

The pgf-cmykshadings package

David Purton*

2019/11/05 v1.2

Abstract

Note: This package is now deprecated. Support for CMYK and grayscale shadings was added to `pgf` in version 3.1.3. Attempting to load it with recent versions of `pgf` only changes the default shading colour model with the `xcolor` `natural` colour model to CMYK. This documentation applies to versions of `pgf` prior to 3.1.3.

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	18
	Change History	33

*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 `pgfcoreshade.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

`pgf-cmykshadings` supports the following package options:

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

`rgb` to use 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`, `cmk`, 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 `cmk` 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`, `cmk`, 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 cmk` 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
}
```

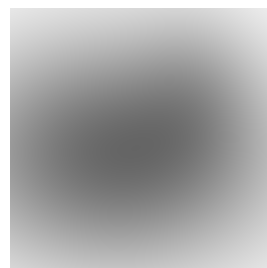
xcolor cmyk model:



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* and optional argument are created at the time they are actually used (using `\pgfuseshading`).

`\pgfdeclarehorizontalshading` `\pgfdeclarehorizontalshading[\color list]{\shading name}{\shading height}{\color specification}`

Declare a horizontal shading.

`\pgfdeclareverticalshading` `\pgfdeclareverticalshading[\color list]{\shading name}{\shading width}{\color specification}`

Declare a vertical shading.

`\pgfdeclareradialshading` `\pgfdeclareradialshading[\color list]{\shading name}{\center point}{\color specification}`

Declare a radial shading.

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

Declare a functional shading.

4.2.2 Using shadings

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

`\pgfuseshading` `\pgfuseshading{\shading name}`

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.

`\pgfshadepath` `\pgfshadepath{\shading name}{\angle}`

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

`\pgfadditionalshadetransform` `\pgfadditionalshadetransform{\transformation}`

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.

`\pgfshadecolor torgb` `\pgfshadecolor torgb { $\langle color name \rangle$ }{ $\langle macro \rangle$ }`

This command takes $\langle color name \rangle$ 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 $\langle macro \rangle$. This macro can then be used inside the $\langle type 4 function \rangle$ argument for `\pgfdeclarefunctionalshading`.

In addition, three macros suffixed with `red`, `green` and `blue` are defined, which store the individual components of $\langle color name \rangle$. These can also be used in the $\langle type 4 function \rangle$ argument.

`\pgfshadecolor tocmyk` `\pgfshadecolor tocmyk { $\langle color name \rangle$ }{ $\langle macro \rangle$ }`

This command is analogous to `\pgfshadecolor torgb` , but stores the colour's cyan/magenta/yellow/black components. Four macros suffixed with `cyan`, `magenta`, `yellow`, and `black` are also defined.

`\pgfshadecolor tgray` `\pgfshadecolor tgray { $\langle color name \rangle$ }{ $\langle macro \rangle$ }`

This command is analogous to `\pgfshadecolor torgb` , but stores the colour's gray component. Although it's not needed, for consistency a second macro suffixed with `gray` is also defined.

`\pgffuncshading rgbtocmyk` `\pgffuncshading rgbtocmyk`

Within the $\langle type 4 function \rangle$ argument of `\pgfdeclarefunctionalshading`, this command can be used to convert the top 3 elements on the stack from RGB to CMYK. In combination with the `\ifpgfshadingmodel cmyk` conditional this macro can be used to make functional shading declarations more portable across colour models.

`\pgffuncshading rgbtogray` `\pgffuncshading rgbtogray`

Within the $\langle type 4 function \rangle$ argument of `\pgfdeclarefunctionalshading`, this command can be used to convert the top 3 elements on the stack from RGB to grayscale. In combination with the `\ifpgfshadingmodel gray` conditional this macro can be used to make functional shading declarations more portable across colour models.

`\pgffuncshading cmyktorgb` `\pgffuncshading cmyktorgb`

Within the $\langle type 4 function \rangle$ argument of `\pgfdeclarefunctionalshading`, this command can be used to convert the top 4 elements on the stack from CMYK to RGB. In combination with the `\ifpgfshadingmodel rgb` conditional this macro can be used to make functional shading declarations more portable across colour models.

`\pgffuncshading cmyktogray` `\pgffuncshading cmyktogray`

Within the $\langle type 4 function \rangle$ argument of `\pgfdeclarefunctionalshading`, this command can be used to convert the top 4 elements on the stack from CMYK to grayscale. In combination with the `\ifpgfshadingmodel gray` conditional this macro can be used to make functional shading declarations more portable across colour models.

`\pgffuncshading graytorgb` `\pgffuncshading graytorgb`

Within the $\langle type 4 function \rangle$ argument of `\pgfdeclarefunctionalshading`, this command can be used to convert the top element on the stack from grayscale to RGB. In combination with the `\ifpgfshadingmodel rgb` 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 [2019/11/05
4   CMYK and grayscale shadings support for PGF (DCP)]
5 \RequirePackage{pgf}
6
7 Set colour model to CMYK by default if version of pgf is at least 3.1.3 then \endinput.
8 \ifpackagelater{pgf}{2019/05/14}{%
9   \PackageWarning{pgf-cmykshadings}{%
10     Package 'pgf-cmykshadings' is now deprecated.}%
11   \newif\ifpgfcmykshadingdefault
12   \DeclareOption{cmyk}{%
13     \pgfcmykshadingdefaulttrue
14   }
15   \DeclareOption{rgb}{%
16     \pgfcmykshadingdefaultfalse
17   }
18   \ExecuteOptions{cmyk}
19   \ProcessOptions\relax
20   \def\pgf@setup@shading@model{%
21     \pgfshadingmodelrgbtrue
22   }
23 }
```

```

20 \pgfshadingmodelcmykfalse
21 \pgfshadingmodelgrayfalse
22 \XC@sdef\pgf@mod@test{\XC@tgt@mod{natural}}%
23 \def\pgf@shading@device{/DeviceRGB}%
24 \def\pgf@shading@ps@device{setrgbcolor}%
25 \def\pgf@shading@functional@range{0 1 0 1 0 1}%
26 \def\pgf@shading@model{rgb}%
27 \ifx\pgf@mod@test\XC@mod@natural
28 \ifpgfcmykshadingdefault
29 \def\pgf@shading@functional@range{0 1 0 1 0 1 0 1}%
30 \def\pgf@shading@device{/DeviceCMYK}%
31 \def\pgf@shading@ps@device{setcmykcolor}%
32 \def\pgf@shading@model{cmyk}%
33 \pgfshadingmodelrgbfalse
34 \pgfshadingmodelcmyktrue
35 \else
36 \def\pgf@shading@functional@range{0 1 0 1 0 1}%
37 \def\pgf@shading@device{/DeviceRGB}%
38 \def\pgf@shading@ps@device{setrgbcolor}%
39 \def\pgf@shading@model{rgb}%
40 \fi
41 \fi
42 \ifx\pgf@mod@test\XC@mod@cmyk
43 \def\pgf@shading@device{/DeviceCMYK}%
44 \def\pgf@shading@ps@device{setcmykcolor}%
45 \def\pgf@shading@functional@range{0 1 0 1 0 1 0 1}%
46 \def\pgf@shading@model{cmyk}%
47 \pgfshadingmodelrgbfalse
48 \pgfshadingmodelcmyktrue
49 \fi
50 \ifx\pgf@mod@test\XC@mod@gray
51 \def\pgf@shading@device{/DeviceGray}%
52 \def\pgf@shading@ps@device{setgray}%
53 \def\pgf@shading@functional@range{0 1}%
54 \def\pgf@shading@model{gray}%
55 \pgfshadingmodelrgbfalse
56 \pgfshadingmodelgraytrue
57 \fi
58 \edef\pgf@sys@driver@dvisvgm{pgfsys-dvisvgm.def}%
59 \ifx\pgfsysdriver\pgf@sys@driver@dvisvgm
60 \def\pgf@shading@model{rgb}%
61 \fi
62 \edef\pgf@sys@driver@texforht{pgfsys-tex4ht.def}%
63 \ifx\pgfsysdriver\pgf@sys@driver@texforht
64 \def\pgf@shading@model{rgb}%
65 \fi
66 }%
67 \endinput}{-}

```

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

```

68 \def\pgf@parsefunc#1{%
69 \edef\temp{#1}%
70 \expandafter\pgf@convertstring\temp%
71 \edef\temp{\pgf@conv}%

```



```
72 \expandafter\pgf@@parsefunc\temp}
```

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

```
73 \def\pgf@@parsefunc#1{%
74 \let\pgf@bounds=\pgfutil@empty%
75 \let\pgf@funcs=\pgfutil@empty%
76 \let\pgf@psfuncs=\pgfutil@empty%
77 \let\pgf@encode=\pgfutil@empty%
78 \let\pgf@sys@shading@ranges=\pgfutil@empty%
79 \pgf@sys@shading@range@num=0\relax%
80 \csname pgf@parsefirst\pgf@shading@model\endcsname[#1; ]%
81 \csname pgf@parselastdom\pgf@shading@model\endcsname[#1; ]%
82 \csname pgf@parsemid\pgf@shading@model\endcsname[#1; ]%
83 \ifx\pgf@bounds\pgfutil@empty%
84 \edef\pgf@pdfparseddomain{0 1}%
85 \edef\pgf@pdfparsedfunction{\pgf@singlefunc\space}%
86 \else%
87 \edef\pgf@pdfparseddomain{\pgf@doma\space\pgf@domb}%
88 \edef\pgf@pdfparsedfunction{%
89 << /FunctionType 3 /Domain [\pgf@doma\space\pgf@domb] /Functions
90 [\pgf@funcs\space] /Bounds [\pgf@bounds] /Encode [0 1 \pgf@encode]
91 >> }% <<
92 \fi%
93 \xdef\pgf@psfuncs{\pgf@psfuncs}%
94 }
```

Define RGB parsing macros.

```
95 \let\pgf@parsefirstrgb\pgf@parsefirst
96 \let\pgf@parselastdomrgb\pgf@parselastdom
97 \let\pgf@parsemidrgb\pgf@parsemid
98 \let\pgf@parserestrgb\pgf@parserest
```

Define new CMYK parsing macros.

```
99 \def\pgf@parsefirstcmyk[cmyk(#1)=(#2,#3,#4,#5)#6]{%
100 \pgfmathsetlength\pgf@x{#1}%
101 \edef\pgf@sys@shading@start@pos{\the\pgf@x}%
102 \pgf@sys@bp@correct\pgf@x%
103 \edef\pgf@doma{\pgf@sys@tonumber{\pgf@x}}%
104 \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
105 \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
106 \edef\pgf@sys@shading@start@cmyk{\pgf@sys@cmyk}%
107 \let\pgf@sys@prevcolor=\pgf@sys@shading@start@cmyk%
108 \let\pgf@sys@prevpos=\pgf@sys@shading@start@pos%
109 \edef\pgf@prevcolor{\pgf@cmyk}%
110 \edef\pgf@firstcolor{\pgf@cmyk}}
111 \def\pgf@parselastdomcmyk[cmyk(#1)=(#2,#3,#4,#5); {%
112 \pgfutil@ifnextchar}{%
113 \pgfmathsetlength\pgf@x{#1}%
114 \edef\pgf@sys@shading@end@pos{\the\pgf@x}%
115 \pgf@max=\pgf@x\relax%
116 \pgf@sys@bp@correct\pgf@x%
117 \edef\pgf@domb{\pgf@sys@tonumber{\pgf@x}}%
118 \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
119 \edef\pgf@sys@shading@end@cmyk{\pgf@sys@cmyk}%
```

```

120 \pgfutil@gobble}{\pgf@parselastdomcmyk[]}
121 \def\pgf@parsemidcmyk[cmyk(#1)=(#2,#3,#4,#5)]; {\pgf@parserestcmyk[]}
122 \def\pgf@parserestcmyk[cmyk(#1)=(#2,#3,#4,#5)]; {%
123 \advance\pgf@sys@shading@range@num by1\relax%
124 \pgfutil@ifnextchar}{%
125 \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
126 \edef\pgf@singlefunc{\space%
127 << /FunctionType 2 /Domain [0 1] /CO
128 [\pgf@prevcolor] /C1 [\pgf@cmyk] /N 1 >> }% <<
129 \edef\pgf@funcs{\pgf@funcs\space%
130 << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /CO
131 [\pgf@prevcolor] /C1 [\pgf@cmyk] /N 1 >> }% <<
132 \edef\pgf@psfuncs{\pgf@prevx\space
133 \pgf@cmyk\space \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
134 \pgfmathsetlength\pgf@x{#1}%
135 \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
136 {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@cmyk}}}%
137 \edef\pgf@sys@prevpos{\the\pgf@x}%
138 \let\pgf@sys@prevcolor=\pgf@sys@cmyk%
139 \pgfutil@gobble}{%
140 \pgfmathsetlength\pgf@x{#1}%
141 \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
142 \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
143 {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@cmyk}}}%
144 \edef\pgf@sys@prevpos{\the\pgf@x}%
145 \let\pgf@sys@prevcolor=\pgf@sys@cmyk%
146 \edef\pgf@psfuncs{\pgf@prevx\space \pgf@cmyk\space
147 \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
148 \pgf@sys@bp@correct\pgf@x%
149 \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
150 \edef\pgf@bounds{\pgf@bounds\space\pgf@sys@tonumber{\pgf@x}}%
151 \edef\pgf@encode{\pgf@encode\space0 1}%
152 \edef\pgf@singlefunc{\space%
153 << /FunctionType 2 /Domain [0 1] /CO
154 [\pgf@prevcolor] /C1 [\pgf@cmyk] /N 1 >> }% <<
155 \edef\pgf@funcs{\pgf@funcs\space%
156 << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /CO
157 [\pgf@prevcolor] /C1 [\pgf@cmyk] /N 1 >> }% <<
158 \edef\pgf@prevcolor{\pgf@cmyk}%
159 \pgf@parserestcmyk[]}
160 \def\pgf@getcmyktuplewithmixin#1#2#3#4{%
161 \pgfutil@definecolor{pgfshadetemp}{cmyk}{#1,#2,#3,#4}%
162 \pgfutil@ifundefined{applycolormixins}{\applycolormixins{pgfshadetemp}}%
163 \pgfutil@extractcolorspec{pgfshadetemp}{\pgf@tempcolor}%
164 \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{cmyk}{\pgf@cmykcolor}%
165 \expandafter\pgf@getcmyk@@\pgf@cmykcolor!}
166 \def\pgf@getcmyk@@#1,#2,#3,#4!{%
167 \def\pgf@cmyk{#1 #2 #3 #4}%
168 \def\pgf@sys@cmyk{{#1}{#2}{#3}{#4}}%
169 }

```

Define new grayscale parsing macros.

```

170 \def\pgf@parsefirstgray[gray(#1)=(#2)#3]{%
171 \pgfmathsetlength\pgf@x{#1}%
172 \edef\pgf@sys@shading@start@pos{\the\pgf@x}%

```

```

173 \pgf@sys@bp@correct\pgf@x%
174 \edef\pgf@doma{\pgf@sys@tonumber{\pgf@x}}%
175 \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
176 \pgf@getgraytuplewithmixin{#2}%
177 \edef\pgf@sys@shading@start@gray{\pgf@sys@gray}%
178 \let\pgf@sys@prevcolor=\pgf@sys@shading@start@gray%
179 \let\pgf@sys@prevpos=\pgf@sys@shading@start@pos%
180 \edef\pgf@prevcolor{\pgf@gray}%
181 \edef\pgf@firstcolor{\pgf@gray}}
182 \def\pgf@parselastdomgray[gray(#1)=(#2); {%
183 \pgfutil@ifnextchar}{%
184 \pgfmathsetlength\pgf@x{#1}%
185 \edef\pgf@sys@shading@end@pos{\the\pgf@x}%
186 \pgf@max=\pgf@x\relax%
187 \pgf@sys@bp@correct\pgf@x%
188 \edef\pgf@domb{\pgf@sys@tonumber{\pgf@x}}%
189 \pgf@getgraytuplewithmixin{#2}%
190 \edef\pgf@sys@shading@end@gray{\pgf@sys@gray}%
191 \pgfutil@gobble}{\pgf@parselastdomgray[]}}
192 \def\pgf@parsemidgray[gray(#1)=(#2); {\pgf@parserestgray[]
193 \def\pgf@parserestgray[gray(#1)=(#2); {%
194 \advance\pgf@sys@shading@range@num by1\relax%
195 \pgfutil@ifnextchar}{%
196 \pgf@getgraytuplewithmixin{#2}%
197 \edef\pgf@singlefunc{\space%
198 << /FunctionType 2 /Domain [0 1] /CO
199 [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
200 \edef\pgf@funcs{\pgf@funcs\space%
201 << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /CO
202 [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
203 \edef\pgf@psfuncs{\pgf@prevx\space \pgf@gray\space
204 \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
205 \pgfmathsetlength\pgf@x{#1}%
206 \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
207 {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@gray}}}%
208 \edef\pgf@sys@prevpos{\the\pgf@x}%
209 \let\pgf@sys@prevcolor=\pgf@sys@gray%
210 \pgfutil@gobble}{%
211 \pgfmathsetlength\pgf@x{#1}%
212 \pgf@getgraytuplewithmixin{#2}%
213 \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
214 {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@gray}}}%
215 \edef\pgf@sys@prevpos{\the\pgf@x}%
216 \let\pgf@sys@prevcolor=\pgf@sys@gray%
217 \edef\pgf@psfuncs{\pgf@prevx\space \pgf@gray\space
218 \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
219 \pgf@sys@bp@correct\pgf@x%
220 \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
221 \edef\pgf@bounds{\pgf@bounds\space\pgf@sys@tonumber{\pgf@x}}%
222 \edef\pgf@encode{\pgf@encode\space0 1}%
223 \edef\pgf@singlefunc{\space%
224 << /FunctionType 2 /Domain [0 1] /CO
225 [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
226 \edef\pgf@funcs{\pgf@funcs\space%

```

```

227     << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /CO
228     [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
229     \edef\pgf@prevcolor{\pgf@gray}%
230     \pgf@parserestgray[]}
231 \def\pgf@getgraytuplewithmixin#1{%
232 \pgfutil@definecolor{pgfshadetemp}{gray}{#1}%
233 \pgfutil@ifundefined{applycolormixins}{\applycolormixins{pgfshadetemp}}%
234 \pgfutil@extractcolorspec{pgfshadetemp}{\pgf@tempcolor}%
235 \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{gray}{\pgf@graycolor}%
236 \expandafter\pgf@getgray@\pgf@graycolor!}
237 \def\pgf@getgray@#1!{%
238 \def\pgf@gray{#1}%
239 \def\pgf@sys@gray{#1}%
240 }

```

Define new colour space agnostic colour specification parsing macros. This includes parsing CMYK colour specifications (i.e., $\text{color}(\langle position \rangle) = (\langle C \rangle, \langle M \rangle, \langle Y \rangle, \langle K \rangle)$).

```

241 \def\pgf@convertstring#1{%
242 \def\pgf@conv{}}
243 \pgf@convert#1}%
244 }
245 \def\pgf@convert{%
246 \pgfutil@ifnextchar]{\pgfutil@gobble}%done!
247 {%
248 \pgfutil@ifnextchar;{\pgf@grabsemicolor}%
249 {%
250 \pgfutil@ifnextchar c{\pgf@gobblec}%
251 {%
252 \pgfutil@ifnextchar g{\pgf@grabgray}%
253 {%
254 \pgfutil@ifnextchar o{\pgf@grabcolor}%
255 {%
256 \pgfutil@ifnextchar m{\pgf@grabcmyk}%
257 {%
258 \pgfutil@ifnextchar r{\pgf@grabrgb}%
259 {\pgferror{Illformed shading
260 specification}\pgf@convert}%
261 }%
262 }%
263 }%
264 }%
265 }%
266 }%
267 }
268 \def\pgf@grabsemicolor;{%
269 \edef\pgf@conv{\pgf@conv; }\pgf@convert}
270 \def\pgf@gobblec c{\pgf@convert}
271 \def\pgf@savecolor#1{%
272 \pgfutil@extractcolorspec{pgf@tempcol}{\pgf@tempcolor}%
273 \expandafter\pgfutil@convertcolorspec\pgf@tempcolor
274 {\pgf@shading@model}{\pgf@color}%
275 \expandafter\pgf@convget@\expandafter{\pgf@color}{#1}%
276 }
277 \def\pgf@grabrgb rgb(#1)=(#2,#3,#4){%

```

```

278 \pgfutil@definecolor{pgf@tempcol}{rgb}{#2,#3,#4}%
279 \pgf@savecolor{#1}%
280 }
281 \def\pgf@grabcmk myk(#1)=(#2,#3,#4,#5){%
282 \pgfutil@definecolor{pgf@tempcol}{cmyk}{#2,#3,#4,#5}%
283 \pgf@savecolor{#1}%
284 }
285 \def\pgf@grabgray gray(#1)=(#2){%
286 \pgfutil@definecolor{pgf@tempcol}{gray}{#2}%
287 \pgf@savecolor{#1}%
288 }
289 \def\pgf@grabcolor olor(#1)=(#2){%
290 \pgfutil@colorlet{pgf@tempcol}{#2}%
291 \pgf@savecolor{#1}%
292 }
293 \def\pgf@convget#1#2{%
294 \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`.

```

295 \newdimen\pgf@xd
296 \def\pgfshadecolortocmyk#1#2{%
297 \pgfutil@colorlet{pgf@tempcol}{#1}%
298 \pgfutil@extractcolorspec{pgf@tempcol}{\pgf@tempcolor}%
299 \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{cmyk}{\pgf@cmykcolor}%
300 \expandafter\pgfshading@cmyk\pgf@cmykcolor\relax%
301 \edef#2{\pgf@sys@tonumber{\pgf@xa}\space\pgf@sys@tonumber{\pgf@xb}\space
302 \pgf@sys@tonumber{\pgf@xc}\space\pgf@sys@tonumber{\pgf@xd}\space}%
303 \c@pgf@counta\escapechar%
304 \escapechar-1\relax%
305 \expandafter\edef\csname\string#2cyan\endcsname{%
306 \pgf@sys@tonumber{\pgf@xa}\space}%
307 \expandafter\edef\csname\string#2magenta\endcsname{%
308 \pgf@sys@tonumber{\pgf@xb}\space}%
309 \expandafter\edef\csname\string#2yellow\endcsname{%
310 \pgf@sys@tonumber{\pgf@xc}\space}%
311 \expandafter\edef\csname\string#2black\endcsname{%
312 \pgf@sys@tonumber{\pgf@xd}\space}%
313 \escapechar\c@pgf@counta
314 }
315 \def\pgfshading@cmyk#1,#2,#3,#4\relax{%
316 \pgf@xa=#1pt%
317 \pgf@xb=#2pt%
318 \pgf@xc=#3pt%
319 \pgf@xd=#4pt%
320 }

```

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

```

321 \def\pgfshadecolortogley#1#2{%
322 \pgfutil@colorlet{pgf@tempcol}{#1}%
323 \pgfutil@extractcolorspec{pgf@tempcol}{\pgf@tempcolor}%
324 \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{gray}{\pgf@graycolor}%
325 \expandafter\pgfshading@gray\pgf@graycolor\relax
326 \edef#2{\pgf@sys@tonumber{\pgf@xa}\space}%

```

```

327 \c@pgf@counta\escapechar
328 \escapechar-1\relax
329 \expandafter\edef\csname\string#2gray\endcsname{%
330   \pgf@sys@tonumber{\pgf@xa}\space}%
331 \escapechar\c@pgf@counta
332 }
333 \def\pgfshading@gray#1\relax{%
334   \pgf@xa=#1pt%
335 }

```

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

```

336 \def\pgfdeclarehorizontalshading{%
337   \pgf@setup@model
338   \pgfutil@ifnextchar{%
339     \pgfdeclarehorizontalshading{\pgfdeclarehorizontalshading[]}}
340 \def\pgfdeclareverticalshading{%
341   \pgf@setup@model
342   \pgfutil@ifnextchar{%
343     \pgfdeclareverticalshading{\pgfdeclareverticalshading[]}}
344 \def\pgfdecleareradialshading{%
345   \pgf@setup@model
346   \pgfutil@ifnextchar{%
347     \pgfdecleareradialshading{\pgfdecleareradialshading[]}}
348 \def\pgfdeclarefunctionalshading{%
349   \pgf@setup@model
350   \pgfutil@ifnextchar{%
351     \pgfdeclarefunctionalshading{\pgfdeclarefunctionalshading[]}}

```

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

```

352 \def\pgfuseshading#1{%
353   \edef\pgf@shadingname{@pgfshading#1}%
354   \edef\pgf@shadingsavedmodel{@pgfshading#1@model}%
355   \pgf@tryextensions{\pgf@shadingname}{\pgf@alternateextension}%
356   \expandafter\pgfutil@ifundefined\expandafter{\pgf@shadingname}%
357   {\pgferror{Undefined shading "#1"}}%
358   {%
359     \pgf@setup@model
360     \pgfutil@globalcolorsfalse
361     \def\pgf@shade@adds{}%
362     \pgfutil@ifundefined{pgf@deps\pgf@shadingname}%
363     {}%
364     {%
365       \edef\@list{\csname pgf@deps\pgf@shadingname\endcsname}%
366       \pgfutil@for\@temp:=\@list\do{%
367         {%
368           \pgfutil@ifundefined{applycolormixins}%
369           {\@temp}}%
370         \pgfutil@extractcolorspec{\@temp}{\pgf@tempcolor}%
371         \expandafter\pgfutil@ifundefined\expandafter{%
372           \pgf@shadingsavedmodel}%
373         {\expandafter\pgfutil@convertcolorspec\pgf@tempcolor%
374           \pgf@shading@model}{\pgf@color}}%

```

```

376         {\expandafter\pgfutil@convertcolorspec\pgf@tempcolor{%
377          \pgf@shadingsavedmodel}{\pgf@color}}}%
378     \xdef\pgf@shade@adds{\pgf@shade@adds,\pgf@color}%
379     }%
380 }%
381 }%
382 \expandafter\pgf@strip@shadename\pgf@shadingname!!%
383 \pgfutil@ifundefined{@pgfshading\pgf@basename\pgf@shade@adds!}%
384 {%
385     {%
386         \expandafter\def\expandafter\@temp\expandafter{%
387          \csname pgf@func\pgf@shadingname\endcsname}%
388         \edef\@args{\pgf@basename\pgf@shade@adds}}}%
389     \expandafter\expandafter\expandafter\def
390     \expandafter\expandafter\expandafter\@@args
391     \expandafter\expandafter\expandafter{%
392         \csname pgf@args\pgf@shadingname\endcsname}%
393     \expandafter\expandafter\expandafter\@temp
394     \expandafter\@args\@@args
395     }%
396 }%
397 {}%
398 \pgf@invoke shading{%
399     \csname @pgfshading\pgf@basename\pgf@shade@adds!\endcsname}%
400 }%
401 }%
402 }

```

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

```

403 \newif\ifpgfshadingmodelrgb
404 \newif\ifpgfshadingmodelcmyk
405 \newif\ifpgfshadingmodelgray

```

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

```

406 \def\pgf@setup@model{%
407     \pgfshadingmodelrgbtrue
408     \pgfshadingmodelcmykfalse
409     \pgfshadingmodelgrayfalse
410     \XC@sdef\pgf@mod@test{\XC@tgt@mod{natural}}%
411     \def\pgf@shading@functional@range{0 1 0 1 0 1}%
412     \def\pgf@shading@device{/DeviceRGB}%
413     \def\pgf@shading@ps@device{setrgbcolor}%
414     \def\pgf@shading@model{rgb}%
415     \ifx\pgf@mod@test\XC@mod@natural
416         \ifpgfcmykshadingdefault
417             \def\pgf@shading@functional@range{0 1 0 1 0 1 0 1}%
418             \def\pgf@shading@device{/DeviceCMYK}%
419             \def\pgf@shading@ps@device{setcmykcolor}%
420             \def\pgf@shading@model{cmyk}%
421             \pgfshadingmodelrgbfalse
422             \pgfshadingmodelcmyktrue
423         \else
424             \def\pgf@shading@functional@range{0 1 0 1 0 1}%
425             \def\pgf@shading@device{/DeviceRGB}%

```

```

426     \def\pgf@shading@ps@device{setrgbcolor}%
427     \def\pgf@shading@model{rgb}%
428     \fi
429   \fi
430   \ifx\pgf@mod@test\XC@mod@cmyk
431     \def\pgf@shading@functional@range{0 1 0 1 0 1 0 1}%
432     \def\pgf@shading@device{/DeviceCMYK}%
433     \def\pgf@shading@ps@device{setcmykcolor}%
434     \def\pgf@shading@model{cmyk}%
435     \pgfshadingmodelrgbfalse
436     \pgfshadingmodelcmyktrue
437   \fi
438   \ifx\pgf@mod@test\XC@mod@gray
439     \def\pgf@shading@functional@range{0 1}%
440     \def\pgf@shading@device{/DeviceGray}%
441     \def\pgf@shading@ps@device{setgray}%
442     \def\pgf@shading@model{gray}%
443     \pgfshadingmodelrgbfalse
444     \pgfshadingmodelgraytrue
445   \fi
446   \edef\pgf@sys@driver@dvisvgm{pgfsys-dvisvgm.def}%
447   \ifx\pgfsysdriver\pgf@sys@driver@dvisvgm
448     \def\pgf@shading@model{rgb}%
449   \fi
450   \edef\pgf@sys@driver@texforht{pgfsys-tex4ht.def}%
451   \ifx\pgfsysdriver\pgf@sys@driver@texforht
452     \def\pgf@shading@model{rgb}%
453   \fi
454 }

```

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

```

455 \def\pgffuncshadingrgbtocmyk{%
456   1.0 exch sub 3 1 roll
457   1.0 exch sub 3 1 roll
458   1.0 exch sub 3 1 roll
459   3 copy
460   2 copy gt { exch } if pop
461   2 copy gt { exch } if pop
462   dup 3 1 roll sub
463   0.0 2 copy lt { exch } if pop
464   1.0 2 copy gt { exch } if pop
465   4 1 roll
466   dup 3 1 roll sub
467   0.0 2 copy lt { exch } if pop
468   1.0 2 copy gt { exch } if pop
469   4 1 roll
470   dup 3 1 roll sub
471   0.0 2 copy lt { exch } if pop
472   1.0 2 copy gt { exch } if pop
473   4 1 roll
474 }
475 \def\pgffuncshadingrgbtogray{%
476   0.11 mul exch 0.59 mul add exch 0.3 mul add

```



```

477 }
478 \def\pgffuncshadingcmyktorgb{%
479   % covert to CMY
480   dup 3 1 roll add
481   1.0 2 copy gt { exch } if pop
482   4 1 roll
483   dup 3 1 roll add
484   1.0 2 copy gt { exch } if pop
485   4 1 roll
486   add
487   1.0 2 copy gt { exch } if pop
488   3 1 roll
489   % covert to RGB
490   1.0 exch sub
491   3 1 roll
492   1.0 exch sub
493   3 1 roll
494   1.0 exch sub
495   3 1 roll
496 }
497 \def\pgffuncshadingcmyktogray{%
498   exch 0.11 mul add exch 0.59 mul add exch 0.3 mul add
499   1.0 2 copy gt { exch } if pop
500   1.0 exch sub
501 }
502 \def\pgffuncshadinggraytorgb{%
503   dup dup
504 }
505 \def\pgffuncshadinggraytocmyk{%
506   0.0 0.0 0.0
507   4 3 roll
508 }

```

Load the correct driver file.

```

509 \def\pgfutilgetcmykshadingsdriver{%
510   \expandafter\pgfutil@getcmykshadingsdriver\pgfsysdriver[%
511 }
512 \def\pgfutil@getcmykshadingsdriver pgfsys-#1[{%
513   \edef\pgfsyscmykshadingsdriver{pgfsys-cmykshadings-#1}%
514 }
515 \pgfutilgetcmykshadingsdriver
516 \pgfutil@InputIfFileExists{\pgfsyscmykshadingsdriver}{-}{-}

```

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

```

517 \newif\ifpgfcmykshadingdefault
518 \DeclareOption{cmyk}{%
519   \pgfcmykshadingdefaulttrue
520 }
521 \DeclareOption{rgb}{%
522   \pgfcmykshadingdefaultfalse
523 }
524 \ExecuteOptions{cmyk}
525 \ProcessOptions\relax

```

526 </package>

5.2 Drivers

pdftex driver

```
527 <*pdftex-driver>
528 \ProvidesFile{pgfsys-cmykshadings-pdftex.def}%
529 [2018/10/24
530   CMYK and grayscale shadings support for PGF pdftex driver (DCP)]
531 \def\pgfsys@horishading#1#2#3{%
532   {%
533     \pgf@parsefunc{#3}%
534     \pgfmathparse{#2}%
535     \setbox\pgfutil@tempboxa=\hbox to\pgf@max{%
536       \vbox to\pgfmathresult pt{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
537     \pgf@process{\pgfpoint{\pgf@max}{#2}}%
538     \immediate\pdfxform resources {%
539       /Shading << /Sh << /ShadingType 2
540       /ColorSpace \pgf@shading@device\space
541       /Domain [\pgf@pdfparseddomain]
542       /Coords [\pgf@doma\space0 \pgf@domb\space0]
543       /Function \pgf@pdfparsedfunction
544       /Extend [false false] >> >>}\pgfutil@tempboxa% <<
545     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
546       \leavevmode\noexpand\pdfrefxform\the\pdflastxform}%
547     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
548       \pgf@shading@model}%
549   }%
550 }
551 \def\pgfsys@vertshading#1#2#3{%
552   {%
553     \pgf@parsefunc{#3}%
554     \pgfmathparse{#2}%
555     \setbox\pgfutil@tempboxa=\hbox to\pgfmathresult pt{%
556       \vbox to\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
557     \pgf@process{\pgfpoint{#2}{\pgf@max}}%
558     \immediate\pdfxform resources {%
559       /Shading << /Sh << /ShadingType 2
560       /ColorSpace \pgf@shading@device\space
561       /Domain [\pgf@pdfparseddomain]
562       /Coords [0 \pgf@doma\space0 \pgf@domb]
563       /Function \pgf@pdfparsedfunction
564       /Extend [false false] >> >>}\pgfutil@tempboxa% <<
565     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
566       \leavevmode\noexpand\pdfrefxform\the\pdflastxform}%
567     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
568       \pgf@shading@model}%
569   }%
570 }
571 \def\pgfsys@radialshading#1#2#3{%
572   {%
573     \pgf@parsefunc{#3}%
574     \setbox\pgfutil@tempboxa=\hbox to2\pgf@max%
```

```

575     \vbox to2\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
576 \pgf@process{#2}%
577 \pgf@xa=\pgf@x
578 \pgf@ya=\pgf@y
579 \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
580 \advance\pgf@xa by \pgf@x
581 \advance\pgf@ya by \pgf@y
582 \pgf@sys@bp@correct{\pgf@x}%
583 \pgf@sys@bp@correct{\pgf@y}%
584 \pgf@sys@bp@correct{\pgf@xa}%
585 \pgf@sys@bp@correct{\pgf@ya}%
586 \immediate\pdfxform resources {%
587 /Shading << /Sh << /ShadingType 3
588 /ColorSpace \pgf@shading@device\space
589 /Domain [\pgf@pdfparseddomain]
590 /Coords [\pgf@sys@tonumber{\pgf@xa}
591         \pgf@sys@tonumber{\pgf@ya}
592         \pgf@doma\space
593         \pgf@sys@tonumber{\pgf@x}
594         \pgf@sys@tonumber{\pgf@y}
595         \pgf@domb]
596 /Function \pgf@pdfparsedfunction
597 /Extend [true false] >> >>}\pgfutil@tempboxa% <<
598 \expandafter\xdef\csname @pgfshading#1!\endcsname{%
599 \leavevmode\noexpand\pdfrefxform\the\pdflastxform}%
600 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
601 \pgf@shading@model}%
602 }%
603 }
604 \def\pgfsys@functionalshading#1#2#3#4{%
605   {%
606     \pgf@process{#2}%
607     \pgf@xa=\pgf@x
608     \pgf@ya=\pgf@y
609     \pgf@process{#3}%
610     \pgf@xb=\pgf@x
611     \pgf@yb=\pgf@y
612     \advance\pgf@x by-\pgf@xa
613     \advance\pgf@y by-\pgf@ya
614     \setbox\pgfutil@tempboxa=\hbox to\pgf@x{%
615       \vbox to\pgf@y{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
616     \pgf@sys@bp@correct{\pgf@xa}%
617     \pgf@sys@bp@correct{\pgf@ya}%
618     \pgf@sys@bp@correct{\pgf@xb}%
619     \pgf@sys@bp@correct{\pgf@yb}%
620     \pgf@xc=-\pgf@xa
621     \pgf@yc=-\pgf@ya
622     % Now build the function
623     \pdfobj
624     stream
625     attr
626     {
627       /FunctionType 4
628       /Domain [\pgf@sys@tonumber{\pgf@xa}\space

```

```

629     \pgf@sys@tonumber{\pgf@xb}\space
630     \pgf@sys@tonumber{\pgf@ya}\space
631     \pgf@sys@tonumber{\pgf@yb}]
632     /Range [\pgf@shading@functional@range]
633 }
634 {{#4}}%
635 \edef\pgf@temp@num{\the\pdf@lastobj}%
636 \pdfxform resources {%
637     /Shading << /Sh << /ShadingType 1
638     /ColorSpace \pgf@shading@device\space
639     /Matrix [1 0 0 1 \pgf@sys@tonumber{\pgf@xc}\space
640             \pgf@sys@tonumber{\pgf@yc}]
641     /Domain [\pgf@sys@tonumber{\pgf@xa}\space
642             \pgf@sys@tonumber{\pgf@xb}\space
643             \pgf@sys@tonumber{\pgf@ya}\space
644             \pgf@sys@tonumber{\pgf@yb}]
645     /Function \pgf@temp@num\space 0 R
646     >> >>}\pgf@util@tempboxa% <<
647 \expandafter\xdef\csname @pgfshading#1!\endcsname{%
648     \leavevmode%
649     \noexpand\pdfrefxform\the\pdf@lastxform%
650     \noexpand\pdfrefobj\pgf@temp@num%
651 }%
652 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
653     \pgf@shading@model}%
654 }%
655 }
656 </pdf@tex-driver>

```

xetex driver

```

657 <*xetex-driver>
658 \ProvidesFile{pgfsys-cmykshadings-xetex.def}%
659 [2018/10/24
660     CMYK and grayscale shadings support for PGF xetex driver (DCP)]
661 \input pgfsys-cmykshadings-dvipdfmx.def
662 </xetex-driver>

```

luatex driver

```

663 <*luatex-driver>
664 \ProvidesFile{pgfsys-cmykshadings-luatex.def}%
665 [2018/10/24
666     CMYK and grayscale shadings support for PGF luatex driver (DCP)]
667 \def\pgfsys@horishading#1#2#3{%
668     {%
669         \pgf@parsefunc{#3}%
670         \pgf@mathparse{#2}%
671         \setbox\pgf@util@tempboxa=\hbox to\pgf@max{%
672             \vbox to\pgf@mathresult pt{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
673         \pgf@process{\pgf@point{\pgf@max}{#2}}%
674         \immediate\saveboxresource resources {%
675             /Shading << /Sh << /ShadingType 2

```

```

676     /ColorSpace \pgf@shading@device\space
677     /Domain [\pgf@pdfparseddomain]
678     /Coords [\pgf@doma\space0 \pgf@domb\space0]
679     /Function \pgf@pdfparsedfunction
680     /Extend [false false] >> >>}\pgfutil@tempboxa% <<
681     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
682     \leavevmode\noexpand\useboxresource\the\lastsavedboxresourceindex}%
683     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
684     \pgf@shading@model}%
685   }%
686 }
687 \def\pgfsys@vertshading#1#2#3{%
688   {%
689     \pgf@parsefunc{#3}%
690     \pgfmathparse{#2}%
691     \setbox\pgfutil@tempboxa=\hbox to\pgfmathresult pt{%
692       \vbox to\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
693     \pgf@process{\pgfpoint{#2}{\pgf@max}}%
694     \immediate\saveboxresource resources {%
695       /Shading << /Sh << /ShadingType 2
696       /ColorSpace \pgf@shading@device\space
697       /Domain [\pgf@pdfparseddomain]
698       /Coords [0 \pgf@doma\space0 \pgf@domb]
699       /Function \pgf@pdfparsedfunction
700       /Extend [false false] >> >>}\pgfutil@tempboxa% <<
701     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
702     \leavevmode\noexpand\useboxresource\the\lastsavedboxresourceindex}%
703     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
704     \pgf@shading@model}%
705   }%
706 }
707 \def\pgfsys@radialshading#1#2#3{%
708   {%
709     \pgf@parsefunc{#3}%
710     \setbox\pgfutil@tempboxa=\hbox to2\pgf@max{%
711       \vbox to2\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
712     \pgf@process{#2}%
713     \pgf@xa=\pgf@x
714     \pgf@ya=\pgf@y
715     \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
716     \advance\pgf@xa by \pgf@x
717     \advance\pgf@ya by \pgf@y
718     \pgf@sys@bp@correct{\pgf@x}%
719     \pgf@sys@bp@correct{\pgf@y}%
720     \pgf@sys@bp@correct{\pgf@xa}%
721     \pgf@sys@bp@correct{\pgf@ya}%
722     \immediate\saveboxresource resources {%
723       /Shading << /Sh << /ShadingType 3
724       /ColorSpace \pgf@shading@device\space
725       /Domain [\pgf@pdfparseddomain]
726       /Coords [\pgf@sys@tonumber{\pgf@xa}
727         \pgf@sys@tonumber{\pgf@ya}
728         \pgf@doma\space
729         \pgf@sys@tonumber{\pgf@x}

```

```

730     \pgf@sys@tonumber{\pgf@y}
731     \pgf@domb]
732     /Function \pgf@pdfparsedfunction
733     /Extend [true false] >> >>}\pgfutil@tempboxa% <<
734     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
735     \leavevmode\noexpand\useboxresource\the\lastsavedboxresourceindex}%
736     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
737     \pgf@shading@model}%
738 }%
739 }
740 \def\pgfsys@functionalshading#1#2#3#4{%
741   {%
742     \pgf@process{#2}%
743     \pgf@xa=\pgf@x
744     \pgf@ya=\pgf@y
745     \pgf@process{#3}%
746     \pgf@xb=\pgf@x
747     \pgf@yb=\pgf@y
748     \advance\pgf@x by-\pgf@xa
749     \advance\pgf@y by-\pgf@ya
750     \setbox\pgfutil@tempboxa=\hbox to\pgf@x{%
751     \vbox to\pgf@y{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
752     \pgf@sys@bp@correct{\pgf@xa}%
753     \pgf@sys@bp@correct{\pgf@ya}%
754     \pgf@sys@bp@correct{\pgf@xb}%
755     \pgf@sys@bp@correct{\pgf@yb}%
756     \pgf@xc=-\pgf@xa
757     \pgf@yc=-\pgf@ya
758     % Now build the function
759     \pdfextension obj
760     stream
761     attr
762     {
763       /FunctionType 4
764       /Domain [\pgf@sys@tonumber{\pgf@xa}\space
765       \pgf@sys@tonumber{\pgf@xb}\space
766       \pgf@sys@tonumber{\pgf@ya}\space
767       \pgf@sys@tonumber{\pgf@yb}]
768       /Range [\pgf@shading@functional@range]
769     }
770     {{#4}}%
771     \edef\pgf@temp@num{\the\numexpr\pdffeedback lastobj\relax}%
772     \saveboxresource resources {%
773       /Shading << /Sh << /ShadingType 1
774       /ColorSpace \pgf@shading@device\space
775       /Matrix [1 0 0 1 \pgf@sys@tonumber{\pgf@xc}\space
776       \pgf@sys@tonumber{\pgf@yc}]
777       /Domain [\pgf@sys@tonumber{\pgf@xa}\space
778       \pgf@sys@tonumber{\pgf@xb}\space
779       \pgf@sys@tonumber{\pgf@ya}\space
780       \pgf@sys@tonumber{\pgf@yb}]
781       /Function \pgf@temp@num\space 0 R
782       >> >>}\pgfutil@tempboxa% <<
783     \expandafter\xdef\csname @pgfshading#1!\endcsname{%

```

```

784     \leavevmode%
785     \noexpand\useboxresource\the\lastsavedboxresourceindex%
786     \noexpand\pdfextension refobj \pgf@temp@num%
787   }%
788   \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
789     \pgf@shading@model}%
790 }%
791 }
792 </luatex-driver>

```

dvipdfmx driver

```

793 <*dvipdfmx-driver>
794 \ProvidesFile{pgfsys-cmykshadings-dvipdfmx.def}%
795 [2018/10/24
796   CMYK and grayscale shadings support for PGF dvipdfmx driver (DCP)]
797 \def\pgfsys@horishading#1#2#3{%
798   {%
799     \pgf@parsefunc{#3}%
800     \pgfmathparse{#2}%
801     \pgf@process{\pgfpoint{\pgf@max}{#2}}%
802     \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
803       \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
804         width \the\pgf@max\space height \pgfmathresult pt}}%
805       \special{pdf:put @resources
806       <<
807         /Shading << /Sh << /ShadingType 2
808         /ColorSpace \pgf@shading@device\space
809         /Domain [\pgf@pdfparseddomain]
810         /Coords [\pgf@doma\space0 \pgf@domb\space0]
811         /Function \pgf@pdfparsedfunction
812         /Extend [false false] >> >>
813       >>}}%
814     \pgfsys@invoke{/Sh sh}%
815     \special{pdf:exobj}}}\@tempa% <<
816     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
817       \hbox to\the\pgf@max{\vbox to\pgfmathresult pt{%
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@vertshading#1#2#3{%
825   {%
826     \pgf@parsefunc{#3}%
827     \pgfmathparse{#2}%
828     \pgf@process{\pgfpoint{\pgf@max}{#2}}%
829     \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
830       \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
831         width \pgfmathresult pt\space height \the\pgf@max}%
832       \special{pdf:put @resources
833       <<
834         /Shading << /Sh << /ShadingType 2

```

```

835     /ColorSpace \pgf@shading@device\space
836     /Domain [\pgf@pdfparseddomain]
837     /Coords [0 \pgf@doma\space0 \pgf@domb]
838     /Function \pgf@pdfparsedfunction
839     /Extend [false false] >> >>
840     >>}%
841     \pgfsys@invoke{/Sh sh}%
842     \special{pdf:exobj}}\@tempa% <<
843     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
844     \hbox to\pgfmathresult pt{\vbox to\the\pgf@max{%
845     \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
846     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
847     \pgf@shading@model}%
848     }%
849     \global\advance\pgfsys@objnum\@ne
850 }
851 \def\pgfsys@radialshading#1#2#3{%
852   {%
853     \pgf@parsefunc{#3}%
854     \pgf@process{#2}%
855     \pgf@xa=\pgf@x
856     \pgf@ya=\pgf@y
857     \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
858     \advance\pgf@xa by \pgf@x
859     \advance\pgf@ya by \pgf@y
860     \pgf@sys@bp@correct{\pgf@x}%
861     \pgf@sys@bp@correct{\pgf@y}%
862     \pgf@sys@bp@correct{\pgf@xa}%
863     \pgf@sys@bp@correct{\pgf@ya}%
864     \pgfutil@tempdima=2\pgf@max
865     \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
866     \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
867     width \the\pgfutil@tempdima\space height \the\pgfutil@tempdima}%
868     \special{pdf:put @resources
869     <<
870     /Shading << /Sh << /ShadingType 3
871     /ColorSpace \pgf@shading@device\space
872     /Domain [\pgf@pdfparseddomain]
873     /Coords [\pgf@sys@tonumber{\pgf@xa}
874     \pgf@sys@tonumber{\pgf@ya}
875     \pgf@doma\space \pgf@sys@tonumber{\pgf@x}
876     \pgf@sys@tonumber{\pgf@y}
877     \pgf@domb]
878     /Function \pgf@pdfparsedfunction
879     /Extend [true false] >> >>
880     >>}%
881     \pgfsys@invoke{/Sh sh}%
882     \special{pdf:exobj}}\@tempa% <<
883     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
884     \hbox to\the\pgfutil@tempdima{\vbox to\the\pgfutil@tempdima{%
885     \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
886     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
887     \pgf@shading@model}%
888     }%

```



```

889 \global\advance\pgfsys@objnum\@ne
890 }
891 \def\pgfsys@functionalshading#1#2#3#4{%
892   {%
893     \pgf@process{#2}%
894     \pgf@xa=\pgf@x
895     \pgf@ya=\pgf@y
896     \pgf@process{#3}%
897     \pgf@xb=\pgf@x
898     \pgf@yb=\pgf@y
899     \advance\pgf@x by-\pgf@xa%
900     \advance\pgf@y by-\pgf@ya%
901     \pgf@sys@bp@correct{\pgf@xa}%
902     \pgf@sys@bp@correct{\pgf@ya}%
903     \pgf@sys@bp@correct{\pgf@xb}%
904     \pgf@sys@bp@correct{\pgf@yb}%
905     \pgf@xc=-\pgf@xa
906     \pgf@yc=-\pgf@ya
907     % Now build the function
908     \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
909       \special{pdf:stream @pgfstream\the\pgfsys@objnum\space({#4})
910         <</FunctionType 4 /Domain [\pgf@sys@tonumber{\pgf@xa}\space
911           \pgf@sys@tonumber{\pgf@xb}\space
912           \pgf@sys@tonumber{\pgf@ya}\space
913           \pgf@sys@tonumber{\pgf@yb}]
914         /Range [\pgf@shading@functional@range]>>}}}\@tempa%
915     \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
916       \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
917         width \the\pgf@x\space height \the\pgf@y}%
918       \special{pdf:put @resources <</Shading <</Sh <</ShadingType 1
919         /ColorSpace \pgf@shading@device\space
920         /Matrix [1 0 0 1 \pgf@sys@tonumber{\pgf@xc}\space
921           \pgf@sys@tonumber{\pgf@yc}]
922         /Domain [\pgf@sys@tonumber{\pgf@xa}\space
923           \pgf@sys@tonumber{\pgf@xb}\space
924           \pgf@sys@tonumber{\pgf@ya}\space
925           \pgf@sys@tonumber{\pgf@yb}]
926         /Function @pgfstream\the\pgfsys@objnum>> >> >>}}%
927     \pgfsys@invoke{/Sh sh}%
928     \special{pdf:exobj}}}\@tempa% <<
929     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
930       \leavevmode\hbox to\the\pgf@x{\vbox to\the\pgf@y{%
931         \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
932     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
933       \pgf@shading@model}%
934   }%
935 \global\advance\pgfsys@objnum\@ne
936 }
937 </dviptdpmx-driver>

```

dvipdfm driver

```

938 <*dvipdfm-driver>
939 \ProvidesFile{pgfsys-cmykshadings-dvipdfm.def}%

```

```

940 [2018/10/24
941 CMYK and grayscale shadings support for PGF dvipdfm driver (DCP)]
942 \def\pgfsys@horishading#1#2#3{%
943   {%
944     \pgf@parsefunc{#3}%
945     \pgf@process{\pgfpoint{\pgf@max}{#2}}%
946     \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
947       \special{pdf: beginxobj @pgfshade#1 width \the\pgf@max\space
948         height \the\pgf@y}}\@temp
949     \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
950       \special{pdf: put @resources <<
951         /Shading << /Sh << /ShadingType 2
952         /ColorSpace \pgf@shading@device\space
953         /Domain [\pgf@pdfparseddomain]
954         /Coords [\pgf@doma\space0 \pgf@domb\space0]
955         /Function \pgf@pdfparsedfunction
956         /Extend [false false] >> >> >>}}\@temp% <<
957     \pgfutil@insertatbegincurrentpage{\special{pdf: content /Sh sh}}%
958     \special{pdf: endxobj}}%
959     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
960       \hbox to\the\pgf@max{\vbox to#2{%
961         \vfil\special{pdf: usexobj @pgfshade#1}}\hfil}}%
962     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
963       \pgf@shading@model}%
964   }%
965 }
966 \def\pgfsys@vertshading#1#2#3{%
967   {%
968     \pgf@parsefunc{#3}%
969     \pgf@process{\pgfpoint{\pgf@max}{#2}}%
970     \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
971       \special{pdf: beginxobj @pgfshade#1 width \the\pgf@y\space
972         height \the\pgf@max\space}}\@temp
973     \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
974       \special{pdf: put @resources <<
975         /Shading << /Sh << /ShadingType 2
976         /ColorSpace \pgf@shading@device\space
977         /Domain [\pgf@pdfparseddomain]
978         /Coords [0 \pgf@doma\space0 \pgf@domb]
979         /Function \pgf@pdfparsedfunction
980         /Extend [false false] >> >> >>}}\@temp% <<
981     \pgfutil@insertatbegincurrentpage{\special{pdf: content /Sh sh}}%
982     \special{pdf: endxobj}}%
983     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
984       \hbox to#2{\vbox to\the\pgf@max{\vfil\special{%
985         pdf: usexobj @pgfshade#1}}\hfil}}%
986     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
987       \pgf@shading@model}%
988   }%
989 }
990 \def\pgfsys@radialshading#1#2#3{%
991   {%
992     \pgf@parsefunc{#3}%
993     \pgf@process{#2}%

```

```

994 \pgf@xa=\pgf@x
995 \pgf@ya=\pgf@y
996 \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
997 \advance\pgf@xa by \pgf@x\relax
998 \advance\pgf@ya by \pgf@y\relax
999 \pgf@sys@bp@correct{\pgf@x}%
1000 \pgf@sys@bp@correct{\pgf@y}%
1001 \pgf@sys@bp@correct{\pgf@xa}%
1002 \pgf@sys@bp@correct{\pgf@ya}%
1003 \pgfutil@tempdima=2\pgf@max\relax
1004 \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
1005 \special{pdf: beginxobj @pgfshade#1 width
1006 \the\pgfutil@tempdima\space height \the\pgfutil@tempdima}}}\@temp%
1007 \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
1008 \special{pdf: put @resources <<
1009 /Shading << /Sh << /ShadingType 3
1010 /ColorSpace \pgf@shading@device\space
1011 /Domain [\pgf@pdfparseddomain]
1012 /Coords [\pgf@sys@tonumber{\pgf@xa} \pgf@sys@tonumber{\pgf@ya}
1013 \pgf@doma\space \pgf@sys@tonumber{\pgf@x} \pgf@sys@tonumber{\pgf@y}
1014 \pgf@domb]
1015 /Function \pgf@pdfparsedfunction
1016 /Extend [true false] >> >> >>}}}\@temp% <<
1017 \pgfutil@insertatbegincurrentpage{\special{pdf: content /Sh sh}}%
1018 \special{pdf: endxobj}}}%
1019 \expandafter\xdef\csname @pgfshade#1!\endcsname{%
1020 \hbox to\the\pgfutil@tempdima{\vbox to\the\pgfutil@tempdima{%
1021 \vfil\special{pdf: usexobj @pgfshade#1}}\hfil}}}%
1022 \expandafter\xdef\csname @pgfshade#1@model!\endcsname{%
1023 \pgf@shading@model}%
1024 }%
1025 }%
1026 </dvipdfm-driver>

```

dvips driver

```

1027 <*dvips-driver>
1028 \ProvidesFile{pgfsys-cmykshadings-dvips.def}%
1029 [2018/10/24
1030 CMYK and grayscale shadings support for PGF dvips driver (DCP)]
1031 \input pgfsys-cmykshadings-common-postscript.def
1032 </dvips-driver>

```

textures driver

```

1033 <*textures-driver>
1034 \ProvidesFile{pgfsys-cmykshadings-textures.def}%
1035 [2018/10/24
1036 CMYK and grayscale shadings support for PGF textures driver (DCP)]
1037 \input pgfsys-cmykshadings-common-postscript.def
1038 </textures-driver>

```

vtex driver

```

1039 <*vtex-driver>
1040 \ProvidesFile{pgfsys-cmykshadings-vtex.def}%
1041 [2018/10/24
1042   CMYK and grayscale shadings support for PGF vtex driver (DCP)]
1043 \input pgfsys-cmykshadings-common-postscript.def
1044 </vtex-driver>

```

PostScript® driver common code

```

1045 <*common-ps-driver>
1046 \ProvidesFile{pgfsys-cmykshadings-common-postscript.def}%
1047 [2018/10/24
1048   CMYK and grayscale shadings support for PGF PostScript driver (DCP)]

```

TeX shading macros.

```

1049 \def\pgfsys@horishading#1#2#3{%
1050   {%
1051     \pgf@parsefunc{#3}%
1052     \pgfmathsetlength\pgf@x{#2}%
1053     \pgf@xa=\pgf@x
1054     \pgf@sys@bp@correct{\pgf@x}%
1055     \pgf@y=\pgf@max
1056     \pgf@sys@bp@correct{\pgf@y}%
1057     \expandafter\xdef\csname @pgfshading#1!\endcsname{\hbox to \the\pgf@max{%
1058       \noexpand\vrule width0pt height\the\pgf@xa
1059       \noexpand\pgfsys@beginpurepicture
1060         \noexpand\pgfsys@rect{0pt}{0pt}{\the\pgf@max}{\the\pgf@xa}%
1061         \noexpand\pgfsys@clipnext
1062         \noexpand\pgfsys@discardpath
1063         \noexpand\pgfsys@invoke{\pgf@domb\space \pgf@sys@tonumber{\pgf@x}
1064           \pgfH\pgf@shading@model\space \pgf@psfuncs\space pop}%
1065         \hss
1066       \noexpand\pgfsys@endpurepicture}}%
1067     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
1068       \pgf@shading@model}%
1069   }%
1070 }
1071 \def\pgfsys@vertshading#1#2#3{%
1072   {%
1073     \pgf@parsefunc{#3}%
1074     \pgfmathsetlength\pgf@x{#2}%
1075     \pgf@xa=\pgf@x
1076     \pgf@sys@bp@correct{\pgf@x}%
1077     \pgf@y=\pgf@max
1078     \pgf@sys@bp@correct{\pgf@y}%
1079     \expandafter\xdef\csname @pgfshading#1!\endcsname{\hbox to\the\pgf@xa{%
1080       \noexpand\vrule width0pt height\the\pgf@max
1081       \noexpand\pgfsys@beginpurepicture
1082         \noexpand\pgfsys@rect{0pt}{0pt}{\the\pgf@xa}{\the\pgf@max}%
1083         \noexpand\pgfsys@clipnext
1084         \noexpand\pgfsys@discardpath
1085         \noexpand\pgfsys@invoke{\pgf@domb\space \pgf@sys@tonumber{\pgf@x}
1086           \pgfV\pgf@shading@model\space \pgf@psfuncs\space pop}%
1087         \hss

```

```

1088     \noexpand\pgfsys@endpurepicture}}%
1089     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
1090     \pgf@shading@model}%
1091 }%
1092 }
1093 \def\pgfsys@radialshading#1#2#3{%
1094   {%
1095     \pgf@parsefunc{#3}%
1096     \pgf@process{#2}%
1097     \pgf@sys@bp@correct{\pgf@x}%
1098     \pgf@sys@bp@correct{\pgf@y}%
1099     \pgf@xa=2\pgf@max
1100     \pgf@sys@bp@correct{\pgf@max}%
1101     \advance\pgf@x by \pgf@max
1102     \advance\pgf@y by \pgf@max
1103     \expandafter\xdef\csname @pgfshading#1!\endcsname{\hbox to \the\pgf@xa{%
1104       \noexpand\vrule width0pt height\the\pgf@xa
1105       \noexpand\pgfsys@beginpurepicture
1106       \noexpand\pgfsys@invoke{%
1107         \pgf@domb\space \pgf@sys@tonumber{\pgf@y}
1108         \pgf@sys@tonumber{\pgf@x}
1109         \pgf@sys@tonumber{\pgf@max} pgfR1\pgf@shading@model\space
1110         \pgf@psfuncs\space \pgf@firstcolor\space \pgf@doma\space
1111         pgfR2\pgf@shading@model}}%
1112       \hss
1113       \noexpand\pgfsys@endpurepicture}}%
1114     \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
1115     \pgf@shading@model}%
1116 }%
1117 }
1118 \def\pgfsys@functionalshading#1#2#3#4{%
1119   {%
1120     \pgf@process{#2}%
1121     \pgf@xa=\pgf@x
1122     \pgf@ya=\pgf@y
1123     \pgf@process{#3}%
1124     \pgf@xb=\pgf@x
1125     \pgf@yb=\pgf@y
1126     \advance\pgf@x by-\pgf@xa
1127     \advance\pgf@y by-\pgf@ya
1128     \pgf@sys@bp@correct{\pgf@xa}%
1129     \pgf@sys@bp@correct{\pgf@ya}%
1130     \pgf@sys@bp@correct{\pgf@xb}%
1131     \pgf@sys@bp@correct{\pgf@yb}%
1132     \pgf@xc=-\pgf@xa
1133     \pgf@yc=-\pgf@ya
1134     \expandafter\xdef\csname @pgfshading#1!\endcsname{%
1135     \hbox to\the\pgf@x{\vbox to\the\pgf@y{\vfil
1136       \noexpand\pgfsys@beginpurepicture
1137       \noexpand\pgfsys@invoke{%
1138         \pgf@sys@tonumber{\pgf@xc} \pgf@sys@tonumber{\pgf@yc} translate
1139         1.1 setlinewidth [] 0 setdash 0 setlinecap
1140         /pgfproc {#4} bind def
1141         \pgf@sys@tonumber{\pgf@ya} 1 \pgf@sys@tonumber{\pgf@yb}

```

```

1142         {
1143             \pgf@sys@tonumber{\pgf@xa} 1 \pgf@sys@tonumber{\pgf@xb}
1144             { 1 index 2 copy pgfproc \pgf@shading@ps@device\space
1145               moveto 1.1 0 rlineto stroke }
1146             for
1147             pop
1148         }
1149         for
1150     }%
1151     \noexpand\pgfsys@endpurepicture
1152 } \hfil}%
1153 }%
1154 \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
1155     \pgf@shading@model}%
1156 }%
1157 }

PostScript® support code.
1158 \g@addto@macro\pgfsys@atbegindocument{%

Define RGB PostScript® shading functions.
1159 \pgf@sys@postscript@header{/pgfHrgb { pgfH } bind def}%
1160 \pgf@sys@postscript@header{/pgfVrgb { pgfV } bind def}%
1161 \pgf@sys@postscript@header{/pgfR1rgb { pgfR1 } bind def}%
1162 \pgf@sys@postscript@header{/pgfR2rgb { pgfR2 } bind def}%

Define CMYK PostScript® shading functions.
1163 \pgf@sys@postscript@header{/pgfHcmyk{
1164     /pgfheight exch def 0.75 setlinewidth [] 0 setdash
1165     /pgfshade {pgfAcmyk} def /pgfdir { dup 0 moveto
1166       dup 6 index lineto } bind def} bind def}%
1167 \pgf@sys@postscript@header{/pgfVcmyk{
1168     /pgfheight exch def 0.75 setlinewidth [] 0 setdash
1169     /pgfshade {pgfAcmyk} def /pgfdir { dup 0 exch moveto dup 6 index
1170       exch lineto } bind def} bind def}%
1171 \pgf@sys@postscript@header{/pgfAcmyk{
1172     /pgfdiff 10 index round cvi 10 index round cvi sub 2 mul 1 add def
1173     3 index 8 index sub pgfdiff div % put cyan-step on stack
1174     3 index 8 index sub pgfdiff div % put magenta-step on stack
1175     3 index 8 index sub pgfdiff div % put yellow-step on stack
1176     3 index 8 index sub pgfdiff div % put black-step on stack
1177     pgfheight 12 index 12 index 12 index 12 index 18 index
1178     pgfdiff {
1179         4 index 4 index 4 index 4 index setcmykcolor % Set color
1180         pgfdir
1181         stroke
1182         5 -1 roll 9 index add % cyan += inccyan
1183         5 -1 roll 8 index add % magenta += incmagenta
1184         5 -1 roll 7 index add % yellow += incyellow
1185         5 -1 roll 6 index add % black += incblack
1186         5 -1 roll .5 sub % x += 0.5
1187     } repeat
1188     mark 19 1 roll cleartomark exch pop % leave only start x on stack
1189 }bind def }%
1190 \pgf@sys@postscript@header{/pgfR1cmyk{
1191     newpath dup dup dup 0 360 arc clip newpath

```

```

1192     dup /pgfendx exch def
1193     /pgfendy exch def
1194     0.875 setlinewidth
1195     [] 0 setdash
1196     /pgfshade {pgfRcmyk} def
1197     /pgfstartx exch def
1198     /pgfstarty exch def
1199     /pgfdiffx pgfendx pgfstartx sub def
1200     /pgfdiffy pgfendy pgfstarty sub def
1201     dup /pgfdomb exch def
1202     }bind def }%
1203     \pgf@sys@postscript@header{/pgfR2cmyk{
1204         newpath 0.5 add pgfcircx pgfcircy 3 2 roll 0 360 arc
1205         setcmykcolor fill pop}bind def }%
1206     \pgf@sys@postscript@header{/pgfRcmyk{
1207         /pgfdiff 10 index round cvi 10 index round cvi sub 4 mul 1 add def
1208         /pgfcircx pgfstartx 11 index pgfdiffx pgfdomb div mul add def
1209         /pgfcircy pgfstarty 11 index pgfdiffy pgfdomb div mul add def
1210         /pgfcircxe pgfstartx 10 index pgfdiffx pgfdomb div mul add def
1211         /pgfcircye pgfstarty 10 index pgfdiffy pgfdomb div mul add def
1212         /pgfxstep pgfcircxe pgfcircx sub pgfdiff div def
1213         /pgfystep pgfcircye pgfcircy sub pgfdiff div def
1214         3 index 8 index sub pgfdiff div % put cyan-step on stack
1215         3 index 8 index sub pgfdiff div % put magenta-step on stack
1216         3 index 8 index sub pgfdiff div % put yellow-step on stack
1217         3 index 8 index sub pgfdiff div % put black-step on stack
1218         11 index 11 index 11 index 17 index
1219         pgfdiff {
1220             4 index 4 index 4 index 4 index setcmykcolor % Set color
1221             pgfcircx pgfcircy 2 index 0 360 arc closepath
1222             stroke
1223             5 -1 roll 8 index add % cyan += inccyan
1224             5 -1 roll 7 index add % magenta += incmagenta
1225             5 -1 roll 6 index add % yellow += incyellow
1226             5 -1 roll 5 index add % black += incblack
1227             5 -1 roll .25 sub % x += 0.25
1228             /pgfcircx pgfcircx pgfxstep add def
1229             /pgfcircy pgfcircy pgfystep add def
1230         } repeat
1231         mark 18 1 roll cleartomark exch pop % leave only start x on stack
1232     }bind def}%

```

Define grayscale PostScript® shading functions.

```

1233     \pgf@sys@postscript@header{/pgfHgray{
1234         /pgfheight exch def 0.75 setlinewidth [] 0 setdash
1235         /pgfshade {pgfAgray} def /pgfdir { dup 0 moveto
1236             dup 3 index lineto } bind def} bind def}%
1237     \pgf@sys@postscript@header{/pgfVgray{
1238         /pgfheight exch def 0.75 setlinewidth [] 0 setdash
1239         /pgfshade {pgfAgray} def /pgfdir { dup 0 exch moveto dup 3 index
1240             exch lineto } bind def} bind def}%
1241     \pgf@sys@postscript@header{/pgfAgray{
1242         /pgfdiff 4 index round cvi 4 index round cvi sub 2 mul 1 add def
1243         dup 2 index sub pgfdiff div % put gray-step on stack
1244         pgfheight 3 index 6 index

```

```

1245     pgfdiff {
1246         1 index setgray % Set color
1247         pgfdir
1248         stroke
1249         exch 3 index add % gray += incgray
1250         exch .5 sub % x += 0.5
1251     } repeat
1252     mark 7 1 roll cleartomark exch pop % leave only start x on stack
1253 }bind def }%
1254 \pgf@sys@postscript@header{/pgfR1gray{
1255     newpath dup dup dup 0 360 arc clip newpath
1256     dup /pgfendx exch def
1257     /pgfendy exch def
1258     0.875 setlinewidth
1259     [] 0 setdash
1260     /pgfshade {pgfRgray} def
1261     /pgfstartx exch def
1262     /pgfstarty exch def
1263     /pgfdiffx pgfendx pgfstartx sub def
1264     /pgfdiffy pgfendy pgfstarty sub def
1265     dup /pgfdomb exch def
1266 }bind def }%
1267 \pgf@sys@postscript@header{/pgfR2gray{
1268     newpath 0.5 add pgfcircx pgfcircy 3 2 roll 0 360 arc
1269     setgray fill pop}bind def }%
1270 \pgf@sys@postscript@header{/pgfRgray{
1271     /pgfdiff 4 index round cvi 4 index round cvi sub 4 mul 1 add def
1272     /pgfcircx pgfstartx 5 index pgfdiffx pgfdomb div mul add def
1273     /pgfcircy pgfstarty 5 index pgfdiffy pgfdomb div mul add def
1274     /pgfcircxe pgfstartx 4 index pgfdiffx pgfdomb div mul add def
1275     /pgfcircye pgfstarty 4 index pgfdiffy pgfdomb div mul add def
1276     /pgfxstep pgfcircxe pgfcircx sub pgfdiff div def
1277     /pgfystep pgfcircye pgfcircy sub pgfdiff div def
1278     dup 2 index sub pgfdiff div % put gray-step on stack
1279     2 index 5 index
1280     pgfdiff {
1281         1 index setgray % Set color
1282         pgfcircx pgfcircy 2 index 0 360 arc closepath
1283         stroke
1284         exch 2 index add % gray += incgray
1285         exch .25 sub % x += 0.25
1286         /pgfcircx pgfcircx pgfxstep add def
1287         /pgfcircy pgfcircy pgfystep add def
1288     } repeat
1289     mark 6 1 roll cleartomark exch pop % leave only start x on stack
1290 }bind def}%
1291 }
1292 </common-ps-driver>

```


Change History

v1.0		Support PostScript® drivers	27	
	General: First public release	1	Support tex4ht driver	16
v1.1		v1.1a		
	General: Fix typo for <code>rgb</code> option . . .	17	General: Fix missing percent sign . . .	12
	Support dvipdfm driver	25	v1.2	
	Support dvisvgm driver	16	General: Deprecate package	1