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

---

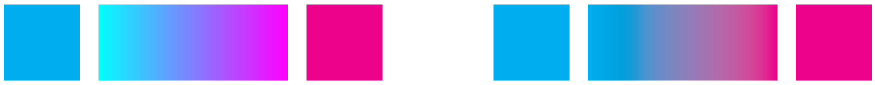
[*] Email: dcpurton@marshwiggle.net

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 respoitory. 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:

`cmyk` (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`, `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 `\pgfuseshading` is called if `\pgfdeclare*shading` was called with an optional argument.

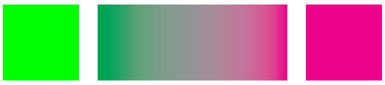
If the xcolor package `natural` colour model is in use then the shading colour model will be CMYK by default (equivalent to passing the `cmyk` option to the pgf-cmykshadings package). RGB shadings can be output by default instead by passing the `rgb` option to the pgf-cmykshadings package.

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.

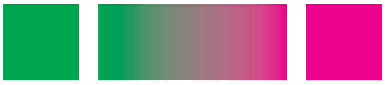
For this reason it is recommended to always load the xcolor package before the pgf-cmykshadings package with either the `rgb`, `cmyk`, or `gray` options to avoid colour surprises.

```
\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}
```

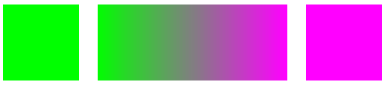
xcolor `natural` colour model:



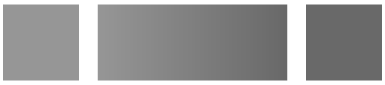
xcolor `cmyk` colour model:



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.

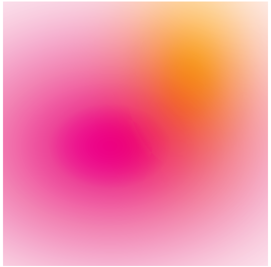
Take particular care that the same colour model is in use at declaration time and use time for functional shadings declared with an optional argument as otherwise the PostScript® data will not match the declared colour space and you will end up with a malformed PDF.

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.

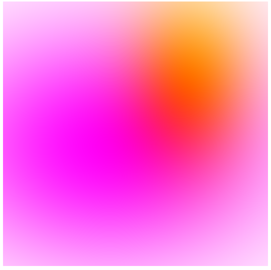
Having said this, it *is* possible to create portable functional shadings by providing conditional code to append colour transformations to the PostScript® data. A variety of `\pgffuncshading*to*` (e.g., `\pgffuncshadingrgbtocmyk`) macros along with `\ifpgfshadingmodel*` (e.g., `\ifpgfshadingmodelcmyk`)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

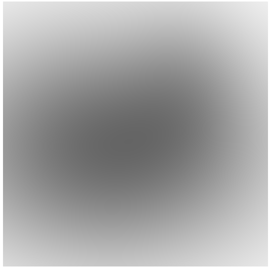
xcolor cmyk model:



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., `cmyk(⟨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.

`\pgfshadecolortorgb`

`\pgfshadecolortorgb{`⟨*color name*⟩`}{`⟨*macro*⟩`}`

This command takes ⟨*color name*⟩ 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 ⟨*macro*⟩. This macro can then be used inside the ⟨*type 4 function*⟩ argument for `\pgfdeclarefunctionalshading`.

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

`\pgfshadecolortocmyk`

`\pgfshadecolortocmyk{`⟨*color name*⟩`}{`⟨*macro*⟩`}`

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

`\pgfshadecolortogray`

`\pgfshadecolortogray{`⟨*color name*⟩`}{`⟨*macro*⟩`}`

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

`\pgffuncshadingrgbtocmyk`

`\pgffuncshadingrgbtocmyk`

Within the ⟨*type 4 function*⟩ 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 `\ifpgfshadingmodelcmyk` conditional this macro can be used to make functional shading declarations more portable across colour models.

`\pgffuncshadingrgbtogray`

`\pgffuncshadingrgbtogray`

Within the ⟨*type 4 function*⟩ 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 `\ifpgfshadingmodelgray` conditional this macro can be used to make functional shading declarations more portable across colour models.

`\pgffuncshadingcmyktorgb`

`\pgffuncshadingcmyktorgb`

Within the ⟨*type 4 function*⟩ 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 `\ifpgfshadingmodelrgb` conditional this macro can be used to make functional shading declarations more portable across colour models.

`\pgffuncshadingcmyktogray`

`\pgffuncshadingcmyktogray`

Within the ⟨*type 4 function*⟩ 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 `\ifpgfshadingmodelgray` conditional this macro can be used to make functional shading declarations more portable across colour models.

`\pgffuncshadinggraytorgb`

`\pgffuncshadinggraytorgb`

Within the ⟨*type 4 function*⟩ argument of `\pgfdeclarefunctionalshading`, this command can be used to convert the top element on the stack from grayscale to RGB. In combination with the `\ifpgfshadingmodelrgb` 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

```
⟨*package⟩
\ProvidesPackage{pgf-cmykshadings}%
  [2019/11/05
   CMYK and grayscale shadings support for PGF (DCP)]
\RequirePackage{pgf}
```

Set colour model to CMYK by default if version of pgf is at least 3.1.3 then `\endinput`.

```
\@ifpackagelater{pgf}{2019/05/14}{%
  \PackageWarning{pgf-cmykshadings}{%
    Package ‘pgf-cmykshadings’ is now deprecated.}%
  \newif\ifpgfcmykshadingdefault
  \DeclareOption{cmyk}{%
    \pgfcmykshadingdefaulttrue
  }
  \DeclareOption{rgb}{%
    \pgfcmykshadingdefaultfalse
  }
  \ExecuteOptions{cmyk}
  \ProcessOptions\relax
  \def\pgf@setup@shading@model{%
    \pgfshadingmodelrgbtrue
  \pgfshadingmodelcmykfalse
  \pgfshadingmodelgrayfalse
  \XC@sdef\pgf@mod@test{\XC@tgt@mod{natural}}%
  \def\pgf@shading@device{/DeviceRGB}%
  \def\pgf@shading@ps@device{setrgbcolor}%
  \def\pgf@shading@functional@range{0 1 0 1 0 1}%
  \def\pgf@shading@model{rgb}%
  \ifx\pgf@mod@test\XC@mod@natural
   \ifpgfcmykshadingdefault
    \def\pgf@shading@functional@range{0 1 0 1 0 1 0 1}%
    \def\pgf@shading@device{/DeviceCMYK}%
    \def\pgf@shading@ps@device{setcmykcolor}%
    \def\pgf@shading@model{cmyk}%
    \pgfshadingmodelrgbfalse
    \pgfshadingmodelcmyktrue
   \else
    \def\pgf@shading@functional@range{0 1 0 1 0 1}%
    \def\pgf@shading@device{/DeviceRGB}%
    \def\pgf@shading@ps@device{setrgbcolor}%
    \def\pgf@shading@model{rgb}%
   \fi
  \fi
  \ifx\pgf@mod@test\XC@mod@cmyk
   \def\pgf@shading@device{/DeviceCMYK}%
   \def\pgf@shading@ps@device{setcmykcolor}%
   \def\pgf@shading@functional@range{0 1 0 1 0 1 0 1}%
   \def\pgf@shading@model{cmyk}%
   \pgfshadingmodelrgbfalse
   \pgfshadingmodelcmyktrue
  \fi
  \ifx\pgf@mod@test\XC@mod@gray
   \def\pgf@shading@device{/DeviceGray}%
   \def\pgf@shading@ps@device{setgray}%
   \def\pgf@shading@functional@range{0 1}%
   \def\pgf@shading@model{gray}%
   \pgfshadingmodelrgbfalse
   \pgfshadingmodelgraytrue
  \fi
  \edef\pgf@sys@driver@dvisvgm{pgfsys-dvisvgm.def}%
  \ifx\pgfsysdriver\pgf@sys@driver@dvisvgm
   \def\pgf@shading@model{rgb}%
  \fi
  \edef\pgf@sys@driver@texforht{pgfsys-tex4ht.def}%
  \ifx\pgfsysdriver\pgf@sys@driver@texforht
   \def\pgf@shading@model{rgb}%
  \fi
 }%
 \endinput}{}
```

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

```
\def\pgf@parsefunc#1{%
 \edef\temp{{#1}}%
 \expandafter\pgf@convertstring\temp%
 \edef\temp{{\pgf@conv}}%
```

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

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

```
\def\pgf@@parsefunc#1{%
  \let\pgf@bounds=\pgfutil@empty%
  \let\pgf@funcs=\pgfutil@empty%
  \let\pgf@psfuncs=\pgfutil@empty%
  \let\pgf@encode=\pgfutil@empty%
  \let\pgf@sys@shading@ranges=\pgfutil@empty%
  \pgf@sys@shading@range@num=0\relax%
  \csname pgf@parsefirst\pgf@shading@model\endcsname[#1; ]%
  \csname pgf@parselastdom\pgf@shading@model\endcsname[#1; ]%
  \csname pgf@parsemid\pgf@shading@model\endcsname[#1; ]%
  \ifx\pgf@bounds\pgfutil@empty%
    \edef\pgf@pdfparseddomain{0 1}%
    \edef\pgf@pdfparsedfunction{\pgf@singlefunc\space}%
  \else%
    \edef\pgf@pdfparseddomain{\pgf@doma\space\pgf@domb}%
    \edef\pgf@pdfparsedfunction{%
      << /FunctionType 3 /Domain [\pgf@doma\space\pgf@domb] /Functions
      [\pgf@funcs\space] /Bounds [\pgf@bounds] /Encode [0 1 \pgf@encode]
      >> }% <<
  \fi%
  \xdef\pgf@psfuncs{\pgf@psfuncs}%
  }
```

Define RGB parsing macros.

```
\let\pgf@parsefirstrgb\pgf@parsefirst
\let\pgf@parselastdomrgb\pgf@parselastdom
\let\pgf@parsemidrgb\pgf@parsemid
\let\pgf@parserestrgb\pgf@parserest
```

Define new CMYK parsing macros.

```
\def\pgf@parsefirstcmyk[cmyk(#1)=(#2,#3,#4,#5)#6]{%
  \pgfmathsetlength\pgf@x{#1}%
  \edef\pgf@sys@shading@start@pos{\the\pgf@x}%
  \pgf@sys@bp@correct\pgf@x%
  \edef\pgf@doma{\pgf@sys@tonumber{\pgf@x}}%
  \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
  \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
  \edef\pgf@sys@shading@start@cmyk{\pgf@sys@cmyk}%
  \let\pgf@sys@prevcolor=\pgf@sys@shading@start@cmyk%
  \let\pgf@sys@prevpos=\pgf@sys@shading@start@pos%
  \edef\pgf@prevcolor{\pgf@cmyk}%
  \edef\pgf@firstcolor{\pgf@cmyk}}
\def\pgf@parselastdomcmyk[cmyk(#1)=(#2,#3,#4,#5); {%
  \pgfutil@ifnextchar]{%
    \pgfmathsetlength\pgf@x{#1}%
    \edef\pgf@sys@shading@end@pos{\the\pgf@x}%
    \pgf@max=\pgf@x\relax%
    \pgf@sys@bp@correct\pgf@x%
    \edef\pgf@domb{\pgf@sys@tonumber{\pgf@x}}%
    \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
    \edef\pgf@sys@shading@end@cmyk{\pgf@sys@cmyk}%
    \pgfutil@gobble}{\pgf@parselastdomcmyk[]}
\def\pgf@parsemidcmyk[cmyk(#1)=(#2,#3,#4,#5); {\pgf@parserestcmyk[}
\def\pgf@parserestcmyk[cmyk(#1)=(#2,#3,#4,#5); {%
  \advance\pgf@sys@shading@range@num by1\relax%
  \pgfutil@ifnextchar]{%
    \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
    \edef\pgf@singlefunc{\space%
      << /FunctionType 2 /Domain [0 1] /C0
      [\pgf@prevcolor] /C1 [\pgf@cmyk] /N 1 >> }% <<
    \edef\pgf@funcs{\pgf@funcs\space%
      << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /C0
      [\pgf@prevcolor] /C1 [\pgf@cmyk] /N 1 >> }% <<
    \edef\pgf@psfuncs{\pgf@prevx\space
      \pgf@cmyk\space \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
    \pgfmathsetlength\pgf@x{#1}%
    \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
      {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@cmyk}}}%
    \edef\pgf@sys@prevpos{\the\pgf@x}%
    \let\pgf@sys@prevcolor=\pgf@sys@cmyk%
    \pgfutil@gobble}{%
    \pgfmathsetlength\pgf@x{#1}%
    \pgf@getcmyktuplewithmixin{#2}{#3}{#4}{#5}%
    \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
      {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@cmyk}}}%
    \edef\pgf@sys@prevpos{\the\pgf@x}%
    \let\pgf@sys@prevcolor=\pgf@sys@cmyk%
    \edef\pgf@psfuncs{\pgf@prevx\space \pgf@cmyk\space
      \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
    \pgf@sys@bp@correct\pgf@x%
    \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
    \edef\pgf@bounds{\pgf@bounds\space\pgf@sys@tonumber{\pgf@x}}%
    \edef\pgf@encode{\pgf@encode\space0 1}%
    \edef\pgf@singlefunc{\space%
      << /FunctionType 2 /Domain [0 1] /C0
      [\pgf@prevcolor] /C1 [\pgf@cmyk] /N 1 >> }% <<
    \edef\pgf@funcs{\pgf@funcs\space%
      << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /C0
      [\pgf@prevcolor] /C1 [\pgf@cmyk] /N 1 >> }% <<
    \edef\pgf@prevcolor{\pgf@cmyk}%
    \pgf@parserestcmyk[]}
\def\pgf@getcmyktuplewithmixin#1#2#3#4{%
  \pgfutil@definecolor{pgfshadetemp}{cmyk}{#1,#2,#3,#4}%
  \pgfutil@ifundefined{applycolormixins}{}{\applycolormixins{pgfshadetemp}}%
  \pgfutil@extractcolorspec{pgfshadetemp}{\pgf@tempcolor}%
  \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{cmyk}{\pgf@cmykcolor}%
  \expandafter\pgf@getcmyk@@\pgf@cmykcolor!}
\def\pgf@getcmyk@@#1,#2,#3,#4!{%
  \def\pgf@cmyk{#1 #2 #3 #4}%
  \def\pgf@sys@cmyk{{#1}{#2}{#3}{#4}}%
}
```

Define new grayscale parsing macros.

```
\def\pgf@parsefirstgray[gray(#1)=(#2)#3]{%
  \pgfmathsetlength\pgf@x{#1}%
  \edef\pgf@sys@shading@start@pos{\the\pgf@x}%
  \pgf@sys@bp@correct\pgf@x%
  \edef\pgf@doma{\pgf@sys@tonumber{\pgf@x}}%
  \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
  \pgf@getgraytuplewithmixin{#2}%
  \edef\pgf@sys@shading@start@gray{\pgf@sys@gray}%
  \let\pgf@sys@prevcolor=\pgf@sys@shading@start@gray%
  \let\pgf@sys@prevpos=\pgf@sys@shading@start@pos%
  \edef\pgf@prevcolor{\pgf@gray}%
  \edef\pgf@firstcolor{\pgf@gray}}
\def\pgf@parselastdomgray[gray(#1)=(#2); {%
  \pgfutil@ifnextchar]{%
    \pgfmathsetlength\pgf@x{#1}%
    \edef\pgf@sys@shading@end@pos{\the\pgf@x}%
    \pgf@max=\pgf@x\relax%
    \pgf@sys@bp@correct\pgf@x%
    \edef\pgf@domb{\pgf@sys@tonumber{\pgf@x}}%
    \pgf@getgraytuplewithmixin{#2}%
    \edef\pgf@sys@shading@end@gray{\pgf@sys@gray}%
    \pgfutil@gobble}{\pgf@parselastdomgray[}}
\def\pgf@parsemidgray[gray(#1)=(#2); {\pgf@parserestgray[}
\def\pgf@parserestgray[gray(#1)=(#2); {%
  \advance\pgf@sys@shading@range@num by1\relax%
  \pgfutil@ifnextchar]{%
    \pgf@getgraytuplewithmixin{#2}%
    \edef\pgf@singlefunc{\space%
      << /FunctionType 2 /Domain [0 1] /C0
      [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
    \edef\pgf@funcs{\pgf@funcs\space%
      << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /C0
      [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
    \edef\pgf@psfuncs{\pgf@prevx\space \pgf@gray\space
      \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
    \pgfmathsetlength\pgf@x{#1}%
    \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
      {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@gray}}}%
    \edef\pgf@sys@prevpos{\the\pgf@x}%
    \let\pgf@sys@prevcolor=\pgf@sys@gray%
    \pgfutil@gobble}{%
    \pgfmathsetlength\pgf@x{#1}%
    \pgf@getgraytuplewithmixin{#2}%
    \edef\pgf@sys@shading@ranges{\pgf@sys@shading@ranges{%
      {\pgf@sys@prevpos}{\the\pgf@x}{\pgf@sys@prevcolor}{\pgf@sys@gray}}}%
    \edef\pgf@sys@prevpos{\the\pgf@x}%
    \let\pgf@sys@prevcolor=\pgf@sys@gray%
    \edef\pgf@psfuncs{\pgf@prevx\space \pgf@gray\space
      \pgf@prevcolor\space pgfshade \pgf@psfuncs}%
    \pgf@sys@bp@correct\pgf@x%
    \edef\pgf@prevx{\pgf@sys@tonumber{\pgf@x}}%
    \edef\pgf@bounds{\pgf@bounds\space\pgf@sys@tonumber{\pgf@x}}%
    \edef\pgf@encode{\pgf@encode\space0 1}%
    \edef\pgf@singlefunc{\space%
      << /FunctionType 2 /Domain [0 1] /C0
      [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
    \edef\pgf@funcs{\pgf@funcs\space%
```

```
      << /FunctionType 2 /Domain [\pgf@doma\space\pgf@domb] /C0
      [\pgf@prevcolor] /C1 [\pgf@gray] /N 1 >> }% <<
    \edef\pgf@prevcolor{\pgf@gray}%
    \pgf@parserestgray[}}
\def\pgf@getgraytuplewithmixin#1{%
  \pgfutil@definecolor{pgfshadetemp}{gray}{#1}%
  \pgfutil@ifundefined{applycolormixins}{}{\applycolormixins{pgfshadetemp}}%
  \pgfutil@extractcolorspec{pgfshadetemp}{\pgf@tempcolor}%
  \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{gray}{\pgf@graycolor}%
  \expandafter\pgf@getgray@@\pgf@graycolor!}
\def\pgf@getgray@@#1!{%
  \def\pgf@gray{#1}%
  \def\pgf@sys@gray{{#1}}%
}
```

Define new colour space agnostic colour specification parsing macros. This includes parsing CMYK colour specifications (i.e., `color(`⟨*postition*⟩`)=(`⟨*C*⟩`,`⟨*M*⟩`,`⟨*Y*⟩`,`⟨*K*⟩`)`.

```
\def\pgf@convertstring#1{%
  \def\pgf@conv{}%
  \pgf@convert#1]%
  }
\def\pgf@convert{%
  \pgfutil@ifnextchar]{\pgfutil@gobble}%done!
  {%
    \pgfutil@ifnextchar;{\pgf@grabsemicolor}%
    {%
      \pgfutil@ifnextchar c{\pgf@gobblec}%
      {%
        \pgfutil@ifnextchar g{\pgf@grabgray}%
        {%
          \pgfutil@ifnextchar o{\pgf@grabcolor}%
          {%
            \pgfutil@ifnextchar m{\pgf@grabcmyk}%
            {%
              \pgfutil@ifnextchar r{\pgf@grabrgb}%
                {\pgferror{Illformed shading
                 specification}\pgf@convert}%
            }%
          }%
        }%
      }%
    }%
  }%
}
\def\pgf@grabsemicolor;{%
  \edef\pgf@conv{\pgf@conv; }\pgf@convert}
\def\pgf@gobblec c{\pgf@convert}

\def\pgf@savecolor#1{%
  \pgfutil@extractcolorspec{pgf@tempcol}{\pgf@tempcolor}%
  \expandafter\pgfutil@convertcolorspec\pgf@tempcolor
    {\pgf@shading@model}{\pgf@color}%
  \expandafter\pgf@convget@\expandafter{\pgf@color}{#1}%
}
\def\pgf@grabrgb rgb(#1)=(#2,#3,#4){%
```

```
  \pgfutil@definecolor{pgf@tempcol}{rgb}{#2,#3,#4}%
  \pgf@savecolor{#1}%
}
\def\pgf@grabcmyk myk(#1)=(#2,#3,#4,#5){%
  \pgfutil@definecolor{pgf@tempcol}{cmyk}{#2,#3,#4,#5}%
  \pgf@savecolor{#1}%
}
\def\pgf@grabgray gray(#1)=(#2){%
  \pgfutil@definecolor{pgf@tempcol}{gray}{#2}%
  \pgf@savecolor{#1}%
}
\def\pgf@grabcolor olor(#1)=(#2){%
  \pgfutil@colorlet{pgf@tempcol}{#2}%
  \pgf@savecolor{#1}%
}
\def\pgf@convget@#1#2{%
  \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`.

```
\newdimen\pgf@xd
\def\pgfshadecolortocmyk#1#2{%
  \pgfutil@colorlet{pgf@tempcol}{#1}%
  \pgfutil@extractcolorspec{pgf@tempcol}{\pgf@tempcolor}%
  \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{cmyk}{\pgf@cmykcolor}%
  \expandafter\pgfshading@cmyk\pgf@cmykcolor\relax%
  \edef#2{\pgf@sys@tonumber{\pgf@xa}\space\pgf@sys@tonumber{\pgf@xb}\space
    \pgf@sys@tonumber{\pgf@xc}\space\pgf@sys@tonumber{\pgf@xd}\space}%
  \c@pgf@counta\escapechar%
  \escapechar-1\relax%
  \expandafter\edef\csname\string#2cyan\endcsname{%
    \pgf@sys@tonumber{\pgf@xa}\space}%
  \expandafter\edef\csname\string#2magenta\endcsname{%
    \pgf@sys@tonumber{\pgf@xb}\space}%
  \expandafter\edef\csname\string#2yellow\endcsname{%
    \pgf@sys@tonumber{\pgf@xc}\space}%
  \expandafter\edef\csname\string#2black\endcsname{%
    \pgf@sys@tonumber{\pgf@xd}\space}%
  \escapechar\c@pgf@counta
}
\def\pgfshading@cmyk#1,#2,#3,#4\relax{%
  \pgf@xa=#1pt%
  \pgf@xb=#2pt%
  \pgf@xc=#3pt%
  \pgf@xd=#4pt%
}
```

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

```
\def\pgfshadecolortogray#1#2{%
  \pgfutil@colorlet{pgf@tempcol}{#1}%
  \pgfutil@extractcolorspec{pgf@tempcol}{\pgf@tempcolor}%
  \expandafter\pgfutil@convertcolorspec\pgf@tempcolor{gray}{\pgf@graycolor}%
  \expandafter\pgfshading@gray\pgf@graycolor\relax
  \edef#2{\pgf@sys@tonumber{\pgf@xa}\space}%
  \c@pgf@counta\escapechar
  \escapechar-1\relax
  \expandafter\edef\csname\string#2gray\endcsname{%
    \pgf@sys@tonumber{\pgf@xa}\space}%
  \escapechar\c@pgf@counta
}
\def\pgfshading@gray#1\relax{%
  \pgf@xa=#1pt%
}
```

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

```
\def\pgfdeclarehorizontalshading{%
  \pgf@setup@model
  \pgfutil@ifnextchar[%
    \pgf@declarehorizontalshading{\pgf@declarehorizontalshading[]}}
\def\pgfdeclareverticalshading{%
  \pgf@setup@model
  \pgfutil@ifnextchar[%
    \pgf@declareverticalshading{\pgf@declareverticalshading[]}}
\def\pgfdeclareradialshading{%
  \pgf@setup@model
  \pgfutil@ifnextchar[%
    \pgf@declareradialshading{\pgf@declareradialshading[]}}
\def\pgfdeclarefunctionalshading{%
  \pgf@setup@model
  \pgfutil@ifnextchar[%
    \pgf@declarefunctionalshading{\pgf@declarefunctionalshading[]}}
```

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

```
\def\pgfuseshading#1{%
  \edef\pgf@shadingname{@pgfshading#1}%
  \edef\pgf@shadingsavedmodel{@pgfshading#1@model}%
  \pgf@tryextensions{\pgf@shadingname}{\pgfalternateextension}%
  \expandafter\pgfutil@ifundefined\expandafter{\pgf@shadingname}%
  {\pgferror{Undefined shading "#1"}}%
  {%
    {%
      \pgf@setup@model
      \pgfutil@globalcolorsfalse
      \def\pgf@shade@adds{}%
      \pgfutil@ifundefined{pgf@deps\pgf@shadingname}%
      {}%
      {%
        \edef\@list{\csname pgf@deps\pgf@shadingname\endcsname}%
        \pgfutil@for\@temp:=\@list\do{%
          {%
            \pgfutil@ifundefined{applycolormixins}%
              {}{\applycolormixins{\@temp}}%
            \pgfutil@extractcolorspec{\@temp}{\pgf@tempcolor}%
            \expandafter\pgfutil@ifundefined\expandafter{%
              \pgf@shadingsavedmodel}%
              {\expandafter\pgfutil@convertcolorspec\pgf@tempcolor{%
                 \pgf@shading@model}{\pgf@color}}%
              {\expandafter\pgfutil@convertcolorspec\pgf@tempcolor{%
                 \pgf@shadingsavedmodel}{\pgf@color}}%
            \xdef\pgf@shade@adds{\pgf@shade@adds,\pgf@color}%
          }%
        }%
      }%
      \expandafter\pgf@strip@shadename\pgf@shadingname!!%
      \pgfutil@ifundefined{@pgfshading\pgf@basename\pgf@shade@adds!}%
      {%
        {%
          \expandafter\def\expandafter\@temp\expandafter{%
            \csname pgf@func\pgf@shadingname\endcsname}%
          \edef\@args{{\pgf@basename\pgf@shade@adds}}%
          \expandafter\expandafter\expandafter\def
          \expandafter\expandafter\expandafter\@@args
          \expandafter\expandafter\expandafter{%
            \csname pgf@args\pgf@shadingname\endcsname}%
          \expandafter\expandafter\expandafter\@temp
            \expandafter\@args\@@args
        }%
      }%
      {}%
      \pgf@invokeshading{%
        \csname @pgfshading\pgf@basename\pgf@shade@adds!\endcsname}%
    }%
  }%
}
```

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

```
\newif\ifpgfshadingmodelrgb
\newif\ifpgfshadingmodelcmyk
\newif\ifpgfshadingmodelgray
```

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

```
\def\pgf@setup@model{%
  \pgfshadingmodelrgbtrue
  \pgfshadingmodelcmykfalse
  \pgfshadingmodelgrayfalse
  \XC@sdef\pgf@mod@test{\XC@tgt@mod{natural}}%
  \def\pgf@shading@functional@range{0 1 0 1 0 1}%
  \def\pgf@shading@device{/DeviceRGB}%
  \def\pgf@shading@ps@device{setrgbcolor}%
  \def\pgf@shading@model{rgb}%
  \ifx\pgf@mod@test\XC@mod@natural
    \ifpgfcmykshadingdefault
      \def\pgf@shading@functional@range{0 1 0 1 0 1 0 1}%
      \def\pgf@shading@device{/DeviceCMYK}%
      \def\pgf@shading@ps@device{setcmykcolor}%
      \def\pgf@shading@model{cmyk}%
      \pgfshadingmodelrgbfalse
      \pgfshadingmodelcmyktrue
    \else
      \def\pgf@shading@functional@range{0 1 0 1 0 1}%
      \def\pgf@shading@device{/DeviceRGB}%
      \def\pgf@shading@ps@device{setrgbcolor}%
      \def\pgf@shading@model{rgb}%
    \fi
  \fi
  \ifx\pgf@mod@test\XC@mod@cmyk
    \def\pgf@shading@functional@range{0 1 0 1 0 1 0 1}%
    \def\pgf@shading@device{/DeviceCMYK}%
    \def\pgf@shading@ps@device{setcmykcolor}%
    \def\pgf@shading@model{cmyk}%
      \pgfshadingmodelrgbfalse
      \pgfshadingmodelcmyktrue
  \fi
  \ifx\pgf@mod@test\XC@mod@gray
    \def\pgf@shading@functional@range{0 1}%
    \def\pgf@shading@device{/DeviceGray}%
    \def\pgf@shading@ps@device{setgray}%
    \def\pgf@shading@model{gray}%
      \pgfshadingmodelrgbfalse
      \pgfshadingmodelgraytrue
  \fi
```

```
  \edef\pgf@sys@driver@dvisvgm{pgfsys-dvisvgm.def}%
  \ifx\pgfsysdriver\pgf@sys@driver@dvisvgm
    \def\pgf@shading@model{rgb}%
  \fi
```

```
  \edef\pgf@sys@driver@texforht{pgfsys-tex4ht.def}%
  \ifx\pgfsysdriver\pgf@sys@driver@texforht
    \def\pgf@shading@model{rgb}%
  \fi
}
```

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

```
\def\pgffuncshadingrgbtocmyk{%
  1.0 exch sub 3 1 roll
  1.0 exch sub 3 1 roll
  1.0 exch sub 3 1 roll
  3 copy
  2 copy gt { exch } if pop
  2 copy gt { exch } if pop
  dup 3 1 roll sub
  0.0 2 copy lt { exch } if pop
  1.0 2 copy gt { exch } if pop
  4 1 roll
  dup 3 1 roll sub
  0.0 2 copy lt { exch } if pop
  1.0 2 copy gt { exch } if pop
  4 1 roll
  dup 3 1 roll sub
  0.0 2 copy lt { exch } if pop
  1.0 2 copy gt { exch } if pop
  4 1 roll
}
\def\pgffuncshadingrgbtogray{%
  0.11 mul exch 0.59 mul add exch 0.3 mul add
}
\def\pgffuncshadingcmyktorgb{%
  % covert to CMY
  dup 3 1 roll add
  1.0 2 copy gt { exch } if pop
  4 1 roll
  dup 3 1 roll add
  1.0 2 copy gt { exch } if pop
  4 1 roll
  add
  1.0 2 copy gt { exch } if pop
  3 1 roll
  % covert to RGB
  1.0 exch sub
  3 1 roll
  1.0 exch sub
  3 1 roll
  1.0 exch sub
  3 1 roll
}
\def\pgffuncshadingcmyktogray{%
  exch 0.11 mul add exch 0.59 mul add exch 0.3 mul add
  1.0 2 copy gt { exch } if pop
  1.0 exch sub
}
\def\pgffuncshadinggraytorgb{%
  dup dup
}
\def\pgffuncshadinggraytocmyk{%
  0.0 0.0 0.0
  4 3 roll
}
```

Load the correct driver file.

```
\def\pgfutilgetcmykshadingsdriver{%
  \expandafter\pgfutil@getcmykshadingsdriver\pgfsysdriver[%
}
\def\pgfutil@getcmykshadingsdriver pgfsys-#1[{%
  \edef\pgfsyscmykshadingsdriver{pgfsys-cmykshadings-#1}%
}
\pgfutilgetcmykshadingsdriver
\pgfutil@InputIfFileExists{\pgfsyscmykshadingsdriver}{}{}
```

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

```
\newif\ifpgfcmykshadingdefault
\DeclareOption{cmyk}{%
  \pgfcmykshadingdefaulttrue
}

\DeclareOption{rgb}{%
  \pgfcmykshadingdefaultfalse
}
\ExecuteOptions{cmyk}
\ProcessOptions\relax
```

⟨/package⟩

5.2 Drivers

pdftex driver

⟨*pdftex-driver⟩

```
\ProvidesFile{pgfsys-cmykshadings-pdftex.def}%
  [2018/10/24
   CMYK and grayscale shadings support for PGF pdftex driver (DCP)]
\def\pgfsys@horishading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgfmathparse{#2}%
    \setbox\pgfutil@tempboxa=\hbox to\pgf@max{%
      \vbox to\pgfmathresult pt{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
    \pgf@process{\pgfpoint{\pgf@max}{#2}}%
    \immediate\pdfxform resources {%
      /Shading << /Sh << /ShadingType 2
      /ColorSpace \pgf@shading@device\space
      /Domain [\pgf@pdfparseddomain]
      /Coords [\pgf@doma\space0 \pgf@domb\space0]
      /Function \pgf@pdfparsedfunction
      /Extend [false false] >> >>}\pgfutil@tempboxa% <<
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \leavevmode\noexpand\pdfrefxform\the\pdflastxform}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
\def\pgfsys@vertshading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgfmathparse{#2}%
    \setbox\pgfutil@tempboxa=\hbox to\pgfmathresult pt{%
      \vbox to\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
    \pgf@process{\pgfpoint{#2}{\pgf@max}}%
    \immediate\pdfxform resources {%
      /Shading << /Sh << /ShadingType 2
      /ColorSpace \pgf@shading@device\space
      /Domain [\pgf@pdfparseddomain]
      /Coords [0 \pgf@doma\space0 \pgf@domb]
      /Function \pgf@pdfparsedfunction
      /Extend [false false] >> >>}\pgfutil@tempboxa% <<
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \leavevmode\noexpand\pdfrefxform\the\pdflastxform}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
\def\pgfsys@radialshading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \setbox\pgfutil@tempboxa=\hbox to2\pgf@max{%
```

```
      \vbox to2\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
    \pgf@process{#2}%
    \pgf@xa=\pgf@x
    \pgf@ya=\pgf@y
    \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
    \advance\pgf@xa by \pgf@x
    \advance\pgf@ya by \pgf@y
    \pgf@sys@bp@correct{\pgf@x}%
    \pgf@sys@bp@correct{\pgf@y}%
    \pgf@sys@bp@correct{\pgf@xa}%
    \pgf@sys@bp@correct{\pgf@ya}%
    \immediate\pdfxform resources {%
      /Shading << /Sh << /ShadingType 3
      /ColorSpace \pgf@shading@device\space
      /Domain [\pgf@pdfparseddomain]
      /Coords [\pgf@sys@tonumber{\pgf@xa}
        \pgf@sys@tonumber{\pgf@ya}
        \pgf@doma\space
        \pgf@sys@tonumber{\pgf@x}
        \pgf@sys@tonumber{\pgf@y}
        \pgf@domb]
      /Function \pgf@pdfparsedfunction
      /Extend [true false] >> >>}\pgfutil@tempboxa% <<
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \leavevmode\noexpand\pdfrefxform\the\pdflastxform}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
\def\pgfsys@functionalshading#1#2#3#4{%
  {%
    \pgf@process{#2}%
    \pgf@xa=\pgf@x
    \pgf@ya=\pgf@y
    \pgf@process{#3}%
    \pgf@xb=\pgf@x
    \pgf@yb=\pgf@y
    \advance\pgf@x by-\pgf@xa
    \advance\pgf@y by-\pgf@ya
    \setbox\pgfutil@tempboxa=\hbox to\pgf@x{%
      \vbox to\pgf@y{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
    \pgf@sys@bp@correct{\pgf@xa}%
    \pgf@sys@bp@correct{\pgf@ya}%
    \pgf@sys@bp@correct{\pgf@xb}%
    \pgf@sys@bp@correct{\pgf@yb}%
    \pgf@xc=-\pgf@xa
    \pgf@yc=-\pgf@ya
    % Now build the function
    \pdfobj
    stream
    attr
    {
      /FunctionType 4
      /Domain [\pgf@sys@tonumber{\pgf@xa}\space
```

```
        \pgf@sys@tonumber{\pgf@xb}\space
        \pgf@sys@tonumber{\pgf@ya}\space
        \pgf@sys@tonumber{\pgf@yb}]
      /Range [\pgf@shading@functional@range]
    }
    {{#4}}%
    \edef\pgf@temp@num{\the\pdflastobj}%
    \pdfxform resources {%
      /Shading << /Sh << /ShadingType 1
      /ColorSpace \pgf@shading@device\space
      /Matrix [1 0 0 1 \pgf@sys@tonumber{\pgf@xc}\space
        \pgf@sys@tonumber{\pgf@yc}]
      /Domain [\pgf@sys@tonumber{\pgf@xa}\space
        \pgf@sys@tonumber{\pgf@xb}\space
        \pgf@sys@tonumber{\pgf@ya}\space
        \pgf@sys@tonumber{\pgf@yb}]
      /Function \pgf@temp@num\space 0 R
      >> >>}\pgfutil@tempboxa% <<
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \leavevmode%
      \noexpand\pdfrefxform\the\pdflastxform%
      \noexpand\pdfrefobj\pgf@temp@num%
    }%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
```

⟨/pdftex-driver⟩

xetex driver

⟨*xetex-driver⟩

```
\ProvidesFile{pgfsys-cmykshadings-xetex.def}%
  [2018/10/24
   CMYK and grayscale shadings support for PGF xetex driver (DCP)]
```

```
\input pgfsys-cmykshadings-dvipdfmx.def
```

⟨/xetex-driver⟩

luatex driver

⟨*luatex-driver⟩

```
\ProvidesFile{pgfsys-cmykshadings-luatex.def}%
  [2018/10/24
   CMYK and grayscale shadings support for PGF luatex driver (DCP)]
```

```
\def\pgfsys@horishading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgfmathparse{#2}%
    \setbox\pgfutil@tempboxa=\hbox to\pgf@max{%
      \vbox to\pgfmathresult pt{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
    \pgf@process{\pgfpoint{\pgf@max}{#2}}%
    \immediate\saveboxresource resources {%
      /Shading << /Sh << /ShadingType 2
      /ColorSpace \pgf@shading@device\space
      /Domain [\pgf@pdfparseddomain]
      /Coords [\pgf@doma\space0 \pgf@domb\space0]
      /Function \pgf@pdfparsedfunction
      /Extend [false false] >> >>}\pgfutil@tempboxa% <<
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \leavevmode\noexpand\useboxresource\the\lastsavedboxresourceindex}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
\def\pgfsys@vertshading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgfmathparse{#2}%
    \setbox\pgfutil@tempboxa=\hbox to\pgfmathresult pt{%
      \vbox to\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
    \pgf@process{\pgfpoint{#2}{\pgf@max}}%
    \immediate\saveboxresource resources {%
      /Shading << /Sh << /ShadingType 2
      /ColorSpace \pgf@shading@device\space
      /Domain [\pgf@pdfparseddomain]
      /Coords [0 \pgf@doma\space0 \pgf@domb]
      /Function \pgf@pdfparsedfunction
      /Extend [false false] >> >>}\pgfutil@tempboxa% <<
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \leavevmode\noexpand\useboxresource\the\lastsavedboxresourceindex}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
\def\pgfsys@radialshading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \setbox\pgfutil@tempboxa=\hbox to2\pgf@max{%
      \vbox to2\pgf@max{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
    \pgf@process{#2}%
    \pgf@xa=\pgf@x
    \pgf@ya=\pgf@y
    \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
    \advance\pgf@xa by \pgf@x
    \advance\pgf@ya by \pgf@y
    \pgf@sys@bp@correct{\pgf@x}%
    \pgf@sys@bp@correct{\pgf@y}%
    \pgf@sys@bp@correct{\pgf@xa}%
    \pgf@sys@bp@correct{\pgf@ya}%
    \immediate\saveboxresource resources {%
      /Shading << /Sh << /ShadingType 3
      /ColorSpace \pgf@shading@device\space
      /Domain [\pgf@pdfparseddomain]
      /Coords [\pgf@sys@tonumber{\pgf@xa}
        \pgf@sys@tonumber{\pgf@ya}
        \pgf@doma\space
        \pgf@sys@tonumber{\pgf@x}
    \pgf@sys@tonumber{\pgf@y}
    \pgf@domb]
   /Function \pgf@pdfparsedfunction
   /Extend [true false] >> >>}\pgfutil@tempboxa% <<
  \expandafter\xdef\csname @pgfshading#1!\endcsname{%
   \leavevmode\noexpand\useboxresource\the\lastsavedboxresourceindex}%
  \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
   \pgf@shading@model}%
 }%
}
\def\pgfsys@functionalshading#1#2#3#4{%
 {%
  \pgf@process{#2}%
  \pgf@xa=\pgf@x
  \pgf@ya=\pgf@y
  \pgf@process{#3}%
  \pgf@xb=\pgf@x
  \pgf@yb=\pgf@y
  \advance\pgf@x by-\pgf@xa
  \advance\pgf@y by-\pgf@ya
  \setbox\pgfutil@tempboxa=\hbox to\pgf@x{%
   \vbox to\pgf@y{\vfil\pgfsys@invoke{/Sh sh}}\hfil}%
  \pgf@sys@bp@correct{\pgf@xa}%
  \pgf@sys@bp@correct{\pgf@ya}%
  \pgf@sys@bp@correct{\pgf@xb}%
  \pgf@sys@bp@correct{\pgf@yb}%
  \pgf@xc=-\pgf@xa
  \pgf@yc=-\pgf@ya
  % Now build the function
  \pdfextension obj
  stream
  attr
  {
   /FunctionType 4
   /Domain [\pgf@sys@tonumber{\pgf@xa}\space
    \pgf@sys@tonumber{\pgf@xb}\space
    \pgf@sys@tonumber{\pgf@ya}\space
    \pgf@sys@tonumber{\pgf@yb}]
   /Range [\pgf@shading@functional@range]
  }
  {{#4}}%
  \edef\pgf@temp@num{\the\numexpr\pdffeedback lastobj\relax}%
  \saveboxresource resources {%
   /Shading << /Sh << /ShadingType 1
   /ColorSpace \pgf@shading@device\space
   /Matrix [1 0 0 1 \pgf@sys@tonumber{\pgf@xc}\space
    \pgf@sys@tonumber{\pgf@yc}]
   /Domain [\pgf@sys@tonumber{\pgf@xa}\space
    \pgf@sys@tonumber{\pgf@xb}\space
    \pgf@sys@tonumber{\pgf@ya}\space
    \pgf@sys@tonumber{\pgf@yb}]
   /Function \pgf@temp@num\space 0 R
   >> >>}\pgfutil@tempboxa% <<
  \expandafter\xdef\csname @pgfshading#1!\endcsname{%
```

```
      \leavevmode%
      \noexpand\useboxresource\the\lastsavedboxresourceindex%
      \noexpand\pdfextension refobj \pgf@temp@num%
    }%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
```

⟨/luatex-driver⟩

dvipdfmx driver

⟨*dvipdfmx-driver⟩

```
\ProvidesFile{pgfsys-cmykshadings-dvipdfmx.def}%
  [2018/10/24
   CMYK and grayscale shadings support for PGF dvipdfmx driver (DCP)]
```

```
\def\pgfsys@horishading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgfmathparse{#2}%
    \pgf@process{\pgfpoint{\pgf@max}{#2}}%
    \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
      \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
        width \the\pgf@max\space height \pgfmathresult pt}%
      \special{pdf:put @resources
      <<
        /Shading << /Sh << /ShadingType 2
        /ColorSpace \pgf@shading@device\space
        /Domain [\pgf@pdfparseddomain]
        /Coords [\pgf@doma\space0 \pgf@domb\space0]
        /Function \pgf@pdfparsedfunction
        /Extend [false false] >> >>
      >>}%
      \pgfsys@invoke{/Sh sh}%
      \special{pdf:exobj}}}\@tempa% <<
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \hbox to\the\pgf@max{\vbox to\pgfmathresult pt{%
        \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
  \global\advance\pgfsys@objnum\@ne%
}
\def\pgfsys@vertshading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgfmathparse{#2}%
    \pgf@process{\pgfpoint{\pgf@max}{#2}}%
    \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
      \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
        width \pgfmathresult pt\space height \the\pgf@max}%
      \special{pdf:put @resources
      <<
        /Shading << /Sh << /ShadingType 2
        /ColorSpace \pgf@shading@device\space
        /Domain [\pgf@pdfparseddomain]
        /Coords [0 \pgf@doma\space0 \pgf@domb]
        /Function \pgf@pdfparsedfunction
        /Extend [false false] >> >>
      >>}%
      \pgfsys@invoke{/Sh sh}%
      \special{pdf:exobj}}}\@tempa% <<
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \hbox to\pgfmathresult pt{\vbox to\the\pgf@max{%
        \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
  \global\advance\pgfsys@objnum\@ne
}
\def\pgfsys@radialshading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgf@process{#2}%
    \pgf@xa=\pgf@x
    \pgf@ya=\pgf@y
    \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
    \advance\pgf@xa by \pgf@x
    \advance\pgf@ya by \pgf@y
    \pgf@sys@bp@correct{\pgf@x}%
    \pgf@sys@bp@correct{\pgf@y}%
    \pgf@sys@bp@correct{\pgf@xa}%
    \pgf@sys@bp@correct{\pgf@ya}%
    \pgfutil@tempdima=2\pgf@max
    \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
      \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
        width \the\pgfutil@tempdima\space height \the\pgfutil@tempdima}%
      \special{pdf:put @resources
      <<
        /Shading << /Sh << /ShadingType 3
        /ColorSpace \pgf@shading@device\space
        /Domain [\pgf@pdfparseddomain]
        /Coords [\pgf@sys@tonumber{\pgf@xa}
          \pgf@sys@tonumber{\pgf@ya}
          \pgf@doma\space \pgf@sys@tonumber{\pgf@x}
          \pgf@sys@tonumber{\pgf@y}
          \pgf@domb]
        /Function \pgf@pdfparsedfunction
        /Extend [true false] >> >>
      >>}%
      \pgfsys@invoke{/Sh sh}%
      \special{pdf:exobj}}}\@tempa% <<
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \hbox to\the\pgfutil@tempdima{\vbox to\the\pgfutil@tempdima{%
        \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
  \global\advance\pgfsys@objnum\@ne
}
\def\pgfsys@functionalshading#1#2#3#4{%
  {%
    \pgf@process{#2}%
    \pgf@xa=\pgf@x
    \pgf@ya=\pgf@y
    \pgf@process{#3}%
    \pgf@xb=\pgf@x
    \pgf@yb=\pgf@y
    \advance\pgf@x by-\pgf@xa%
    \advance\pgf@y by-\pgf@ya%
    \pgf@sys@bp@correct{\pgf@xa}%
    \pgf@sys@bp@correct{\pgf@ya}%
    \pgf@sys@bp@correct{\pgf@xb}%
    \pgf@sys@bp@correct{\pgf@yb}%
    \pgf@xc=-\pgf@xa
    \pgf@yc=-\pgf@ya
    % Now build the function
    \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
      \special{pdf:stream @pgfstream\the\pgfsys@objnum\space({#4})
        <</FunctionType 4 /Domain [\pgf@sys@tonumber{\pgf@xa}\space
          \pgf@sys@tonumber{\pgf@xb}\space
          \pgf@sys@tonumber{\pgf@ya}\space
          \pgf@sys@tonumber{\pgf@yb}]
        /Range [\pgf@shading@functional@range]>>}}}\@tempa%
    \edef\@tempa{\noexpand\pgfutil@insertatbegincurrentpagefrombox{%
      \special{pdf:bxobj @pgfshade\the\pgfsys@objnum\space
        width \the\pgf@x\space height \the\pgf@y}%
      \special{pdf:put @resources <</Shading <</Sh <</ShadingType 1
        /ColorSpace \pgf@shading@device\space
        /Matrix [1 0 0 1 \pgf@sys@tonumber{\pgf@xc}\space
          \pgf@sys@tonumber{\pgf@yc}]
        /Domain [\pgf@sys@tonumber{\pgf@xa}\space
          \pgf@sys@tonumber{\pgf@xb}\space
          \pgf@sys@tonumber{\pgf@ya}\space
          \pgf@sys@tonumber{\pgf@yb}]
        /Function @pgfstream\the\pgfsys@objnum>> >> >>}%
      \pgfsys@invoke{/Sh sh}%
      \special{pdf:exobj}}}\@tempa% <<
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \leavevmode\hbox to\the\pgf@x{\vbox to\the\pgf@y{%
        \vfil\special{pdf:uxobj @pgfshade\the\pgfsys@objnum}}\hfil}}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
  \global\advance\pgfsys@objnum\@ne
}
```

⟨/dvipdfmx-driver⟩

dvipdfm driver

⟨*dvipdfm-driver⟩

```
\ProvidesFile{pgfsys-cmykshadings-dvipdfm.def}%
```

```
  [2018/10/24
   CMYK and grayscale shadings support for PGF dvipdfm driver (DCP)]
\def\pgfsys@horishading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgf@process{\pgfpoint{\pgf@max}{#2}}%
    \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
      \special{pdf: beginxobj @pgfshade#1 width \the\pgf@max\space
        height \the\pgf@y}}}\@temp
    \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
        \special{pdf: put @resources <<
      /Shading << /Sh << /ShadingType 2
      /ColorSpace \pgf@shading@device\space
      /Domain [\pgf@pdfparseddomain]
      /Coords [\pgf@doma\space0 \pgf@domb\space0]
      /Function \pgf@pdfparsedfunction
      /Extend [false false] >> >> >>}}}\@temp% <<
      \pgfutil@insertatbegincurrentpage{\special{pdf: content /Sh sh}%
      \special{pdf: endxobj}}%
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \hbox to\the\pgf@max{\vbox to#2{%
        \vfil\special{pdf: usexobj @pgfshade#1}}\hfil}}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
\def\pgfsys@vertshading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgf@process{\pgfpoint{\pgf@max}{#2}}%
    \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
      \special{pdf: beginxobj @pgfshade#1 width \the\pgf@y\space
        height \the\pgf@max\space}}}\@temp
    \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
        \special{pdf: put @resources <<
      /Shading << /Sh << /ShadingType 2
      /ColorSpace \pgf@shading@device\space
      /Domain [\pgf@pdfparseddomain]
      /Coords [0 \pgf@doma\space0 \pgf@domb]
      /Function \pgf@pdfparsedfunction
      /Extend [false false] >> >> >>}}}\@temp% <<
      \pgfutil@insertatbegincurrentpage{\special{pdf: content /Sh sh}%
      \special{pdf: endxobj}}%
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \hbox to#2{\vbox to\the\pgf@max{\vfil\special{%
        pdf: usexobj @pgfshade#1}}\hfil}}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
\def\pgfsys@radialshading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgf@process{#2}%
    \pgf@xa=\pgf@x
    \pgf@ya=\pgf@y
    \pgf@process{\pgfpoint{\pgf@max}{\pgf@max}}%
    \advance\pgf@xa by \pgf@x\relax
    \advance\pgf@ya by \pgf@y\relax
    \pgf@sys@bp@correct{\pgf@x}%
    \pgf@sys@bp@correct{\pgf@y}%
    \pgf@sys@bp@correct{\pgf@xa}%
    \pgf@sys@bp@correct{\pgf@ya}%
    \pgfutil@tempdima=2\pgf@max\relax
    \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
      \special{pdf: beginxobj @pgfshade#1 width
        \the\pgfutil@tempdima\space height \the\pgfutil@tempdima}}}\@temp%
    \edef\@temp{\noexpand\pgfutil@insertatbegincurrentpage{%
        \special{pdf: put @resources <<
      /Shading << /Sh << /ShadingType 3
      /ColorSpace \pgf@shading@device\space
      /Domain [\pgf@pdfparseddomain]
      /Coords [\pgf@sys@tonumber{\pgf@xa} \pgf@sys@tonumber{\pgf@ya}
        \pgf@doma\space \pgf@sys@tonumber{\pgf@x} \pgf@sys@tonumber{\pgf@y}
        \pgf@domb]
      /Function \pgf@pdfparsedfunction
      /Extend [true false] >> >> >>}}}\@temp% <<
      \pgfutil@insertatbegincurrentpage{\special{pdf: content /Sh sh}%
      \special{pdf: endxobj}}%
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \hbox to\the\pgfutil@tempdima{\vbox to\the\pgfutil@tempdima{%
        \vfil\special{pdf: usexobj @pgfshade#1}}\hfil}}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}%
```

⟨/dvipdfm-driver⟩

dvips driver

⟨*dvips-driver⟩

```
\ProvidesFile{pgfsys-cmykshadings-dvips.def}%
  [2018/10/24
   CMYK and grayscale shadings support for PGF dvips driver (DCP)]
```

```
\input pgfsys-cmykshadings-common-postscript.def
```

⟨/dvips-driver⟩

textures driver

⟨*textures-driver⟩

```
\ProvidesFile{pgfsys-cmykshadings-textures.def}%
  [2018/10/24
   CMYK and grayscale shadings support for PGF textures driver (DCP)]
```

```
\input pgfsys-cmykshadings-common-postscript.def
```

⟨/textures-driver⟩

vtex driver

```
⟨*vtex-driver⟩
\ProvidesFile{pgfsys-cmykshadings-vtex.def}%
  [2018/10/24
   CMYK and grayscale shadings support for PGF vtex driver (DCP)]
\input pgfsys-cmykshadings-common-postscript.def
⟨/vtex-driver⟩
```

PostScript® driver common code

```
⟨*common-ps-driver⟩
\ProvidesFile{pgfsys-cmykshadings-common-postscript.def}%
  [2018/10/24
   CMYK and grayscale shadings support for PGF PostScript driver (DCP)]
```

TeX shading macros.

```
\def\pgfsys@horishading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgfmathsetlength\pgf@x{#2}%
    \pgf@xa=\pgf@x
    \pgf@sys@bp@correct{\pgf@x}%
    \pgf@y=\pgf@max
    \pgf@sys@bp@correct{\pgf@y}%
    \expandafter\xdef\csname @pgfshading#1!\endcsname{\hbox to \the\pgf@max{%
        \noexpand\vrule width0pt height\the\pgf@xa
        \noexpand\pgfsys@beginpurepicture
          \noexpand\pgfsys@rect{0pt}{0pt}{\the\pgf@max}{\the\pgf@xa}%
          \noexpand\pgfsys@clipnext
          \noexpand\pgfsys@discardpath
          \noexpand\pgfsys@invoke{\pgf@domb\space \pgf@sys@tonumber{\pgf@x}
            pgfH\pgf@shading@model\space \pgf@psfuncs\space pop}%
          \hss
        \noexpand\pgfsys@endpurepicture}}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
\def\pgfsys@vertshading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgfmathsetlength\pgf@x{#2}%
    \pgf@xa=\pgf@x
    \pgf@sys@bp@correct{\pgf@x}%
    \pgf@y=\pgf@max
    \pgf@sys@bp@correct{\pgf@y}%
    \expandafter\xdef\csname @pgfshading#1!\endcsname{\hbox to\the\pgf@xa{%
        \noexpand\vrule width0pt height\the\pgf@max
        \noexpand\pgfsys@beginpurepicture
          \noexpand\pgfsys@rect{0pt}{0pt}{\the\pgf@xa}{\the\pgf@max}%
          \noexpand\pgfsys@clipnext
          \noexpand\pgfsys@discardpath
          \noexpand\pgfsys@invoke{\pgf@domb\space \pgf@sys@tonumber{\pgf@x}
            pgfV\pgf@shading@model\space \pgf@psfuncs\space pop}%
          \hss
        \noexpand\pgfsys@endpurepicture}}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
\def\pgfsys@radialshading#1#2#3{%
  {%
    \pgf@parsefunc{#3}%
    \pgf@process{#2}%
    \pgf@sys@bp@correct{\pgf@x}%
    \pgf@sys@bp@correct{\pgf@y}%
    \pgf@xa=2\pgf@max
    \pgf@sys@bp@correct{\pgf@max}%
    \advance\pgf@x by \pgf@max
    \advance\pgf@y by \pgf@max
    \expandafter\xdef\csname @pgfshading#1!\endcsname{\hbox to \the\pgf@xa{%
        \noexpand\vrule width0pt height\the\pgf@xa
        \noexpand\pgfsys@beginpurepicture
          \noexpand\pgfsys@invoke{%
            \pgf@domb\space \pgf@sys@tonumber{\pgf@y}
            \pgf@sys@tonumber{\pgf@x}
            \pgf@sys@tonumber{\pgf@max} pgfR1\pgf@shading@model\space
            \pgf@psfuncs\space \pgf@firstcolor\space \pgf@doma\space
            pgfR2\pgf@shading@model}%
          \hss
        \noexpand\pgfsys@endpurepicture}}%
    \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
      \pgf@shading@model}%
  }%
}
\def\pgfsys@functionalshading#1#2#3#4{%
  {%
    \pgf@process{#2}%
    \pgf@xa=\pgf@x
    \pgf@ya=\pgf@y
    \pgf@process{#3}%
    \pgf@xb=\pgf@x
    \pgf@yb=\pgf@y
    \advance\pgf@x by-\pgf@xa
    \advance\pgf@y by-\pgf@ya
    \pgf@sys@bp@correct{\pgf@xa}%
    \pgf@sys@bp@correct{\pgf@ya}%
    \pgf@sys@bp@correct{\pgf@xb}%
    \pgf@sys@bp@correct{\pgf@yb}%
    \pgf@xc=-\pgf@xa
    \pgf@yc=-\pgf@ya
    \expandafter\xdef\csname @pgfshading#1!\endcsname{%
      \hbox to\the\pgf@x{\vbox to\the\pgf@y{\vfil
          \noexpand\pgfsys@beginpurepicture
          \noexpand\pgfsys@invoke{%
            \pgf@sys@tonumber{\pgf@xc} \pgf@sys@tonumber{\pgf@yc} translate
            1.1 setlinewidth [] 0 setdash 0 setlinecap
            /pgfproc {#4} bind def
            \pgf@sys@tonumber{\pgf@ya} 1 \pgf@sys@tonumber{\pgf@yb}
      {
       \pgf@sys@tonumber{\pgf@xa} 1 \pgf@sys@tonumber{\pgf@xb}
       { 1 index 2 copy pgfproc \pgf@shading@ps@device\space
        moveto 1.1 0 rlineto stroke }
       for
       pop
      }
      for
     }%
    \noexpand\pgfsys@endpurepicture
   }\hfil}%
  }%
  \expandafter\xdef\csname @pgfshading#1@model!\endcsname{%
   \pgf@shading@model}%
 }%
}
```

PostScript® support code.

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

Define RGB PostScript® shading functions.

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

Define CMYK PostScript® shding functions.

```
 \pgf@sys@postscript@header{/pgfHcmyk{
   /pgfheight exch def 0.75 setlinewidth [] 0 setdash
   /pgfshade {pgfAcmyk} def /pgfdir { dup 0 moveto
    dup 6 index lineto } bind def} bind def}%
 \pgf@sys@postscript@header{/pgfVcmyk{
   /pgfheight exch def 0.75 setlinewidth [] 0 setdash
   /pgfshade {pgfAcmyk} def /pgfdir { dup 0 exch moveto dup 6 index
    exch lineto } bind def} bind def}%
 \pgf@sys@postscript@header{/pgfAcmyk{
   /pgfdiff 10 index round cvi 10 index round cvi sub 2 mul 1 add def
   3 index 8 index sub pgfdiff div % put cyan-step on stack
   3 index 8 index sub pgfdiff div % put magenta-step on stack
   3 index 8 index sub pgfdiff div % put yellow-step on stack
   3 index 8 index sub pgfdiff div % put black-step on stack
   pgfheight 12 index 12 index 12 index 12 index 18 index
   pgfdiff {
    4 index 4 index 4 index 4 index setcmykcolor % Set color
    pgfdir
    stroke
    5 -1 roll 9 index add % cyan += inccyan
    5 -1 roll 8 index add % magenta += incmagenta
    5 -1 roll 7 index add % yellow += incyellow
    5 -1 roll 6 index add % black += incblack
    5 -1 roll .5 sub % x += 0.5
   } repeat
   mark 19 1 roll cleartomark exch pop % leave only start x on stack
  }bind def }%
 \pgf@sys@postscript@header{/pgfR1cmyk{
   newpath dup dup dup 0 360 arc clip newpath
```

```
      dup /pgfendx exch def
      /pgfendy exch def
      0.875 setlinewidth
      [] 0 setdash
      /pgfshade {pgfRcmyk} def
      /pgfstartx exch def
      /pgfstarty exch def
      /pgfdiffx pgfendx pgfstartx sub def
      /pgfdiffy pgfendy pgfstarty sub def
      dup /pgfdomb exch def
    }bind def }%
  \pgf@sys@postscript@header{/pgfR2cmyk{
      newpath 0.5 add pgfcircx pgfcircy 3 2 roll 0 360 arc
      setcmykcolor fill pop}bind def }%
  \pgf@sys@postscript@header{/pgfRcmyk{
      /pgfdiff 10 index round cvi 10 index round cvi sub 4 mul 1 add def
      /pgfcircx pgfstartx 11 index pgfdiffx pgfdomb div mul add def
      /pgfcircy pgfstarty 11 index pgfdiffy pgfdomb div mul add def
      /pgfcircxe pgfstartx 10 index pgfdiffx pgfdomb div mul add def
      /pgfcircye pgfstarty 10 index pgfdiffy pgfdomb div mul add def
      /pgfxstep pgfcircxe pgfcircx sub pgfdiff div def
      /pgfystep pgfcircye pgfcircy sub pgfdiff div def
      3 index 8 index sub pgfdiff div % put cyan-step on stack
      3 index 8 index sub pgfdiff div % put magenta-step on stack
      3 index 8 index sub pgfdiff div % put yellow-step on stack
      3 index 8 index sub pgfdiff div % put black-step on stack
      11 index 11 index 11 index 11 index 17 index
      pgfdiff {
        4 index 4 index 4 index 4 index setcmykcolor % Set color
        pgfcircx pgfcircy 2 index 0 360 arc closepath
        stroke
        5 -1 roll 8 index add % cyan += inccyan
        5 -1 roll 7 index add % magenta += incmagenta
        5 -1 roll 6 index add % yellow += incyellow
        5 -1 roll 5 index add % black += incblack
        5 -1 roll .25 sub % x += 0.25
        /pgfcircx pgfcircx pgfxstep add def
        /pgfcircy pgfcircy pgfystep add def
      } repeat
      mark 18 1 roll cleartomark exch pop % leave only start x on stack
    }bind def}%
```

Define grayscale PostScript® shding functions.

```
  \pgf@sys@postscript@header{/pgfHgray{
      /pgfheight exch def 0.75 setlinewidth [] 0 setdash
      /pgfshade {pgfAgray} def /pgfdir { dup 0 moveto
        dup 3 index lineto } bind def} bind def}%
  \pgf@sys@postscript@header{/pgfVgray{
      /pgfheight exch def 0.75 setlinewidth [] 0 setdash
      /pgfshade {pgfAgray} def /pgfdir { dup 0 exch moveto dup 3 index
        exch lineto } bind def} bind def}%
  \pgf@sys@postscript@header{/pgfAgray{
      /pgfdiff 4 index round cvi 4 index round cvi sub 2 mul 1 add def
      dup 2 index sub pgfdiff div % put gray-step on stack
      pgfheight 3 index 6 index
      pgfdiff {
        1 index setgray % Set color
        pgfdir
        stroke
        exch 3 index add % gray += incgray
        exch .5 sub % x += 0.5
      } repeat
      mark 7 1 roll cleartomark exch pop % leave only start x on stack
    }bind def }%
  \pgf@sys@postscript@header{/pgfR1gray{
      newpath dup dup dup 0 360 arc clip newpath
      dup /pgfendx exch def
      /pgfendy exch def
      0.875 setlinewidth
      [] 0 setdash
      /pgfshade {pgfRgray} def
      /pgfstartx exch def
      /pgfstarty exch def
      /pgfdiffx pgfendx pgfstartx sub def
      /pgfdiffy pgfendy pgfstarty sub def
      dup /pgfdomb exch def
    }bind def }%
  \pgf@sys@postscript@header{/pgfR2gray{
      newpath 0.5 add pgfcircx pgfcircy 3 2 roll 0 360 arc
      setgray fill pop}bind def }%
  \pgf@sys@postscript@header{/pgfRgray{
      /pgfdiff 4 index round cvi 4 index round cvi sub 4 mul 1 add def
      /pgfcircx pgfstartx 5 index pgfdiffx pgfdomb div mul add def
      /pgfcircy pgfstarty 5 index pgfdiffy pgfdomb div mul add def
      /pgfcircxe pgfstartx 4 index pgfdiffx pgfdomb div mul add def
      /pgfcircye pgfstarty 4 index pgfdiffy pgfdomb div mul add def
      /pgfxstep pgfcircxe pgfcircx sub pgfdiff div def
      /pgfystep pgfcircye pgfcircy sub pgfdiff div def
      dup 2 index sub pgfdiff div % put gray-step on stack
      2 index 5 index
      pgfdiff {
        1 index setgray % Set color
        pgfcircx pgfcircy 2 index 0 360 arc closepath
        stroke
        exch 2 index add % gray += incgray
        exch .25 sub % x += 0.25
        /pgfcircx pgfcircx pgfxstep add def
        /pgfcircy pgfcircy pgfystep add def
      } repeat
      mark 6 1 roll cleartomark exch pop % leave only start x on stack
    }bind def}%
}
⟨/common-ps-driver⟩
```

Change History