

The pgf-cmykshadings package

David Purton*

2018/10/23 v1.1

Abstract

The `pgf-cmykshadings` package provides support for CMYK and grayscale shadings for the `pgf` package. By default `pgf` only supports RGB shadings. `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
	Change History	31

*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 file `pgfcoresshade.code.tex` along with the driver files `pgfsys-*.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
  }
```

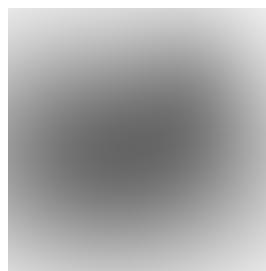
xcolor cmyk model:



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

Declare a horizontal shading.

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

Declare a vertical shading.

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

Declare a radial shading.

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

Declare a functional shading.

4.2.2 Using shadings

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

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

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

<code>\pgfadditionalshadetransform</code>	<code>\pgfadditionalshadetransform{\transformation}</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 [2018/10/23 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@cmyk] /N 1 >> }% <<
69 \edef\pgf@psfuncs{\pgf@prevx\space
70     \pgf@cmyk\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@cmyk}}}%
74 \edef\pgf@sys@prevpos{\the\pgf@x}%
75 \let\pgf@sys@prevcolor=\pgf@sys@cmyk%
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@cmyk}}}%
81 \edef\pgf@sys@prevpos{\the\pgf@x}%
82 \let\pgf@sys@prevcolor=\pgf@sys@cmyk%
83 \edef\pgf@psfuncs{\pgf@prevx\space \pgf@cmyk\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] /C0
91     [\pgf@prevcolor] /C1 [\pgf@cmyk] /N 1 >> }% <<
92 \edef\pgf@funcs{\pgf@funcs\space%
93     << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /C0
94     [\pgf@prevcolor] /C1 [\pgf@cmyk] /N 1 >> }% <<
95 \edef\pgf@prevcolor{\pgf@cmyk}%
96 \pgf@parserestcmyk[]}
97 \def\pgf@getcmyktuplewithmixin#1#2#3#4{%
98     \pgfutil@definecolor{pgfshadetemp}{cmyk}{#1,#2,#3,#4}%
99     \pgfutil@ifundefined{applycolormixins}{\applycolormixins{pgfshadetemp}}%
100     \pgfutil@extractcolorspec{pgfshadetemp}{\pgf@tempcolor}%
101     \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{cmyk}{\pgf@cmykcolor}%
102     \expandafter\pgf@getcmyk@@\pgf@cmykcolor!}
103 \def\pgf@getcmyk@@#1,#2,#3,#4!{%
104     \def\pgf@cmyk{#1 #2 #3 #4}%
105     \def\pgf@sys@cmyk{{#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@parserelastdomgray[]}
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] /C0
136 [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
137 \edef\pgf@funcs{\pgf@funcs\space%
138 << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /C0
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] /C0
162 [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
163 \edef\pgf@funcs{\pgf@funcs\space%
164 << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /C0
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(\langle position \rangle)=(\langle C \rangle,\langle M \rangle,\langle Y \rangle,\langle K \rangle)`).

```

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@grabcmyk}%
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@grabcmyk myk(#1)=(#2,#3,#4,#5){%
219   \pgfutil@definecolor{\pgf@tempcol}{cmyk}{#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\pgfshadecolortocmyk#1#2{%
234 \pgfutil@colorlet{pgf@tempcol}{#1}%
235 \pgfutil@extractcolorspec{pgf@tempcol}{\pgf@tempcolor}%
236 \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{cmyk}{\pgf@cmykcolor}%
237 \expandafter\pgfshading@cmyk\pgf@cmykcolor\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@cmyk#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\pgfshadecolorctogray#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
331         \expandafter\@args\@args%
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@sdef\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@ps@device{setrgbcolor}%
351     \def\pgf@shading@model{rgb}%
352     \ifx\pgf@mod@test\XC@mod@natural
353         \ifpgfcmykshadingdefault
354             \def\pgf@shading@functional@range{0 1 0 1 0 1 0 1}%
355             \def\pgf@shading@device{/DeviceCMYK}%
356         \def\pgf@shading@ps@device{setcmykcolor}%
357         \def\pgf@shading@model{cmyk}%
358         \pgfshadingmodelrgbfalse
359         \pgfshadingmodelcmyktrue
360     \else
361         \def\pgf@shading@functional@range{0 1 0 1 0 1}%
362         \def\pgf@shading@device{/DeviceRGB}%
363         \def\pgf@shading@ps@device{setrgbcolor}%
364         \def\pgf@shading@model{rgb}%
365     \fi
366 \fi
367 \ifx\pgf@mod@test\XC@mod@cmyk
368     \def\pgf@shading@functional@range{0 1 0 1 0 1 0 1}%
369     \def\pgf@shading@device{/DeviceCMYK}%
370     \def\pgf@shading@ps@device{setcmykcolor}%
371     \def\pgf@shading@model{cmyk}%
372     \pgfshadingmodelrgbfalse
373     \pgfshadingmodelcmyktrue
374 \fi

```

```

375 \ifx\pgf@mod@test\XC@mod@gray
376   \def\pgf@shading@functional@range{0 1}%
377   \def\pgf@shading@device{/DeviceGray}%
378   \def\pgf@shading@ps@device{setgray}%
379   \def\pgf@shading@model{gray}%
380   \pgfshadingmodelrgbfalse
381   \pgfshadingmodelgraytrue
382 \fi
383 \edef\pgf@sys@driver@dvisvgm{pgfsys-dvisvgm.def}%
384 \ifx\pgfsysdriver\pgf@sys@driver@dvisvgm
385   \def\pgf@shading@model{rgb}%
386 \fi
387 \edef\pgf@sys@driver@texforht{pgfsys-tex4ht.def}%
388 \ifx\pgfsysdriver\pgf@sys@driver@texforht
389   \def\pgf@shading@model{rgb}%
390 \fi
391 }

```

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

```

392 \def\pgffuncshadingrgbtocmyk{%
393   1.0 exch sub 3 1 roll
394   1.0 exch sub 3 1 roll
395   1.0 exch sub 3 1 roll
396   3 copy
397   2 copy gt { exch } if pop
398   2 copy gt { exch } if pop
399   dup 3 1 roll sub
400   0.0 2 copy lt { exch } if pop
401   1.0 2 copy gt { exch } if pop
402   4 1 roll
403   dup 3 1 roll sub
404   0.0 2 copy lt { exch } if pop
405   1.0 2 copy gt { exch } if pop
406   4 1 roll
407   dup 3 1 roll sub
408   0.0 2 copy lt { exch } if pop
409   1.0 2 copy gt { exch } if pop
410   4 1 roll
411 }
412 \def\pgffuncshadingrgbtogray{%
413   0.11 mul exch 0.59 mul add exch 0.3 mul add
414 }
415 \def\pgffuncshadingcmyktorgb{%
416   % covert to CMY
417   dup 3 1 roll add
418   1.0 2 copy gt { exch } if pop
419   4 1 roll
420   dup 3 1 roll add
421   1.0 2 copy gt { exch } if pop
422   4 1 roll
423   add
424   1.0 2 copy gt { exch } if pop
425   3 1 roll

```

```

426 % covert to RGB
427 1.0 exch sub
428 3 1 roll
429 1.0 exch sub
430 3 1 roll
431 1.0 exch sub
432 3 1 roll
433 }
434 \def\pgffuncshadingcmyktogray{%
435   exch 0.11 mul add exch 0.59 mul add exch 0.3 mul add
436   1.0 2 copy gt { exch } if pop
437   1.0 exch sub
438 }
439 \def\pgffuncshadinggraytorgb{%
440   dup dup
441 }
442 \def\pgffuncshadinggraytocmyk{%
443   0.0 0.0 0.0
444   4 3 roll
445 }

```

Load the correct driver file.

```

446 \def\pgfutilgetcmykshadingsdriver{%
447   \expandafter\pgfutil@getcmykshadingsdriver\pgfsysdriver[
448 ]
449 \def\pgfutil@getcmykshadingsdriver pgfsys-#1[{%
450   \edef\pgfsyscmykshadingsdriver{pgfsys-cmykshadings-#1}%
451 }
452 \pgfutilgetcmykshadingsdriver
453 \pgfutil@InputIfFileExists{\pgfsyscmykshadingsdriver}{-}{-}

```

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

```

454 \newif\ifpgfcmykshadingdefault
455 \DeclareOption{cmyk}{%
456   \pgfcmykshadingdefaulttrue
457 }
458 \DeclareOption{rgb}{%
459   \pgfcmykshadingdefaultfalse
460 }
461 \ExecuteOptions{cmyk}
462 \ProcessOptions\relax
463 </package>

```

5.2 Drivers

pdftex driver

```

464 <*pdftex-driver>
465 \ProvidesFile{pgfsys-cmykshadings-pdftex.def}%
466 [2018/10/17 CMYK shadings support for PGF pdftex driver (DCP)]
467 \def\pgfsys@horishading#1#2#3{%
468   {%
469     \pgf@parsefunc{#3}%

```



```

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

```

```

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

```

```

578     \pgf@sys@tonumber{\pgf@xb}\space
579     \pgf@sys@tonumber{\pgf@ya}\space
580     \pgf@sys@tonumber{\pgf@yb}]
581     /Function \pgf@temp@num\space 0 R
582     >> >>}\pgfutil@tempboxa% <<
583     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
584     \leavevmode%
585     \noexpand\pdfrefxform\the\pdflastxform%
586     \noexpand\pdfrefobj\pgf@temp@num%
587     }%
588     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
589     \pgf@shading@model}%
590     }%
591 }
592 </pdftex-driver>

```

xetex driver

```

593 <*xetex-driver>
594 \ProvidesFile{pgfsys-cmykshadings-xetex.def}%
595 [2018/10/17 CMYK shadings support for PGF xetex driver (DCP)]
596 \input pgfsys-cmykshadings-dvipdfmx.def
597 </xetex-driver>

```

luatex driver

```

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

```

```

625 \setbox\pgfutil@tempboxa=\hbox to\pgfmathresult pt{%
626 \vbox to\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
627 \pgf@process{\pgfpoint{#2}{\pgf@max}}%
628 \immediate\saveboxresource resources {%
629 /Shading << /Sh << /ShadingType 2
630 /ColorSpace \pgf@shading@device\space
631 /Domain [\pgf@pdfparseddomain]
632 /Coords [0 \pgf@doma\space0 \pgf@domb]
633 /Function \pgf@pdfparsedfunction
634 /Extend [false false] >> >>}\pgfutil@tempboxa% <<
635 \expandafter\xdef\csname @pgfshading#1!\endcsname{%
636 \leavevmode\noexpand\useboxresource\the\lastsavedboxresourceindex}%
637 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
638 \pgf@shading@model}%
639 }%
640 }
641 \def\pgfsys@radialshading#1#2#3{%
642 {%
643 \pgf@parsefunc{#3}%
644 \setbox\pgfutil@tempboxa=\hbox to2\pgf@max{%
645 \vbox to2\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
646 \pgf@process{#2}%
647 \pgf@xa=\pgf@x%
648 \pgf@ya=\pgf@y%
649 \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
650 \advance\pgf@xa by \pgf@x%
651 \advance\pgf@ya by \pgf@y%
652 \pgf@sys@bp@correct{\pgf@x}%
653 \pgf@sys@bp@correct{\pgf@y}%
654 \pgf@sys@bp@correct{\pgf@xa}%
655 \pgf@sys@bp@correct{\pgf@ya}%
656 \immediate\saveboxresource resources {%
657 /Shading << /Sh << /ShadingType 3
658 /ColorSpace \pgf@shading@device\space
659 /Domain [\pgf@pdfparseddomain]
660 /Coords [\pgf@sys@tonumber{\pgf@xa}
661 \pgf@sys@tonumber{\pgf@ya}
662 \pgf@doma\space
663 \pgf@sys@tonumber{\pgf@x}
664 \pgf@sys@tonumber{\pgf@y}
665 \pgf@domb]
666 /Function \pgf@pdfparsedfunction
667 /Extend [true false] >> >>}\pgfutil@tempboxa% <<
668 \expandafter\xdef\csname @pgfshading#1!\endcsname{%
669 \leavevmode\noexpand\useboxresource\the\lastsavedboxresourceindex}%
670 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
671 \pgf@shading@model}%
672 }%
673 }
674 \def\pgfsys@functionalshading#1#2#3#4{%
675 {%
676 \pgf@process{#2}%
677 \pgf@xa=\pgf@x%
678 \pgf@ya=\pgf@y%

```

```

679 \pgf@process{#3}%
680 \pgf@xb=\pgf@x%
681 \pgf@yb=\pgf@y%
682 \advance\pgf@x by-\pgf@xa%
683 \advance\pgf@y by-\pgf@ya%
684 \setbox\pgfutil@tempboxa=\hbox to\pgf@x{%
685   \vbox to\pgf@y{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
686 \pgf@sys@bp@correct{\pgf@xa}%
687 \pgf@sys@bp@correct{\pgf@ya}%
688 \pgf@sys@bp@correct{\pgf@xb}%
689 \pgf@sys@bp@correct{\pgf@yb}%
690 \pgf@xc=-\pgf@xa%
691 \pgf@yc=-\pgf@ya%
692 % Now build the function
693 \pdfextension obj
694 stream
695 attr
696 {
697   /FunctionType 4
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   /Range [\pgf@shading@functional@range]
703 }
704 {{#4}}%
705 \edef\pgf@temp@num{\the\numexpr\pdffeedback lastobj\relax}%
706 \saveboxresource resources {%
707   /Shading << /Sh << /ShadingType 1
708   /ColorSpace \pgf@shading@device\space
709   /Matrix [1 0 0 1 \pgf@sys@tonumber{\pgf@xc}\space
710     \pgf@sys@tonumber{\pgf@yc}]
711   /Domain [\pgf@sys@tonumber{\pgf@xa}\space
712     \pgf@sys@tonumber{\pgf@xb}\space
713     \pgf@sys@tonumber{\pgf@ya}\space
714     \pgf@sys@tonumber{\pgf@yb}]
715   /Function \pgf@temp@num\space 0 R
716   >> >>}\pgfutil@tempboxa% <<
717 \expandafter\xdef\csname @pgfshading#1!\endcsname{%
718   \leavevmode%
719   \noexpand\useboxresource\the\lastsavedboxresourceindex%
720   \noexpand\pdfextension refobj \pgf@temp@num%
721 }%
722 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
723   \pgf@shading@model}%
724 }%
725 }
726 </luatex-driver>

```

dvipdfmx driver

```

727 <*dvipdfmx-driver>
728 \ProvidesFile{pgfsys-cmykshadings-dvipdfmx.def}%
729 [2018/10/17 CMYK shadings support for PGF dvipdfmx driver (DCP)]

```

```

730 \def\pgfsys@horishading#1#2#3{%
731   {%
732     \pgf@parsefunc{#3}%
733     \pgfmathparse{#2}%
734     \pgf@process{\pgfpoint{\pgf@max}{#2}}%
735     \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
736       \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
737         width \the\pgf@max\space height \pgfmathresult pt}%
738       \special{pdf:put @resources
739       <<
740         /Shading << /Sh << /ShadingType 2
741         /ColorSpace \pgf@shading@device\space
742         /Domain [\pgf@pdfparseddomain]
743         /Coords [\pgf@doma\space0 \pgf@domb\space0]
744         /Function \pgf@pdfparsedfunction
745         /Extend [false false] >> >>
746       >>}%
747       \pgfsys@invoke{/Sh sh}%
748       \special{pdf:exobj}}}\@tempa% <<
749       \expandafter\xdef\csname @pgfshading#1!\endcsname{%
750         \hbox to\the\pgf@max{\vbox to\pgfmathresult pt{%
751           \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
752       \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
753         \pgf@shading@model}%
754     }%
755     \global\advance\pgfsys@objnum\@ne%
756   }
757 \def\pgfsys@vertshading#1#2#3{%
758   {%
759     \pgf@parsefunc{#3}%
760     \pgfmathparse{#2}%
761     \pgf@process{\pgfpoint{\pgf@max}{#2}}%
762     \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
763       \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
764         width \pgfmathresult pt\space height \the\pgf@max}%
765       \special{pdf:put @resources
766       <<
767         /Shading << /Sh << /ShadingType 2
768         /ColorSpace \pgf@shading@device\space
769         /Domain [\pgf@pdfparseddomain]
770         /Coords [0 \pgf@doma\space0 \pgf@domb]
771         /Function \pgf@pdfparsedfunction
772         /Extend [false false] >> >>
773       >>}%
774       \pgfsys@invoke{/Sh sh}%
775       \special{pdf:exobj}}}\@tempa% <<
776       \expandafter\xdef\csname @pgfshading#1!\endcsname{%
777         \hbox to\pgfmathresult pt{\vbox to\the\pgf@max{%
778           \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
779       \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
780         \pgf@shading@model}%
781     }%
782     \global\advance\pgfsys@objnum\@ne%
783   }

```

```

784 \def\pgfsys@radialshading#1#2#3{%
785   {%
786     \pgf@parsefunc{#3}%
787     \pgf@process{#2}%
788     \pgf@xa=\pgf@x%
789     \pgf@ya=\pgf@y%
790     \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
791     \advance\pgf@xa by \pgf@x%
792     \advance\pgf@ya by \pgf@y%
793     \pgf@sys@bp@correct{\pgf@x}%
794     \pgf@sys@bp@correct{\pgf@y}%
795     \pgf@sys@bp@correct{\pgf@xa}%
796     \pgf@sys@bp@correct{\pgf@ya}%
797     \pgfutil@tempdima=2\pgf@max%
798     \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
799       \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
800         width \the\pgfutil@tempdima\space height \the\pgfutil@tempdima}%
801       \special{pdf:put @resources
802         <<
803           /Shading << /Sh << /ShadingType 3
804           /ColorSpace \pgf@shading@device\space
805           /Domain [\pgf@pdfparseddomain]
806           /Coords [\pgf@sys@tonumber{\pgf@xa}
807             \pgf@sys@tonumber{\pgf@ya}
808             \pgf@doma\space \pgf@sys@tonumber{\pgf@x}
809             \pgf@sys@tonumber{\pgf@y}
810             \pgf@domb]
811           /Function \pgf@pdfparsedfunction
812           /Extend [true false] >> >>
813         >>}%
814         \pgfsys@invoke{/Sh sh}%
815         \special{pdf:exobj}}}\@tempa% <<
816     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
817       \hbox to\the\pgfutil@tempdima{\vbox to\the\pgfutil@tempdima{%
818         \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
819     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
820       \pgf@shading@model}%
821   }%
822   \global\advance\pgfsys@objnum\@ne%
823 }
824 \def\pgfsys@functionalshading#1#2#3#4{%
825   {%
826     \pgf@process{#2}%
827     \pgf@xa=\pgf@x%
828     \pgf@ya=\pgf@y%
829     \pgf@process{#3}%
830     \pgf@xb=\pgf@x%
831     \pgf@yb=\pgf@y%
832     \advance\pgf@x by-\pgf@xa%
833     \advance\pgf@y by-\pgf@ya%
834     \pgf@sys@bp@correct{\pgf@xa}%
835     \pgf@sys@bp@correct{\pgf@ya}%
836     \pgf@sys@bp@correct{\pgf@xb}%
837     \pgf@sys@bp@correct{\pgf@yb}%

```

```

838 \pgf@xc=-\pgf@xa%
839 \pgf@yc=-\pgf@ya%
840 % Now build the function
841 \edef\@tempa{\noexpand\pgfutil@insertatbeginincurrentpagefrombox{%
842 \special{pdf:stream @pgfstream\the\pgfsys@objnum\space({#4})
843 <</FunctionType 4 /Domain [\pgf@sys@tonumber{\pgf@xa}\space
844 \pgf@sys@tonumber{\pgf@xb}\space
845 \pgf@sys@tonumber{\pgf@ya}\space
846 \pgf@sys@tonumber{\pgf@yb}]
847 /Range [\pgf@shading@functional@range]>>}}}\@tempa%
848 \edef\@tempa{\noexpand\pgfutil@insertatbeginincurrentpagefrombox{%
849 \special{pdf:bbox @pgfshade\the\pgfsys@objnum\space
850 width \the\pgf@x\space height \the\pgf@y}%
851 \special{pdf:put @resources <</Shading <</Sh <</ShadingType 1
852 /ColorSpace \pgf@shading@device\space
853 /Matrix [1 0 0 1 \pgf@sys@tonumber{\pgf@xc}\space
854 \pgf@sys@tonumber{\pgf@yc}]
855 /Domain [\pgf@sys@tonumber{\pgf@xa}\space
856 \pgf@sys@tonumber{\pgf@xb}\space
857 \pgf@sys@tonumber{\pgf@ya}\space
858 \pgf@sys@tonumber{\pgf@yb}]
859 /Function @pgfstream\the\pgfsys@objnum>> >> >>}}%
860 \pgfsys@invoke{/Sh sh}%
861 \special{pdf:exobj}}}\@tempa% <<
862 \expandafter\xdef\csname @pgfshading#1!\endcsname{%
863 \leavevmode\hbox to\the\pgf@x{\vbox to\the\pgf@y{%
864 \vfil\special{pdf:uobj @pgfshade\the\pgfsys@objnum}}\hfil}}}%
865 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
866 \pgf@shading@model}%
867 }%
868 \global\advance\pgfsys@objnum\@ne%
869 }
870 </dvipdfm-driver>

```

dvipdfm driver

```

871 <*dvipdfm-driver>
872 \ProvidesFile{pgfsys-cmykshadings-dvipdfm.def}%
873 [2018/10/18 CMYK shadings support for PGF dvipdfm driver (DCP)]
874 \def\pgfsys@horishading#1#2#3{%
875 {%
876 \pgf@parsefunc{#3}%
877 \pgf@process{\pgfpoint{\pgf@max}{#2}}%
878 \edef\@temp{\noexpand\pgfutil@insertatbeginincurrentpage{%
879 \special{pdf: beginxobj @pgfshade#1 width \the\pgf@max\space
880 height \the\pgf@y}}}\@temp%
881 \edef\@temp{\noexpand\pgfutil@insertatbeginincurrentpage{%
882 \special{pdf: put @resources <<
883 /Shading << /Sh << /ShadingType 2
884 /ColorSpace \pgf@shading@device\space
885 /Domain [\pgf@pdfparseddomain]
886 /Coords [\pgf@doma\space0 \pgf@domb\space0]
887 /Function \pgf@pdfparsedfunction
888 /Extend [false false] >> >> >>}}}\@temp% <<

```



```

889     \pgfutil@insertatbegincurrentpage{\special{pdf: content /Sh sh}}%
890     \special{pdf: endxobj}}}%
891     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
892     \hbox to\the\pgf@max{\vbox to#2{%
893     \vfil\special{pdf: usexobj @pgfshade#1}}\hfil}}}%
894     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
895     \pgf@shading@model}}%
896 }%
897 }
898 \def\pgfsys@vertshading#1#2#3{%
899 {%
900     \pgf@parsefunc{#3}%
901     \pgf@process{\pgfpoint{\pgf@max}{#2}}}%
902     \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
903     \special{pdf: beginxobj @pgfshade#1 width \the\pgf@y\space
904     height \the\pgf@max\space}}}\@temp%
905     \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
906     \special{pdf: put @resources <<
907     /Shading << /Sh << /ShadingType 2
908     /ColorSpace \pgf@shading@device\space
909     /Domain [\pgf@pdfparseddomain]
910     /Coords [0 \pgf@doma\space0 \pgf@domb]
911     /Function \pgf@pdfparsedfunction
912     /Extend [false false] >> >> >>}}}\@temp% <<
913     \pgfutil@insertatbegincurrentpage{\special{pdf: content /Sh sh}}%
914     \special{pdf: endxobj}}}%
915     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
916     \hbox to#2{\vbox to\the\pgf@max{\vfil\special{%
917     pdf: usexobj @pgfshade#1}}\hfil}}}%
918     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
919     \pgf@shading@model}}%
920 }%
921 }
922 \def\pgfsys@radialshading#1#2#3{%
923 {%
924     \pgf@parsefunc{#3}%
925     \pgf@process{#2}%
926     \pgf@xa=\pgf@x%
927     \pgf@ya=\pgf@y%
928     \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}}%
929     \advance\pgf@xa by \pgf@x\relax%
930     \advance\pgf@ya by \pgf@y\relax%
931     \pgf@sys@bp@correct{\pgf@x}%
932     \pgf@sys@bp@correct{\pgf@y}%
933     \pgf@sys@bp@correct{\pgf@xa}%
934     \pgf@sys@bp@correct{\pgf@ya}%
935     \pgfutil@tempdima=2\pgf@max\relax%
936     \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
937     \special{pdf: beginxobj @pgfshade#1 width
938     \the\pgfutil@tempdima\space height \the\pgfutil@tempdima}}}\@temp%
939     \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
940     \special{pdf: put @resources <<
941     /Shading << /Sh << /ShadingType 3
942     /ColorSpace \pgf@shading@device\space

```

```

943 /Domain [\pgf@pdfparseddomain]
944 /Coords [\pgf@sys@tonumber{\pgf@xa} \pgf@sys@tonumber{\pgf@ya}
945 \pgf@doma\space \pgf@sys@tonumber{\pgf@x} \pgf@sys@tonumber{\pgf@y}
946 \pgf@domb]
947 /Function \pgf@pdfparsedfunction
948 /Extend [true false] >> >> >>}}\@temp% <<
949 \pgfutil@insertatbegincurrentpage{\special{pdf: content /Sh sh}}%
950 \special{pdf: endxobj}}}%
951 \expandafter\xdef\csname @pgfshading#1!\endcsname{%
952 \hbox to\the\pgfutil@tempdima{\vbox to\the\pgfutil@tempdima{%
953 \vfil\special{pdf: usexobj @pgfshade#1}}\hfil}}}%
954 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
955 \pgf@shading@model}}%
956 }%
957 }%
958 </dvipdfm-driver>

```

dvips driver

```

959 <*dvips-driver>
960 \ProvidesFile{pgfsys-cmykshadings-dvips.def}%
961 [2018/10/23 CMYK shadings support for PGF dvips driver (DCP)]
962 \input pgfsys-cmykshadings-common-postscript.def
963 </dvips-driver>

```

textures driver

```

964 <*textures-driver>
965 \ProvidesFile{pgfsys-cmykshadings-textures.def}%
966 [2018/10/23 CMYK shadings support for PGF textures driver (DCP)]
967 \input pgfsys-cmykshadings-common-postscript.def
968 </textures-driver>

```

vtex driver

```

969 <*vtex-driver>
970 \ProvidesFile{pgfsys-cmykshadings-vtex.def}%
971 [2018/10/23 CMYK shadings support for PGF vtex driver (DCP)]
972 \input pgfsys-cmykshadings-common-postscript.def
973 </vtex-driver>

```

PostScript® driver common code

```

974 <*common-ps-driver>
975 \ProvidesFile{pgfsys-cmykshadings-common-postscript.def}%
976 [2018/10/23 CMYK shadings support for PGF PostScript driver (DCP)]
977 \TeX shading macros.
978 \def\pgfsys@horishading#1#2#3{%
979 {%
980 \pgf@parsefunc{#3}%
981 \pgfmathsetlength\pgf@x{#2}%
982 \pgf@xa=\pgf@x%

```

```

982 \pgf@sys@bp@correct{\pgf@x}%
983 \pgf@y=\pgf@max%
984 \pgf@sys@bp@correct{\pgf@y}%
985 \expandafter\xdef\csname @pgfshading#1!\endcsname{\hbox to \the\pgf@max{%
986 \noexpand\vrule width0pt height\the\pgf@xa%
987 \noexpand\pgfsys@beginpurepicture%
988 \noexpand\pgfsys@rect{0pt}{0pt}{\the\pgf@max}{\the\pgf@xa}%
989 \noexpand\pgfsys@clipnext%
990 \noexpand\pgfsys@discardpath%
991 \noexpand\pgfsys@invoke{\pgf@domb\space \pgf@sys@tonumber{\pgf@x}
992 pgfH\pgf@shading@model\space \pgf@psfuncs\space pop}%
993 \hss%
994 \noexpand\pgfsys@endpurepicture}}%
995 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
996 \pgf@shading@model}%
997 }%
998 }
999 \def\pgfsys@vertshading#1#2#3{%
1000 {%
1001 \pgf@parsefunc{#3}%
1002 \pgfmathsetlength\pgf@x{#2}%
1003 \pgf@xa=\pgf@x%
1004 \pgf@sys@bp@correct{\pgf@x}%
1005 \pgf@y=\pgf@max%
1006 \pgf@sys@bp@correct{\pgf@y}%
1007 \expandafter\xdef\csname @pgfshading#1!\endcsname{\hbox to\the\pgf@xa{%
1008 \noexpand\vrule width0pt height\the\pgf@max%
1009 \noexpand\pgfsys@beginpurepicture%
1010 \noexpand\pgfsys@rect{0pt}{0pt}{\the\pgf@xa}{\the\pgf@max}%
1011 \noexpand\pgfsys@clipnext%
1012 \noexpand\pgfsys@discardpath%
1013 \noexpand\pgfsys@invoke{\pgf@domb\space \pgf@sys@tonumber{\pgf@x}
1014 pgfV\pgf@shading@model\space \pgf@psfuncs\space pop}%
1015 \hss%
1016 \noexpand\pgfsys@endpurepicture}}%
1017 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
1018 \pgf@shading@model}%
1019 }%
1020 }
1021 \def\pgfsys@radialshading#1#2#3{%
1022 {%
1023 \pgf@parsefunc{#3}%
1024 \pgf@process{#2}%
1025 \pgf@sys@bp@correct{\pgf@x}%
1026 \pgf@sys@bp@correct{\pgf@y}%
1027 \pgf@xa=2\pgf@max%
1028 \pgf@sys@bp@correct{\pgf@max}%
1029 \advance\pgf@x by \pgf@max%
1030 \advance\pgf@y by \pgf@max%
1031 \expandafter\xdef\csname @pgfshading#1!\endcsname{\hbox to \the\pgf@xa{%
1032 \noexpand\vrule width0pt height\the\pgf@xa%
1033 \noexpand\pgfsys@beginpurepicture%
1034 \noexpand\pgfsys@invoke{%
1035 \pgf@domb\space \pgf@sys@tonumber{\pgf@y}

```

```

1036         \pgf@sys@tonumber{\pgf@x}
1037         \pgf@sys@tonumber{\pgf@max} pgfR1\pgf@shading@model\space
1038         \pgf@psfuncs\space \pgf@firstcolor\space \pgf@doma\space
1039         pgfR2\pgf@shading@model}%
1040     \hss%
1041     \noexpand\pgfsys@endpurepicture}}%
1042     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
1043         \pgf@shading@model}%
1044     }%
1045 }
1046 \def\pgfsys@functionalshading#1#2#3#4{%
1047     {%
1048         \pgf@process{#2}%
1049         \pgf@xa=\pgf@x%
1050         \pgf@ya=\pgf@y%
1051         \pgf@process{#3}%
1052         \pgf@xb=\pgf@x%
1053         \pgf@yb=\pgf@y%
1054         \advance\pgf@x by-\pgf@xa%
1055         \advance\pgf@y by-\pgf@ya%
1056         \pgf@sys@bp@correct{\pgf@xa}%
1057         \pgf@sys@bp@correct{\pgf@ya}%
1058         \pgf@sys@bp@correct{\pgf@xb}%
1059         \pgf@sys@bp@correct{\pgf@yb}%
1060         \pgf@xc=-\pgf@xa%
1061         \pgf@yc=-\pgf@ya%
1062         \expandafter\xdef\csname @pgfshading#1!\endcsname{%
1063             \hbox to\the\pgf@x{\vbox to\the\pgf@y{\vfil
1064                 \noexpand\pgfsys@beginpurepicture%
1065                 \noexpand\pgfsys@invoke{%
1066                     \pgf@sys@tonumber{\pgf@xc} \pgf@sys@tonumber{\pgf@yc} translate
1067                     1.1 setlinewidth [] 0 setdash 0 setlinecap
1068                     /pgfproc {#4} bind def
1069                     \pgf@sys@tonumber{\pgf@ya} 1 \pgf@sys@tonumber{\pgf@yb}
1070                     {
1071                         \pgf@sys@tonumber{\pgf@xa} 1 \pgf@sys@tonumber{\pgf@xb}
1072                         { 1 index 2 copy pgfproc \pgf@shading@ps@device\space
1073                             moveto 1.1 0 rlineto stroke }
1074                     for
1075                     pop
1076                 }
1077                 for
1078             }%
1079             \noexpand\pgfsys@endpurepicture
1080         }\hfil}%
1081     }%
1082     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
1083         \pgf@shading@model}%
1084     }%
1085 }

```

PostScript® support code.

```

1086 \g@addto@macro\pgfsys@atbegindocument{%

```

Define RGB PostScript® shading functions.

```

1087 \pgf@sys@postscript@header{/pgfHrgb { pgfH } bind def}%
1088 \pgf@sys@postscript@header{/pgfVrgb { pgfV } bind def}%
1089 \pgf@sys@postscript@header{/pgfR1rgb { pgfR1 } bind def}%
1090 \pgf@sys@postscript@header{/pgfR2rgb { pgfR2 } bind def}%

Define CMYK PostScript® shading functions.

1091 \pgf@sys@postscript@header{/pgfHcmyk{%
1092     /pgfheight exch def 0.75 setlinewidth [] 0 setdash
1093     /pgfshade {pgfAcmyk} def /pgfdir { dup 0 moveto
1094         dup 6 index lineto } bind def} bind def}%
1095 \pgf@sys@postscript@header{/pgfVcmyk{%
1096     /pgfheight exch def 0.75 setlinewidth [] 0 setdash
1097     /pgfshade {pgfAcmyk} def /pgfdir { dup 0 exch moveto dup 6 index
1098         exch lineto } bind def} bind def}%
1099 \pgf@sys@postscript@header{/pgfAcmyk{
1100     /pgfdiff 10 index round cvi 10 index round cvi sub 2 mul 1 add def
1101     3 index 8 index sub pgfdiff div % put cyan-step on stack
1102     3 index 8 index sub pgfdiff div % put magenta-step on stack
1103     3 index 8 index sub pgfdiff div % put yellow-step on stack
1104     3 index 8 index sub pgfdiff div % put black-step on stack
1105     pgfheight 12 index 12 index 12 index 12 index 18 index
1106     pgfdiff {
1107         4 index 4 index 4 index 4 index setcmykcolor % Set color
1108         pgfdir
1109         stroke
1110         5 -1 roll 9 index add % cyan += inccyan
1111         5 -1 roll 8 index add % magenta += incmagenta
1112         5 -1 roll 7 index add % yellow += incyellow
1113         5 -1 roll 6 index add % black += incblack
1114         5 -1 roll .5 sub % x += 0.5
1115     } repeat
1116     mark 19 1 roll cleartomark exch pop % leave only start x on stack
1117 }bind def }
1118 \pgf@sys@postscript@header{/pgfR1cmyk{
1119     newpath dup dup dup 0 360 arc clip newpath
1120     dup /pgfendx exch def
1121     /pgfendy exch def
1122     0.875 setlinewidth
1123     [] 0 setdash
1124     /pgfshade {pgfRcmyk} def
1125     /pgfstartx exch def
1126     /pgfstarty exch def
1127     /pgfdiffx pgfendx pgfstartx sub def
1128     /pgfdiffy pgfendy pgfstarty sub def
1129     dup /pgfdomb exch def
1130 }bind def }
1131 \pgf@sys@postscript@header{/pgfR2cmyk{
1132     newpath 0.5 add pgfcircx pgfcircy 3 2 roll 0 360 arc
1133     setcmykcolor fill pop}bind def }%
1134 \pgf@sys@postscript@header{/pgfRcmyk{
1135     /pgfdiff 10 index round cvi 10 index round cvi sub 4 mul 1 add def
1136     /pgfcircx pgfstartx 11 index pgfdiffx pgfdomb div mul add def
1137     /pgfcircy pgfstarty 11 index pgfdiffy pgfdomb div mul add def
1138     /pgfcircxe pgfstartx 10 index pgfdiffx pgfdomb div mul add def
1139     /pgfcircye pgfstarty 10 index pgfdiffy pgfdomb div mul add def

```

```

1140 /pgfxstep pgfcircxe pgfcircx sub pgfdiff div def
1141 /pgfystep pgfcircye pgfcircy sub pgfdiff div def
1142 3 index 8 index sub pgfdiff div % put cyan-step on stack
1143 3 index 8 index sub pgfdiff div % put magenta-step on stack
1144 3 index 8 index sub pgfdiff div % put yellow-step on stack
1145 3 index 8 index sub pgfdiff div % put black-step on stack
1146 11 index 11 index 11 index 11 index 17 index
1147 pgfdiff {
1148   4 index 4 index 4 index 4 index setcmykcolor % Set color
1149   pgfcircx pgfcircy 2 index 0 360 arc closepath
1150   stroke
1151   5 -1 roll 8 index add % cyan += inccyan
1152   5 -1 roll 7 index add % magenta += incmagenta
1153   5 -1 roll 6 index add % yellow += incyellow
1154   5 -1 roll 5 index add % black += incblack
1155   5 -1 roll .25 sub % x += 0.25
1156   /pgfcircx pgfcircx pgfxstep add def
1157   /pgfcircy pgfcircy pgfystep add def
1158 } repeat
1159 mark 18 1 roll cleartomark exch pop % leave only start x on stack
1160 }bind def}

```

Define grayscale PostScript® shading functions.

```

1161 \pgf@sys@postscript@header{/pgfHgray{%
1162   /pgfheight exch def 0.75 setlinewidth [] 0 setdash
1163   /pgfshade {pgfAgray} def /pgfdir { dup 0 moveto
1164     dup 3 index lineto } bind def} bind def}%
1165 \pgf@sys@postscript@header{/pgfVgray{%
1166   /pgfheight exch def 0.75 setlinewidth [] 0 setdash
1167   /pgfshade {pgfAgray} def /pgfdir { dup 0 exch moveto dup 3 index
1168     exch lineto } bind def} bind def}%
1169 \pgf@sys@postscript@header{/pgfAgray{
1170   /pgfdiff 4 index round cvi 4 index round cvi sub 2 mul 1 add def
1171   dup 2 index sub pgfdiff div % put gray-step on stack
1172   pgfheight 3 index 6 index
1173   pgfdiff {
1174     1 index setgray % Set color
1175     pgfdir
1176     stroke
1177     exch 3 index add % gray += incgray
1178     exch .5 sub % x += 0.5
1179   } repeat
1180   mark 7 1 roll cleartomark exch pop % leave only start x on stack
1181 }bind def }
1182 \pgf@sys@postscript@header{/pgfR1gray{
1183   newpath dup dup dup 0 360 arc clip newpath
1184   dup /pgfendx exch def
1185   /pgfendy exch def
1186   0.875 setlinewidth
1187   [] 0 setdash
1188   /pgfshade {pgfRgray} def
1189   /pgfstartx exch def
1190   /pgfstarty exch def
1191   /pgfdiffx pgfendx pgfstartx sub def
1192   /pgfdiffy pgfendy pgfstarty sub def

```

```

1193     dup /pgfdomb exch def
1194   }bind def }
1195 \pgf@sys@postscript@header{/pgfR2gray{
1196   newpath 0.5 add pgfcircx pgfcircy 3 2 roll 0 360 arc
1197   setgray fill pop}bind def }%
1198 \pgf@sys@postscript@header{/pgfRgray{
1199   /pgfdiff 4 index round cvi 4 index round cvi sub 4 mul 1 add def
1200   /pgfcircx pgfstartx 5 index pgfdiffx pgfdomb div mul add def
1201   /pgfcircy pgfstarty 5 index pgfdiffy pgfdomb div mul add def
1202   /pgfcircxe pgfstartx 4 index pgfdiffx pgfdomb div mul add def
1203   /pgfcircye pgfstarty 4 index pgfdiffy pgfdomb div mul add def
1204   /pgfxstep pgfcircxe pgfcircx sub pgfdiff div def
1205   /pgfystep pgfcircye pgfcircy sub pgfdiff div def
1206   dup 2 index sub pgfdiff div % put gray-step on stack
1207   2 index 5 index
1208   pgfdiff {
1209     1 index setgray % Set color
1210     pgfcircx pgfcircy 2 index 0 360 arc closepath
1211     stroke
1212     exch 2 index add % gray += incgray
1213     exch .25 sub % x += 0.25
1214     /pgfcircx pgfcircx pgfxstep add def
1215     /pgfcircy pgfcircy pgfystep add def
1216   } repeat
1217   mark 6 1 roll cleartomark exch pop % leave only start x on stack
1218   }bind def}
1219 }
1220 </common-ps-driver>

```

Change History

v1.0		Support dvipdfm driver	24	
	General: First public release	1	Support dvisvgm driver	15
v1.1		Support PostScript® drivers	26	
	General: Fix typo for rgb option . . .	16	Support tex4ht driver	15