

GfxCon

Dirk Farin

COLLABORATORS

	<i>TITLE :</i> GfxCon		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Dirk Farin	April 13, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	GfxCon	1
1.1	contents	1
1.2	the purpose of GfxCon	2
1.3	news	2
1.4	remarks	3
1.5	bugs	4
1.6	requirements	4
1.7	installation	4
1.8	using as shell-command	5
1.9	entering filenames	7
1.10	selection the output format	8
1.11	modifying size and orientation	8
1.12	modifying colors	9
1.13	the conversion-process	10
1.14	showing information about the input-image	10
1.15	supported formats	11
1.16	problemes	12
1.17	setting the tooltypes	12
1.18	automatic format-recognition	13
1.19	memory usage	13
1.20	virtual memory	14
1.21	IFF - ILBM Format	15
1.22	PCX - format	15
1.23	JPEG - format	16
1.24	Postscript	16
1.25	RGB - raw - files	16
1.26	RGB8 / RGBN - format	17
1.27	IMG - format	17
1.28	BMP/RLE4/RLE8 - formats	17
1.29	GIF - format	18

1.30	TIFF - format	18
1.31	Targa - format	19
1.32	position and size of Postscript output	19
1.33	about GfxCon	19
1.34	registration	20
1.35	copyright	21

Chapter 1

GfxCon

1.1 contents

GfxCon - image format converter

=====

V1.8c (3 July 1997)

Dirk Farin

Introduction

what is GfxCon ?

what's new ?

Remarks

bugs ?

installation

requirements

installation

usage

entering the filenames

selection the output-format

modifying size and orientation

modifying colors

the conversion-process

showing information about the input image

using GfxCon as shell-command NEW
reference

supported formats

problems that may occur

about the program

copyright

(registration)

1.2 the purpose of GfxCon

what is GfxCon ?

=====

GfxCon makes it possible to convert images to the most popular image formats in a user-friendly way. Simplicity of usage and a high quality have been the primary goals while developing GfxCon.

Besides this it is possible to modify the image in many ways like changing the resolution, contrast of the image or numerous color-effects.

1.3 news

news

version 1.8c

- GIF loader doesn't get into an endless loop on corrupt files
- 24bit-BMP images with odd width now load correctly

version 1.8b

- 24bit-BMP loader
- chooses best filemode for saving automatically
(i.e. a JPEG-file saves to 24bit output (if possible) by default)
- BOXFITALL working now / enforcer hit in BOXFIT/BOXFITALL removed

version 1.8

- Much more TIFF-images can be loaded. Especially LZW compressed images are supported now.
 - 24bit-PCX-images (version 5) can be loaded.
 - all known bugs have been removed in the TIFF,PCX and BMP-loaders
(valid BMP-files were refused because GfxCon thought they are corrupted, TIFF-files were sometimes read up to behind the end)
-

- new shell commands: CROP, CENTERBOX, OFFSET, SORTDIR, UNUSED, RESIZEH, RESIZEV

version 1.7

- RGB-Raw input, output and brightness, contrast may be used with the Shell-interface
- output format = input format by default in the shell
- GIF89a transparent color is now saved and loaded too (not only background color)
- Guru with bad input files removed
- progress-information in the Shell may be switched off

version 1.6

- two ways to proportionally resize an image (shell only)
- it's now possible to query the image-infos from shell

version 1.5

- may now be used as shell-command
- saves background-color if known (ILBM<->GIF) can only be changed by hand with shell-parameter
- runs under Kick 1.2 :) (shell only)
- bug fixed: changing image-geometry and saving as GIF at once was faulty

version 1.4

- bug in PCX-loader fixed (images with odd number of bytes per row)
- bug in PCX-saver fixed (BytesPerRow in PCX-header was set incorrectly)
- 68020-version doesn't need 68881 any more (uses IEEE-libraries now)

version 1.3

- HAM6 is working again
- virtual memory improved (new Tooltypes)
- TIFF-RGB-images and bug-fix

version 1.2

- HAM8-mode (finally)
- Targa-CLUT-images

1.4 remarks

I've written GfxCon a long time ago (before the release of ↵ WB2.1).

Therefore some things like localisation have been solved in a different way (with tooltypes instead of using the WB2.1-catalogs). But this allows even users of WB1.2 to use the feature of localisation :)

For the same reason GfxCon does not support the use of datatypes (introduces in Kick3.0).

Since upgrading the program to use all these would exceed my free time at

the moment, it is unlikely that an update will be released. But a completely new program (of course much better than this) may be possible in the near future.

see here for more information...

1.5 bugs

known bugs:

- sometimes the font in one of the button-gadgets changes to a smaller one "without any sensible reason". But this does not affect functionality in any way.

If you discover a bug, please let me know. I will try to fix it, but I cannot promise to succeed, as this program is nearly 2 years old by now (!) and has not been looked at in the meantime.

my address

1.6 requirements

requirements

=====

This program requires to run:

Kickstart 1.2
20000 Bytes stack

Recommended:

At LEAST a 68020 !

4 MB memory

hard-disk

1.7 installation

installation

=====

To install this program simply move the drawer containing the program (or only the program) to a place of your choice.

To configure the program to your system, it is recommended to

change the
 tooltypes
 to your needs.

Thats all.

1.8 using as shell-command

GfxCon can now (since version 1.5) be used as shell-command. This works even with Kickstart 1.2 (or lower :-)

usage:

```
GfxCon inputfile [TO outputfile] [FORMAT f] [COLORS c] [SIZE x y] [FLIPX] [...]
```

If you don't specify an outputfile, the old suffix will be replaced with the suffix of the output-format.

parameter:

```
TO destname set destination filename
FORMAT output format { ILBM,GIF,PCX,JPEG,Postscript,RGB-Raw }

COLORS n use 'n' colors max.
DITHER use floyd-steinberg dithering
FASTDITHER use fast floyd dithering
BACKGROUND n use color 'n' as background color (only ILBM and GIF)
UNUSED r g b fill empty CLUT entries with (r,g,b)
OFFSET n don't use first 'n' colors of CLUT (see below)
SORTDIR n sort CLUT-colors (n=1 darkest to brightest/n=-1) (see below)

CROP x1 y1 x2 y2 crops the image to the specifies rectangle
SIZE w h resize image
RESIZE f proportionally resize image with factor f (>0)
RESIZEH f resize image horizontally with factor f (>0)
RESIZEV f resize image vertically with factor f (>0)
BOXFIT w h proportionally resize image to fit into the box specified
BOXFITALL w h same as BOXFIT, but enlarges image, if smaller than the box
CENTERBOX w h r g b creates image with the size (w,h), centers the
loaded image into the box and fills the border with
the color (r,g,b)

QUALITY q set JPEG-quality
CLUT select IFF-CLUT mode\
HAM IFF-HAM mode \_ use only with 'FORMAT ILBM'
HAM8 IFF-HAM8 mode /
24BIT IFF-24bit mode/

INVERS invert colors
BW create black and white only
GRAYSCALE produce a grayscale (not simply BW!) output
NORED remove red
```

```

NOGREEN remove green
NOBLUE  remove blue

BRIGHTNESS n  change brightness by n (255 to shift black to white)
CONTRAST n   change contrast by n (percent)

FLIPX   flip horizontally
FLIPY   flip vertically
ROTATELEFT rotate left (90 degrees)
ROTATERIGHT rotate right (90 degrees)

GREEN n set name of green-rgb-raw-file to n (RGB-Raw only)
BLUE n  set name of blue -rgb-raw-file to n (RGB-Raw only)
INPUTWIDTH n force input-width to be 'n' pixels (RGB-Raw only)

INFO      show image-information only
NOPROGRESS don't show how much the conversion has progressed
HELP / -help / -h / ? / -?  this help-page :-)
```

remarks:

The shell-interface becomes much more powerful if combined with the 'foreach'-program (by me; available e.g. on aminet 'util/cli/foreach.lha').

Examples:

```

Convert all IFFs to GIFs:  foreach #?.iff GfxCon & FORMAT GIF
Produce a series of JPEGs with different quality-settings:
    foreach {30,50,70,90} GfxCon Pic.iff TO Pic.&.jpg FORMAT JPEG QUALITY &
```

The shell-functions and the Kick1.2-support have been implemented after the program has been finished with a lot of dirty tricks and is therefore everything but perfect. It is mainly to offer these capabilities until GfxCon2.0 is finished and to finally give a graphics-converter to those poor Kick1.2 people :)

NEW (v1.8) usage of UNUSED

```

-----
If you put the UNUSED-parameter after the COLORS-parameter,
the number of colors not to be used will be subtracted
from the number you specified with the COLORS parameter.
If you place the two parameters the other way, they will
be added. This shall mean:
```

```
gfxcon ... COLORS 32 UNUSED 10 ...
```

creates an image with 32 colors, 10 of which are not used.

```
gfxcon ... UNUSED 10 COLORS 32 ...
```

creates an image with 42 colors in total.

IMPORTANT: UNUSED only works with ILBM-output. Behaviour
with other output formats is undefined.

SORTDIR

 If you write "SORTDIR 1", colors will be sorted that the darkest colors will be in the first color table entry and the brightest color in the last. If you write "SORTDIR -1" it just sorts in the other direction.

using CROP/RESIZE/BOXFIT... at once

The operators that change the image size are executed independent of the order on the command line in this order:

- 1) CROP
- 2) RESIZE, RESIZEH, RESIZEV
- 3) SIZE
- 4) BOXFIT, BOXFITALL

It is not allowed to use several RESIZE* or BOXFIT* commands in one program run.

(v1.7) How to load RGB-Raw-files: type the red-file as input file and add the two other files with the GREEN and BLUE parameter.

Example: To load the files "pic.red", "pic.green", "pic.blue", type

```
GfxCon pic.red GREEN pic.green BLUE pic.blue ...
```

If you have to override the width, GfxCon guesses for the width, use the INPUTWIDTH parameter.

The suffices ".red", ".green" and ".blue" are used for RGB-Raw file output. If you specify an output filename with "TO", these suffices are appended too.

caution:

Kick1.3 users CANNOT use the Postscript-output. Sorry!

1.9 entering filenames

entering filenames

=====

general case

There is a box 'Load' at the top left and a box 'Save' at the right to enter the filenames for the input- and output-filename respective. Both boxes can be used in the same way.

To enter a filename simply type it into the string-gadget or choose a file using the filerequester which can be opened by pressing the button to the left of each string-gadget.

If you entered the input-filename (and have pressed RETURN or the TAB-key or you choose it with the filerequester, GfxCon tries to determine the format of the input-file and enters the name of the format into the popup-gadget below the string-gadgets.

RGB-files (raw)

If GfxCon doesn't recognise the input-format, it supposes that it is a raw

RGB-file

(consisting of three files for red,green and blue).

As three input-files are needed to load a RGB-raw-image, GfxCon enables the two string-gadgets below the normal one to let you enter the filenames for the green and blue part of the image. The character at the right of each string-gadget shows which filename to enter (the red-part at R, the green at G and the blue at B of course).

1.10 selection the output format

selection the output format

=====

selecting the file-format

You can select the format using the "popup-gadget" in the 'Save'-box.

selecting the color mode (only ILBM)

As the

ILBM-format

supports several color modes (colortable, 24 bit, HAM6, HAM8), you may choose the color mode with the popup-gadget 'colormode'.

setting number of colors

The number of colors allowed in the output-file may be specified with the gadget 'colors'. You may also enter unusual numbers like 28 or 53. As most image-formats only allow numbers that are a power of 2, GfxCon rounds this number up to the next possible number of colors and leaves the remaining colors unused. (If you specify more colors than the format allows, like 512 für PCX, GfxCon uses the maximum colors possible with that format).

1.11 modifying size and orientation

modifying size and orientation

=====

resizing

Switch the button 'custom' in the box 'size' to on. The two gadgets

'width' and 'height' will become enabled with the original size of the image in pixel already filled in. Modify these values to the desired ones.

If you enlarge the image and don't want that "big pixels" are created, set the switch 'interpolated'. The conversion will take longer but the result will be a smoother image.

NOTE: Interpolating an image creates new colors. It makes no sense to interpolate a image when the output will only have 2 colors for example.

NOTE: To enlarge the picture, GfxCon need more
memory
than normal.

flipping the image

you may flip the image horizontally or vertically (or both, what results in rotating the image 180 degrees) using the gadget 'flip' in the 'transform' box.

rotating the image

you may rotate the image 90 degrees (counterclockwise) or -90 degrees (clockwise) using the gadget 'rotate' in the 'transform' box.

NOTE: for rotating the image GfxCon needs more
memory
than normal.

1.12 modifying colors

modifying colors

=====

brightness and contrast

At the bottom of the box 'colors' there are two gadgets to change the brightness and contrast of the whole image. The value for the brightness must be in the range between -255 and +255 inclusive. The value for contrast should of course be at least 1%. A value between 1% and 99% reduces the contrast, a value above 100% enhances contrast.

Brightness is always changed first.

I.e.: $output = (input + brightness) * contrast$

color effects

There are several special color effects, which you can choose using the popup-gadget 'effects' in the 'colors' box:

convert to black-white : choose 'black/white'

```
invert image      :      'inverse'
black-white-negative :      'inperse & bw'
```

Besides this you can select filters. In the 'effects'-popup-gadget there are filters in these colors:

red, green, blue, yellow, magenta and cyan.

grayscale

If you want the output image to be colored using a linear grayscale (or images using only the colors black and white as needed for printing) choose the item 'grayscale' from the 'effects'-gadget. But remember that this only works if you are converting to a image with a format using a CLUT (colortable).

dithering

If you are reducing the number of colors of a picture with many colors (like JPEG) it is very important to use dithering. If you don't, awful looking "wrong" colors will stay in the output. The principle of dithering is to use several pixels of different colors to achieve the effect that these colors look like the missing one if looked at from a distance.

GfxCon uses at the moment only the 'Floyd-Steinberg'-dithering, that is very time-consuming but produces very good images.

Select the Floyd-Steinberg-dithering with the gadget 'dithering'.

1.13 the conversion-process

the conversion-process

=====

The conversion can be started by clicking the big gadget 'convert' at the bottom left. A window will open telling you what GfxCon is doing at the moment (rotating, saving...) in the lower part and showing how much of this action is finished with the bar in the upper part of the window.

It may be that another window will open, where you may enter additional parameters (like the quality of
 JPEG-output
 or the
 size of the image
 with
 Postscript-output
).

1.14 showing information about the input-image

showing information about the input-image

=====

With the big gadget 'infos' at the bottom, you can display some details about the input-image. What is displayed depends on the format of the input-image and what is really stored in the image. This information can be very complex (like TIFF) or very scarce (like IMG).

1.15 supported formats

supported formats

=====

input formats

output formats

ILBM

ILBM

LBM

PCX

RGB8 / RGBN

JPEG

PCX

RGB-Raw

IMG

Postscript

BMP / RLE4 / RLE8

GIF

GIF

TIFF

JPEG

Targa

RGB-Raw

1.16 problemes

any problems ?

=====

There are files called 'GFXCON...' in a directory of my harddisc. Have they any use?

These files are created if the program needs to save parts of the image it is working on onto the hard-disc (see
virtual memory
).

Normally, these files are automatically deleted after use. But if you reset your computer while GfxCon is working it doesn't have a chance to do this.

Please delete these files by hand or start GfxCon again to delete these files automatically.

After starting the program I only get the message 'window does not fit on the screen'.

GfxCon is font-sensitive. You have selected too big fonts for the gadgets, which would require a window that is too large to fit on the screen.

Please select smaller fonts in the
Tooltypes
.

1.17 setting the tooltypes

These tooltypes are used by GfxCon:

LANGUAGE - lets you select the language of GfxCon. Currently possible values are: 'english', 'german' or 'englisch', 'deutsch'

TEMPDIR - As GfxCon uses a big amount of memory, it swaps part of the image to hard-disc if more memory is needed than available (see
virtual memory
) to let you even convert pictures that are too large to fit in the memory.

With this tooltype you specify the directory into which GfxCon writes these temporary files.

NOTE: This string will be used as prefix to the

filename. Therefore do NOT enter 'sys:t' but 'sys:t/'.

HINT: Insert a command in your s:user-startup to clear this directory during startup, as files may stay in there if a strange error occurs or the system crashes.

STDLOADPATH - Most people have most of their image in the same directory. To prevent you from entering the path to this directory each time you start GfxCon, you may specify a default directory which will automatically be used if you use the filerequester.

STDSAVEPATH - The same as STDLOADPATH for the output-file.

GADGETFONT - With this tooltype you may specify the font for most of the gadgets and nearly all output. The format of this tooltype is: '<name> <size>'. For example: 'topaz 8' or 'helvetica 18'.

TITLEFONT - The font for titles and big gadgets. The format is the same as that of 'GADGETFONT'.

MINMEM - How much memory (in bytes) GfxCon may not use for its image-data.

MINMEMBLOCK - GfxCon only allocates memory if there is a mem-block with at least "MINMEMBLOCK" bytes.

1.18 automatic format-recognition

GfxCon determines the input-format by itself by scanning the input for unequivocal characteristics. It does not look at the suffix of the filename (as most programs do).

The format-recognition will be called when you leave to topmost string-gadget in the 'load'-box.

The algorithm works fine in almost every case. But theoretically it may be, that the wrong format is guessed. Then you have to set it manually using the popup-gadget.

1.19 memory usage

This program needs very much memory, as all images are stored internally using 24 bit. Additionally the program needs one more byte for each pixel for special data.

You may estimate the memory-usage using this equation:

$$\begin{array}{c} +-----+ \\ | \\ | \text{ memory usage} = \text{height in pixels} * \text{width in pixel} * 4 \text{ Bytes} | \\ | \\ +-----+ \end{array}$$

For example: a 320x 256 image needs 320 kByte
 640x 512 1.25 MB
 800x 600 ca. 1.80 MB
 1024x 768 3 MB
 1280x1024 5 MB

exceptions

Some effects (resize, rotate) need two buffers at the same time. But the memory that is needed is not the memory of these two buffers added together, as GfxCon knows what parts of the image it does not need any more. These parts will be freed when it is not needed any more.

virtual memory

To make it possible to convert images that do not fit into the memory, parts of the image, that are not needed at the moment are written to hard-disc (see virtual memory).

1.20 virtual memory

If more memory is needed to process an image than available, parts of the image are saved to hard-disc to use the memory to do other calculations there. ←

If a image would for example require 5 MB to be processed, but you only have 3 MB free memory, GfxCon will swap at least 2 MB to hard-disc during processing the image.

These files are of course deleted just when they are not needed any more, not to fill the hard-disc with useless files.

NOTE: If you reset the computer while GfxCon is using such temporary files, GfxCon cannot delete this files. Beginning with version 1.3, GfxCon deletes this files automatically at start.

DO NOT delete any file, while GfxCon is using them!!!

1.21 IFF - ILBM Format

IFF - ILBM format

load

- images with any number of colors, 24 bit, HAM6/8 and ExtraHalfBrite
- the PC-version of the ILBM-format with 16 or 256 colors
- compressed and uncompressed

save

- images with any number of colors (even more than 256)
- images with 24 Bit or HAM6/8
- all images are compressed
- not supported at the moment is ExtraHalfbrite

1.22 PCX - format

PCX - format

load

- all PCX-images
- compressed and uncompressed

save

- images with 2, 16 or 256 (all possibilities)
 - all images are compressed
 - PCX-version 3.0 is used
-

1.23 JPEG - format

JPEG format

load

- all color JPEG/JFIF images, greyscale JPEG is not supported

save

- the quality of the output can be altered in the range between 25% and 100%

1.24 Postscript

Postscript - format

load

- not supported at the moment

save

- output is intended to drive a Postscript-printer

- it is not intended as exchange-format (EPS)

- the
size and position
of the picture in the output can be specified in 'cm'.

- only black/white output (color images are automatically converted to black/white)

1.25 RGB - raw - files

RGB - raw - files (3 files with the red, green and blue parts of the image)

load

- As the size of the picture cannot be determined by looking at the input-file. Therefore GfxCon guesses the width of the picture. If

it guesses wrong, you can correct the width in pixel.

save

- yes

1.26 RGB8 / RGBN - format

IFF - RGB8 / RGBN - format (Imagine)

load

- RGBN (4096 colors) and RGB8 (truecolor, 24 bit) images

save

- no

1.27 IMG - format

IMG (GEM Image File Format)

load

- all IMG-files

- IMG files do not contain a colortable. GfxCon uses the PC default colors.

save

- no

1.28 BMP/RLE4/RLE8 - formats

BMP / RLE4 / RLE8 (MS-Windows)

load

- all formats
- compression is supported

save

- no

1.29 GIF - format

GIF - format

load

- all images (even those with fewer than 256 colors)
- raster and interlace format
- only the first picture of the file is read

save

- always saves GIF87a / raster-images

1.30 TIFF - format

TIFF - format

load

- uncompressed: black/white (2 colors)
 16 colors (with and without bitplanes)
 256 colors (with and without bitplanes)
 RGB-images (without bitplanes)
 - CCITT3 (FAX) compressed: black/white
 - RunLength black/white
 256 colors
 - LZW black/white
-

256 colors
 RGB-images

- JPEG is not supported.

save

- no

1.31 Targa - format

Targa (TGA) - format

load

- only truecolor pictures
 - 15,16,24 or 32 bit
 - uncompressed or compressed with RLE

save

- no

1.32 position and size of Postscript output

position and size of Postscript output
 =====

The window to enter the position and size of the Postscript-output opens automatically. You must specify the position of the top left corner of the picture measured from the bottom left corner of the paper in portrait-orientation (may depend on your printer) and the width and height of the image (in cm).

The values that are by default filled into the gadgets produce a picture 18cm wide and a height according to the aspect of the picture.

1.33 about GfxCon

NOTE: As usual, this version 1.8b is of course the last one. And ↔
 of
 course I am sure it will be the last :-)

But the successor is under development (for some time now). Those

who are brave and want to test it can find preview versions of it on my WWW page.

author
=====

Dirk Farin
Kapellenweg 15
72070 Tübingen
Germany

E-Mail: farindk@trick.informatik.uni-stuttgart.de
WWW: <http://tick.informatik.uni-stuttgart.de/~farindk>

copyright

(registration)

I would be happy to receive any proposals and bug-reports (←
E-Mail
preferred). I will answer to all of these if possible.

If you miss your favourite format (and probably do have any documentation about it), I will try to implement this format too in future versions.

This program has been written on an Amiga3000T/25Mhz with 5MB, Kick2.04 and graphics-card. It has been compiled with SAS/C V6.55.

It has been successfully tested on these systems:

Amiga 3000T / OS2.04
Amiga 3000 / OS2.04
Amiga 2000 / OS2.04
Amiga 500 / OS2.04
Amiga 4000 / OS3.0
Amiga 1200 / OS3.0
... :-))

thanx go to:

Fred Fish for his Fish-discs / -CDs.
Matthew Dillon for his DME-editor.
Bob Richardson who persuaded me to write version 1.8
The maintainers of Aminet
All users of GfxCon who did send me E-mails, postcards letters or/and
did sent bug reports !!!

1.34 registration

(registration)

This program is Cardware / EMailware.

As this program is very old and not up to date, I think it would be unfair to sell it as Shareware.

But if you use this program sometimes I would be very happy to receive a postcard or even an EMail, in which you can tell me, if you like this program or what could be done better.

Of course I will answer any questions concerning the program.

my address

1.35 copyright

copyright:

GfxCon is

Cardware / EMailware.

It may be included on Public-Domain-discs, AmiNet and spread in ↔
any
other noncommercial way. No part of GfxCon may be used for commercial
purpose without my permission.

The Graphics Interchange Format(c) is the Copyright property of
CompuServe Incorporated. GIF(sm) is a Service Mark property of
CompuServe Incorporated.
