gfx-php Documentation

Michael Billington

Jun 10, 2018

User Documentation

1 Features 3
2 Installation 5
3 Contribute 7
4 Navigation 9
5 Indices and tables 49

This library implements input, output and processing of raster images in pure PHP, so that image processing PHP extensions are not required.

This allows developers to eliminate some portability issues from their applications.

The basic usage is like this:

```php
<?php
use Mike42\GfxPhp\Image;
$img = Image::fromFile("colorwheel256.png");
$img -> write("test.gif");
```

CHAPTER 1

Features

- Format support includes PNG, GIF, BMP and the Netpbm formats.
- Support for scaling, cropping, format conversion and colorspace transformations.
- Pure PHP: This library does not require Gd, ImageMagick or GraphicsMagick extensions.

CHAPTER 2

Installation

Install gfx-php by running:

```
composer install mike42/gfx-php
```

CHAPTER 3

Contribute

- Issue Tracker: https://github.com/mike42/gfx-php/issues
- Source Code: https://github.com/mike42/gfx-php

CHAPTER 4

Navigation

4.1 File formats

- *Input formats*
  - *PNG*
  - *Netpbm Formats*
- *Output formats*
  - *PNG*
  - *GIF*
  - *BMP*
  - *Netpbm Formats*

4.1.1 Input formats

Files are read from a file or URL by using the *`Image::fromFile()`* function:

```
use Mike42\GfxPhp\Image;
$tux = Image::fromFile("tux.png")
```

If the your image is not being read from a file, then *`Image::fromBlob()`* can load it from a binary string:

```
use Mike42\GfxPhp\Image;
$tuxStr = "...";
$tux = Image::fromBlob($tuxStr, "tux.png");
```

In either case, the input format is determined using the file's magic number.

PNG

The PNG codec is used where the input has the `png` file extension.

All valid PNG files can be read, including:

- RGB or RGBA images
- Indexed images
- Monochrome images, from 1 to 16 bits per pixel
- Interlaced images

This library currently has limited support for transparency, and will discard any alpha channel from a PNG file when it is loaded.

Netpbm Formats

The Netpbm formats are a series of uncompressed bitmap formats, which can represent most types of image. These formats can be read by `gfx-php`:

PNM This is a file extension only. Files carrying `.pnm` extension can carry any of the below formats.

PPM This is a color raster format. A PPM file is identified by the P6 magic number, and will be loaded into an instance of *`RgbRasterImage`*.

PGM This is a monochrome raster format. A PGM file is identified by the P5 magic number, and will be loaded instance of *`GrayscaleRasterImage`*.

PBM This is a 1-bit bitmap format. A PBM file is identified by the P4 header, and loaded into an instance of *`BlackAndWhiteRasterImage`*.

Each of these formats has both a binary and text encoding. `gfx-php` only supports the binary encodings at this stage.

4.1.2 Output formats

When you write a *`RasterImage`* to a file, you need to specify a filename. The extension on this file is used to determine the desired output format.

There is currently no mechanism to write a file directly to a string.

PNG

The PNG format is selected by using the `png` file extension when you call *`RasterImage::write()`*.

```
$tux -> write("tux.png");
```

This library will currently output PNG files as RGB data. If you write to PNG from an instance of *`RgbRasterImage`*, then no conversion has to be done, so the output is significantly faster.

GIF

The GIF format is selected by using the `gif` file extension.

```
$tux -> write("tux.gif");
```

This format is limited to using a 256-color palette.

- If your image is not an *IndexedRasterImage*, then it is indexed when you write.
- If the image uses more than 256 colors, then it will be converted to an 8-bit RGB representation (3 bits red, 3 bits green, 2 bits blue), which introduces some distortions.

When you are creating GIF images, then you can avoid these conversions by using a `IndexedRasterImage` with a palette of fewer than 256 colors.

There is no encoder for multi-image GIF files at this stage.

BMP

The BMP format is selected by using the `bmp` file extension.

```
$tux -> write("tux.bmp");
```

This library will currently output BMP files using an uncompressed 24-bit RGB representation of the image.

Netpbm Formats

The Netpbm formats can be used for output. Each format is identified by their respective file extension:

```
$tux -> write("tux.ppm");
$tux -> write("tux.pgm";
$tux -> write("tux.pbm");
```

Since each of these formats has a different raster data representation, you should be aware that

PPM For this output format, the file is converted to a `RgbRasterImage` and typically written with a 24 bit color depth. In some cases, a 48 bit color depth will be used.

PGM The file is converted to a `GrayscaleRasterImage` and written with a depth of 8 or 16 bits per pixel.

PPM The file is converted to a `BlackAndWhiteRasterImage` and written with 1 bit per pixel.

If you want to avoid these conversions, then you should use the `pnm` extension to write your files. Since files with this extension can hold any of the above formats, the output encoder will avoid converting the raster data where possible.

```
$tux -> write("tux.pnm");
```

4.2 Image types

Every raster image in `gfx-php` implements the `RasterImage` interface.

There are several classes that implement this interface, which handle different representations of the image data.

`RgbRasterImage` Holds RGB data.

`GrayscaleRasterImage` Holds monochrome data.

`BlackAndWhiteRasterImage` Holds 1-bit raster data.

`IndexedRasterImage` Holds an image and associated palette.

4.2.1 Creating an image

Each of these classes has a static method which can be used to create an image of that type.

These only require a `width` and `height`.

```
use Mike42\GfxPhp\BlackAndWhiteRasterImage;
$image = BlackAndWhiteRasterImage::create(50, 100);
```

4.2.2 Converting between image types

You can convert between image types. This is similar to performing a *color-space conversion* in an image editor.

```
use Mike42\GfxPhp\Image;
$image = Image::load("tux.ppm");
$image -> toBlackAndWhite();
```

The methods to use are:

- *`RasterImage::toBlackAndWhite()`*
- *`RasterImage::toGrayscale()`*
- *`RasterImage::toIndexed()`*
- *`RasterImage::toRgb()`*

Each of these returns an image of the requested type. They work by instantiating a new image, then copying across the data as accurately as possible. As a result, the original image is unmodified.

4.2.3 Implicit conversions

Some file formats only accept specific types of raster data, so the *`RasterImage::write()`* method will need to convert it. For example, this `.pbm` will be limited to 2 colors, which is achieved by using `RasterImage:toBlackAndWhite` in the background:

```
use Mike42\GfxPhp\Image;
$wheel = Image::load("colorwheel.ppm");
$wheel -> write("wheel.pbm");
```

Since converting the color space creates a new image, the image stored in `$wheel` is unmodified.

4.3 Image operations

You can use this library to perform basic raster operations, such as:

- Crop
- Scale / resize
- Color-space conversions

See the `example/` sub-folder on GitHub for usage.

4.4 License

```
                  GNU LESSER GENERAL PUBLIC LICENSE
                       Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it.  You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

  When we speak of free software, we are referring to freedom of use,
not price.  Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

  To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights.  These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

  For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you.  You must make sure that they, too, receive or can get the source
code.  If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it.  And you must show them these terms so they know their rights.

  We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

  To protect each distributor, we want to make it very clear that
there is no warranty for the free library.  Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
```

(continues on next page)

(continued from previous page)

```
author's reputation will not be affected by problems that might be
introduced by others.

  Finally, software patents pose a constant threat to the existence of
any free program.  We wish to make sure that a company cannot
effectively restrict the users of a free program by obtaining a
restrictive license from a patent holder.  Therefore, we insist that
any patent license obtained for a version of the library must be
consistent with the full freedom of use specified in this license.

  Most GNU software, including some libraries, is covered by the
ordinary GNU General Public License.  This license, the GNU Lesser
General Public License, applies to certain designated libraries, and
is quite different from the ordinary General Public License.  We use
this license for certain libraries in order to permit linking those
libraries into non-free programs.

  When a program is linked with a library, whether statically or using
a shared library, the combination of the two is legally speaking a
combined work, a derivative of the original library.  The ordinary
General Public License therefore permits such linking only if the
entire combination fits its criteria of freedom.  The Lesser General
Public License permits more lax criteria for linking other code with
the library.

  We call this license the "Lesser" General Public License because it
does Less to protect the user's freedom than the ordinary General
Public License.  It also provides other free software developers Less
of an advantage over competing non-free programs.  These disadvantages
are the reason we use the ordinary General Public License for many
libraries.  However, the Lesser license provides advantages in certain
special circumstances.

  For example, on rare occasions, there may be a special need to
encourage the widest possible use of a certain library, so that it becomes
a de-facto standard.  To achieve this, non-free programs must be
allowed to use the library.  A more frequent case is that a free
library does the same job as widely used non-free libraries.  In this
case, there is little to gain by limiting the free library to free
software only, so we use the Lesser General Public License.

  In other cases, permission to use a particular library in non-free
programs enables a greater number of people to use a large body of
free software.  For example, permission to use the GNU C Library in
non-free programs enables many more people to use the whole GNU
operating system, as well as its variant, the GNU/Linux operating
system.

  Although the Lesser General Public License is Less protective of the
users' freedom, it does ensure that the user of a program that is
linked with the Library has the freedom and the wherewithal to run
that program using a modified version of the Library.

  The precise terms and conditions for copying, distribution and
modification follow.  Pay close attention to the difference between a
"work based on the library" and a "work that uses the library".  The
former contains code derived from the library, whereas the latter must
```

(continues on next page)

(continued from previous page)

```
be combined with the library in order to run.

                  GNU LESSER GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License Agreement applies to any software library or other
program which contains a notice placed by the copyright holder or
other authorized party saying it may be distributed under the terms of
this Lesser General Public License (also called "this License").
Each licensee is addressed as "you".

  A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

  The "Library", below, refers to any such software library or work
which has been distributed under these terms.  A "work based on the
Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language.  (Hereinafter, translation is
included without limitation in the term "modification".)

  "Source code" for a work means the preferred form of the work for
making modifications to it.  For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control compilation
and installation of the library.

  Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope.  The act of
running a program using the Library is not restricted, and output from
such a program is covered only if its contents constitute a work based
on the Library (independent of the use of the Library in a tool for
writing it).  Whether that is true depends on what the Library does
and what the program that uses the Library does.

  1. You may copy and distribute verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that
you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact
all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the
Library.

  You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a
fee.

  2. You may modify your copy or copies of the Library or any portion
of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

    a) The modified work must itself be a software library.

    b) You must cause the files modified to carry prominent notices
```

(continues on next page)

(continued from previous page)

```
    stating that you changed the files and the date of any change.

    c) You must cause the whole of the work to be licensed at no
    charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a
    table of data to be supplied by an application program that uses
    the facility, other than as an argument passed when the facility
    is invoked, then you must make a good faith effort to ensure that,
    in the event an application does not supply such function or
    table, the facility still operates, and performs whatever part of
    its purpose remains meaningful.

    (For example, a function in a library to compute square roots has
    a purpose that is entirely well-defined independent of the
    application.  Therefore, Subsection 2d requires that any
    application-supplied function or table used by this function must
    be optional: if the application does not supply it, the square
    root function must still compute square roots.)

These requirements apply to the modified work as a whole.  If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works.  But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library.  To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License.  (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.)  Do not make any other change in
these notices.

  Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.
```

(continues on next page)

(continued from previous page)

```
  4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library".  Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library".  The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library.  The
threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work.  (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

  Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

  6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

  You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License.  You must supply a copy of this License.  If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License.  Also, you must do one
```

(continues on next page)

(continued from previous page)

```
of these things:

    a) Accompany the work with the complete corresponding
    machine-readable source code for the Library including whatever
    changes were used in the work (which must be distributed under
    Sections 1 and 2 above); and, if the work is an executable linked
    with the Library, with the complete machine-readable "work that
    uses the Library", as object code and/or source code, so that the
    user can modify the Library and then relink to produce a modified
    executable containing the modified Library.  (It is understood
    that the user who changes the contents of definitions files in the
    Library will not necessarily be able to recompile the application
    to use the modified definitions.)

    b) Use a suitable shared library mechanism for linking with the
    Library.  A suitable mechanism is one that (1) uses at run time a
    copy of the library already present on the user's computer system,
    rather than copying library functions into the executable, and (2)
    will operate properly with a modified version of the library, if
    the user installs one, as long as the modified version is
    interface-compatible with the version that the work was made with.

    c) Accompany the work with a written offer, valid for at
    least three years, to give the same user the materials
    specified in Subsection 6a, above, for a charge no more
    than the cost of performing this distribution.

    d) If distribution of the work is made by offering access to copy
    from a designated place, offer equivalent access to copy the above
    specified materials from the same place.

    e) Verify that the user has already received a copy of these
    materials or that you have already sent this user a copy.

  For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it.  However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

  It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system.  Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

  7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

    a) Accompany the combined library with a copy of the same work
```

(continues on next page)

(continued from previous page)

```
    based on the Library, uncombined with any other library
    facilities.  This must be distributed under the terms of the
    Sections above.

    b) Give prominent notice with the combined library of the fact
    that part of it is a work based on the Library, and explaining
    where to find the accompanying uncombined form of the same work.

  8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License.  Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License.  However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

  9. You are not required to accept this License, since you have not
signed it.  However, nothing else grants you permission to modify or
distribute the Library or its derivative works.  These actions are
prohibited by law if you do not accept this License.  Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

  10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions.  You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

  11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all.  For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices.  Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
```

(continues on next page)

(continued from previous page)

```
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded.  In such case, this License incorporates the limitation as if
written in the body of this License.

  13. The Free Software Foundation may publish revised and/or new
versions of the Lesser General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number.  If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation.  If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

  14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission.  For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this.  Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

                            NO WARRANTY

  15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU.  SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
```

(continues on next page)

(continued from previous page)

```
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

                     END OF TERMS AND CONDITIONS

           How to Apply These Terms to Your New Libraries

  If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change.  You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

  To apply these terms, attach the following notices to the library.  It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301
    USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random
  Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!
```

4.5 Contributor guide

This project is open source, and is built on a voluntary basis by developers like you. You can help to improve it by contributing code & documentation to the repository, or by interacting with other developers and users on the issue tracker.

4.5.1 Where to ask a question

If you need to seek clarification, then you are welcome to post questions about the library to the gfx-php issue tracker. They will be tagged `question`, and closed once there is an answer.

If your question is incomplete or not specific to this project, then it will be tagged `invalid` and closed with a short explanation. We do this to keep the issue tracker focussed, on-topic and actionable by project contributors.

The StackOverflow help page How do I ask a good question? contains advice about writing complete questions.

4.5.2 How to report a bug

We track bugs as GitHub issues. If something does not work as documented, then you are welcome to make a bug report on the gfx-php issue tracker

When posting a bug:

- Check for open issues first. If the same issue has already been reported, then you should post any additional information to the existing thread.
- Describe what you are trying to do, and how the actual behaviour of the library differs from what you expected.
- Include a self-contained code snippet that demonstrates the issue, as PHP code formatted in a code block
- Try to use images from the repository to demonstrate the problem. If you need to use a specific example image, then attach it to the issue in a `.zip` file.

Bug reports will stay open as long as they are actionable. Generally, this means that they can be replicated on the current stable release, and there is some expectation that the issue is solvable.

The title and tags on your bug may be edited so that it can be distinguished from other bugs.

4.5.3 Feature requests

You can also make suggestions for new features on the gfx-php issue tracker. These are tagged `enhancement`.

Please keep the scope and resources of the project in mind when making suggestions.

4.5.4 Development process

The project is hosted online on the services below:

Code https://github.com/mike42/gfx-php

Continuous integration https://travis-ci.org/mike42/gfx-php

Code coverage reporting https://coveralls.io/github/mike42/gfx-php

Documentation https://gfx-php.readthedocs.io/

Package manager https://packagist.org/packages/mike42/gfx-php

For a change to be accepted, it will first need to meet some basic technical criteria, such as passing existing unit tests, a style check, and not breaking any of the examples.

Secondly, it will need to pass a human review, to confirm that it improves the overall product. You are encouraged to submit changes which address one open issue, so that this review can be as constructive as possible.

Commands to use locally

To make code changes, fork the repository on GitHub, and set up your local copy with composer.

```
composer install
```

To run unit tests, execute:

```
php vendor/bin/phpunit --coverage-text
```

To test all examples:

```
mkdir -p tmp && (cd tmp && find ../example -name '*.php' -print0 | xargs -n 1 -0 sh -c 'echo $0; php $0 || exit 255')
```

To run a style check and fix formatting issues:

```
php vendor/bin/phpcs --standard=psr2 src/ -n
php vendor/bin/phpcbf --standard=psr2 src/ -n
```

Submitting changes

Changes should be submitted as a GitHub pull request to the `master` branch.

Licensing considerations

You are not required to assign copyright for contributions to this project, but you do need ensure that your changes are suitable for release under the project's *copyleft* license.

If you hold the copyright to the submitted code, then indicate this in your pull request.

If you are thinking of including code which you do not hold the copyright to, please post relevant details to the issue tracker first. Only works which are licensed under an LGPL-2.1-compatible license may be combined with *gfx-php* code.

Release process

Git tags are automatically reflected as releases in packagist.

Release numbers approximately follow semantic versioning, and do not follow a particular schedule.

Updates are not typically made to old releases.

4.6 API documentation

4.6.1 Codec namespace

Codec::Common namespace

DataBlobInputStream

Qualified name `Mike42\GfxPhp\Codec\Common\DataBlobInputStream`

Implements *DataInputStream*

class DataBlobInputStream

__construct (*string $data*)

Parameters

- **$data** (*string*) –

advance (*int $bytes*)

Parameters

- **$bytes** (*int*) –

isEof ()

peek (*int $bytes*)

Parameters

- **$bytes** (*int*) –

read (*int $bytes*)

Parameters

- **$bytes** (*int*) –

static fromBlob (*string $blob*)

Parameters

- **$blob** (*string*) –

static fromFilename (*string $filename*)

Parameters

- **$filename** (*string*) –

DataInputStream

Qualified name `Mike42\GfxPhp\Codec\Common\DataInputStream`

interface DataInputStream

advance (*int $bytes*)

Parameters

- **$bytes** (*int*) –

isEof ()

peek (*int $bytes*)

Parameters

- **$bytes** (*int*) –

read (*int $bytes*)

Parameters

- **$bytes** (*int*) –

Codec::Png namespace

FilterDecoder

Qualified name `Mike42\GfxPhp\Codec\Png\FilterDecoder`

class FilterDecoder

unfilterImage (*string $binData*, *int $scanlineBytes*, *int $channels*, *int $bitDepth*)

Parameters

- **$binData** (*string*) –
- **$scanlineBytes** (*int*) –
- **$channels** (*int*) –
- **$bitDepth** (*int*) –

unfilterScanline (*array $currentFiltered*, *array $prior*, *int $filterType*, *int $bpp*)

Unfilter an individual scanline

Parameters

- **$currentFiltered** (*array*) –
- **$prior** (*array*) –
- **$filterType** (*int*) –
- **$bpp** (*int*) –

InterlaceDecoder

Qualified name `Mike42\GfxPhp\Codec\Png\InterlaceDecoder`

class InterlaceDecoder

__construct (*FilterDecoder $filterDecoder*)

Parameters

- **$filterDecoder** (FilterDecoder) –

decode (*PngHeader $header*, *string $binData*)

Parameters

- **$header** (PngHeader) –
- **$binData** (*string*) –

PngChunk

Qualified name `Mike42\GfxPhp\Codec\Png\PngChunk`

class PngChunk

__construct (*string $type*, *string $data*)

Parameters

- **$type** (*string*) –
- **$data** (*string*) –

getCrc ()

getData ()

getType ()

toBin ()

toString ()

static fromBin (*DataInputStream $in*)

Parameters

- **$in** (DataInputStream) –

static isValidChunkName (*string $name*)

Parameters

- **$name** (*string*) –

PngHeader

Qualified name `Mike42\GfxPhp\Codec\Png\PngHeader`

class PngHeader

__construct (*int $width*, *int $height*, *int $bitDepth*, *int $colorType*, *int $compression*, *int $filter*, *int $interlace*)

Parameters

- **$width** (*int*) –
- **$height** (*int*) –
- **$bitDepth** (*int*) –
- **$colorType** (*int*) –
- **$compression** (*int*) –
- **$filter** (*int*) –
- **$interlace** (*int*) –

allowsPalette ()

getBitDepth ()

getChannels()

getColorType()

getCompresssion()

getFilter()

getHeight()

getInterlace()

getWidth()

requiresPalette()

toString()

static fromChunk(*PngChunk $chunk*)

Parameters

- **$chunk** (PngChunk) –

PngImage

Qualified name `Mike42\GfxPhp\Codec\Png\PngImage`

class PngImage

toRasterImage()

static combineBytes16Bpp(*array $in*)

Takes 8-bit samples, and produces half as many 16-bit samples.

Parameters

- **$in** (*array*) –

static expandBytes1Bpp(*array $in*, *int $width*)

Takes 8-bit samples, and produces eight times as many 1-bit samples, dropping padding bits along the way.

Parameters

- **$in** (*array*) –
- **$width** (*int*) –

static expandBytes2Bpp(*array $in*, *int $width*)

Takes 8-bit samples, and produces four times as many 2-bit samples, dropping padding bits along the way.

Parameters

- **$in** (*array*) –
- **$width** (*int*) –

static expandBytes4Bpp(*array $in*, *int $width*)

Takes 8-bit samples, and produces twice as many 4-bit samples, dropping padding bits along the way.

Parameters

- **$in** (*array*) –
- **$width** (*int*) –

static fromBinary(*DataInputStream $data*)

Parameters

- **$data** (DataInputStream) –

BmpCodec

Qualified name `Mike42\GfxPhp\Codec\BmpCodec`

Implements *ImageEncoder*

class BmpCodec

encode(*RasterImage $image*, *string $format*)

Parameters

- **$image** (RasterImage) –
- **$format** (*string*) –

getEncodeFormats()

static getInstance

GifCodec

Qualified name `Mike42\GfxPhp\Codec\GifCodec`

Implements *ImageEncoder*

class GifCodec

encode(*RasterImage $image*, *string $format*)

Parameters

- **$image** (RasterImage) –
- **$format** (*string*) –

getEncodeFormats()

static getInstance

ImageCodec

Qualified name `Mike42\GfxPhp\Codec\ImageCodec`

class ImageCodec

__construct(*array $encoders*, *array $decoders*)

Parameters

- **$encoders** (*array*) –
- **$decoders** (*array*) –

getDecoderForFormat(*string $format*)

Parameters

- **`$format`** (*`string`*) –

`getEncoderForFormat` (*string $format*)

Parameters

- **`$format`** (*`string`*) –

`identify` (*string $blob*)

Parameters

- **`$blob`** (*`string`*) –

`static getInstance`

ImageDecoder

Qualified name `Mike42\GfxPhp\Codec\ImageDecoder`

`interface ImageDecoder`

`decode` (*string $blob*)

Parameters

- **`$blob`** (*`string`*) –

`getDecodeFormats` ()

`identify` (*string $blob*)

Parameters

- **`$blob`** (*`string`*) –

ImageEncoder

Qualified name `Mike42\GfxPhp\Codec\ImageEncoder`

`interface ImageEncoder`

`encode` (*RasterImage $image*, *string $format*)

Parameters

- **`$image`** (`RasterImage`) –
- **`$format`** (*`string`*) –

`getEncodeFormats` ()

PngCodec

Qualified name `Mike42\GfxPhp\Codec\PngCodec`

Implements *`ImageEncoder`*, *`ImageDecoder`*

`class PngCodec`

decode (*string $blob*)

Parameters

- **$blob** (*string*) –

encode (*RasterImage $image*, *string $format*)

Parameters

- **$image** (RasterImage) –
- **$format** (*string*) –

encodeRgb (*RgbRasterImage $image*)

Parameters

- **$image** (RgbRasterImage) –

getDecodeFormats ()

getEncodeFormats ()

identify (*string $blob*)

Parameters

- **$blob** (*string*) –

static getInstance

PnmCodec

Qualified name `Mike42\GfxPhp\Codec\PnmCodec`

Implements *ImageDecoder*, *ImageEncoder*

class PnmCodec

decode (*string $blob*)

Parameters

- **$blob** (*string*) –

encode (*RasterImage $image*, *string $format*)

Parameters

- **$image** (RasterImage) –
- **$format** (*string*) –

getDecodeFormats ()

getEncodeFormats ()

identify (*string $blob*)

Parameters

- **$blob** (*string*) –

static getInstance

4.6.2 Util namespace

AbstractLzwDictionary

Qualified name `Mike42\GfxPhp\Util\AbstractLzwDictionary`

class AbstractLzwDictionary

__construct (*int $minCodeSize*)

Parameters

- **$minCodeSize** (*int*) –

add (*string $entry*)

Parameters

- **$entry** (*string*) –

clear ()

getClearCode () → number

Returns number –

getEodCode () → number

Returns number –

getSize ()

LzwCompression

Utility classes to decode or encode entire strings.

Qualified name `Mike42\GfxPhp\Util\LzwCompression`

class LzwCompression

static compress (*string $inp*, *int $minCodeSize*)

Parameters

- **$inp** (*string*) –
- **$minCodeSize** (*int*) –

static decompress (*string $inp*, *int $minCodeSize*)

Parameters

- **$inp** (*string*) –
- **$minCodeSize** (*int*) –

LzwDecodeBuffer

Treat incoming string as a stream of bits

Qualified name `Mike42\GfxPhp\Util\LzwDecodeBuffer`

class LzwDecodeBuffer

__construct (*string $contents*)

Parameters

- **$contents** (*string*) –

read (*int $readBits*)

Parameters

- **$readBits** (*int*) – Number of bits to read

readBit (*int $i*)

Parameters

- **$i** (*int*) –

LzwDecodeDictionary

Qualified name `Mike42\GfxPhp\Util\LzwDecodeDictionary`

Extends *AbstractLzwDictionary*

class LzwDecodeDictionary

__construct (*int $minCodeSize*)

Parameters

- **$minCodeSize** (*int*) –

add (*string $entry*)

Parameters

- **$entry** (*string*) –

clear ()

contains (*int $code*)

Parameters

- **$code** (*int*) –

get (*int $code*)

Parameters

- **$code** (*int*) –

getClearCode () → number

Returns number –

getEodCode () → number

Returns number –

getSize ()

LzwEncodeBuffer

Qualified name `Mike42\GfxPhp\Util\LzwEncodeBuffer`

class LzwEncodeBuffer

__construct()

add(*int $code*, *int $bits*)

Parameters

- **$code** (*int*) –
- **$bits** (*int*) –

asString()

LzwEncodeDictionary

Qualified name `Mike42\GfxPhp\Util\LzwEncodeDictionary`

Extends *AbstractLzwDictionary*

class LzwEncodeDictionary

__construct(*int $minCodeSize*)

Parameters

- **$minCodeSize** (*int*) –

add(*string $entry*)

Parameters

- **$entry** (*string*) –

clear()

contains(*string $code*)

Parameters

- **$code** (*string*) –

get(*string $code*)

Parameters

- **$code** (*string*) –

getClearCode() → number

Returns number –

getEodCode() → number

Returns number –

getSize()

4.6.3 AbstractRasterImage

Qualified name `Mike42\GfxPhp\AbstractRasterImage`

Implements *RasterImage*

class AbstractRasterImage

compose (*RasterImage $source*, *int $startX*, *int $startY*, *int $destStartX*, *int $destStartY*, *int $width*, *int $height*)

Parameters

- **$source** (RasterImage) –
- **$startX** (*int*) –
- **$startY** (*int*) –
- **$destStartX** (*int*) –
- **$destStartY** (*int*) –
- **$width** (*int*) –
- **$height** (*int*) –

getHeight () → int

Get the height of the image in pixels.

Returns int – The height of the image in pixels.

getPixel (*int $x*, *int $y*) → int

Get the value of a given pixel. The meaning of the integer value of this pixel is implementation-dependent.

Parameters

- **$x** (*int*) – X co-ordinate
- **$y** (*int*) – Y co-ordinate

Returns int – The value of the pixel at ($x, $y).

getRasterData () → string

Get a binary string representing the underlying image data. The formatting of this data is implementation-dependent.

Returns string – A binary string representation of the raster data for this image.

getWidth () → int

Get the width of the image in pixels.

Returns int – The width of the image in pixels.

rect (*$startX*, *$startY*, *$width*, *$height*[, *$filled*, *$outline*, *$fill*])

Produce a rectangle with the given properties.

Parameters

- **$startX** –
- **$startY** –
- **$width** –
- **$height** –
- **$filled** – Default: `false`

- **`$outline`** – Default: `1`
- **`$fill`** (*`int`*) – Default: `1`

`scale`(*int $width*, *int $height*) → RasterImage

Produce a new *`RasterImage`* based on this one. The new image will be scaled to the requested dimensions via resampling.

Parameters

- **`$width`** (*`int`*) – The width of the returned image.
- **`$height`** (*`int`*) – The height of the returned image.

Returns *`RasterImage`* – A scaled version of the image.

`setPixel`(*int $x*, *int $y*, *int $value*)

Set the value of a given pixel.

Parameters

- **`$x`** (*`int`*) – X co-ordinate
- **`$y`** (*`int`*) – Y co-ordinate
- **`$value`** (*`int`*) – Value to set

`subImage`(*int $startX*, *int $startY*, *int $width*, *int $height*)

Parameters

- **`$startX`** (*`int`*) –
- **`$startY`** (*`int`*) –
- **`$width`** (*`int`*) –
- **`$height`** (*`int`*) –

`toBlackAndWhite`() → BlackAndWhiteRasterImage

Produce a copy of this *`RasterImage`* in a pure black-and-white colorspace.

Returns *`BlackAndWhiteRasterImage`* – a black and white version of the image.

`toGrayscale`() → GrayscaleRasterImage

Produce a copy of this *`RasterImage`* in a monochrome colorspace.

Returns *`GrayscaleRasterImage`* – A monochrome version of the image.

`toIndexed`() → IndexedRasterImage

Produce a copy of this *`RasterImage`* as an indexed image with an associated palette of unique colors.

Returns *`IndexedRasterImage`* – An paletted version of the image.

`toRgb`() → RgbRasterImage

Produce a copy of this *`RasterImage`* in the RGB colorspace.

Returns *`RgbRasterImage`* – An RGB version of the image.

`write`(*string $filename*)

Write the image to a file. The output format is determined by the file extension.

Parameters

- **`$filename`** (*`string`*) – Filename to write to.

4.6.4 BlackAndWhiteRasterImage

Small implementation of basic raster operations on PBM files to support creation of placeholder glyphs

Qualified name `Mike42\GfxPhp\BlackAndWhiteRasterImage`

Extends *AbstractRasterImage*

class BlackAndWhiteRasterImage

clear()

compose(*RasterImage $source*, *int $startX*, *int $startY*, *int $destStartX*, *int $destStartY*, *int $width*, *int $height*)

Parameters

- **$source** (RasterImage) –
- **$startX** (*int*) –
- **$startY** (*int*) –
- **$destStartX** (*int*) –
- **$destStartY** (*int*) –
- **$width** (*int*) –
- **$height** (*int*) –

getHeight() → int

Get the height of the image in pixels.

Returns int – The height of the image in pixels.

getPixel(*int $x*, *int $y*) → int

Get the value of a given pixel. The meaning of the integer value of this pixel is implementation-dependent.

Parameters

- **$x** (*int*) – X co-ordinate
- **$y** (*int*) – Y co-ordinate

Returns int – The value of the pixel at ($x, $y).

getRasterData() → string

Get a binary string representing the underlying image data. The formatting of this data is implementation-dependent.

Returns string – A binary string representation of the raster data for this image.

getWidth() → int

Get the width of the image in pixels.

Returns int – The width of the image in pixels.

invert()

mapColor(*int $srcColor*, *RasterImage $destImage*)

Parameters

- **$srcColor** (*int*) –
- **$destImage** (RasterImage) –

rect (*$startX*, *$startY*, *$width*, *$height*[, *$filled*, *$outline*, *$fill*])
Produce a rectangle with the given properties.

Parameters

- **$startX** –
- **$startY** –
- **$width** –
- **$height** –
- **$filled** – Default: `false`
- **$outline** – Default: `1`
- **$fill** (*int*) – Default: `1`

scale (*int $width*, *int $height*) → RasterImage
Produce a new `RasterImage` based on this one. The new image will be scaled to the requested dimensions via resampling.

Parameters

- **$width** (*int*) – The width of the returned image.
- **$height** (*int*) – The height of the returned image.

Returns `RasterImage` – A scaled version of the image.

setPixel (*int $x*, *int $y*, *int $value*)
Set the value of a given pixel.

Parameters

- **$x** (*int*) – X co-ordinate
- **$y** (*int*) – Y co-ordinate
- **$value** (*int*) – Value to set

subImage (*int $startX*, *int $startY*, *int $width*, *int $height*)

Parameters

- **$startX** (*int*) –
- **$startY** (*int*) –
- **$width** (*int*) –
- **$height** (*int*) –

toBlackAndWhite () → BlackAndWhiteRasterImage
Produce a copy of this `RasterImage` in a pure black-and-white colorspace.

Returns `BlackAndWhiteRasterImage` – a black and white version of the image.

toGrayscale () → GrayscaleRasterImage
Produce a copy of this `RasterImage` in a monochrome colorspace.

Returns `GrayscaleRasterImage` – A monochrome version of the image.

toIndexed () → IndexedRasterImage
Produce a copy of this `RasterImage` as an indexed image with an associated palette of unique colors.

Returns `IndexedRasterImage` – An paletted version of the image.

toRgb() → RgbRasterImage

Produce a copy of this `RasterImage` in the RGB colorspace.

Returns `RgbRasterImage` – An RGB version of the image.

toString()

write(*string $filename*)

Write the image to a file. The output format is determined by the file extension.

Parameters

- **$filename** (*string*) – Filename to write to.

static create(*$width*, *$height*[, *array $data*])

Parameters

- **$width** –
- **$height** –
- **$data** (*array*) – Default: `null`

4.6.5 GrayscaleRasterImage

Qualified name `Mike42\GfxPhp\GrayscaleRasterImage`

Extends `AbstractRasterImage`

class GrayscaleRasterImage

compose(*RasterImage $source*, *int $startX*, *int $startY*, *int $destStartX*, *int $destStartY*, *int $width*, *int $height*)

Parameters

- **$source** (`RasterImage`) –
- **$startX** (*int*) –
- **$startY** (*int*) –
- **$destStartX** (*int*) –
- **$destStartY** (*int*) –
- **$width** (*int*) –
- **$height** (*int*) –

getHeight() → int

Get the height of the image in pixels.

Returns int – The height of the image in pixels.

getMaxVal()

getPixel(*int $x*, *int $y*) → int

Get the value of a given pixel. The meaning of the integer value of this pixel is implementation-dependent.

Parameters

- **$x** (*int*) – X co-ordinate
- **$y** (*int*) – Y co-ordinate

Returns int – The value of the pixel at ($x, $y).

getRasterData() → string

Get a binary string representing the underlying image data. The formatting of this data is implementation-dependent.

Returns string – A binary string representation of the raster data for this image.

getWidth() → int

Get the width of the image in pixels.

Returns int – The width of the image in pixels.

mapColor(*int $srcColor*, *RasterImage $destImage*)

Parameters

- **$srcColor** (*int*) –
- **$destImage** (RasterImage) –

rect(*$startX*, *$startY*, *$width*, *$height*[, *$filled*, *$outline*, *$fill*])

Produce a rectangle with the given properties.

Parameters

- **$startX** –
- **$startY** –
- **$width** –
- **$height** –
- **$filled** – Default: `false`
- **$outline** – Default: `1`
- **$fill** (*int*) – Default: `1`

scale(*int $width*, *int $height*) → RasterImage

Produce a new *RasterImage* based on this one. The new image will be scaled to the requested dimensions via resampling.

Parameters

- **$width** (*int*) – The width of the returned image.
- **$height** (*int*) – The height of the returned image.

Returns *RasterImage* – A scaled version of the image.

setPixel(*int $x*, *int $y*, *int $value*)

Set the value of a given pixel.

Parameters

- **$x** (*int*) – X co-ordinate
- **$y** (*int*) – Y co-ordinate
- **$value** (*int*) – Value to set

subImage(*int $startX*, *int $startY*, *int $width*, *int $height*)

Parameters

- **$startX** (*int*) –

- **$startY** (*int*) –
- **$width** (*int*) –
- **$height** (*int*) –

toBlackAndWhite() → BlackAndWhiteRasterImage

Produce a copy of this `RasterImage` in a pure black-and-white colorspace.

Returns `BlackAndWhiteRasterImage` – a black and white version of the image.

toGrayscale() → GrayscaleRasterImage

Produce a copy of this `RasterImage` in a monochrome colorspace.

Returns `GrayscaleRasterImage` – A monochrome version of the image.

toIndexed() → IndexedRasterImage

Produce a copy of this `RasterImage` as an indexed image with an associated palette of unique colors.

Returns `IndexedRasterImage` – An paletted version of the image.

toRgb() → RgbRasterImage

Produce a copy of this `RasterImage` in the RGB colorspace.

Returns `RgbRasterImage` – An RGB version of the image.

write(*string $filename*)

Write the image to a file. The output format is determined by the file extension.

Parameters

- **$filename** (*string*) – Filename to write to.

static create(*$width*, *$height*[, *array $data*, *$maxVal*])

Parameters

- **$width** –
- **$height** –
- **$data** (*array*) – Default: `null`
- **$maxVal** – Default: `255`

4.6.6 Image

Qualified name `Mike42\GfxPhp\Image`

class Image

static create(*int $width*, *int $height*[, *int $impl*])

Parameters

- **$width** (*int*) –
- **$height** (*int*) –
- **$impl** (*int*) – Default: `self::IMAGE_BLACK_WHITE`

static fromBlob(*string $blob*[, *string $filename*])

Parameters

- **$blob** (*string*) –

- **$filename** (*string*) – Default: `null`

static fromFile (*string $filename*)

Parameters

- **$filename** (*string*) –

4.6.7 IndexedRasterImage

Qualified name `Mike42\GfxPhp\IndexedRasterImage`

Extends *AbstractRasterImage*

class IndexedRasterImage

allocateColor (*array $color*)

Parameters

- **$color** (*array*) –

compose (*RasterImage $source*, *int $startX*, *int $startY*, *int $destStartX*, *int $destStartY*, *int $width*, *int $height*)

Parameters

- **$source** (RasterImage) –
- **$startX** (*int*) –
- **$startY** (*int*) –
- **$destStartX** (*int*) –
- **$destStartY** (*int*) –
- **$width** (*int*) –
- **$height** (*int*) –

deallocateColor (*array $color*)

Parameters

- **$color** (*array*) –

getHeight () → int

Get the height of the image in pixels.

Returns int – The height of the image in pixels.

getMaxVal ()

getPalette ()

getPixel (*int $x*, *int $y*) → int

Get the value of a given pixel. The meaning of the integer value of this pixel is implementation-dependent.

Parameters

- **$x** (*int*) – X co-ordinate
- **$y** (*int*) – Y co-ordinate

Returns int – The value of the pixel at ($x, $y).

getRasterData() → string

Get a binary string representing the underlying image data. The formatting of this data is implementation-dependent.

Returns string – A binary string representation of the raster data for this image.

getTransparentColor()

getWidth() → int

Get the width of the image in pixels.

Returns int – The width of the image in pixels.

indexToRgb(*int $index*)

Parameters

- **$index** (*int*) –

rect(*$startX*, *$startY*, *$width*, *$height*[, *$filled*, *$outline*, *$fill*])

Produce a rectangle with the given properties.

Parameters

- **$startX** –
- **$startY** –
- **$width** –
- **$height** –
- **$filled** – Default: `false`
- **$outline** – Default: `1`
- **$fill** (*int*) – Default: `1`

rgbToIndex(*array $rgb*)

Parameters

- **$rgb** (*array*) –

scale(*int $width*, *int $height*) → RasterImage

Produce a new *RasterImage* based on this one. The new image will be scaled to the requested dimensions via resampling.

Parameters

- **$width** (*int*) – The width of the returned image.
- **$height** (*int*) – The height of the returned image.

Returns *RasterImage* – A scaled version of the image.

setMaxVal(*int $maxVal*)

Parameters

- **$maxVal** (*int*) –

setPalette(*array $palette*)

Parameters

- **$palette** (*array*) –

setPixel(*int $x*, *int $y*, *int $value*)
Set the value of a given pixel.

Parameters

- **$x** (*int*) – X co-ordinate
- **$y** (*int*) – Y co-ordinate
- **$value** (*int*) – Value to set

setTransparentColor([ ])

Parameters

- **$color** (*int*) – Default: `null`

subImage(*int $startX*, *int $startY*, *int $width*, *int $height*)

Parameters

- **$startX** (*int*) –
- **$startY** (*int*) –
- **$width** (*int*) –
- **$height** (*int*) –

toBlackAndWhite() → BlackAndWhiteRasterImage
Produce a copy of this *RasterImage* in a pure black-and-white colorspace.

Returns *BlackAndWhiteRasterImage* – a black and white version of the image.

toGrayscale() → GrayscaleRasterImage
Produce a copy of this *RasterImage* in a monochrome colorspace.

Returns *GrayscaleRasterImage* – A monochrome version of the image.

toIndexed() → IndexedRasterImage
Produce a copy of this *RasterImage* as an indexed image with an associated palette of unique colors.

Returns *IndexedRasterImage* – An paletted version of the image.

toRgb() → RgbRasterImage
Produce a copy of this *RasterImage* in the RGB colorspace.

Returns *RgbRasterImage* – An RGB version of the image.

write(*string $filename*)
Write the image to a file. The output format is determined by the file extension.

Parameters

- **$filename** (*string*) – Filename to write to.

static create(*int $width*, *int $height*[, *array $data*, *array $palette*, *int $maxVal*])

Parameters

- **$width** (*int*) –
- **$height** (*int*) –
- **$data** (*array*) – Default: `null`
- **$palette** (*array*) – Default: `[]`
- **$maxVal** (*int*) – Default: `255`

4.6.8 PaletteGenerator

Qualified name `Mike42\GfxPhp\PaletteGenerator`

class PaletteGenerator

static blackAndWhitePalette

static colorPalette

static monochromePalette

static whitePalette

4.6.9 RasterImage

Generic interface to raster images.

Qualified name `Mike42\GfxPhp\RasterImage`

interface RasterImage

getHeight() → int

Get the height of the image in pixels.

Returns int – The height of the image in pixels.

getPixel(*int $x*, *int $y*) → int

Get the value of a given pixel. The meaning of the integer value of this pixel is implementation-dependent.

Parameters

- **$x** (*int*) – X co-ordinate
- **$y** (*int*) – Y co-ordinate

Returns int – The value of the pixel at ($x, $y).

getRasterData() → string

Get a binary string representing the underlying image data. The formatting of this data is implementation-dependent.

Returns string – A binary string representation of the raster data for this image.

getWidth() → int

Get the width of the image in pixels.

Returns int – The width of the image in pixels.

scale(*int $width*, *int $height*) → RasterImage

Produce a new *RasterImage* based on this one. The new image will be scaled to the requested dimensions via resampling.

Parameters

- **$width** (*int*) – The width of the returned image.
- **$height** (*int*) – The height of the returned image.

Returns *RasterImage* – A scaled version of the image.

setPixel(*int $x*, *int $y*, *int $value*)

Set the value of a given pixel.

Parameters

- **`$x`** (*`int`*) – X co-ordinate
- **`$y`** (*`int`*) – Y co-ordinate
- **`$value`** (*`int`*) – Value to set

`toBlackAndWhite``()` → BlackAndWhiteRasterImage

Produce a copy of this *`RasterImage`* in a pure black-and-white colorspace.

Returns *`BlackAndWhiteRasterImage`* – a black and white version of the image.

`toGrayscale``()` → GrayscaleRasterImage

Produce a copy of this *`RasterImage`* in a monochrome colorspace.

Returns *`GrayscaleRasterImage`* – A monochrome version of the image.

`toIndexed``()` → IndexedRasterImage

Produce a copy of this *`RasterImage`* as an indexed image with an associated palette of unique colors.

Returns *`IndexedRasterImage`* – An paletted version of the image.

`toRgb``()` → RgbRasterImage

Produce a copy of this *`RasterImage`* in the RGB colorspace.

Returns *`RgbRasterImage`* – An RGB version of the image.

`write``(`*string $filename*`)`

Write the image to a file. The output format is determined by the file extension.

Parameters

- **`$filename`** (*`string`*) – Filename to write to.

4.6.10 RgbRasterImage

Qualified name `Mike42\GfxPhp\RgbRasterImage`

Extends *`AbstractRasterImage`*

`class RgbRasterImage`

`compose``(`*RasterImage $source*, *int $startX*, *int $startY*, *int $destStartX*, *int $destStartY*, *int $width*, *int $height*`)`

Parameters

- **`$source`** (`RasterImage`) –
- **`$startX`** (*`int`*) –
- **`$startY`** (*`int`*) –
- **`$destStartX`** (*`int`*) –
- **`$destStartY`** (*`int`*) –
- **`$width`** (*`int`*) –
- **`$height`** (*`int`*) –

`getHeight``()` → int

Get the height of the image in pixels.

Returns int – The height of the image in pixels.

getMaxVal()

getPixel(*int $x*, *int $y*) → int

Get the value of a given pixel. The meaning of the integer value of this pixel is implementation-dependent.

Parameters

- **$x** (*int*) – X co-ordinate
- **$y** (*int*) – Y co-ordinate

Returns int – The value of the pixel at ($x, $y).

getRasterData() → string

Get a binary string representing the underlying image data. The formatting of this data is implementation-dependent.

Returns string – A binary string representation of the raster data for this image.

getWidth() → int

Get the width of the image in pixels.

Returns int – The width of the image in pixels.

indexToRgb(*int $val*)

Parameters

- **$val** (*int*) –

mapColor(*int $srcColor*, *RasterImage $destImage*)

Parameters

- **$srcColor** (*int*) –
- **$destImage** (RasterImage) –

rect(*$startX*, *$startY*, *$width*, *$height*[, *$filled*, *$outline*, *$fill*])

Produce a rectangle with the given properties.

Parameters

- **$startX** –
- **$startY** –
- **$width** –
- **$height** –
- **$filled** – Default: `false`
- **$outline** – Default: `1`
- **$fill** (*int*) – Default: `1`

rgbToIndex(*array $val*)

Parameters

- **$val** (*array*) –

scale(*int $width*, *int $height*) → RasterImage

Produce a new *RasterImage* based on this one. The new image will be scaled to the requested dimensions via resampling.

Parameters

- **$width** (*int*) – The width of the returned image.
- **$height** (*int*) – The height of the returned image.

Returns *RasterImage* – A scaled version of the image.

setPixel (*int $x*, *int $y*, *int $value*)
Set the value of a given pixel.

Parameters

- **$x** (*int*) – X co-ordinate
- **$y** (*int*) – Y co-ordinate
- **$value** (*int*) – Value to set

subImage (*int $startX*, *int $startY*, *int $width*, *int $height*)

Parameters

- **$startX** (*int*) –
- **$startY** (*int*) –
- **$width** (*int*) –
- **$height** (*int*) –

toBlackAndWhite () → BlackAndWhiteRasterImage
Produce a copy of this *RasterImage* in a pure black-and-white colorspace.

Returns *BlackAndWhiteRasterImage* – a black and white version of the image.

toGrayscale () → GrayscaleRasterImage
Produce a copy of this *RasterImage* in a monochrome colorspace.

Returns *GrayscaleRasterImage* – A monochrome version of the image.

toIndexed () → IndexedRasterImage
Produce a copy of this *RasterImage* as an indexed image with an associated palette of unique colors.

Returns *IndexedRasterImage* – An paletted version of the image.

toRgb () → RgbRasterImage
Produce a copy of this *RasterImage* in the RGB colorspace.

Returns *RgbRasterImage* – An RGB version of the image.

write (*string $filename*)
Write the image to a file. The output format is determined by the file extension.

Parameters

- **$filename** (*string*) – Filename to write to.

static convertDepth (*& $item*, *$key*, *array $data*)

Parameters

- **$item** (*&*) –
- **$key** –
- **$data** (*array*) –

static create (*$width*, *$height*[, *array $data*, *$maxVal*])

Parameters

- **`$width`** –
- **`$height`** –
- **`$data`** (*`array`*) – Default: `null`
- **`$maxVal`** – Default: `255`

`static intToRgb`(*$in*)

Parameters

- **`$in`** –

`static rgbToInt`(*int $r*, *int $g*, *int $b*)

Parameters

- **`$r`** (*`int`*) –
- **`$g`** (*`int`*) –
- **`$b`** (*`int`*) –

CHAPTER 5

Indices and tables

- genindex
- search

Index

Symbols

__construct() (AbstractLzwDictionary method), 31
__construct() (DataBlobInputStream method), 24
__construct() (ImageCodec method), 28
__construct() (InterlaceDecoder method), 25
__construct() (LzwDecodeBuffer method), 32
__construct() (LzwDecodeDictionary method), 32
__construct() (LzwEncodeBuffer method), 33
__construct() (LzwEncodeDictionary method), 33
__construct() (PngChunk method), 26
__construct() (PngHeader method), 26

A

AbstractLzwDictionary (class), 31
AbstractRasterImage (class), 34
add() (AbstractLzwDictionary method), 31
add() (LzwDecodeDictionary method), 32
add() (LzwEncodeBuffer method), 33
add() (LzwEncodeDictionary method), 33
advance() (DataBlobInputStream method), 24
advance() (DataInputStream method), 24
allocateColor() (IndexedRasterImage method), 41
allowsPalette() (PngHeader method), 26
asString() (LzwEncodeBuffer method), 33

B

blackAndWhitePalette() (PaletteGenerator method), 44
BlackAndWhiteRasterImage (class), 36
BmpCodec (class), 28

C

clear() (AbstractLzwDictionary method), 31
clear() (BlackAndWhiteRasterImage method), 36
clear() (LzwDecodeDictionary method), 32
clear() (LzwEncodeDictionary method), 33
colorPalette() (PaletteGenerator method), 44
combineBytes16Bpp() (PngImage method), 27
compose() (AbstractRasterImage method), 34
compose() (BlackAndWhiteRasterImage method), 36
compose() (GrayscaleRasterImage method), 38
compose() (IndexedRasterImage method), 41
compose() (RgbRasterImage method), 45
compress() (LzwCompression method), 31
contains() (LzwDecodeDictionary method), 32
contains() (LzwEncodeDictionary method), 33
convertDepth() (RgbRasterImage method), 47
create() (BlackAndWhiteRasterImage method), 38
create() (GrayscaleRasterImage method), 40
create() (Image method), 40
create() (IndexedRasterImage method), 43
create() (RgbRasterImage method), 47

D

DataBlobInputStream (class), 24
DataInputStream (interface), 24
deallocateColor() (IndexedRasterImage method), 41
decode() (ImageDecoder method), 29
decode() (InterlaceDecoder method), 25
decode() (PngCodec method), 29
decode() (PnmCodec method), 30
decompress() (LzwCompression method), 31

E

encode() (BmpCodec method), 28
encode() (GifCodec method), 28
encode() (ImageEncoder method), 29
encode() (PngCodec method), 30
encode() (PnmCodec method), 30
encodeRgb() (PngCodec method), 30
expandBytes1Bpp() (PngImage method), 27
expandBytes2Bpp() (PngImage method), 27
expandBytes4Bpp() (PngImage method), 27

F

FilterDecoder (class), 25
fromBin() (PngChunk method), 26
fromBinary() (PngImage method), 27
fromBlob() (DataBlobInputStream method), 24

fromBlob() (Image method), 40
fromChunk() (PngHeader method), 27
fromFile() (Image method), 41
fromFilename() (DataBlobInputStream method), 24

G

get() (LzwDecodeDictionary method), 32
get() (LzwEncodeDictionary method), 33
getBitDepth() (PngHeader method), 26
getChannels() (PngHeader method), 26
getClearCode() (AbstractLzwDictionary method), 31
getClearCode() (LzwDecodeDictionary method), 32
getClearCode() (LzwEncodeDictionary method), 33
getColorType() (PngHeader method), 27
getCompresssion() (PngHeader method), 27
getCrc() (PngChunk method), 26
getData() (PngChunk method), 26
getDecodeFormats() (ImageDecoder method), 29
getDecodeFormats() (PngCodec method), 30
getDecodeFormats() (PnmCodec method), 30
getDecoderForFormat() (ImageCodec method), 28
getEncodeFormats() (BmpCodec method), 28
getEncodeFormats() (GifCodec method), 28
getEncodeFormats() (ImageEncoder method), 29
getEncodeFormats() (PngCodec method), 30
getEncodeFormats() (PnmCodec method), 30
getEncoderForFormat() (ImageCodec method), 29
getEodCode() (AbstractLzwDictionary method), 31
getEodCode() (LzwDecodeDictionary method), 32
getEodCode() (LzwEncodeDictionary method), 33
getFilter() (PngHeader method), 27
getHeight() (AbstractRasterImage method), 34
getHeight() (BlackAndWhiteRasterImage method), 36
getHeight() (GrayscaleRasterImage method), 38
getHeight() (IndexedRasterImage method), 41
getHeight() (PngHeader method), 27
getHeight() (RasterImage method), 44
getHeight() (RgbRasterImage method), 45
getInstance() (BmpCodec method), 28
getInstance() (GifCodec method), 28
getInstance() (ImageCodec method), 29
getInstance() (PngCodec method), 30
getInstance() (PnmCodec method), 30
getInterlace() (PngHeader method), 27
getMaxVal() (GrayscaleRasterImage method), 38
getMaxVal() (IndexedRasterImage method), 41
getMaxVal() (RgbRasterImage method), 45
getPalette() (IndexedRasterImage method), 41
getPixel() (AbstractRasterImage method), 34
getPixel() (BlackAndWhiteRasterImage method), 36
getPixel() (GrayscaleRasterImage method), 38
getPixel() (IndexedRasterImage method), 41
getPixel() (RasterImage method), 44
getPixel() (RgbRasterImage method), 46
getRasterData() (AbstractRasterImage method), 34
getRasterData() (BlackAndWhiteRasterImage method), 36
getRasterData() (GrayscaleRasterImage method), 39
getRasterData() (IndexedRasterImage method), 41
getRasterData() (RasterImage method), 44
getRasterData() (RgbRasterImage method), 46
getSize() (AbstractLzwDictionary method), 31
getSize() (LzwDecodeDictionary method), 32
getSize() (LzwEncodeDictionary method), 33
getTransparentColor() (IndexedRasterImage method), 42
getType() (PngChunk method), 26
getWidth() (AbstractRasterImage method), 34
getWidth() (BlackAndWhiteRasterImage method), 36
getWidth() (GrayscaleRasterImage method), 39
getWidth() (IndexedRasterImage method), 42
getWidth() (PngHeader method), 27
getWidth() (RasterImage method), 44
getWidth() (RgbRasterImage method), 46
GifCodec (class), 28
GrayscaleRasterImage (class), 38

I

identify() (ImageCodec method), 29
identify() (ImageDecoder method), 29
identify() (PngCodec method), 30
identify() (PnmCodec method), 30
Image (class), 40
ImageCodec (class), 28
ImageDecoder (interface), 29
ImageEncoder (interface), 29
IndexedRasterImage (class), 41
indexToRgb() (IndexedRasterImage method), 42
indexToRgb() (RgbRasterImage method), 46
InterlaceDecoder (class), 25
intToRgb() (RgbRasterImage method), 48
invert() (BlackAndWhiteRasterImage method), 36
isEof() (DataBlobInputStream method), 24
isEof() (DataInputStream method), 24
isValidChunkName() (PngChunk method), 26

L

LzwCompression (class), 31
LzwDecodeBuffer (class), 31
LzwDecodeDictionary (class), 32
LzwEncodeBuffer (class), 33
LzwEncodeDictionary (class), 33

M

mapColor() (BlackAndWhiteRasterImage method), 36
mapColor() (GrayscaleRasterImage method), 39
mapColor() (RgbRasterImage method), 46
monochromePalette() (PaletteGenerator method), 44

P

PaletteGenerator (class), 44
peek() (DataBlobInputStream method), 24
peek() (DataInputStream method), 24
PngChunk (class), 26
PngCodec (class), 29
PngHeader (class), 26
PngImage (class), 27
PnmCodec (class), 30

R

RasterImage (interface), 44
read() (DataBlobInputStream method), 24
read() (DataInputStream method), 24
read() (LzwDecodeBuffer method), 32
readBit() (LzwDecodeBuffer method), 32
rect() (AbstractRasterImage method), 34
rect() (BlackAndWhiteRasterImage method), 36
rect() (GrayscaleRasterImage method), 39
rect() (IndexedRasterImage method), 42
rect() (RgbRasterImage method), 46
requiresPalette() (PngHeader method), 27
RgbRasterImage (class), 45
rgbToIndex() (IndexedRasterImage method), 42
rgbToIndex() (RgbRasterImage method), 46
rgbToInt() (RgbRasterImage method), 48

S

scale() (AbstractRasterImage method), 35
scale() (BlackAndWhiteRasterImage method), 37
scale() (GrayscaleRasterImage method), 39
scale() (IndexedRasterImage method), 42
scale() (RasterImage method), 44
scale() (RgbRasterImage method), 46
setMaxVal() (IndexedRasterImage method), 42
setPalette() (IndexedRasterImage method), 42
setPixel() (AbstractRasterImage method), 35
setPixel() (BlackAndWhiteRasterImage method), 37
setPixel() (GrayscaleRasterImage method), 39
setPixel() (IndexedRasterImage method), 42
setPixel() (RasterImage method), 44
setPixel() (RgbRasterImage method), 47
setTransparentColor() (IndexedRasterImage method), 43
subImage() (AbstractRasterImage method), 35
subImage() (BlackAndWhiteRasterImage method), 37
subImage() (GrayscaleRasterImage method), 39
subImage() (IndexedRasterImage method), 43
subImage() (RgbRasterImage method), 47

T

toBin() (PngChunk method), 26
toBlackAndWhite() (AbstractRasterImage method), 35
toBlackAndWhite() (BlackAndWhiteRasterImage method), 37
toBlackAndWhite() (GrayscaleRasterImage method), 40
toBlackAndWhite() (IndexedRasterImage method), 43
toBlackAndWhite() (RasterImage method), 45
toBlackAndWhite() (RgbRasterImage method), 47
toGrayscale() (AbstractRasterImage method), 35
toGrayscale() (BlackAndWhiteRasterImage method), 37
toGrayscale() (GrayscaleRasterImage method), 40
toGrayscale() (IndexedRasterImage method), 43
toGrayscale() (RasterImage method), 45
toGrayscale() (RgbRasterImage method), 47
toIndexed() (AbstractRasterImage method), 35
toIndexed() (BlackAndWhiteRasterImage method), 37
toIndexed() (GrayscaleRasterImage method), 40
toIndexed() (IndexedRasterImage method), 43
toIndexed() (RasterImage method), 45
toIndexed() (RgbRasterImage method), 47
toRasterImage() (PngImage method), 27
toRgb() (AbstractRasterImage method), 35
toRgb() (BlackAndWhiteRasterImage method), 37
toRgb() (GrayscaleRasterImage method), 40
toRgb() (IndexedRasterImage method), 43
toRgb() (RasterImage method), 45
toRgb() (RgbRasterImage method), 47
toString() (BlackAndWhiteRasterImage method), 38
toString() (PngChunk method), 26
toString() (PngHeader method), 27

U

unfilterImage() (FilterDecoder method), 25
unfilterScanline() (FilterDecoder method), 25

W

whitePalette() (PaletteGenerator method), 44
write() (AbstractRasterImage method), 35
write() (BlackAndWhiteRasterImage method), 38
write() (GrayscaleRasterImage method), 40
write() (IndexedRasterImage method), 43
write() (RasterImage method), 45
write() (RgbRasterImage method), 47