Butterfly

SimpleAuto

Home Download license Contact Me

What is Butterfly?

Butterfly is a enhanced version of FLY(by Martin Gleeson), code by C as a command-file interface that creates and modifies BMP, PNG, JPEG,TIFF,WBMP,TGA, BIN or GIF images. Using Thomas Boutell's gd graphics library with some enhanced functions for fast image creation and modification.

Contents

What different between Butterfly and Fly

goto top

- **Fly provided 256 color graphic only,**
- **Butterfly uses truecolor for Bmp, Jpeg, Tga, Bin, Tiff and Png images.**
- **Supports standard and Freetype characters functions with libiconv to enable non English string can be drawn by freetype funcions**

properly.

- Almost all of the standard drawing funcions supports shaded color option.
- Some chart, filter and transparent functions was provided also.

how to use Butterfly? goto top

The standard manner used to invoke is:

```
Butterfly -i <input file> -o <output file>
```

If you omit the input file name, butterfly takes its directives from stdin, and if you omit the output file name, the output goes to stdout. Output file name can be defined by directive name also.

- Using the `-q` switch sets butterfly to turn on screen reporting.
- A quick reference to directives may be seen by using the `-h` switch.

The directive script file goto top

Directive parses policy for directive script file:

1:If first non-blank character less than char. 'A' will be treated
as a comment line.
2:First word must be a directive name and following is(are) parameters.
3:Directive and parameters were separated from by blank,',' and Tab.
4:In case of parameter has blank or start by special char.
This parameter can use a less than 'A' char. as a Quotation mark.
5:0~9,*,+,-,blank,tab,',' chars. have special meaning, can not be a
Quotation mark.

Ex.
directive 'This is first parameter' ^This is sec. par.^

The directive script file uses a number of directives. It must start with new,existing,GifAnim as initial directive, The detail information as the following:

- **Modifying an existing image**
  existing filename_with_path [,width,height]
  Modifying an existing image with exist image file name. In case of existing file format is bin, image width and height must be provided also.

- **create a new image**
  new[,x,y,type]
  New dirctive must follow by size(second) and type(third). Size and type parameters may follow by new directive or in second and third line. Define as bmp, png, jpg, tif, tga, bin type will create a truecolor image. The image size is x,y and image type is one of bmp,gif,png,jpeg,jpg,tif,tiff,wbmp,tga,bin,gd,gd2.
  OR
  new
  size x,y
  type bmp|gif|png|jpeg|tif|tiff|wbmp|tga|bin|gd|gd2
  size Creating a new image of width x pixels, height y pixels. The next directive must specify the image type.
  type Support image format is bmp , png , jpng , jpg , tif , tiff , gif , wbmp , tga , bin , gd and gd2
  (* Bin format(binary output) is for GNUPlot image import purpose only.)**

- **create a new GIF animated image(Only three directives for GIF animated image.)**
  GifAnim first_GIF_name, out_name, Loops
  GifAnim dirctive must follows by the first GIF file name , the out file name and Loops count. Only two directive, AnimAdd and AnimEnd, can be used after GifAnim dirctive.

AnimAdd add_GIF_name, LeftOfs, TopOfs, Delay
AnimAdd directive adds a GIF image, must follows the add GIF file name with left and top offset and delay time
AnimEnd
The end of Animative GIF definition.

After the initial directive, any of the directives below may be used. To create more than one image from a directive file, use the directive end (not a necessary directive), followed by existing, new or GifAnim directive means to start a new image.

existing filename_with_path [,width,height]
for modifying an another existing image, or

new new[,x,y,type]
size x,y
type bmp|gif|png|jpeg|tif|tiff|wbmp|tga|bin|gd|gd2
for creating an another new image. or

GifAnim first_GIF_name, out_name, Loops
for create an another animated image.

Note:

*** All x,y values are in pixels measured from the top left of the image. For a 256x256 image, top left is 0,0 and bottom right is 255,255. All x1,y1,x2,y2 pairs must specify the top left and bottom right of the shape, where appropriate.**
*** All RGB colour values are in integer format, not hexadecimal.**
*** Arc sweeps are clockwise.**

Basic Directives goto top

SetShaded on|off|n,^R1,^G1,^B1,...(n colours)

on

:for turn on at second time only.
off:turn off SetShaded
option.
n:define number of Shaded colors.
^R1,^G1,^B1,...
:define Shaded colors.

Define shaded colors, or turn on/off shaded option after SetShaded was defined. When SetShaded was set, all functions support SetShaded will show shaded color by SetShaded colors definition.

Ex.
line x1,y1,x2,y2[,R,G,B]

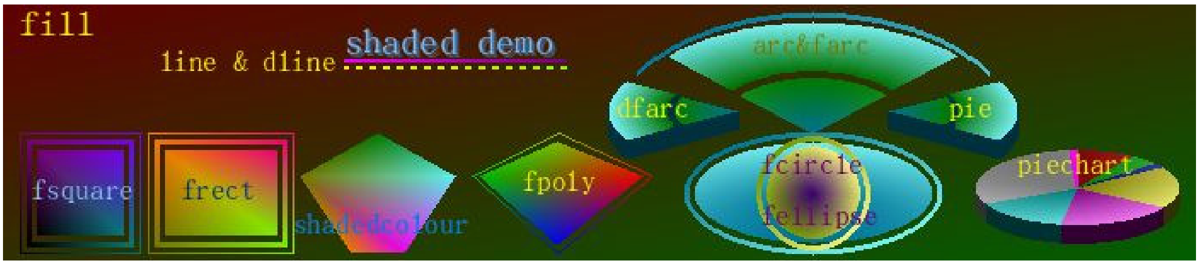
line directive define a line from x1,y1 to x2,y2 with color R,G,B, in case of SetShaded is on, the line will show the color at x1,y1, with R,G,B color and changes the color step by step, the color at x2,y2 will be R+^R1,G+^G1,B+^B1.

In case of SetShaded define one color only and drawing directive has over 2 locations, the colors will be setting as original color, original color+1/2(^R1,^G1,^B1), original color+(^R1,^G1,^B1) and so on.

For arc, circle and ellipse directives, only first shaded color was used, and the max. shaded color angle will be defined by shaded-angle. The default angle is 45 degree(max.x, max.y). For farc, circle and ellipse, center color can be specified also, the default will be the same value.

For polygon fill drawing, Butterfly will draw by triangle solution. It means Butterfly will draw first triangle (x1,y1,color1 x2,y2,color2 and x3,y3,color3). Then draws second triangle (x1,y1,color1 x3,y3,color3 and x4,y4,color4), and so on. Keep in mind that the triangle will be x1,y1, xn-1,yn-1 and xn,yn.

Here is a demonstration to show how it works.



```
line x1,y1,x2,y2[,R,G,B]
```

Creates a line from coordinates x1,y1 to coordinates x2,y2 with colour R,G,B. If colour R,G,B is absent, default color will be used.

```
dline x1,y1,x2,y2[,R,G,B]
```

Creates a dashed line from coordinates x1,y1 to coordinates x2,y2 with colour R,G,B. If colour R,G,B is absent, default color will be used.

```
rect x1,y1,x2,y2[,R,G,B]
```

Creates a rectangle from coordinates x1,y1 to coordinates x2,y2 with edging of colour R,G,B. If colour R,G,B is absent, default color will be used.

```
arect x1,y1,x2,y2[,R,G,B,[a]]
```

Creates a arch rectangle from coordinates x1,y1 to coordinates x2,y2 with edging of colour R,G,B. If colour R,G,B is absent, default color will be used. a = arch ratio, default = 5(1/5).

```
frect x1,y1,x2,y2[,R,G,B]
```

Creates a filled rectangle from coordinates x1,y1 to coordinates x2,y2 filled with colour R,G,B .If colour R,G,B is absent, default color will be used.

```
farect x1,y1,x2,y2[,R,G,B[,a]]
```

Creates a filled arch rectangle from coordinates x1,y1 to coordinates x2,y2 filled with colour R,G,B .If colour R,G,B is absent, default color will be used. a = arch ratio, default = 5(1/5).

```
square x1,y1,s[,R,G,B]
```

Creates a square qith the top left corner at coordinates x1,y1 , with side s in length, with edge of colour R,G,B. If colour R,G,B is absent, default color will

be used.

fsquare x1,y1,s[,R,G,B]

Creates a square qith the top left corner at coordinates x1,y1 , with side s in length, filled with colour R,G,B If colour R,G,B is absent, default color will be used.

poly R,G,B,x1,y1...,xn,yn

Creates a polygon (has to be closed) through the points x1,y1 to x2,y2 to ... to xn,yn, with edge of colour R,G,B.

Note that the colour values appear before the coordinates in this directive.

fpoly R,G,B,x1,y1...,xn,yn

Creates a polygon (has to be closed) through the points x1,y1 to x2,y2 to ... to xn,yn filled with colour R,G,B.

arc x1,y1,w,h,start,finish[,R,G,B][,shaded-angle]

Creates an arc with colour R,G,B centered at coordinates x1,y1, of width w and height h, starting at start degrees and finishing at finish degrees.

In case of SetShaded is on, shaded-angle define the max. shaded color angle. If colour R,G,B is absent, default color will be used.

farc x1,y1,w,h,start,finish[,R,G,B][,shaded-angle[,center-r,c-g,c-b]][,st

Creates an filled arc with colour R,G,B centered at coordinates x1,y1, of width w and he ight h, starting at start degrees and finishing at finish degrees.

In case of SetShaded is on, shaded-angle define the max. shaded color angle, and center-r,c-g,c-b define the color of center. The default value of shaded-angle is 45 degree, and center color will be the same as R,G,B. If colour R,G,B is absent, default color will be used.
Default start radius is zero. In case of start-radius was presented, field area

will from start-radius to 1.0.

dfarc x1,y1,w,h,start,finish,R1,G1,B1,depth,R2,G2,B2[,shaded-angle [,center-r,c-g,c-b]][,start-radius(0.0~1.0)]

pie x1,y1,w,h,start,finish,R1,G1,B1,depth,R2,G2,B2[,shaded-angle [,center-r,c-g,c-b]][,start-radius(0.0~1.0)]

Creates an 3D pie of colour R,G,B centered at coordinates x1,y1, of width w and height h, starting at start degrees and finishing at finish degrees.

In case of SetShaded is on, shaded-angle define the max. shaded color angle, and center-r,c-g,c-b define the color of center. The default value of shaded-angle is 45 degree, and center color will be the same as R,G,B. Default start radius is zero. In case of start-radius was presented, field area will from start-radius to 1.0.

ellipse x1,y1,w,h[,R,G,B][,shaded-angle]

Creates an ellipse with colour R,G,B centered at coordinates x1,y1, of width w and height h.

In case of SetShaded is on, shaded-angle define the max. shaded color angle. The default value of shaded-angle is 45 degree. If colour R,G,B is absent, default color will be used.

fellipse x1,y1,w,h[,R,G,B][,shaded-angle[,center-r,c-g,c-b]]

Creates an ellipse filled with colour R,G,B with centered at coordinates x1,y1, of width w and height h.

In case of SetShaded is on, shaded-angle define the max. shaded color angle, and center-r,c-g,c-b define the color of center. The default value of shaded-angle is 45 degree, and center color will be the same as R,G,B. If colour R,G,B is absent, default color will be used.

circle x1,y1,d[,R,G,B][,shaded-angle]

Creates a circle of colour R,G,B centered at coordinates x1,y1, of diameter d. If colour R,G,B is absent, default color will be used.

In case of SetShaded is on, shaded-angle define the max. shaded color angle.

fcircle x1,y1,d[,R,G,B][,shaded-angle[,center-r,c-g,c-b]]

Creates a circle centered at coordinates x1,y1, of diameter d, filled with colour R,G,B. If colour R,G,B is absent, default color will be used.

In case of SetShaded is on, shaded-angle define the max. shaded color angle, and center-r,c-g,c-b define the color of center. The default value of shaded-angle is 45 degree, and center color will be the same as R,G,B.

fill x,y,R,G,B

Flood fills with the colour R,G,B from the coordinates x,y to the edge of the area of the original color of x,y.

filltoborder x,y,R1,G1,B1,R2,B2,G2

Flood fills with colour R2,G2,B2 from x,y to the border of colour R1,G1,B1.

string R,G,B,x,y,<size>,<string>

Writes a string starting at x,y (in the colour R,G,B), of font size <size>, where size can be one of tiny (5x8), small (6x12), medium (7x13, bold), large (8x16) or giant (9x15, bold).

stringup R,G,B,x,y,<size>,<string>

Writes a string vertically starting at x,y (in the colour R,G,B), of font size <size>, where size can be one of tiny, small, medium, large or giant. The string will go up from the coordinates supplied.

```
getmask R-L,G-L,B-L,Intensity-L (any value in the range,
 [R-H,G-H,B-H,Intensity-H] do nothing while value = 0.)
```

Convert graphic image to mask file. Mask range was defined by the R, G, B colors or intensity. Color value must great than zero to active the range check.

R-L,G-L,B-L,Intensity-L define the range of low limit.
R-H,G-H,B-H,Intensity-H define the range of high limit.

Ex.

```
GetMask 0,0,0,0,10       // Mask area was defined as all color that
                          // red value less than 10.
```

```
resize new_x,new_y
```

Resize an image. new_x,new_y is the new size. In case of one of the value is less than or equal 0, the value will proportional to old image size and the other value.

```
copy x,y,x1,y1,x2,y2,[source-filename|`command`]
```

Copies region x1,y1 - x2,y2 of source-filename or the image output of command to the coordinates x,y of the image being created/modified. If x1,y1,x2,y2 are all -1, program will copy the entire image.

```
copysampled x,y,w,h,srcx,srcy,srcw,srch,source-filename
```

Copies region srcx,srcy ~ srcx+srcw,srcy+srch of source-filename to the coordinates x,y of the image being created/modified. If w,h,srcw,srch is 0, program will copy the size of the image. If w,h,srcw,srch is -1, program will copy the minus size of the image.

Ex.

```
existing original.png       // Flip image horizontal
name FlipHorizontal.png
copysampled 0,0,0,0,-1,0,-1,0,original.png

existing original.png      // Flop image vertical
name FlopVertical.png
```

copysampled 0,0,0,0,0,-1,0,-1,original.png

---

copyresized x1,y1,x2,y2,dx1,dy1,dx2,dy2,source-filename

Copies region x1,y1 - x2,y2 of source-filename to the area dx1,dy1 - dx2,dy2 of the image being created/modified, resizing the image to fit. If x1,y1,x2,y2 are all -1, program will copy the entire image.
If dx1,dy1 are -1, the start location will be set to 0,0.
If dx2,dy2 are -1, original image size will be setting.
If one of dx2,dy2 is -1, this value will proportional setting by original image size and another value.

copymerge x,y,x1,y1,x2,y2,pct|plus|minus|xor|atop|overlay|diff,source-f

Copies region x1,y1 - x2,y2 of source-filename to the coordinates x,y of the image being created/modified. If x1,y1,x2,y2 are all -1, program will copy the entire image.
pct|plus|minus|xor|atop|diff
pct:The two images will be merged according to pct which can range from 0 to 100. When pct = 0, no action is taken, when 100 this function behaves identically to copy for pallete images, while it implements alpha transparency for true colour images.
plus:Adds the colors of the source-filename to the image.
minus:The color of the image is subtracted from the colors of the source-filename.
xor:Base on alpha channel, overlay the two images together, but clear the area shared.
atop:The image colors was added by any non-transparent parts of the source-filename image.
overlay:The image colors was modified by the intensity of the source-filename image. Default base color intensity is 255, if base was specified, all color greater than base will be more bright.
diff:The resulting image is the absolute difference in the color values.

copymergegray x,y,x1,y1,x2,y2,pct,source-filename

Copies region x1,y1 - x2,y2 of source-filename to the coordinates x,y of the image being created/modified. If x1,y1,x2,y2 are all -1, program will copy the

entire image.
This function is identical to copymerge except that when merging it preserves the hue of the source by converting the destination pixels to gray scale before the copy operation.
The two images will be merged according to pct which can range from0 to 100. When pct = 0, no action is taken, when 100 this function behaves identically to copy for pallete images, while it implements alpha transparency for true colour images.

copymask x,y,x1,y1,x2,y2,filename,maskname

Copies region x1,y1 - x2,y2 of filename to the coordinates x,y of the image being created/modified base on relative position at mask file pixel intensity. In case of pixel intensities in mask file are all 0, this function just the same as copy. In case of pixel intensity is 255, program will do nothing, For other value, it will do partial copy base on the value. If x1,y1,x2,y2 are all -1, program will copy the entire image.

copyframe x,y,x1,y1,x2,y2,framename[,threshold]

Copies region x1,y1 - x2,y2 of framename to the coordinates x,y of the image being created/modified base on relative position at framename pixel intensity. In case of pixel intensities in framename greater than threshold value the pixel will be copied.

getpixel x,y

Dump image pixel information at location x,y.

setpixel x,y,R,G,B

Sets the point at x,y to the colour R,G,B.

colourchange R1,G1,B1,R2,G2,B2[,different[,shift(0|1)]]

Compare all pixel colour with R1,G1,B1. In case of the different value less than different the pixel of colour will be changed to R2,G2,B2. If shift was set, new value will be R2,G2,B2 plus the different between old colour and R1,G1,B1.

setbrush filename

Sets the current "brush" to filename. Subsequent directives of line, dline, rect, poly and arc will use the selected "brush" to draw their lines, until a call of killbrush.

killbrush

Turns off the brush selection. Calls to line, dline, rect, poly and arc will then use the standard single-pixel width brush.

settile filename

ets the current "tile" to filename Subsequent directives of fill, filltoborder and fpoly will use the selected "tile" as a fill pattern, until a call of killtile.

killtile

Turns off the tile selection. Calls to fill, filltoborder and fpoly will then use the specified colour for fills.

setstyle [R1,G1,B1, R2, G2, B2, ...,Rn,Gn,Bn]

Various line calls can use a style, specified by one or more colour settings for each pixel, that is repeated for the length of the "line". All subsequent directives of line, dline, rect, poly and arc will use the selected "style" to draw theirlines, until a call of killstyle.

killstyle

Turns off the style selection. Calls to line, dline, rect, poly and arc will then use the standard single-pixel width brush.

transparent R,G,B

Define the colour R,G,B as transparent colour.

```
rotate degz[, degx, degy, cx, cy]
```

Rotates the image degx degrees along X axis, degy degrees along Y axis, degz degrees(clockwise) along Z axis.
cx, cy defines the center of the rotate operation. Default cx, cy is at the center of the image.

```
interlace
```

Makes the image output by butterfly an interlaced GIF.

```
thick value
```

Setting the line thick value, default value is 1 (pixel).

```
Info
```

Dump image basic information.

```
end
```

End of this image drawing, The result of drawing image will be writed out right away.

Gdlib Filter Functions goto top

```
FlipV
```

Flip image vertically.

```
FlipH
```

Flip image horizontally.

FlipBoth

Flip image both vertically and horizontally.

negate

Replace every pixel with its complementary color.

Brightness brightness(>0, <256)

Image will process by Gdlib brightness filter .

MeanRemoval

Image will process by 3x3 MeanRemoval filter.

GrayScale

No parameter the image will process by Gdlib GrayScale fileter.

Emboss

No parameter the image will process by Gdlib Emboss fileter.

Contrast (float)contrast(0~100)

Image will process by Gdlib Contrast fileter.

gaussianblur

Image will process by Gdlib gaussianblur fileter.

Smooth (float)weight

Image will process by Gdlib Smooth fileter.

SelectiveBlur

Image will process by Gdlib SelectiveBlur fileter.

Convolution float filter[3][3], float filter_div, float offset

Image will process by Gdlib Convolution fileter.

EdgeDetectQuick

Image will process by Gdlib EdgeDetectQuick fileter.

Scatter sub, plus (sub < plus)

Image will process by Gdlib Scatter fileter.

scale new_width, new_height

Image will process by Gdlib scale fileter.

scalebilinear new_width, new_height

Image will process by Gdlib scalebilinear fileter.

scalebicubicfixed width, height

Image will process by Gdlib scalebicubicfixed fileter.

scalenearestneighbour width, height

Image will process by Gdlib scalenearestneighbour fileter.

rotatebilinear angle, bgcolor_r, bgcolor_g, bgcolor_b

Image will process by Gdlib rotatebilinear fileter.

rotateinterpolated angle, bgcolor_r, bgcolor_g, bgcolor_b

Image will process by Gdlib rotateinterpolated fileter.

rotatebicubicfixed angle, bgcolor_r, bgcolor_g, bgcolor_b

Image will process by Gdlib rotatebicubicfixed fileter.

rotatenearestneighbour angle, bgcolor_r, bgcolor_g, bgcolor_b

Image will process by Gdlib rotatenearestneighbour fileter.

rotategeneric angle, bgcolor_r, bgcolor_g, bgcolor_b

Image will process by Gdlib rotategeneric fileter.

Butterfly Filter Functions

goto top

FilterRange x-min, ymin, x-max, y-max

Define filter function range from x-min,y-min to x-max,y-max.

FilterMask [MaskName, FilterThred]

Define filter function mask file name.
FilterThred: All pixel intensity of mask file less than FilterThred will effect almost all filter function operations.
In case of no file name was provided, the function will disable.
*** Shadow directive will override and disable this function ***

slope x-shift,y-shift

Image were sloped by parameters x-shift and y-shift.

Brightnessx brightness(>0, <256)

Shift 50 percent glay color value from 128 to brightness value.

Chrome [exposure(0.0~1.0)]

Image will process by chrome filter.

ColorShift shift-r, shift-g, shift-b, shift-a[, r, g, b, different]

Shift image color by shift-r, shift-g, shift-b and shift-a. If different was specidied, only the color distance to r, g, b less than different will be processed.

GrayScale method(-1~2)[, bits(1|2|4|8) (for method>0), threshold(for

Convert image to Gray by method -1 to 2. The image may down scale by bits value for method > 0. In case of bits = 1, varying threshold value so as to get the mask closer to the image proper.

Dither (int)method(0~5), bits(1~8)

Image will processes by **dither** filter. There are six filter matrixes(method) were applied. The image may down scale by **bits** value.

Method:

0 Bayer
1 Half tone
2 Screw 1
3 Screw 2
4 Central stress
5 Dot concentrate

DeltaSigma (int)method(0~1)[, bits(1|2|4|8), threshold(for bits=1)]

Image will process by **DeltaSigma** filter. Two sub-method was provided. The image may down scale by **bits** value. In case of **bits** = 1, varying **threshold** value so as to get the mask closer to the image proper.

Gamma gamma-r, gamma-g, gamma-b(0.1~5.0)

Image will processes by **Gamma** filter with gamma-r, gamma-g, gamma-b for red, green, blue color.

EdgeDetect [bias(default = 0), threshold]

Image will process by **3x3 EdgeDetect** filter to find out the Edge of the image. If **threshold** not equal to zero, then all value difference between **bias** less than **threshold** will be set as **bias**.

EdgeDetect2 [size, mode(0|1)]

All pixels will process by sizexsize **EdgeDetect2** process to find out the maximum difference of the image block.
If **mode** value is not zero then the output color will be reversed.

Contour [size, mode(0|1)]

All pixels will process by sizexsize **Contour** filter process.
If **mode** value is not zero then the output color will be reversed.

Dilate [size]

All pixels will process by sizexsize Dilate filter process.

Emboss [size(3|5|7|9), angle(0|45|90|135|180|225|270}315), Bias]

Image will process by 3x3, 5x5, 7x7 or 9x9 Emboss filter with angle angle value.

Default value:
size = 3.
angle = 0.
bias = 127.

Darkness amount(0~255)

Image darkness will be re-calculated by amount value.

Normalize bias, peak [smooth(3~15)]

Adjust image intensity as:
Change minimum intensity to bias.
Change maximum intensity to peak.
In case of smooth was provided, a smooth*smooth smooth filter was executed after normalize.

Smoothx size(3~15), (float)weight[, different]

Image will process by sizexsize Smooth filter. Weight value can be specified to overwide the default value. If different was specidied, only the color distance less than different was processed by Smooth filter.

Noise [F|B]

For no parameter or wirh F parameter, program will add an uniform noise to the image.

For parameter B, program will build a noise image.

AddNoise level

Program will add an uniform noise to the image. The noise level is defined by level.

Gaussian [weight(for 3x3) | size(>2) weight(for bigger than 3x3)]

Image will process by 3x3 Gaussian filter or use size value to define filter diamension. weight value can be specified to overwide to default value.

Gaussian15

Image will process by 15x15 flat Gaussian filter.

Sharp radius, amount(0.0~2.0), threshold(0~255)

sharpen the image.
radius: width in pixels of the blurring effect. Range: >2; default = 5.
amount: strength of the filter. Range: 0.0 (none) to 2.0 (max); default = 1.0
threshold: difference to trigger the filter.
Range: 0 (always triggered) to 255 (never triggered); default = 0.

Sharpen Pct(>0)

sharpen the image by simple 3x3 sharp filter. Pct is sharpening percentage, and can be greater than 100.

Shadow shift-x, shift-y[, red, green, blue, smooth(0|1, default = 1)]

simulate an image shadow.
shift-x, shift-y is the location shift of the shadow.
red, green, blue: are the shadow color, default value is 0x00202020.
smooth: After merge shadow and original image, program will do a edge smooth operation. Set smooth value 0 to skip the smooth operation.

Mosic [Size[|Size-w, Size-h]]

Image will process by Mosic filter with Size X Size or Size-w X Size-h size. The default size = 10 X 10.

ChannelMix angle

Shift image color system by angle value.

Diffuse scale(>0, pixels)

Diffuses the image by moving its pixels in random directions with scale factor.

Gain [Gain(0.0~1.0), Bias(0.0~1.0)]

Changes the gain and bias of the image - similar to ContrastFilter.

Contrastx (float)contrast(0~100)

Butterfly Contrast fileter.

Shade gray[0|1], azimuth(0~360), elevation(0~360)

shade the image using a distant light source by azimuth(0~360) and elevation(0~360). If gray was set, the image colour will change to gray.

Filter (float)filter[3][3],[(float)filter_div][,(float)offset]

Image will process by the filter matrix provided by filter[3][3]. The value of size was fixed by 3. Divisor was defined by filter_div(default = 1.0), and offset was defined by offset(default = 0.0).

FilterX size(3~15),(float)filter[size][size],[(float)filter_div][,(float)offse

Image will process by the filter matrix provided by filter[size][size]. Filter dimension was defined by size. The value of size was fixed from 3 to 15. Divisor was defined by filter_div(default = 1.0), and offset was defined by offset(default = 0.0).

Watermark functions goto top

Watermark functions enable you to set some hidden strings to protect your images. Watermark functions will float the original color +/- by 1. It is difficult to find out the different, but can be showed by program ChkMark.

ChkMark reads in the input image and output ChkMark.png. The file will highlight Watermark informations in the output image.

It is to be note that Butterfly set a version information Watermark in the corner of left bottom, to show that the image was built by Butterfly.

```
stringw x,y,size,string
```

Writes a watermark string starting at x,y with font size <size>, where size can be one of tiny (5x8), small (6x12), medium (7x13, bold), large (8x16) or giant (9x15, bold).

```
ftstringw x,y,fontfilename,fontsize,angle,string
```

Writes a watermark string starting at x,y, font size was specified by integer, character angle specific by real number, Character font name shall be found in O.S. font diretory or specified the file name with full path. In case of non-English Character code was used, Use CodeConvert directive to specify that convert the code to unicode before processed.

```
copyimagew x,y,imagename
```

Copy a watermark image starting at location x,y. The watermark image just something like a mask file to make a hidden stamp in the image.

TrueColour Functions goto top

hfill x1,y1,R1,G1,B1,x2,y2,R2,G2,B2[,Alpha1,Alpha2]

Fill background color from x1,y1to x2,y2. The color will change horizontally from R1,G1,B1 to R2,G2,B2.
The default alpha value were set to zero. In case of alpha1 and alpha2 were specified, There two value will be applied to R1,G1,B1 and R2,G2,B2.

vfill x1,y1,R1,G1,B1,x2,y2,R2,G2,B2[,Alpha1,Alpha2]

Fill background color from x1,y1 to x2,y2. The color will change vertically from R1,G1,B1 to R2,G2,B2.
The default alpha value were set to zero. In case of alpha1 and alpha2 were specified, There two value will be applied to R1,G1,B1 and R2,G2,B2.

shadedcolour n,x1,y1,R1,G1,B1,x2,y2,R2,G2,B2,x3,y3,R3,G3,B3,..

Creates a shaded color polygon through the points x1,y1 to x2,y2 to ... to xn,yn. The color of coordinates xn,yn was defined by Rn,Gn,Bn.

Free Type String functions

goto top

ftoutline True|False, (default is false)

Set outline image output mode for free type string drawing functions.

ftthick value

Set FTOutline directive thick value(default = 1 [pixel]).

SetNColors no-of-colors

SetNColors: Set number of antialised colors for indexed bitmaps as no-of-colors(For Free Type string functions).

CodeConvert from-code [to-code]

```
CodeConvert off
```

Enable iconv code convert or turn off the function for free type string. The default to-code is UTF-8. This function is useful for non-english country they use multi-byte characters with the string codes are not in unicode. For example Taiwan O.S. use traditional chinese characters with BIG5 code, but for normal cases the font index was ordered by unicode(UTF-8). So, a code convert must apply to enable non-unicode string drawing.

Ex.

```
CodeConvert BIG5
```

```
ftstring R,G,B,x,y,fontfilename,fontsize,string_angle,string
ftstringx R,G,B,x,y,fontfilename,fontsize,string_angle,Chr._angle,strin
ftstringcircle R,G,B,x,y,radius,TEXTradius,fontfilename,fontsize,fillP
```

Writes a string starting at x,y(with the colour R,G,B), font size was specified by integer, character angle specific by real number, Character font name shall be found in O.S. font diretory or specified the file name with full path. Use ftstringx directive in case of string direction is different from the character direction. Ftstringcircle will make character string in curve.

For manipulating direct UTF-8 binary code string(Not UTF-8 ASCII code string), GDLib provides HTML4.0 parser entities in decimal form, e.g. &#197; or in hexadecimal form, e.g. &#x6C34; . This function is useful in direct specific the code value, specially the code value is bigger than 0XFFFF.

Ex.

```
ftstring 0,0,0, 40,120,kaiu,32,0.0,"10000:"
ftstring 0,0,0,290,120,CODE2001.TTF,32,0.0,"&#X10000;  &#X10001;
&#X10002;  &#X10003;"
```

Chart Functions goto top

```
piechart X,Y,W,H,No_Pie,depth,Titleflag[,fontfilename,fontsize][,shad
R1,G1,B1,value,[R2,G2,B2(depth colour if any)][,Title(if any)] (one v
...
```

```
 -- Loop for No_Pie Times --
...
```

Creates a Pie chart start at x1,y1 with W width and H high. In case of depth, Titleflag are not zero, depth colour/Title must be specified. For SetShaded directive was on, shaded-angle shall be specified.

From second line, Every Pie needs to define pie color and Title(if any).

```
drawscale scale_type [,No_depth], X, Y, w, h, depth, Titleflag [,Titlasi
xInc,yInc,R1,G1,B1[,R2,G2,B2(If depth)]
[R,G,B,fontfilename,No_Scale_XValue,XValue1,XValue2,XValue3,XV
[R,G,B,fontfilename,No_Scale_YValue,YValue1,YValue2,YValue3,YV
```

Drawscale will draw a 2D or 3D scale image for linechart/barchart /columnchart. Any way, Drawscale was decouple from chart functions will be more flaxable for image arrange. For example, draw more than one chart on one scale. scale_type will be set for 3D linechart, and No_depth must be setting for special depth define in case of scale_type(if scale_type=1 then total depth = No_depth * depth; else total depth = depth).

Scale is start at X,Y, with w wide and h high. depth will be defined as zreo for 2D scale and great then zreo for 3D scale. Titleflag must specified to define draw title or not. In case of title flag was set, title string must be defined in line 3 and line 4 for X and Y scale title.

Second line defines how many pixel width draw a grid line both X and Y axis, and grid shall be specified also. Line 3 and 4 defines scale value string and title in case of Titleflag and/or ValueFlag was set.

```
barchart x,y,w,h,No_bar,No_Value,depth
[R,G,B,R,G,B,...(One colour per Bar.)][,R,G,B,...(if 3D that 3D Colou
CurveName_flag[,fontfilename,fontsize,Curve_name(per Curve),...]
base_value,Full_scale_value
Value1,Value2,Value3,Value4,...(loop for No_Bar)
...
...-- Loop No_Value Time --
...
```

Draw a Barchart start at X,Y, with w wide and h high. depth will be defined as zreo for 2D scale and great then zreo for 3D scale. The chart has No_bar Bar

and every Bar has No_Value value.

Second line defines the Bar color(One colour per Bar). In case of depth is not zero, 3D color shall specified too. If colour R,G,B is absent, default color will be used.

Third line defines Curve Name and Forth line defines Max. and Min scale value.

The last are bar values.

```
linechart x,y,w,h,No_Line,No_Value,depth
[R,G,B,R,G,B,...(One colour per Parm.)][,R,G,B,..(if 3D that 3D Col
CurveName_flag[,fontfilename,fontsize,Curve_name(per Curve),...]
base_value,Full_scale_value
Value1,Value2,Value3,Value4,...(loop for No_Line)
...
...-- Loop No_Value Time --
...

If depth = -1: 2D BAR
 -2: 3D BAR
 -3: Cylinder BAR Type 0
 -4: Cylinder BAR Type 1
```

Draw a Linechart start at X,Y, with w wide and h high. depth will be defined as zreo for 2D scale and great then zreo for 3D scale. The chart has No_Line Line and every Line has No_Value value.

Second line defines the Line color(One colour per Line). In case of depth is not zero, 3D color shall specified too If colour R,G,B is absent, default color will be used..

Third line defines Curve Name and Forth line defines Max. and Min scale value.

The last are Line values.

```
columnchart x,y,w,h,No_Column,No_Value,depth
[R,G,B,R,G,B,...(One colour per Parm.)][,R,G,B,..(if 3D that 3D Col
CurveName_flag[,fontfilename,fontsize,Curve_name(per Curve),...]
base_value,Full_scale_value
Value1,Value2,Value3,Value4,...(loop for No_Column)
...
...-- Loop No_Value Time --
...

If depth = -1: 2D BAR
 -2: 3D BAR
 -3: Cylinder BAR Type 0
 -4: Cylinder BAR Type 1
```

Draw a columnchart start at X,Y, with w wide and h high. depth will be defined as zreo for 2D scale and great then zreo for 3D scale. The chart has No_Column Column and every Column has No_Value value.

Second line defines the Column color(One colour per Column). In case of depth is not zero, 3D color shall specified too. If colour R,G,B is absent, default color will be used.

Third line defines Curve Name and Forth line defines Max. and Min scale value.

The last are Column values.

GIF animated image Functions

goto top

```
GifAnim first_GIF_name, out_name, Loops
```

GifAnim dirctive must follows by the first GIF file name , the out file name and Loops count. Only two directive, AnimAdd and AnimEnd, can be used after GifAnim dirctive.

```
AnimAdd add_GIF_name, LeftOfs, TopOfs, Delay
```

AnimAdd directive adds a GIF image, must follows the add GIF file name with left and top offset and delay time.

```
AnimEnd
```

The end of Animative GIF definition.

Ex.

```
GifAnim SimpleAuto-S.png SimpleAuto.gif 999999
AnimAdd SimpleAuto-S.png  0 0 100
AnimAdd SimpleAuto-M.png 0 0 100
AnimAdd SimpleAuto-L.png  0 0 100
AnimEnd
```

Alpha Channel(Set) Functions (For PNG Only) goto top

SaveAlpha ON|OFF

Define save alpha channel flag for png image.

Set AlphaBlending On|Off

AlphaBlending dirctive turns On or Off Gdlib AlphaBlending Flag.

Set Alpha alpha

Alpha: Set default alpha channel value.

Set AlphaDepth alpha

AlphaDepth: Set default 3D alpha color value for pie function.

Set AlphaCenter alpha

AlphaCenter: Set default center alpha color value for pie/fellipse functions.

Set AlphaColor R,G,B[,alpha]

AlphaColor: Define R,G,B color for all alpha channel value great than aplha pixels.

Set Transparent alpha(0~127)

Transparent dirctive set whole image transparent value to alpha.

Set TransColor R,G,B,alpha(0~127)[,threshold]

TransColor dirctive set color R.G.B has transparent value alpha.
In case of threshold was specified, the difference between any color and R.G.B less than threshold have transparent value alpha.

Set TransMask MaskName[, reverse(0|1)]

Set image alpha channel value by the intensity of mask image. If reverse value not equal zero, intensity value will reverse before setting.

Set TransVertical x1,y1,alpha1(0~127),x2,y2,alpha2

TransVertical: Transparent value changes vertically from alpha1 at location x1,y1 to alpha2 at location x2,y2.

Set TransHorozon x1,y1,alpha1(0~127),x2,y2,alpha2

TransHorizon: Transparent value changes horizontally from alpha1 at location x1,y1 to alpha2 at location x2,y2.

Set transLine x1,y1,x2,y2[,alpha1][,alpha2]

transLine: Set Shaded Line alpha channel color from alpha1 at x1,y1 to alpha2 at x2,y2. If alpha value(s) is/are absent, Default value was defined by Alpha will be used.

Set transTriangle x1,y1,x2,y2,x3,y3[,alpha1][,alpha2][,alpha3]

transTriangle: Set Shaded Triangle alpha channel color by alpha1, alpha2 and alpha3. If alpha value(s) is/are absent, Default value was defined by Alpha will be used.

```
Set transRect x1,y1,x2,y2[,alpha1][,alpha2]
```

transRect: Set Shaded rectangle alpha channel color with edge is alpha1 and center is alph2. If alpha value(s) is/are absent, Default value was defined by Alpha and AplhaCenter will be used.

```
Set transRectX x1,y1,x2,y2,offset[,alpha][,alphacenter]
```

transRectX: Set the edge of rectangle alpha channel color as alpha and the offset of rectangle alpha channel color as alphacenter. If alpha value(s) is/are absent, Default value was defined by AlphaCenter and Alpha will be used.

```
Set transfillarc x,y,w,h,start,end[,alpha1(0~127)][,alpha2]
```

transfillarc: Set the edge of filled-arc alpha channel color as alpha1 and the center of ellipse alpha channel color as alpha2. Arc angle start at **start** and end at **end**. If alpha value is/are absent, Default value was defined by Alpha and AlphaCenter will be used.

```
Set transfillarcx x,y,w,h,start,end,offset[,alpha(0~127)][,alphacenter]
```

transfillarcx: Set the edge of filled-arc alpha channel color as alpha and the offset of ellipse alpha channel color as alphacenter. Arc angle start at **start** and end at **end**. Offset value shall be from 0 to 99 (%). If alpha value is/are absent, Default value was defined by Alpha and AlphaCenter will be used.

```
Set transellipse x,y,w,h[,alpha1(0~127)][,alpha2]
```

transellipse: Set the edge of ellipse alpha channel color as alpha1 and the center of ellipse alpha channel color as alpha2. If alpha value is/are absent, Default value was defined by Alpha and AlphaCenter will be used.

```
Set transellipsex x,y,w,h,offset[,alpha1(0~127)][,alpha2]
```

transellipsex: Set the edge of ellipse alpha channel color as alpha1 and the offset of ellipse alpha channel color as alpha2. Offset value shall be from 0 to

99 (%). If alpha value is/are absent, Default value was defined by Alpha and AlphaCenter will be used.

Ex. Make a ellipse image with transparent outside.
(Example Image size is 329,356)

```
//
//    Build a ellipse Mask Image file
//
```

```
new,329,356,png
name mask.png

fill 0,0, 255,255,255
fellipse 167,182, 233,263  ,0, 0, 0
```

```
//
//   Make a ellipse  image with transparent outside.
//
```

```
existing,Ogiginal.jpg
name,TransImage.png
//   Transparent the outside of ellipse
Set  transmask  mask.png 0
//   Tarnsparent ellipse edge
Set  transEllipsex 165,180,240,270, 70,127,0
```

Plot Functions

goto top

```
Plot grid true|false [r,g,b[rB.gB,bB(for 3D background)]]
```

Specified plot grid or not for Plot directive.
true: Specified plot grid with color r,g,b.
false: Specified do not plot a grid.

```
Plot h1_2dbar x1,y1,x2,y2,R,g,b,shadow[0|1]
```

h1_2dbar: Plot one horizontal 2D Bar.
shadow: If specified as one, means plot shadow also.

```
Plot v1_2dbar x1,y1,x2,y2,R,g,b,shadow[0|1]
```

v1_2dbar: Plot one vertical 2D Bar.
shadow: If specified as one, means plot shadow also.

```
Plot P1_3dbar x1,y1,x2,y2,R,g,b
```

P1_3dbar: Plot one 3D Bar.

```
Plot Cylinder1_HBar x1,y1,x2,y2,R,g,b,ColorType[0|1]
```

Cylinder1_HBar: Plot one horizontal cylinder Bar.
ColorType: Specified 3D color type(two type only).

```
Plot Cylinder1_VBar x1,y1,x2,y2,R,g,b,ColorType[0|1]
```

Cylinder1_VBar: Plot one vertical cylinder Bar.
ColorType: Specified 3D color type(two type only).

```
Plot HBAR datafile,x,y,w,h,type[0~3][,shadow(0|1 for type 0)]
```

HBAR: Plot a horizontal Bar chart start at location x,y with w,h size.
datafile: Specified the data file name.
Type: Specified chart type(0~3, 0:2D Bar, 1:3D Bar, 2,3:cylinder Bar).
shadow: Specified plot shadow also(0|1 for type 0).

```
Plot VBAR datafile,x,y,w,h,type[0~3][,shadow(0|1 for type 0)]
```

VBAR: Plot a vertical Bar chart start at location x,y with w,h size.
datafile: Specified the data file name.
Type: Specified chart type(0~3, 0:2D Bar, 1:3D Bar, 2,3:cylinder Bar).
shadow: Specified plot shadow also(0|1 for type 0).

Created: 12/22/2007, Last modified: 12/04/2013,