk{#1 #2 #3 #4}%
105     \def\pgf@sys@cmk{{#1}{#2}{#3}{#4}}%
106 }

```

Define new grayscale parsing macros.

```

107 \def\pgf@parsefirstgray[gray(#1)=(#2)#3]{%
108     \pgfmathsetlength\pgf@x{#1}%
109     \edef\pgf@sys@shading@start@pos{\the\pgf@x}%
110     \pgf@sys@bp@correct\pgf@x%
111     \edef\pgf@doma{\pgf@sys@tonumber{\pgf@x}}%
112     \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
113     \pgf@getgraytuplewithmixin{#2}%
114     \edef\pgf@sys@shading@start@gray{\pgf@sys@gray}%
115     \let\pgf@sys@prevcolor=\pgf@sys@shading@start@gray%
116     \let\pgf@sys@prevpos=\pgf@sys@shading@start@pos%
117     \edef\pgf@prevcolor{\pgf@gray}%
118     \edef\pgf@firstcolor{\pgf@gray}}
119 \def\pgf@parselastdomgray[gray(#1)=(#2); {%
120     \pgfutil@ifnextchar]{%

```

```

121 \pgfmathsetlength\pgf@x{#1}%
122 \edef\pgf@sys@shading@end@pos{\the\pgf@x}%
123 \pgf@max=\pgf@x\relax%
124 \pgf@sys@bp@correct\pgf@x%
125 \edef\pgf@domb{\pgf@sys@tonumber{\pgf@x}}%
126 \pgf@getgraytuplewithmixin{#2}%
127 \edef\pgf@sys@shading@end@gray{\pgf@sys@gray}%
128 \pgfutil@gobble{\pgf@parserestgray{}}
129 \def\pgf@parsemidgray[gray(#1)=(#2); {\pgf@parserestgray{}
130 \def\pgf@parserestgray[gray(#1)=(#2); {%
131 \advance\pgf@sys@shading@range@num by1\relax%
132 \pgfutil@ifnextchar]{%
133 \pgf@getgraytuplewithmixin{#2}%
134 \edef\pgf@singlefunc{\space%
135 << /FunctionType 2 /Domain [0 1] /CO
136 [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
137 \edef\pgf@funcs{\pgf@funcs\space%
138 << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /CO
139 [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
140 \edef\pgf@psfuncs{\pgf@prevx\space \pgf@gray\space
141 \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
142 \pgfmathsetlength\pgf@x{#1}%
143 \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
144 {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@gray}}}%
145 \edef\pgf@sys@prevpos{\the\pgf@x}%
146 \let\pgf@sys@prevcolor=\pgf@sys@gray%
147 \pgfutil@gobble{%
148 \pgfmathsetlength\pgf@x{#1}%
149 \pgf@getgraytuplewithmixin{#2}%
150 \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
151 {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@gray}}}%
152 \edef\pgf@sys@prevpos{\the\pgf@x}%
153 \let\pgf@sys@prevcolor=\pgf@sys@gray%
154 \edef\pgf@psfuncs{\pgf@prevx\space \pgf@gray\space
155 \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
156 \pgf@sys@bp@correct\pgf@x%
157 \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
158 \edef\pgf@bounds{\pgf@bounds\space\pgf@sys@tonumber{\pgf@x}}%
159 \edef\pgf@encode{\pgf@encode\space0 1}%
160 \edef\pgf@singlefunc{\space%
161 << /FunctionType 2 /Domain [0 1] /CO
162 [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
163 \edef\pgf@funcs{\pgf@funcs\space%
164 << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /CO
165 [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
166 \edef\pgf@prevcolor{\pgf@gray}%
167 \pgf@parserestgray{}}
168 \def\pgf@getgraytuplewithmixin#1{%
169 \pgfutil@definecolor{pgfshadetemp}{gray}{#1}%
170 \pgfutil@ifundefined{applycolormixins}{\applycolormixins{pgfshadetemp}}%
171 \pgfutil@extractcolorspec{pgfshadetemp}{\pgf@tempcolor}%
172 \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{gray}{\pgf@graycolor}%
173 \expandafter\pgf@getgray@@\pgf@graycolor!}
174 \def\pgf@getgray@@#1!{%

```

```

175 \def\pgf@gray{#1}%
176 \def\pgf@sys@gray{#{#1}}%
177 }

```

Define new colour space agnostic colour specification parsing macros. This includes parsing CMYK colour specifications (i.e., `color(<position>)=(<C>,<M>,<Y>,<K>)`).

```

178 \def\pgf@convertstring#1{%
179   \def\pgf@conv{%
180     \pgf@convert#1}%
181   }
182 \def\pgf@convert{%
183   \pgfutil@ifnextchar]{\pgfutil@gobble}%done!
184   {%
185     \pgfutil@ifnextchar;{\pgf@grabsemicolor}%
186     {%
187       \pgfutil@ifnextchar c{\pgf@gobblec}%
188       {%
189         \pgfutil@ifnextchar g{\pgf@grabgray}%
190         {%
191           \pgfutil@ifnextchar o{\pgf@grabcolor}%
192           {%
193             \pgfutil@ifnextchar m{\pgf@grabcmk}%
194             {%
195               \pgfutil@ifnextchar r{\pgf@grabrgb}%
196               {\pgferror{Illformed shading
197                 specification}\pgf@convert}%
198             }%
199           }%
200         }%
201       }%
202     }%
203   }%
204 }
205 \def\pgf@grabsemicolor;{%
206   \edef\pgf@conv{\pgf@conv; }\pgf@convert}
207 \def\pgf@gobblec c{\pgf@convert}
208 \def\pgf@savecolor#1{
209   \pgfutil@extractcolorspec{\pgf@tempcol}{\pgf@tempcolor}%
210   \expandafter\pgfutil@convertcolorspec\pgf@tempcolor%
211   {\pgf@shading@model}{\pgf@color}%
212   \expandafter\pgf@convget@\expandafter{\pgf@color}{#1}%
213 }
214 \def\pgf@grabrgb rgb(#1)=(#2,#3,#4){%
215   \pgfutil@definecolor{\pgf@tempcol}{rgb}{#2,#3,#4}%
216   \pgf@savecolor{#1}%
217 }
218 \def\pgf@grabcmk myk(#1)=(#2,#3,#4,#5){%
219   \pgfutil@definecolor{\pgf@tempcol}{cmk}{#2,#3,#4,#5}%
220   \pgf@savecolor{#1}%
221 }
222 \def\pgf@grabgray gray(#1)=(#2){%
223   \pgfutil@definecolor{\pgf@tempcol}{gray}{#2}%
224   \pgf@savecolor{#1}%
225 }
226 \def\pgf@grabcolor olor(#1)=(#2){%

```

```

227 \pgfutil@colorlet{pgf@tempcol}{#2}%
228 \pgf@savecolor{#1}%
229 }
230 \def\pgf@convget@#1#2{%
231 \edef\pgf@conv{\pgf@conv \pgf@shading@model(#2)=(#1)}\pgf@convert}

```

New macros to convert CMYK colours to a format suitable for use in the *<type 4 function>* argument of `\pgfdeclarefunctionalshading`.

```

232 \newdimen\pgf@xd
233 \def\pgfshadecolorfromcmymk#1#2{%
234 \pgfutil@colorlet{pgf@tempcol}{#1}%
235 \pgfutil@extractcolorspec{pgf@tempcol}{\pgf@tempcolor}%
236 \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{cmymk}{\pgf@cmymkcolor}%
237 \expandafter\pgfshading@cmymk\pgf@cmymkcolor\relax%
238 \edef#2{\pgf@sys@tonumber{\pgf@xa}\space\pgf@sys@tonumber{\pgf@xb}\space
239 \pgf@sys@tonumber{\pgf@xc}\space\pgf@sys@tonumber{\pgf@xd}\space}%
240 \c@pgf@counta\escapechar%
241 \escapechar-1\relax%
242 \expandafter\edef\csname\string#2cyan\endcsname{%
243 \pgf@sys@tonumber{\pgf@xa}\space}%
244 \expandafter\edef\csname\string#2magenta\endcsname{%
245 \pgf@sys@tonumber{\pgf@xb}\space}%
246 \expandafter\edef\csname\string#2yellow\endcsname{%
247 \pgf@sys@tonumber{\pgf@xc}\space}%
248 \expandafter\edef\csname\string#2black\endcsname{%
249 \pgf@sys@tonumber{\pgf@xd}\space}%
250 \escapechar\c@pgf@counta%
251 }
252 \def\pgfshading@cmymk#1,#2,#3,#4\relax{%
253 \pgf@xa=#1pt%
254 \pgf@xb=#2pt%
255 \pgf@xc=#3pt%
256 \pgf@xd=#4pt%
257 }

```

New macros to convert grayscale colours to a format suitable for use in the *<type 4 function>* argument of `\pgfdeclarefunctionalshading`.

```

258 \def\pgfshadecolorfromgray#1#2{%
259 \pgfutil@colorlet{pgf@tempcol}{#1}%
260 \pgfutil@extractcolorspec{pgf@tempcol}{\pgf@tempcolor}%
261 \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{gray}{\pgf@graycolor}%
262 \expandafter\pgfshading@gray\pgf@graycolor\relax%
263 \edef#2{\pgf@sys@tonumber{\pgf@xa}\space}%
264 \c@pgf@counta\escapechar%
265 \escapechar-1\relax%
266 \expandafter\edef\csname\string#2gray\endcsname{%
267 \pgf@sys@tonumber{\pgf@xa}\space}%
268 \escapechar\c@pgf@counta%
269 }
270 \def\pgfshading@gray#1\relax{%
271 \pgf@xa=#1pt%
272 }

```

Ensure colour model is set up based on the current xcolor colour model when declaring shadings.

```

273 \def\pgfdeclarehorizontalshading{%
274   \pgf@setup@model
275   \pgfutil@ifnextchar[%
276     \pgf@declarehorizontalshading{\pgf@declarehorizontalshading[]}}
277 \def\pgfdeclareverticalshading{%
278   \pgf@setup@model
279   \pgfutil@ifnextchar[%
280     \pgf@declareverticalshading{\pgf@declareverticalshading[]}}
281 \def\pgfdeclareradialshading{%
282   \pgf@setup@model
283   \pgfutil@ifnextchar[%
284     \pgf@declareradialshading{\pgf@declareradialshading[]}}
285 \def\pgfdeclarefunctionalshading{%
286   \pgf@setup@model
287   \pgfutil@ifnextchar[
288     \pgf@declarefunctionalshading{\pgf@declarefunctionalshading[]}}

```

Ensure colour model is set up based on the current xcolor colour model when using shadings.

```

289 \def\pgfuseshading#1{%
290   \edef\pgf@shadingname{@pgfshading#1}%
291   \edef\pgf@shadingsavedmodel{@pgfshading#1@model}%
292   \pgf@tryextensions{\pgf@shadingname}{\pgf@alternateextension}%
293   \expandafter\pgfutil@ifundefined\expandafter{\pgf@shadingname}%
294   {\pgferror{Undefined shading "#1"}}}%
295   {%
296     {%
297       \pgf@setup@model
298       \pgfutil@globalcolorsfalse%
299       \def\pgf@shade@adds{}%
300       \pgfutil@ifundefined{pgf@deps\pgf@shadingname}%
301       {}%
302       {%
303         \edef\@list{\csname pgf@deps\pgf@shadingname\endcsname}%
304         \pgfutil@for\@temp:=\@list\do{%
305           {%
306             \pgfutil@ifundefined{applycolormixins}%
307             {}{\applycolormixins{\@temp}}}%
308             \pgfutil@extractcolorspec{\@temp}{\pgf@tempcolor}%
309             \expandafter\pgfutil@ifundefined\expandafter{%
310               \pgf@shadingsavedmodel}
311               {\expandafter\pgfutil@convertcolorspec\pgf@tempcolor{%
312                 \pgf@shading@model}{\pgf@color}}}%
313                 {\expandafter\pgfutil@convertcolorspec\pgf@tempcolor{%
314                   \pgf@shadingsavedmodel}{\pgf@color}}}%
315                   \xdef\pgf@shade@adds{\pgf@shade@adds,\pgf@color}%
316                 }%
317               }%
318             }%
319             \expandafter\pgf@strip@shadename\pgf@shadingname!!%
320             \pgfutil@ifundefined{@pgfshading\pgf@basename\pgf@shade@adds!}%
321             {%
322               {%
323                 \expandafter\def\expandafter\@temp\expandafter{%
324                   \csname pgf@func\pgf@shadingname\endcsname}%

```

```

325         \edef\@args{{\pgf@basename\pgf@shade@adds}}%
326         \expandafter\expandafter\expandafter\def%
327         \expandafter\expandafter\expandafter\@args%
328         \expandafter\expandafter\expandafter{%
329             \csname pgf@args\pgf@shadingname\endcsname}%
330         \expandafter\expandafter\expandafter\@temp\expandafter\@args\@args%
331         %
332     }%
333 }%
334 {}%
335 \pgf@invoke shading{%
336     \csname @pgfshading\pgf@basename\pgf@shade@adds!\endcsname}%
337 }%
338 }%
339 }

```

Conditionals for use in the *<type 4 function>* argument of `\pgfdeclarefunctionalshading` to test for the currently active xcolor colour model.

```

340 \newif\ifpgfshadingmodelrgb
341 \newif\ifpgfshadingmodelcmyk
342 \newif\ifpgfshadingmodelgray

```

Shading colour space property set up based on the currently active xcolor colour model.

```

343 \def\pgf@setup@model{%
344     \pgfshadingmodelrgbtrue
345     \pgfshadingmodelcmykfalse
346     \pgfshadingmodelgrayfalse
347     \XC@edef\pgf@mod@test{\XC@tgt@mod{natural}}%
348     \def\pgf@shading@functional@range{0 1 0 1 0 1}%
349     \def\pgf@shading@device{/DeviceRGB}%
350     \def\pgf@shading@model{rgb}%
351     \ifx\pgf@mod@test\XC@mod@natural
352         \ifpgfcmykshadingdefault
353             \def\pgf@shading@functional@range{0 1 0 1 0 1}%
354             \def\pgf@shading@device{/DeviceCMYK}%
355             \def\pgf@shading@model{cmyk}%
356             \pgfshadingmodelrgbfalse
357             \pgfshadingmodelcmyktrue
358         \else
359             \def\pgf@shading@functional@range{0 1 0 1 0 1}%
360             \def\pgf@shading@device{/DeviceRGB}%
361             \def\pgf@shading@model{rgb}%
362         \fi
363     \fi
364     \ifx\pgf@mod@test\XC@mod@cmyk
365         \def\pgf@shading@functional@range{0 1 0 1 0 1}%
366         \def\pgf@shading@device{/DeviceCMYK}%
367         \def\pgf@shading@model{cmyk}%
368         \pgfshadingmodelrgbfalse
369         \pgfshadingmodelcmyktrue
370     \fi
371     \ifx\pgf@mod@test\XC@mod@gray
372         \def\pgf@shading@functional@range{0 1}%
373         \def\pgf@shading@device{/DeviceGray}%
374         \def\pgf@shading@model{gray}%

```

```

375     \pgfshadingmodelrgbfalse
376     \pgfshadingmodelgraytrue
377 \fi
378 }

```

Converters for use in the *<type 4 function>* argument of `\pgfdeclarefunctionalshading`. These macros use the same algorithms as `xcolor`.

```

379 \def\pgffuncshadingrgbtocmyk{%
380     1.0 exch sub 3 1 roll
381     1.0 exch sub 3 1 roll
382     1.0 exch sub 3 1 roll
383     3 copy
384     2 copy gt { exch } if pop
385     2 copy gt { exch } if pop
386     dup 3 1 roll sub
387     0.0 2 copy lt { exch } if pop
388     1.0 2 copy gt { exch } if pop
389     4 1 roll
390     dup 3 1 roll sub
391     0.0 2 copy lt { exch } if pop
392     1.0 2 copy gt { exch } if pop
393     4 1 roll
394     dup 3 1 roll sub
395     0.0 2 copy lt { exch } if pop
396     1.0 2 copy gt { exch } if pop
397     4 1 roll
398 }
399 \def\pgffuncshadingrgbtogray{%
400     0.11 mul exch 0.59 mul add exch 0.3 mul add
401 }
402 \def\pgffuncshadingcmyktorgb{%
403     % covert to CMY
404     dup 3 1 roll add
405     1.0 2 copy gt { exch } if pop
406     4 1 roll
407     dup 3 1 roll add
408     1.0 2 copy gt { exch } if pop
409     4 1 roll
410     add
411     1.0 2 copy gt { exch } if pop
412     3 1 roll
413     % covert to RGB
414     1.0 exch sub
415     3 1 roll
416     1.0 exch sub
417     3 1 roll
418     1.0 exch sub
419     3 1 roll
420 }
421 \def\pgffuncshadingcmyktogray{%
422     exch 0.11 mul add exch 0.59 mul add exch 0.3 mul add
423     1.0 2 copy gt { exch } if pop
424     1.0 exch sub
425 }
426 \def\pgffuncshadinggraytorgb{%

```

```

427   dup dup
428 }
429 \def\pgffuncshadinggraytocmyk{%
430   0.0 0.0 0.0
431   4 3 roll
432 }

```

Load the correct driver file.

```

433 \def\pgfutilgetcmkshadingsdriver{%
434   \expandafter\pgfutil@getcmkshadingsdriver\pgfsysdriver[
435 ]
436 \def\pgfutil@getcmkshadingsdriver pgfsys-#1[{%
437   \edef\pgfsyscmkshadingsdriver{pgfsys-cmykshadings-#1}%
438 }
439 \pgfutilgetcmkshadingsdriver
440 \input\pgfsyscmkshadingsdriver

```

Style options to use CMYK shadings by default or not when the selected xcolor colour model is natural.

```

441 \newif\ifpgfcmkshadingdefault
442 \DeclareOption{cmyk}{%
443   \pgfcmkshadingdefaulttrue
444 }
445 \DeclareOption{rgb}{%
446   \pgfcmkshadingdefaultfalse
447 }
448 \ExecuteOptions{cmyk}
449 \ProcessOptions\relax
450 \end{package}

```

5.2 Drivers

5.2.1 pdftex driver

```

451 (*pdftex-driver)
452 \ProvidesFile{pgfsys-cmykshadings-pdftex.def}%
453   [2018/10/17 CMYK shadings support for PGF pdftex driver (DCP)]
454 \def\pgfsys@horishading#1#2#3{%
455   {%
456     \pgf@parsefunc{#3}%
457     \pgfmathparse{#2}%
458     \setbox\pgfutil@tempboxa=\hbox to\pgf@max{%
459       \vbox to\pgfmathresult pt{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
460     \pgf@process{\pgfpoint{\pgf@max}{#2}}%
461     \immediate\pdfxform resources {%
462       /Shading << /Sh << /ShadingType 2
463       /ColorSpace \pgf@shading@device\space
464       /Domain [\pgf@pdfparseddomain]
465       /Coords [\pgf@doma\space0 \pgf@domb\space0]
466       /Function \pgf@pdfparsedfunction
467       /Extend [false false] >> >>}\pgfutil@tempboxa% <<
468     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
469       \leavevmode\noexpand\pdfrefxform\the\pdfastxform}%
470     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
471       \pgf@shading@model}%

```



```

472 }%
473 }
474 \def\pgfsys@vertshading#1#2#3{%
475 {%
476   \pgf@parsefunc{#3}%
477   \pgfmathparse{#2}%
478   \setbox\pgfutil@tempboxa=\hbox to\pgfmathresult pt{%
479     \vbox to\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
480   \pgf@process{\pgfpoint{#2}{\pgf@max}}%
481   \immediate\pdfxform resources {%
482     /Shading << /Sh << /ShadingType 2
483     /ColorSpace \pgf@shading@device\space
484     /Domain [\pgf@pdfparseddomain]
485     /Coords [0 \pgf@doma\space0 \pgf@domb]
486     /Function \pgf@pdfparsedfunction
487     /Extend [false false] >> >>}\pgfutil@tempboxa% <<
488   \expandafter\xdef\csname @pgfshading#1!\endcsname{%
489     \leavevmode\noexpand\pdfrefxform\the\pdflastxform}%
490   \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
491     \pgf@shading@model}%
492   }%
493 }
494 \def\pgfsys@radialshading#1#2#3{%
495 {%
496   \pgf@parsefunc{#3}%
497   \setbox\pgfutil@tempboxa=\hbox to2\pgf@max{%
498     \vbox to2\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
499   \pgf@process{#2}%
500   \pgf@xa=\pgf@x%
501   \pgf@ya=\pgf@y%
502   \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
503   \advance\pgf@xa by \pgf@x%
504   \advance\pgf@ya by \pgf@y%
505   \pgf@sys@bp@correct{\pgf@x}%
506   \pgf@sys@bp@correct{\pgf@y}%
507   \pgf@sys@bp@correct{\pgf@xa}%
508   \pgf@sys@bp@correct{\pgf@ya}%
509   \immediate\pdfxform resources {%
510     /Shading << /Sh << /ShadingType 3
511     /ColorSpace \pgf@shading@device\space
512     /Domain [\pgf@pdfparseddomain]
513     /Coords [\pgf@sys@tonumber{\pgf@xa}
514       \pgf@sys@tonumber{\pgf@ya}
515       \pgf@doma\space
516       \pgf@sys@tonumber{\pgf@x}
517       \pgf@sys@tonumber{\pgf@y}
518       \pgf@domb]
519     /Function \pgf@pdfparsedfunction
520     /Extend [true false] >> >>}\pgfutil@tempboxa% <<
521   \expandafter\xdef\csname @pgfshading#1!\endcsname{%
522     \leavevmode\noexpand\pdfrefxform\the\pdflastxform}%
523   \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
524     \pgf@shading@model}%
525   }%

```

```

526 }
527 \def\pgfsys@functionalshading#1#2#3#4{%
528   {%
529     \pgf@process{#2}%
530     \pgf@xa=\pgf@x%
531     \pgf@ya=\pgf@y%
532     \pgf@process{#3}%
533     \pgf@xb=\pgf@x%
534     \pgf@yb=\pgf@y%
535     \advance\pgf@x by-\pgf@xa%
536     \advance\pgf@y by-\pgf@ya%
537     \setbox\pgfutil@tempboxa=\hbox to\pgf@x{%
538       \vbox to\pgf@y{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
539     \pgf@sys@bp@correct{\pgf@xa}%
540     \pgf@sys@bp@correct{\pgf@ya}%
541     \pgf@sys@bp@correct{\pgf@xb}%
542     \pgf@sys@bp@correct{\pgf@yb}%
543     \pgf@xc=-\pgf@xa%
544     \pgf@yc=-\pgf@ya%
545     % Now build the function
546     \pdfobj
547     stream
548     attr
549     {
550       /FunctionType 4
551       /Domain [\pgf@sys@tonumber{\pgf@xa}\space
552         \pgf@sys@tonumber{\pgf@xb}\space
553         \pgf@sys@tonumber{\pgf@ya}\space
554         \pgf@sys@tonumber{\pgf@yb}]
555       /Range [\pgf@shading@functional@range]
556     }
557     {{#4}}%
558     \edef\pgf@temp@num{\the\pdf@lastobj}%
559     \pdfxform resources {%
560       /Shading << /Sh << /ShadingType 1
561       /ColorSpace \pgf@shading@device\space
562       /Matrix [1 0 0 1 \pgf@sys@tonumber{\pgf@xc}\space
563         \pgf@sys@tonumber{\pgf@yc}]
564       /Domain [\pgf@sys@tonumber{\pgf@xa}\space
565         \pgf@sys@tonumber{\pgf@xb}\space
566         \pgf@sys@tonumber{\pgf@ya}\space
567         \pgf@sys@tonumber{\pgf@yb}]
568       /Function \pgf@temp@num\space 0 R
569       >> >>}\pgfutil@tempboxa% <<
570     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
571       \leavevmode%
572       \noexpand\pdfrefxform\the\pdf@lastxform%
573       \noexpand\pdfrefobj\pgf@temp@num%
574     }%
575     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
576       \pgf@shading@model}%
577   }%
578 }
579 (/pdfTeX-driver)

```

5.2.2 xetex driver

```
580 <*xetex-driver>
581 \ProvidesFile{pgfsys-cmykshadings-xetex.def}%
582 [2018/10/17 CMYK shadings support for PGF xetex driver (DCP)]
583 \input pgfsys-cmykshadings-dvipdfmx.def
584 </xetex-driver>
```

5.2.3 luatex driver

```
585 <*luatex-driver>
586 \ProvidesFile{pgfsys-cmykshadings-luatex.def}%
587 [2018/10/17 CMYK shadings support for PGF luatex driver (DCP)]
588 \def\pgfsys@horishading#1#2#3{%
589   {%
590     \pgf@parsefunc{#3}%
591     \pgfmathparse{#2}%
592     \setbox\pgfutil@tempboxa=\hbox to\pgf@max{%
593       \vbox to\pgfmathresult pt{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
594     \pgf@process{\pgfpoint{\pgf@max}{#2}}%
595     \immediate\saveboxresource resources {%
596       /Shading << /Sh << /ShadingType 2
597       /ColorSpace \pgf@shading@device\space
598       /Domain [\pgf@pdfparseddomain]
599       /Coords [\pgf@doma\space0 \pgf@domb\space0]
600       /Function \pgf@pdfparsedfunction
601       /Extend [false false] >> >>}\pgfutil@tempboxa% <<
602     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
603       \leavevmode\noexpand\useboxresource\the\lastsavedboxresourceindex}%
604     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
605       \pgf@shading@model}%
606   }%
607 }
608 \def\pgfsys@vertshading#1#2#3{%
609   {%
610     \pgf@parsefunc{#3}%
611     \pgfmathparse{#2}%
612     \setbox\pgfutil@tempboxa=\hbox to\pgfmathresult pt{%
613       \vbox to\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
614     \pgf@process{\pgfpoint{#2}{\pgf@max}}%
615     \immediate\saveboxresource resources {%
616       /Shading << /Sh << /ShadingType 2
617       /ColorSpace \pgf@shading@device\space
618       /Domain [\pgf@pdfparseddomain]
619       /Coords [0 \pgf@doma\space0 \pgf@domb]
620       /Function \pgf@pdfparsedfunction
621       /Extend [false false] >> >>}\pgfutil@tempboxa% <<
622     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
623       \leavevmode\noexpand\useboxresource\the\lastsavedboxresourceindex}%
624     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
625       \pgf@shading@model}%
626   }%
627 }
```

```

628 \def\pgfsys@radialshading#1#2#3{%
629   {%
630     \pgf@parsefunc{#3}%
631     \setbox\pgfutil@tempboxa=\hbox to2\pgf@max{%
632       \vbox to2\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
633     \pgf@process{#2}%
634     \pgf@xa=\pgf@x%
635     \pgf@ya=\pgf@y%
636     \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
637     \advance\pgf@xa by \pgf@x%
638     \advance\pgf@ya by \pgf@y%
639     \pgf@sys@bp@correct{\pgf@x}%
640     \pgf@sys@bp@correct{\pgf@y}%
641     \pgf@sys@bp@correct{\pgf@xa}%
642     \pgf@sys@bp@correct{\pgf@ya}%
643     \immediate\saveboxresource resources {%
644       /Shading << /Sh << /ShadingType 3
645       /ColorSpace \pgf@shading@device\space
646       /Domain [\pgf@pdfparseddomain]
647       /Coords [\pgf@sys@tonumber{\pgf@xa}
648         \pgf@sys@tonumber{\pgf@ya}
649         \pgf@doma\space
650         \pgf@sys@tonumber{\pgf@x}
651         \pgf@sys@tonumber{\pgf@y}
652         \pgf@domb]
653       /Function \pgf@pdfparsedfunction
654       /Extend [true false] >> >>}\pgfutil@tempboxa% <<
655     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
656       \leavevmode\noexpand\useboxresource\the\lastsavedboxresourceindex}%
657     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
658       \pgf@shading@model}%
659   }%
660 }
661 \def\pgfsys@functionalshading#1#2#3#4{%
662   {%
663     \pgf@process{#2}%
664     \pgf@xa=\pgf@x%
665     \pgf@ya=\pgf@y%
666     \pgf@process{#3}%
667     \pgf@xb=\pgf@x%
668     \pgf@yb=\pgf@y%
669     \advance\pgf@x by-\pgf@xa%
670     \advance\pgf@y by-\pgf@ya%
671     \setbox\pgfutil@tempboxa=\hbox to\pgf@x{%
672       \vbox to\pgf@y{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
673     \pgf@sys@bp@correct{\pgf@xa}%
674     \pgf@sys@bp@correct{\pgf@ya}%
675     \pgf@sys@bp@correct{\pgf@xb}%
676     \pgf@sys@bp@correct{\pgf@yb}%
677     \pgf@xc=-\pgf@xa%
678     \pgf@yc=-\pgf@ya%
679     % Now build the function
680     \pdfextension obj
681     stream

```

```

682 attr
683 {
684   /FunctionType 4
685   /Domain [\pgf@sys@tonumber{\pgf@xa}\space
686     \pgf@sys@tonumber{\pgf@xb}\space
687     \pgf@sys@tonumber{\pgf@ya}\space
688     \pgf@sys@tonumber{\pgf@yb}]
689   /Range [\pgf@shading@functional@range]
690 }
691 {{#4}}%
692 \edef\pgf@temp@num{\the\numexpr\pdffeedback lastobj\relax}%
693 \saveboxresource resources {%
694   /Shading << /Sh << /ShadingType 1
695   /ColorSpace \pgf@shading@device\space
696   /Matrix [1 0 0 1 \pgf@sys@tonumber{\pgf@xc}\space
697     \pgf@sys@tonumber{\pgf@yc}]
698   /Domain [\pgf@sys@tonumber{\pgf@xa}\space
699     \pgf@sys@tonumber{\pgf@xb}\space
700     \pgf@sys@tonumber{\pgf@ya}\space
701     \pgf@sys@tonumber{\pgf@yb}]
702   /Function \pgf@temp@num\space 0 R
703   >> >>}\pgfutil@tempboxa% <<
704 \expandafter\xdef\csname @pgfshading#1!\endcsname{%
705   \leavevmode%
706   \noexpand\useboxresource\the\lastsavedboxresourceindex%
707   \noexpand\pdfextension refobj \pgf@temp@num%
708 }%
709 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
710   \pgf@shading@model}%
711 }%
712 }
713 </luatex-driver>

```

5.2.4 dviPDFmx driver

```

714 <*dviPDFmx-driver>
715 \ProvidesFile{pgfsys-cmykshadings-dviPDFmx.def}%
716 [2018/10/17 CMYK shadings support for PGF dviPDFmx driver (DCP)]
717 \def\pgfsys@horishading#1#2#3{%
718   {%
719     \pgf@parsefunc{#3}%
720     \pgfmathparse{#2}%
721     \pgf@process{\pgfpoint{\pgf@max}{#2}}%
722     \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
723       \special{pdf:bobj @pgfshade\the\pgfsys@objnum\space
724         width \the\pgf@max\space height \pgfmathresult pt}%
725       \special{pdf:put @resources
726         <<
727           /Shading << /Sh << /ShadingType 2
728           /ColorSpace \pgf@shading@device\space
729           /Domain [\pgf@pdfparseddomain]
730           /Coords [\pgf@doma\space0 \pgf@domb\space0]
731           /Function \pgf@pdfparsedfunction
732           /Extend [false false] >> >>

```

```

733     >>}%
734     \pgfsys@invoke{/Sh sh}%
735     \special{pdf:exobj}}}\@tempa% <<
736     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
737     \hbox to\the\pgf@max{\vbox to\pgfmathresult pt{%
738     \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
739     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
740     \pgf@shading@model}%
741     }%
742     \global\advance\pgfsys@objnum\@ne%
743 }
744 \def\pgfsys@vertshading#1#2#3{%
745     {%
746     \pgf@parsefunc{#3}%
747     \pgfmathparse{#2}%
748     \pgf@process{\pgfpoint{\pgf@max}{#2}}%
749     \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
750     \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
751     width \pgfmathresult pt\space height \the\pgf@max}%
752     \special{pdf:put @resources
753     <<
754     /Shading << /Sh << /ShadingType 2
755     /ColorSpace \pgf@shading@device\space
756     /Domain [\pgf@pdfparseddomain]
757     /Coords [0 \pgf@doma\space0 \pgf@domb]
758     /Function \pgf@pdfparsedfunction
759     /Extend [false false] >> >>
760     >>}%
761     \pgfsys@invoke{/Sh sh}%
762     \special{pdf:exobj}}}\@tempa% <<
763     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
764     \hbox to\pgfmathresult pt{\vbox to\the\pgf@max{%
765     \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
766     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
767     \pgf@shading@model}%
768     }%
769     \global\advance\pgfsys@objnum\@ne%
770 }
771 \def\pgfsys@radialshading#1#2#3{%
772     {%
773     \pgf@parsefunc{#3}%
774     \pgf@process{#2}%
775     \pgf@xa=\pgf@x%
776     \pgf@ya=\pgf@y%
777     \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
778     \advance\pgf@xa by \pgf@x%
779     \advance\pgf@ya by \pgf@y%
780     \pgf@sys@bp@correct{\pgf@x}%
781     \pgf@sys@bp@correct{\pgf@y}%
782     \pgf@sys@bp@correct{\pgf@xa}%
783     \pgf@sys@bp@correct{\pgf@ya}%
784     \pgfutil@tempdima=2\pgf@max%
785     \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
786     \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space

```

```

787     width \the\pgfutil@tempdima\space height \the\pgfutil@tempdima}%
788 \special{pdf:put @resources
789 <<
790 /Shading << /Sh << /ShadingType 3
791 /ColorSpace \pgf@shading@device\space
792 /Domain [\pgf@pdfparseddomain]
793 /Coords [\pgf@sys@tonumber{\pgf@xa}
794 \pgf@sys@tonumber{\pgf@ya}
795 \pgf@doma\space \pgf@sys@tonumber{\pgf@x}
796 \pgf@sys@tonumber{\pgf@y}
797 \pgf@domb]
798 /Function \pgf@pdfparsedfunction
799 /Extend [true false] >> >>
800 >>}%
801 \pgf@sys@invoke{/Sh sh}%
802 \special{pdf:exobj}}}\@tempa% <<
803 \expandafter\xdef\csname @pgfshading#1!\endcsname{%
804 \hbox to\the\pgfutil@tempdima{\vbox to\the\pgfutil@tempdima{%
805 \vfil\special{pdf:uxobj @pgfshade\the\pgf@sys@objnum}}\hfil}}%
806 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
807 \pgf@shading@model}%
808 }%
809 \global\advance\pgf@sys@objnum\@ne%
810 }
811 \def\pgf@sys@functionalshading#1#2#3#4{%
812 {%
813 \pgf@process{#2}%
814 \pgf@xa=\pgf@x%
815 \pgf@ya=\pgf@y%
816 \pgf@process{#3}%
817 \pgf@xb=\pgf@x%
818 \pgf@yb=\pgf@y%
819 \advance\pgf@x by-\pgf@xa%
820 \advance\pgf@y by-\pgf@ya%
821 \pgf@sys@bp@correct{\pgf@xa}%
822 \pgf@sys@bp@correct{\pgf@ya}%
823 \pgf@sys@bp@correct{\pgf@xb}%
824 \pgf@sys@bp@correct{\pgf@yb}%
825 \pgf@xc=-\pgf@xa%
826 \pgf@yc=-\pgf@ya%
827 % Now build the function
828 \edef\@tempa{\noexpand\pgfutil@insertatbeginincurrentpagefrombox{%
829 \special{pdf:stream @pgfstream\the\pgf@sys@objnum\space({#4})
830 <</FunctionType 4 /Domain [\pgf@sys@tonumber{\pgf@xa}\space
831 \pgf@sys@tonumber{\pgf@xb}\space
832 \pgf@sys@tonumber{\pgf@ya}\space
833 \pgf@sys@tonumber{\pgf@yb}]
834 /Range [\pgf@shading@functional@range]>>}}}\@tempa%
835 \edef\@tempa{\noexpand\pgfutil@insertatbeginincurrentpagefrombox{%
836 \special{pdf:bxobj @pgfshade\the\pgf@sys@objnum\space
837 width \the\pgf@x\space height \the\pgf@y}%
838 \special{pdf:put @resources <</Shading <</Sh <</ShadingType 1
839 /ColorSpace \pgf@shading@device\space
840 /Matrix [1 0 0 1 \pgf@sys@tonumber{\pgf@xc}\space

```

```

841         \pgf@sys@tonumber{\pgf@yc}]
842     /Domain  [ \pgf@sys@tonumber{\pgf@xa}\space
843         \pgf@sys@tonumber{\pgf@xb}\space
844         \pgf@sys@tonumber{\pgf@ya}\space
845         \pgf@sys@tonumber{\pgf@yb}]
846     /Function @pgfstream\the\pgfsys@objnum>> >> >>}%
847     \pgfsys@invoke{/Sh sh}%
848     \special{pdf:exobj}}}\@tempa% <<
849     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
850     \leavevmode\hbox to\the\pgf@x{\vbox to\the\pgf@y{%
851         \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}}%
852     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
853         \pgf@shading@model}%
854     }%
855     \global\advance\pgfsys@objnum\@ne%
856 }
857 </dvipdfmx-driver>

```

Change History

v1.0

General: First public release 1