

Image Alchemy

Version 1.5

November 11, 1991

Handmade Software, Inc.
15951 Los Gatos Blvd., Suite 17
Los Gatos, CA 95032

+1 408 358-1292
+1 408 356-4143 fax
+1 408 356-3297 BBS

Internet: hsi@netcom.COM
CompuServe: 71330,3136

Shareware Fundamentals and Specifics

We encourage you to freely copy and distribute the shareware version of Image Alchemy provided that:

1. No fee beyond normal media, duplication, and shipping costs may be charged.
2. Shareware vendors and computer user groups who charge less than \$7.00 per disk may distribute Image Alchemy, but Handmade Software must be supplied with a copy of the first catalogue issue offering each new revision of Image Alchemy.
3. Others may only distribute Image Alchemy with the written permission of Handmade Software. In all cases, it must be clearly stated to the purchaser that he or she is receiving an unregistered copy of a shareware product
4. The distribution files must be distributed in their original forms. The unrestricted version may not be distributed.

If you received Image Alchemy as shareware (i.e. you downloaded from a bulletin board, you received it from a friend, it came with some hardware you purchased, or you bought a disk from a software library), and if you use it beyond a two week trial period, you must register the program using the accompanying order form.

For the registration fee, you will receive the current, retail version of Image Alchemy without the 640 by 480 image size restriction and a typeset and printed manual (complete with a table of contents and an index). You will be notified of significant upgrades to Image Alchemy, you will be placed on a mailing list to receive information about future products from Handmade Software, and you will be entitled to phone and email support.

For ordering information please see the order form at the end of this manual or in the file order.frm.

Notice Handmade Software, Inc. makes no warranty of any kind either expressed or implied. In particular we make no warranty as to merchantability or fitness for a particular purpose.

In no event shall Handmade Software, Inc. be liable for any errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of the Image Alchemy product or documentation.

This document contains proprietary information which is protected by copyright. No part of this document may be photocopied, reproduced, or translated without the prior written consent of Handmade Software, Inc.

The information in this document is subject to change without notice.

Copyright (c) 1990-1991 Handmade Software, Inc.

All Rights Reserved

Author credits

Image Alchemy Marcos H. Woehrmann
was written by Allan N. Hessenflow
David Kettmann

Other credits

Marc Schneider Who provided assistance with the Sun
implementation of Image Alchemy including Beta
testing and answering questions about the
internal format of Sun Raster files.

Jack, Norm, Who proofread the manual (any remaining mistakes
Erwin, and Dave are because we made changes after they read it
for the final time).

Everyone Else Who gave us advice and assistance and especially
to those people who sent us sample image files.

Trademarks Image Alchemy is a trademark of Handmade
Software, Inc.

All other products or services mentioned in this
manual, including: IBM PC, IBM PC AT, 80286,
80386, 80486, VGA, 8514/A, Paradise, Everex,
Trident, Video 7, Tseng Labs, Western Digital,
MS-DOS, PC-DOS, SPARC, Sun, SPARCstation,
SPARCserver, SunOS, Targa, PostScript, EPS,
Encapsulated PostScript, GIF, ILBM, IFF,
Macintosh, Silicon Graphics, SGI, PCX, TIFF,
Windows, Windows BitMaP, EGA, PCL, HP, AI, PS/2,
HAM, PC Paintbrush, MacBinary, PHIPS, NeXT, C-
Cube, Storm Technology, Radius, ColorSqueeze,
VFCTool, Amiga, CompuServe, LaserJet, Melior,
and Gill Sans are trademarks, registered
trademarks, service marks, or registered service
marks of their respective companies or
organizations.

Introduction to Image Alchemy

What is Image Alchemy? Image Alchemy is a software utility that manipulates computer image files.

The main thing that Image Alchemy does is to convert between various graphics file formats. Image Alchemy can translate between a large variety of file formats including industry standards such as GIF and TIFF and vendor specific file formats such as Sun Raster and Scodl. Currently Alchemy supports over 30 different formats, and new formats are always being added; in fact, our goal is to have Image Alchemy be able to read and write every graphic file in the world.

Image Alchemy can also make changes in an image. For example, Image Alchemy can re-size an image, change the number of colours in an image, change an image from colour to black and white, and change the colour space an image uses.

Finally, Image Alchemy performs JPEG compression. This is a new standard for image compression that can achieve much higher compression ratios than conventional compression techniques. It achieves this high compression ratio by not entirely preserving the original image (this is referred to as "lossy" compression). For further information see Appendix C, What is JPEG Compression.

About this manual This manual is divided into 8 chapters, 11 appendices, a glossary, and references.

Chapter 0 Introduction and Conventions

Chapter 1 Installation Instructions

Chapter 2 Introduction to Alchemy

Chapter 3 Output Options

Chapter 4 General Options

Chapter 5 Colour and Palette Options

Chapter 6 Scaling Options

Chapter 7 Viewing Options

Appendix A Answers to Frequently Asked Questions

Appendix B Colour and Dithering

Appendix C JPEG Description

Appendix D Customer Support

Appendix E Binary Information Files (BIF)

Appendix F HSI Raw Files

Appendix G Undercolour Removal Files

Appendix H PAL Files

Appendix I Version History

Appendix J Acknowledgements

Appendix K Other Useful Software

Glossary

References

Pathnames Because the MS-DOS and UNIX operating systems use different conventions for path names users of UNIX will have to substitute forward slashes, "/", for the back slashes, "\", found in the examples in this manual.

Unintentional wildcard expansion UNIX users should also be aware that the UNIX shell they are using may be performing wildcard expansion on certain characters (generally "*" and "?"). Since these are options which Alchemy uses they need to be escaped to prevent the wildcard substitution. This is done by using a back slash, "\", before the character (so -? becomes -\?).

Installing Image Alchemy

Overview Installation of Image Alchemy is simple; it involves copying the Alchemy program off of the floppy disk or tape onto your hard drive or network.

You need to be familiar with the copy command if doing a DOS installation and the tar command if doing a UNIX installation. If you are not familiar with these commands you may wish to read the manuals which came with your computer or ask someone to assist you.

The installation instructions are divided into different sections for IBM PC, Sun-4 (Sparc), and Sun-3. Please refer to the section which corresponds to your hardware.

IBM PC

Required equipment At a minimum you must have the following hardware and software to run Image Alchemy.

Computer An MS-DOS computer equipped with an 80286, 80386, or 80486.

Many of the conversions that Alchemy does are cpu intensive, so a faster computer is definitely an advantage.

Memory At least 380k of free memory.

Image Alchemy will run much faster if you have more than 420k of memory available. Some conversions and some images require even more memory (Alchemy will attempt to use all available system memory, so if you get out of memory errors or warnings try removing as many resident programs as you can (also, installing MS-DOS 5.0 will free up more memory)).

Alchemy does not currently make use of expanded or extended memory.

Hard drive A hard drive with at least as much free space as four times the size of the image being converted (i.e. a 640x480 image will require approximately 1.2 megabytes of free space).

Display A supported SVGA or 8514/A board, if you wish to view images.

Supported SVGA boards include those with the Paradise, Everex, Trident, Video 7, and the Tseng Labs chipsets. Supported 8514/A boards include IBM and those with the Western Digital chipset.

Operating MS-DOS or PC-DOS 3.x or greater (because of the system additional free memory available use of MS-DOS 5.0 is highly recommended).

Optional equipment The following hardware and software is optional to run Image Alchemy.

VESA VGA board To get full use out of viewing images on a SVGA board a VESA driver is needed. Without a VESA driver Alchemy is limited to displaying images in 320x200x256, 360x480x256, and 640x400x256 mode; with a VESA driver the resolution is limited only by your SVGA board and monitor. Resolutions up to 1280x1024x256 are possible.

The VESA driver is supplied by the manufacturer of your SVGA board. It may already be included in the BIOS on your SVGA board, or it may have been shipped on a floppy disk with your SVGA board. Consult the documentation that came with the SVGA board. If there is no mention of a VESA driver in the documentation contact your dealer or the manufacturer of your SVGA board; VESA drivers are currently available for VGA boards using chipsets from Cirrus Logic, ATI Technologies, Chips and Technologies, Everex Systems, Genoa Systems, Paradise Logic, Sigma Designs, STB Systems, Tecmar, Headland Technology (Video 7), Orchid Technology, Appian Technology, Trident Microsystems, and Oak Technology.

Refer to your SVGA documentation on how to install the VESA driver.

Disk cache You can greatly increase the speed of many of Image Alchemy's conversions by installing a disk cache that postpones writes. This is because Alchemy uses the hard drive to temporarily store data during many conversions. A write postponing cache will use extended and/or expanded memory to make this more efficient.

An example of a write postponing cache, and the one we use at Handmade Software, is the Norton Utilities 6.01 supplied ncache.exe (the cache supplied with MS-DOS 5.0, smartdrv.sys, is NOT a write postponing cache, and while it will help somewhat, it is not nearly as effective).

Refer to the documentation that came with your cache on how to install it.

Math A math coprocessor will not affect the speed of coprocessor most operations; the only operations that use much floating point math are scaling types c and d.

Packing list The enclosed diskette contains the following files:

ALCHEMY.EXE The Alchemy program.

READ.ME A text document describing any last minute revisions.

SAMPLES A directory containing sample data files and images. See the READ.ME file in this directory for further information.

Installation instructions Using the MS-DOS copy command copy the program ALCHEMY.EXE to a directory in your path. There are no support or configuration files which need to be copied. (For more information on the copy command or the path command see the DOS user's manual which came with MS-DOS).

Note to advanced users: Alchemy uses the environment variable TMP to determine where to open its temporary files. If you have a big enough ram drive you will want to specify it using the TMP variable. Be aware that Alchemy needs up to 4 times as much space on that drive as the size of the image (a 640 by 480 image requires up to 1.2 Megabytes). An example of setting the TMP variable to drive e: would be "set TMP=e:".

Sun-4

Required equipment At a minimum you must have the following hardware and software to run Image Alchemy.

Computer A SPARC equipped Sun (either a Sun-4, SPARCstation, or SPARCserver).

Disk space Alchemy uses disk space while converting images. You should have at least 4 times as much disk space available as the image you are converting (i.e. a 640x480 image will require approximately 1.2 megabytes of disk space).

Operating system SunOS 4.0.3 or greater.

Packing list The enclosed diskette or tape is in tar format and contains the following files:

alchemy The Alchemy program.

read.me A text document describing any last minute revisions.

samples A directory containing sample data files and images. See the read.me file in this directory for further information.

Installation instructions Use the Sun supplied program tar to copy the files from the distribution disk or tape to the current directory.

To install the software from diskette use:

```
tar xvf /dev/fd0
```

For tape installation replace /dev/fd0 with the name of the tape device. The name of the tape device varies between different models and configurations of Sun systems; ask your system administrator if you don't know the name of your tape device.

Note to advanced users Alchemy uses the environment variable TMPDIR to determine where to put its temporary files.

This is usually set to /usr/tmp or /tmp, but if you are converting very large images there may not be enough space available in the partition those directories are on. In that case you may want to set the environment variable TMPDIR to a different partition. For example, to set the temporary file directory to the directory /home/images use "setenv TMPDIR /home/images". Contact your system administrator if you have problems with Alchemy running out of disk space while converting images.

Sun-3

Required equipment At a minimum you must have the following hardware and software to run Image Alchemy.

Computer A 68020 or 68030 equipped Sun (generally called a Sun-3).

Disk space Alchemy uses disk space while converting images. You should have at least 4 times as much disk space available as the image you are converting (i.e. a 640x480 image will require approximately 1.2 megabytes of disk space).

Operating system SunOS 4.0.3 or greater.

Packing list The enclosed diskette or tape is in tar format and contains the following files:

alchemy The Alchemy program.

read.me A text document describing any last minute revisions.

samples A directory containing sample data files and images. See the read.me file in this directory for further information.

Installation instructions Use the Sun supplied program tar to copy the files from the distribution disk or tape to the current directory.

To install the software from diskette use:

```
tar xvf /dev/fd0
```

For tape installation replace /dev/fd0 with the name of the tape device. The name of the tape device varies between different models and configurations of Sun systems; ask your system administrator if you don't know the name of your tape device.

Note to advanced users Alchemy uses the environment variable TMPDIR to determine where to put its temporary files.

This is usually set to /usr/tmp or /tmp, but if you are converting very large images there may not be enough space available in the partition those directories are on. In that case you may want to set the environment variable TMPDIR to a different partition. For example, to set the temporary file directory to the directory /home/images use "setenv TMPDIR /home/images". Contact your system administrator if you have problems with Alchemy running out of disk space while converting images.

Introduction

Basic instructions Image Alchemy is a command line driven program. This is different from a menu driven or mouse driven program in that you instruct Alchemy what to do only when you execute it from the command line. Once the program is running there are no further choices or user interaction.

The basic Image Alchemy usage instructions are:

```
alchemy -option [-option ...] inputFileName
[outputFileName] [outputPathName]
```

This simply means that you type the name of the program, Alchemy, followed by at least one option, followed by the input file name. You may optionally specify an output file name or an output path name.

Options Options are the commands that you give Alchemy so that it knows what you want it to do. So that Alchemy can distinguish between options and file names, options are preceded by a dash ("-").

The only option that is required is the output file format (or the viewing option, for MS-DOS users). Image Alchemy will make reasonable decisions for all of the other options.

Some options take parameters. The parameters may immediately follow the option or be separated by a space. For example, either -c128 or -c 128 is acceptable.

The options are documented in Chapters 3 through 7.

Note that options can appear anywhere in the command line and generally they can be in any order (certain options take parameters; in those cases the parameters must follow the option). The case of the options is significant.

InputFileName The inputFileName is any valid file name. This is the name of an existing image file that you are converting from or viewing. The inputFileName is required. It may include an optional drive and/or path.

OutputFileName The outputFileName is the name of the file you are converting the image to. The outputFileName is optional; if it is not specified Image Alchemy generates one by substituting an appropriate extension to the input file name. If you specify an outputFileName and it does not include an extension one will be added. The outputFileName may include an optional drive and/or path. If you do not supply a path the current directory will be used as the destination directory.

OutputPathName The outputPathName is the location where you want to put the output that Alchemy will create. The outputPathName is optional; if it is not specified Alchemy places the output in the current directory or in the directory specified as part of the outputFileName.

Limitations on filenames Since Alchemy lets you optionally enter a space between an option and its parameter it is possible to confuse Alchemy if one of the filenames starts with a number. In particular, if you use an option which has an optional parameter, you do not use a parameter, and you follow that option immediately with a filename which starts with a number, Alchemy doesn't realize that the filename is not the parameter. While it sounds unlikely that this would ever be a problem it actually happens quite often.

For example, if you wanted to convert the file 12.gif to a Targa file with the name output.tga you would have to be careful of the order you specified things.

If you say

```
alchemy -a 12.gif output.tga
```

Alchemy would misinterpret that as

```
alchemy -a12 .gif output.tga
```

and would generate an error.

The easiest way around this problem is to always put the filenames first

```
alchemy 12.gif output.tga -a
```

or don't use filenames which begin with a number.

Output path The output path name is the location where Alchemy will place its output.

The outputPathName can be specified as part of the outputFileName if you are specifying an outputFileName.

For example

```
alchemy sample.jpg -g test.gif \images
```

is the same as

```
alchemy sample.jpg -g \images\test.gif
```

However,

```
alchemy sample.jpg -g
```

is not the same as

```
alchemy sample.jpg -g \images
```

When using the wildcard option, Alchemy allows wildcards and multiple file names. In this case the use of an outputFileName is not allowed, but the use of the outputPathName is. See the wildcard option below for more information.

Using multiple Sometimes you may know what you want to runs of Alchemy accomplish but not how to specify the correct combination of options. For example, you may wish to re-size a true colour Targa file that you have scanned and convert it to a 16 colour GIF file. Let's say that the input file name is file.tga and you want to generate a file with the name file.gif. In this case you could type:

```
alchemy file.tga -Xb640 -Yb480 -c16 -g
```

However, there would be no penalty in quality (and little in speed) if you did things in two steps:

```
alchemy file.tga -Xb640 -Yb480 -r temp.raw
```

```
alchemy temp.raw -c16 -g file.gif
```

In this case you are telling Alchemy to use a temporary raw file called temp.raw. It turns out that Alchemy would have done that automatically in the first case. Of course you will have to manually delete the file temp.raw when the conversion is finished.

The order of steps is important in many cases. For example, reversing the order of the two operations in the previous example:

```
alchemy file.tga -c16 -g temp.raw
```

```
alchemy temp.raw -Xb640 -Yb480 -g file
```

would give different results. This is because the scaling operation has to temporarily convert the image to true colour, but the GIF file you are generating has to be paletted, so the second operation would re-dither the image.

Sometimes you will have to perform operations using multiple steps because there are some combinations of options that Alchemy explicitly does not allow. These combinations of options are not allowed because the results would not be what you expect.

For example, using the spiff option, -S, in combination with the false colour option, -F, would spiff the image first and then false colour it, which would give the same results as just using the false colour option. Since that isn't something that you would ever want to do, Alchemy will complain if you specify both of those options at the same time.

In this case you could false colour the image first, generating a temporary image, and then spiff that image.

Output Options

Introduction The one option which is always required when running Image Alchemy is the output image file type. Even if you are just re-sizing an image, or changing the number of colours in an image, Alchemy needs to know what type of image you are creating.

The file types that Image Alchemy supports are listed below. In addition to the syntax required to generate the file, any known restrictions or limitations are listed. If you have trouble reading an image in one of the file formats we claim to support please contact us (see Appendix D, Customer Support).

The common output options consist of a single letter. The option, like all Alchemy options, is preceded by a dash, "-". The less common output options consist of a letter preceded by two dashes, "--".

Variations Some of the output formats have several variations; in those cases you specify which variation you want with an optional letter and/or number after the output option.

Example The option to generate a Windows Bitmap file is -w. There are two types of Bitmap files: uncompressed and Run-Length-Encoded (RLE). To write out an uncompressed Bitmap file use -w0; to write out a RLE Bitmap file use -w1 (the default Bitmap file is uncompressed, so a -w without any parameter following it would also generate an uncompressed Bitmap file). Note that Alchemy allows spaces between the option and parameter, so typing -w 1 would be the same as -w1.

Variations Be aware that the other options specified on the command line also affect the type of file that is generated.

Example Within the Windows Bitmap file type there are 1 bit, 4 bit, 8 bit, and 24 bit files.

Alchemy always generates a file using the best match of the file type and the output image. So, in the case of Windows Bitmap files, if the output image is black and white a 1 bit file is generated. If the output image is paletted with 16 colours or less a 4 bit file is generated. If the output image is paletted with more than 16 colours an 8 bit file is generated. And if the output image is true colour a 24 bit file is generated.

You can explicitly force any of these file types by using other Alchemy options. For example, if you wanted a 1 bit Windows Bitmap file you would specify -c2 -b -w. To force a 4 bit file use -c16 -w. To force an 8 bit file use -c256 -w. And to force a true colour file use -24 -w.

Identifying image files Image Alchemy identifies the type of file being read by checking various magic numbers and other information that varies from format to format. Unfortunately some formats do not have a magic number; in those cases Alchemy guesses as to the image type. It is possible for Image Alchemy to incorrectly identify an image; if this happens please contact us.

MacBinary When reading images Alchemy automatically recognizes and reads MacBinary files (MacBinary files are generated when you accidentally leave MacBinary mode on when transferring a file from a Macintosh).

Other information Alchemy will preserve as much information in each file as is reasonable; this always includes the height and width of the image and the number of colours in the image. Some file types include other data, such as the name of the image, the aspect ratio of the image, the date the image was created, etc. Since most of these items are only supported by a few file formats, Alchemy discards everything but the height, width, number of colours, gamma, aspect ratio, and resolution values.

Output Options The individual output options supported by Alchemy are described in alphabetical order on the following pages. The descriptions follow the template given overleaf.

Name of format	-option
----------------	---------

Overview of file format.

Syntax	Description of syntax.
--------	------------------------

Parameter	Description of the parameters.
-----------	--------------------------------

Extensions The extensions commonly used for this image format. When multiple extensions are listed Alchemy writes files using the first one, but will check for files using all extensions (in the order listed). Four letter extensions are skipped on MS-DOS systems.

Creator	The company or individual who created this image format. Please contact them for more information on the format.
---------	--

Used by Programs or types of software that use this image format.

Variations A list of the supported variations.

Limitations Any known limitations that Alchemy has when reading or writing this image format.

Comments	Miscellaneous things you should be aware of.
----------	--

Related options Other Alchemy options that affect the reading or writing of this image format. Note that -8, -24 (and, for some formats, -15, -16, and -32), -c, and -b options have an effect for most image formats and are not listed explicitly.

Example Example conversions involving this image format.

ADEX --A

ADEX files are used by the ADEX Corporation
ChromaGraph series of graphics cards.

Syntax --A compressionType

Parameter compressionType:
0:None
1:Run Length Coded
The default is None.

Extensions .img
.rle

Creator ADEX Corporation

Used by ADEX ChromaGraph cards.

Variations 4 bit and 8 bit images.

Comments Some ADEX files don't contain a palette; in those cases there's usually a second ADEX file which contains the palette to be used. To read those images that don't have palettes, use the -F false colour option to read the palette from a separate file.

Related options -F False colour

Example Convert the file test.gif to an uncompressed ADEX file called test.img:

alchemy --A test.gif

There are quite a few programs which produce image files which contain just pixel data. These image files do not have a header and hence do not include enough information to allow Alchemy to read them.

BIF files are a method which can be used to allow Alchemy to read these images. BIF files can also be created to allow images to be read by software which expects images to be just pixels. Since required information, such as the height and width of the image, are not present in these files you must supply it.

Syntax -B

Extensions .bif For ASCII file describing image.
 .raw For actual image data.

Creator Handmade Software, Inc.

Used by Image Alchemy
 Various image processing software

Variations 24 bit true colour, 8 bit gray-scale, and 1 bit
 black and white.

Limitations Paletted files can not be read in (a work around
 is to generate a .PAL file and then false colour
 the gray-scale image using the -F option).

Comments BIF files are used to read and write files which
 consist entirely of image data. You have to
 generate a text file which describes the format
 of the data you are trying to read in. This
 file is called a BIF file. The format of BIF
 files is documented in Appendix E, Binary
 Information Files. You then instruct Alchemy to
 read the image data by giving it the name of the
 .BIF file.

Related options -F False colour

Examples Convert the file data to a GIF file:

 alchemy data.bif -g

Encapsulated PostScript (EPS)

-e

EPS files are a subset of PostScript; they may be included by other PostScript files without requiring that the importing software be able to interpret the file.

Syntax -e previewType

Parameter previewType:
 0:None
 1:Device independent
 2:TIFF
The default is device independent.

Extensions .epsi
 .epi

Creator Adobe Systems, Inc.

Used by PostScript printers

Variations Gray-scale, RGB, and indexed images.

Limitations Alchemy can only write, not read, EPS images.

Comments If the output is gray-scale, it will work with any PostScript device. If it's true colour, then the CMYK extensions or a level 2 device is required. For paletted files, a level 2 device is always required.

Examples Convert the file input.gif to a colour EPS file called input.eps which will not require level 2 PostScript (but will require CMYK extensions), with no preview:

```
alchemy -e0 -24 input.gif
```

Convert the file input.gif to a gray-scale EPS file called gray.eps, with a device independent preview:

```
alchemy -e -b input.gif gray.eps
```

Erdas LAN/GIS

--e

Erdas files are used by Erdas image processing software.

Syntax --e

Extensions .lan
 .gis

Creator Erdas Inc.

Used by Erdas remote sensing software.

Variations Reads and writes 1 and 3 band files.

 Reads 4, 8, and 16 bit files. Writes 8 bit files.

Limitations When writing Erdas files Alchemy does not change the extension depending on the number of bands in the image; according to the specification gray-scale files should have the extension .gis and true colour files should have the extension .lan. Alchemy always uses .lan.

Comments 1 band files are read in as gray-scale images.

 3 band files are read in as true colour images.
 The colour mapping between RGB and bands 1, 2, and 3 is Red=Band 1, Green=Band 2, and Blue=Band 3.

Examples Convert the GIS file texas.gis to a Sun raster file:

 alchemy texas.gis -s

 Convert the file satellite.image to a GIS file.

 alchemy -b --e satellite.image
 satellite.gis

 Convert the file satellite.image to a LAN file.

 alchemy --e satellite.image satellite.lan

Freedom of the Press

--f

Freedom of the Press is a PostScript interpreter from Custom Applications that converts PostScript files to raster files. The Freedom of the Press format is one of the file types it can create.

Syntax --f

Extension .fop

Creator Custom Applications

Limitations Output only.

Only CMYK 1 bit per component per pixel supported.

Comments Freedom of the Press images are actually two files, a data file and an info file. You specify the name of the data file and Alchemy automatically generates the name of the info file. The output file is normally output.001, output.002, etc. Alchemy will strip the first part of the name and replace it with 'info', so if you specified an output filename of output.005 there will be another file created called info.005. If you don't specify an extension, Alchemy will use .fop, so you'll get two files named filename.fop and info.fop. Alchemy will overwrite info files without warning.

Related options -C Undercolour removal

Example Convert the file image.tga to a Freedom of the Press image called output.003 and info.003, controlling the undercolour removal process using sample.ucr, scaling the image to 2500 pixels across (and scaling proportionately vertically) using nearest neighbor scaling, and conserving memory:

```
alchemy --f -Csample.ucr -X2500 -+ -$  
image.tga output.003
```

GEM VDI Image File

--g

VDI files are black and white files that were developed by Digital Research to use with GEM.

Syntax --g

Extension .img

Creator Digital Research Inc.

Used by GEM

Variations Reads and writes 1 bit, black and white files.

Limitations The files which are written are not optimally compressed.

Comments Since GEM VDI files are always 1 bit, black and white files, Alchemy assumes the use of the -b, -c2, and -8 options.

Examples Convert the image scan.pcx to a GEM file:

```
alchemy scan.pcx --g
```

Convert the image bigscan.tga to a 640x480 GEM VDI file, using nearest neighbor scaling and type 2 dithering:

```
alchemy --g bigscan.tga -X640 -Y480 -d2
```


GIF

-g

GIF files were originally developed by CompuServe as a machine independent image file format. GIF files are clearly the most popular way of storing 8 bit, scanned or digitized images. In addition the compression ratio achieved by GIF files is usually better than any other 8 bit format in common use. GIF89A files were introduced in 1990 as a method for including text and simple animations.

Syntax -g version

Parameter version:
 0: GIF87A
 1: GIF89A
 The default is GIF87A.

Extension .gif

Creator CompuServe, Incorporated

Used by CompuServe
 Everyone

Variations Reads 1 through 8 bit GIF87A and GIF89A
 interleaved and non-interleaved files.

Writes 1 through 8 bit GIF87A and GIF89A non-interleaved files.

Limitations When reading GIF89A files only the first image
 in the file is read. Any text, overlays,
 pauses, palette changes, etc. are ignored.

Because GIF files only store the size of the palette to the nearest power of 2 the exact palette size is lost when converting to and from GIF files. For example, if you convert a 240 colour Sun Raster file to a GIF file and back to a Sun Raster file the resulting Sun Raster file will have 256 colours.

Comments When writing a file you probably want to use the GIF87A variation, since the GIF89A extensions aren't necessary to store single images and a lot of other software still can't read GIF89A images. The only advantage to GIF89A is that the aspect ratio of the image is preserved (GIF87A does not have a provision for storing aspect ratio).

The GIF format includes a field for storing the colour to be used for the background when viewing files. Alchemy does not make use of this value. Alchemy sets the background colour to the darkest colour in the palette when viewing files and organizes the palette such that the first colour is the darkest colour when writing GIF files, if the palette is created by Alchemy (you can override this by using the -z option).

Related options -z Palette Selection

Examples Convert the image test.pcx to a GIF87A image.

```
alchemy test.pcx -g
```

Convert the file input.tga to a 16 colour GIF89A file:

```
alchemy input.tga -c16 -g1
```

HP PCL files are used by HP LaserJets and compatible printers.

Syntax -P compressionType

Parameter compressionType:
 0: Uncompressed
 2: TIFF compressed
 The default is uncompressed.

Extension .pcl

Creator Hewlett-Packard Company

Used by HP LaserJet printers
 HP compatible laser printers

Variations PCL files are always 1 bit per pixel, black and white.

 Reads and writes uncompressed and TIFF compressed files.

Limitations In addition to raster images, PCL files can include text and vector graphics information. When reading Alchemy only pays attention to raster images in the file and attempts to skip everything else. See Appendix A, Answers to Frequently Asked Questions, for further discussion of this.

 The only resolutions allowed in PCL files are 75 DPI, 100 DPI, 150 DPI, and 300 DPI; the X and Y resolution must be the same. If you specify any other resolution Alchemy automatically alters the resolution to the next higher resolution. If no resolution is specified the default is 300 DPI.

Comments Since PCL files are always 1 bit, black and white files, Alchemy assumes the use of the -b, -c2, and -8 options.

 When converting colour or gray-scale images to PCL you will probably want to scale the output so the image will be larger than the input image. This will allow the dithering to preserve more detail in the image.

PCL files can be used to generate output which can be printed on HP LaserJet and compatible printers. The easiest method is to simply generate a .PCL file and then copy it to the printer by using the copy command (when using the copy command from MS-DOS you will have to use a /B to make sure the entire file is copied to the printer; see the example below for more information).

All HP laser printers can print uncompressed files. The newer models, including the HP LaserJet III series, can print TIFF compressed files. TIFF compressed files are smaller, and therefore are faster to transmit to the printer.

Examples Convert the image image.gif to a HP PCL file, using no compression:

```
alchemy image.gif -P
```

Convert the image small.gif to a HP PCL file called out.pcl with dimensions of 2000 by 2000 at 300 DPI:

```
alchemy small.gif -P -X2000 -Y2000 -D 300  
300 out
```

Print the image generated by the previous example (for MS-DOS machines only):

```
copy /B out.pcl prn:
```

Convert the image small.gif to a HP PCL file called out2.pcl with dimensions of 2000 by 2000 at 300 DPI, using TIFF compression:

```
alchemy small.gif -P2 -X2000 -Y2000 -D 300  
300 out2
```

RTL files are used by HP colour raster printers and plotters.

Syntax --r type

Parameter type:
 0:PaintJet
 1:HP7600 uncompressed
 2:HP7600 PackBits compression
 3:HP7600 planar uncompressed
 4:HP7600 planar PackBits compression
 5:HP7600 planar Group III compression
The default is HP7600 PackBits compression.

Extension .rtl

Creator Hewlett-Packard Company

Used by HP raster plotters and printers including
 PaintJets and HP 7600 Series plotters.

Variations CMYK, 1 bit per component per pixel.

Limitations Output only.

The output file is always colour, even if the image is black & white.

Comments RTL files can be used to produce output which can be printed on HP colour printers and raster plotters. The file can be printed by sending the file to the plotter. There is no additional setup required for the PaintJet. HP7600 series plotters should be in HP-GL/2 mode; best results will generally be achieved with compensation off. To get colour plots from the HP7600 series the plotter must be in 4 or 5 pass mode. The file must be sent to the printer/plotter in binary mode (for MS-DOS systems, use the /B option with the copy command).

If the input is black and white, you probably want to do the conversion without an undercolour removal file and with dithering off. See below for an example.

If the input is gray-scale, you probably do want to use an undercolour removal file to perform density correction, but with 100% black removal (the black removal tables should contain 0 through 255, increasing by one each line) so that the output won't contain cyan, magenta, or yellow. The sample directory on the distribution diskette has a UCR file called gray.ucr which has 100% black removal.

Related options -C Undercolour Removal File
-d Specify dither type

Examples Convert the black and white image test.wpg to a RTL file for a PaintJet called test.rtl, not using a UCR file and with dithering off:

```
alchemy --r0 -d0 test.wpg
```

Convert the file image.tga to a RTL file for a PaintJet called image.rtl, using the undercolour removal file sample.ucr:

```
alchemy --r0 -Csample.ucr image.tga
```

Convert the file image.tga to a planar RTL file called image.rtl using PackBits compression, controlling the undercolour removal process using sample.ucr, scaling the image to 3000 pixels across using good quality scaling, preserving the aspect ratio (by proportionately scaling the image vertically), and conserving memory:

```
alchemy --r4 -Csample.ucr -Xb3000 -+ -$  
image.tga
```

Plot the image generated by the previous example (for MS-DOS machines with the plotter connected to a port which is mapped to prn:):

```
copy /b image.rtl prn:
```

HSI JPEG

--j

The HSI JPEG format is a variation of the JPEG format that was designed by Handmade Software to better compress paletted images. Paletted images often have large areas where the image consists of 1 or 2 colours; JPEG compression does a poor job on these sections when compared to LZW compression. HSI JPEG files are a combination of JPEG and LZW compression.

Syntax --j

Extension .jpg

Creator Handmade Software, Inc.

Used by Image Alchemy
 GIF2JPG (another Handmade Software product)

Variations 8 bit paletted

Comments HSI JPEG files are not compatible with JPEG or JFIF files. If you intend to transfer files to other systems do not use this format, use the standard JPEG format instead (using the -j option).

Example Convert the file madonna.gif to a HSI JPEG file:

alchemy --j madonna.gif

PAL files are palettes which are ASCII files that can be edited with a text editor.

Syntax -l (lower case L)

Extension .pal

Creator Handmade Software, Inc.

Used by Image Alchemy

Variations Palette files are always ASCII files.

Limitations .PAL files contain only a palette.

Viewing a .PAL file or converting a .PAL file to another image format is not allowed.

Comments The format of PAL files is described in Appendix H.

Related options -f Match image to specified palette
 -F False colour with specified palette

Examples Extract the palette from the GIF file
 madonna.gif:

alchemy madonna.gif -l

Convert the file image.tga to a GIF file,
matching the palette found in standard.pal:

alchemy image.tga -g -f standard.pal

HSI Raw

-r

HSI Raw files are used internally by Image Alchemy when converting between certain combinations of image formats. If you are interested in converting custom format images to be used with Image Alchemy we suggest using HSI Raw Files.

Syntax -r

Extension .raw

Creator Handmade Software Inc.

Used by Image Alchemy

Variations 8 bit paletted and 24 bit true colour,
uncompressed, not packed.

Comments This format is used internally as temporary files by Alchemy when doing certain image conversions; it can also be explicitly read and written. This format is described in Appendix F.

Examples Convert the file test.lbm to a raw file:

```
alchemy test.lbm -r
```

Convert the raw file, test.raw, to a 24 bit Targa file called output.tga:

```
alchemy -24 -a test.raw output.tga
```

IFF (Interchange File Format) files are used by Amiga computers for storing a number of types of data, including images, text, and music; ILBM (InterLeaved BitMap) is a type of IFF file used to store images.

Syntax -i

Extensions .lbm
 .iff
 .ilbm

Creator Commodore-Amiga Corp.

Used by Amiga
 Deluxe Paint

Variations Reads 1 through 8 bit, 24 bit, and HAM images

 Writes 1 through 8 bit and 24 bit images.

Limitations Dynamic Hi-Res images are not supported.

Does not write images in any of the Amiga specific display modes.

Comments If you're writing an ILBM file for use on an Amiga, you probably want to write either a paletted file with 32 colours or a 24 bit file. 24 bit ILBM files can then be converted to one of the Amiga specific display modes with various third-party utilities.

Example Convert the file input.pcx to an IFF/ILBM file called output.lbm with 32 colours:

```
alchemy -i -c32 input.pcx output.lbm
```

JPEG/JFIF

-j

JPEG is a new type of image file format that uses a lossy compression technique to achieve high compression ratios. See Appendix C, JPEG Compression, for more information.

Syntax -j[coding] quality

Parameters coding:
 Specify the type of entropy coding to perform.
 none: default Huffman coding
 h: optimum Huffman coding
 quality:
 1 through 100 (larger is higher quality)
 The default quality is 32.

Extension .jpg

Creator Joint Photographic Experts Group (JPEG)

Used by Lossy compression of photographic images.

Variations Gray-scale images are saved as single channel JPEG files; colour images are saved as three channel JPEG files.

Reads and writes baseline JPEG with CCIR-601 YCbCr colour space, interleaved components, Huffman coded.

Alchemy can read files with any component sub-sampling up to 4x4; it always writes 2h:1v 1h:1v 1h:1v.

Alchemy JPEG files comply with the industry standard 'JFIF' interchange format.

Limitations Can not read non-interleaved JPEG files.

JPEG files are always lossy, which means that the compressed image is not identical to the original image. At high quality factors (32 and above) this loss is generally so slight as to be barely noticeable. There is no quality factor which is guaranteed to be lossless.

Comments By default, Image Alchemy uses a fixed set of Huffman tables to compress an image. If the -j is immediately followed by an 'h', Alchemy will generate a set of custom tables optimized for the image and quality factor. This usually produces 5-20% better compression (depending on the image content and quality factor) but requires an additional pass over the image data, so it takes a little longer to compress (there's no effect on the decompression time).

Quality may vary between 1 and 100; the default is 32. The higher the number the higher the quality of the image and the lower the compression ratio. Quality factors below 10 will produce images with significant loss of quality.

JPEG files are based on the Joint Photographic Experts Group (JPEG) CD 10918-1 draft standard.

Since JPEG compression was designed for use with continuous tone images (such as those produced by a scanner or digitizer), poor results can be expected when compressing line drawings.

Related options -q Apply Smoothing when decompressing a JPEG image.

Because JPEG compression works on 8x8 pixel blocks there may be discontinuities at the edges of these blocks producing block artifacts. Smoothing attempts to reduce these artifacts. Smoothing is really only necessary at very low quality settings (less than 10); even then the effects of smoothing are not particularly significant.

Examples Convert the file photo.tga to a JPEG file called photo.jpg, using a high quality setting:

```
alchemy -j70 photo.tga
```

Convert the JPEG file, lores.jpg, to a PCX file using smoothing:

```
alchemy lores.jpg -p -q
```

Jovian VI

--J

Jovian VI files are created by the Jovian Logic video capture boards.

Syntax --J

Extensions .vi

Creator Jovian Logic Corp.

Used by Jovian Logic

Variations Reads 1, 4, 6, and 8 bit gray-scale images, 4 and 8 bit colour paletted images, and 16 and 24 bit true colour images.

Writes 8 bit gray-scale, 4 and 8 bit colour paletted images, and 16 and 24 bit true colour images.

Limitations Reads files with 6 and 8 bit palettes, always writes 6 bit palettes.

Gray-scale files are always 8 bit.

Comments When writing a VI file the palette always starts at 0, but will not necessarily be black (which is the way that Jovian VI files are written).

Example Convert the GIF file, test.gif, to a 16 colour VI file:

```
alchemy test.gif --J -c16
```

Macintosh PICT/PICT2

-m

PICT files were created by Apple Computer as a common format for Macintosh applications to use. Virtually every Macintosh application can use PICT files.

Syntax -m

Extensions .pict
 .pic

Creator Apple Computer, Inc.

Used by Macintosh computers

Variations Reads 1, 2, 4, 8, 16, and 32 bit PICT and PICT2 images

Writes 1, 2, 4, 8, and 32 bit PICT2 images.

Limitations Only pays attention to pixMaps in the image; attempts to skip everything else.

Comments Due to the enormous number of options allowed in PICT files, reading PICTs may not always work. See Appendix A, Answers to Frequently Asked Questions, for more information.

Example Convert the file input1.gif to a Mac PICT file called input1.pic:

```
alchemy -m input1.gif
```

--M

MTV files are used by the MTV RayTracer, a public domain ray tracer for Suns and other workstations.

Syntax --M

Extension	.mtv
-----------	------

Creator Mark T. VandeWettering

Used by MTV Raytracer

Variations 24 bit true colour.

Comments MTV is a public domain ray-tracer available free
of charge via anonymous ftp from
drizzle.cs.uoregon.edu or via floppy disk from
us.

Example Convert the file `spheres.img` to a MTV file:

```
alchemy spheres.img --M
```


The Pictor format was designed so that an image could be loaded into an IBM graphics adapter very quickly; it does this by almost exactly duplicating the organization of the graphics adapter memory. This makes the format hardware dependent.

Syntax -A type

Parameter type:
 0:320x200x4 CGA*
 1:320x200x16 PCjr/Tandy*
 2:640x200x2 CGA*
 3:640x200x16 EGA
 4:640x350x2 EGA
 5:640x350x4 EGA
 6:640x350x16 EGA
 7:720x348x2 Hercules*
 8:640x350x16 VGA
 9:320x200x16 EGA
 10:640x400x2 AT&T/Toshiba*
 11:320x200x256 VGA/MCGA
 12:640x480x16 VGA
 13:720x348x16 Hercules InColor*
 14:640x480x2 VGA/MCGA
 15:800x600x2 EGA/VGA
 16:800x600x16 EGA/VGA
 17:640x400x256 SVGA
 18:640x480x256 SVGA
 19:800x600x256 SVGA

 20:1024x768x2 SVGA
 21:1024x768x16 SVGA
 22:360x480x256 VGA
 23:1024x768x256 SVGA
 The default is 640x480x256 SVGA.
 *These modes are not yet supported.

Extension .pic
 .clp

Creator John Bridges

Used by PCPAINT
 GRASP

Variations There are variations for most IBM and third party graphics adapter display modes.

Limitations Only the EGA and VGA modes are supported at this time. Text modes are not supported.

Comments Some Pictor files do not contain palettes. For those files Alchemy will default to using a standard palette appropriate to the display mode the file was saved in. However, the image may not use the default palette; in that case you can read the palette from another file with the -F false colour option.

Related options -F False colour

Example Convert the file image.pcx to a Pictor file called image.pic, for 800x600x256 SVGA mode:

alchemy -A19 image.pcx

PCX

-p

PCX files are used extensively by MS-DOS machines. Originally created by ZSoft for use by their paint packages, PCX files can be read and written by almost all MS-DOS paint software and desktop publishing software.

A new variation of PCX file, DCX, is used by many MS-DOS fax boards.

PCX files suffer from two problems: the compression ratio is poor for 8 and 24 bit images and PCX files are often written out incorrectly; see the Limitations section below for details.

Syntax -p type

Parameter type:
 0:Standard PCX
 1:DCX
 The default is standard PCX.

Extension .pcx

Creator ZSoft Corporation

Used by PC Paint
 Publisher's Paintbrush
 Most paint and desktop publishing software can read and write PCX files.
 Fax board software uses the DCX variation of PCX.

Variations 1, 4, 8, and 24 bits per pixel for standard PCX files.
 1 bit per pixel for DCX files.

Limitations PCX format files are often written out incorrectly; Alchemy attempts to figure out what is wrong and make intelligent decisions (things Alchemy can deal with include PCX files without palettes, files missing the last line of image data, and files with illegal (and incorrect) combinations of bits per pixel and planes).

24 bit PCX files are very new and most other software can not read them. Therefore, unless you are sure that the software you are using can read a 24 bit PCX file, you probably want to use the -8 option to force Alchemy to write a paletted file when generating a PCX file.

DCX files are multiple page PCX images which are used by various manufacturers of fax boards and fax software. Alchemy only reads the first page of a DCX file. Alchemy always writes single page DCX files which are black and white.

Comments Because so many software packages can read and write PCX files we are especially interested in supporting as many variations as possible. If you have any PCX files which Alchemy does not read correctly please contact us.

Since DCX files are always 1 bit, black and white images, Alchemy assumes the use of -b -c2 -8 when writing the DCX variation of PCX.

ZSoft recently changed some of the information in the header of PCX files so they now include image resolution. Some fax board software makes use of this information when transmitting PCX or DCX files as faxes. See the example section below for an example of how to specify image resolution when writing a PCX file.

Related options -D Specify image resolution.

Example Convert the GIF file, lush.gif, to a PCX file:

```
alchemy lush.gif -p
```

Convert the scanned image, page1.tif, to a DCX file:

```
alchemy -p1 page1.tif
```

Convert the scanned image, page2.tif, to a DCX file, specifying an image resolution of 200x100 (a common resolution for fax images):

```
alchemy -p1 page2.tif -D 200 100
```

The Portable BitMap format was developed by Jef Poskanzer to allow the transferring of black and white image files between different workstations. The PBM format has grown to include black and white, gray-scale, and true colour images, a large set of programs to convert various other image formats to and from PBM, and a set of image manipulation tools. The PBM tools are available free of charge (unfortunately, due to their memory requirements, many are not useable on MS-DOS machines).

Syntax -k

Extensions .pnm Portable aNyMap (Any of those below)
 .pbm Portable BitMap (Black and white)
 .pgm Portable GrayMap (Gray-scale)
 .ppm Portable PixelMap (True colour)

Creator Jef Poskanzer

Used by Portable BitMap Package
 Various workstation graphic programs

Variations Reads and writes 1, 8, and 24 bit RAWBITS
 (binary) images.

To write out a PBM file use -b -c2.
To write out a PGM file use -b -c256.
To write out a PPM file use -24.

Limitations When writing a PBM file Alchemy always uses the
 .pnm extension (the extension should be changed
 based on the type of file being written).

Comments The PBM package is a set of image manipulation
 tools which run on various workstations. The
 software is available for free via anonymous ftp
 from expo.lcs.mit.edu as contrib/pbmplus.tar.Z,
 ftp.ee.lbl.gov as pbmplus.tar.Z, or via floppy
 disk from us.

Examples Convert the file sun.im32 to a PBM file:

alchemy sun.im32 -k -b -c2

Q0

--q

The Q0 format is apparently commonly used by various Japanese scanning, painting, and viewing software to store 24 bit images. Handmade Software has no information other than a basic description of the format and some sample images; if you have further information on the Q0 format please contact us.

Syntax --q

Extensions .q0 For pixel data
 .rgb For pixel data
 .fal For image header information

Creator Unknown

Used by Various Japanese image processing software.

Variations 24 bits per pixel

Comments Q0 files are actually two files, one with the extension .rgb or .q0 and the other with the extension .fal. The .rgb or .q0 file contains the actual image data and the .fal file contains the header information (primarily the height and width of the image). You specify the name of the .rgb or .q0 file and Alchemy automatically generates the name of the .fal file.

When writing a Q0 file Alchemy will overwrite, without warning, existing .fal files.

Example Convert the file dogcow.gif to a Q0 file:

```
alchemy dogcow.gif --q
```

QDV

--D

The QDV format is used by Giffer, a Macintosh program which displays and converts image files.

Syntax --D

Extension .qdv

Creator Steve Blackstock

Used by Giffer

Variations QDV files are always 8 bits per pixel.

Comments Giffer is a great shareware (Beerwaretm, actually) program for the Macintosh that converts between various image file formats and allows viewing of graphics files.

Since QDV files are always paletted, Alchemy assumes the use of the -8 option when writing QDV files.

Example Convert the file input.tga to a qdv file:

alchemy input.tga --D

QRT

--T

QRT files are generated by the QRT Ray Tracer, a public domain ray-tracer for Amiga, Macintosh, and IBM PC computers.

Syntax --T

Extension .raw

Creator Steve Korn

Used by QRT Ray Tracer

Variations 24 bits per pixel

Comments Since QRT files are always true colour, Alchemy assumes use of the -24 option.

Example Convert the file spheres.gif to a QRT file called spheres.raw:

alchemy --T spheres.gif

Silicon Graphics Image

-n

Silicon Graphics Image files are used by Silicon Graphics workstations.

Syntax -n

Extension .sgi

Creator Silicon Graphics, Inc.

Used by Silicon Graphics workstations.

Variations Reads 1, 8 (gray-scale), and 24 bit verbatim (uncompressed) and RLE files.

Writes 1, 8 (gray-scale), and 24 bit verbatim files.

Limitations Only uncompressed (verbatim) files can be written.

Comments Only gray-scale images may be 8 bit files. Alchemy will automatically switch to 24 bit mode when writing a colour image.

Example Convert the Sun raster file sun.im8 to a SGI file called sgiout:

alchemy -n sun.im8 sgiout -.

Stork -K

Stork files are CMYK images used by Stork's colour proofing machines.

Syntax -K compressionType

Parameter compressionType:
0:None
1:Run length coded
The default is none.

Extensions .idx Index file
.pre Image data
.tab Colour lookup table

Creator Stork Colorproofing B.V.

Used by Stork Colorproofing machines

Variations Reads and writes 32 KCMY, 32 KCMY RLC, 16 CLU, and 16 CLU RLC images (type 100, 101, 300, and 301, respectively).

Limitations Alchemy can't write paletted files with more than 256 colours.

When reading paletted files with more than 256 colours they are treated as true colour.

Comments Stork images are stored in two or three files (depending on whether or not there's a colour lookup table associated with the image). The filename given to Alchemy should be the name of the data file (normally with a suffix of .pre); Alchemy will generate the names of the other files by stripping the extension and appending .idx for the index file and .tab for the colour lookup table (if any). Alchemy will overwrite existing .idx and .tab files without warning when creating Stork files.

Related options -C Undercolour Removal File

Example Convert the file image.tga to an uncompressed Stork image called image.pre and image.idx, using the undercolour removal file sample.ucr:

```
alchemy -K -Csample.ucr image.tga
```

Sun Raster

-s

Sun Raster files are used by Sun Microsystems workstations.

Syntax -s

Extensions .rast

.ras

.im

.im1

.im8

.im24

.im32

Creator Sun Microsystems, Inc.

Used by Sun workstations

Variations Reads 1, 8, 24, and 32 bit Standard, BGR, RGB, and Byte Encoded (RLE) files.

Writes 1, 8, and 32 bit Standard files.

Limitations For a short time a version of the PBM toolkit wrote Sun Raster files which had the wrong RGB order. Unfortunately there seem to be many images which were generated with this incorrect RGB order. Please be aware that Alchemy's Sun Raster reading and writing capability has been extensively tested and compared to various other tools; the RGB order is correct.

Comments Sun Raster files are normally not compressed and so take up lots of disk space.

There is no standard extension for Sun Raster files; the extensions that Alchemy uses seem to be the most common.

Example Convert the SGI file sgiout to a sun raster file called sun.im8:

```
alchemy -s sgiout sun.im8
```

TIFF is designed to be a universal raster image format; it's very popular with desktop publishing packages.

Syntax -t compressionType

Parameter compressionType:
 0:None
 1:LZW
 2:PackBits
 3:Group III Fax
 4:Group IV Fax
 5:CCITT RLE
The default is LZW Compression.

Extensions .tiff
 .tif

Creator Aldus Corp.
 Microsoft Corp.

Used by Various desktop publishing and scanning software.

Variations Reads TIFF class B, G, R, and most class P files.

Reads 1, 4, 8, 12, 24, and 32 bit images (ignoring the alpha channel for 32 bit images).

Input compression types supported are raw, LZW, PackBits, Group III fax, Group IV fax, CCITT RLE (byte and word aligned), NeXT, Thunderscan, PICIO, and SGI RLE.

Writes class B, G, P, and R files, depending on the input file and options specified.

Writes 1, 4, 8, and 24 bit images.

Limitations Class P TIFF files can only be read if they have 1, 4, or 8 bits per pixel.

Comments 1,4, and 8 bit output files are paletted unless the palette is all gray, in which case the output is a gray-scale file.

If you have TIFF files with 3, 5, 6, or 7 bits per pixel please contact us.

Example Convert the file input.gif to an uncompressed
gray-scale TIFF file called output.tif:

```
alchemy -t0 -b input.gif output.tif
```

Targa

-a

Targa files were created to support the line of Targa graphics cards. The Targa format is popular with scanners and high end paint packages.

Syntax -a outputType

Parameter outputType:
 0:uncompressed
 1:Run Length Coded
 10:uncompressed, no footer
 11:Run Length Coded, no footer
The default is 0 (uncompressed).

Extension .tga

Creator Truevision, Inc.

Used by Various scanning and paint software.

Variations Reads 8, 15, 16, 24, and 32 bit images, ignoring the alpha channel for 32 bit images.

Writes 8, 15, 16, 24, and 32 bit images, writing an empty alpha channel for 32 bit images.

Comments 15 and 16 bit output are actually the same except for one field in the header.

Targa files allow a footer containing additional information such as aspect ratio. However some software is unable to read Targa files which have a footer, so Alchemy allows all valid combinations to be written. The most common variant for software to be able to read is 24 bit uncompressed (specify -a0 and -24).

Example Convert the file input.tif to an uncompressed 24 bit Targa file:

```
alchemy input.tif -a -24
```

Convert the file input.tif to an uncompressed 15 bit Targa file called output.tga with no footer:

```
alchemy -a10 -15 input.tif output.tga
```

The Utah Raster Toolkit is a set of public domain utilities for manipulating and converting images for various workstations. The source code is freely available (see below).

Syntax --u

Extension .rle

Creator The University of Utah
 The University of Michigan

Used by Utah RLE toolkit

Variations Reads and writes 1 and 3 channel 8 bits per pixel files; the Alpha Channel is ignored during reading.

Limitations While reading, files which are 1 channel and have either no colour map or a single channel colour map are assumed to be gray-scale images. The colour map, if present, will be used as a gamma correction table.

Files which are 1 channel and have a 3 channel colour map are assumed to be paletted colour files.

Files which are 3 channel are assumed to be true colour.

When writing RLE files Alchemy will generate a 1 channel file with a 3 channel colour map for paletted images and a 3 channel file with no colour map for true colour images.

Comments Alchemy does not write optimally compressed images.

The Utah Raster Toolkit is available free of charge as pub/urt-3.0.tar.Z via anonymous ftp from cs.utah.edu, weedeater.math.yale.edu, or freebie.engin.umich.edu or via floppy disk from us.

Example Convert the PBM file, image.ppm, to a Utah RLE file:

```
alchemy image.ppm --U
```


Vivid --I

Vivid is a shareware ray-tracer for MS-DOS computers

Syntax --I (upper case i)

Extension .img

Creator Steven B. Coy

Used by Vivid Ray Tracer

Variations Reads and writes 24 bit RLE files.

Comments The Vivid Ray Tracer is a shareware program for PCs and is available from
Stephen Coy
15205 NE 13th Pl., #2904
Bellevue, WA 98007

Example Convert the file spheres.qrt to a Vivid file:

alchemy spheres.qrt --I

Windows Bitmap (BMP)

-w

Windows BMP files are used by Microsoft Windows.

Syntax -w compressionType

Parameter compressionType:
 0:None
 1:RLE
 The default is none.

Extension .bmp

Creator Microsoft Corp.

Used by Microsoft Windows

Variations Reads 1, 4, 8, and 24 bit RGB (raw), RLE4, and
 RLE8 files.

 Writes 1, 4, 8, and 24 bit RGB (raw), RLE4, and
 RLE8 files.

Limitations Several of the programs which read and write RLE
 files do not do so correctly; we do not
 recommend writing RLE files unless you have
 verified that they work with your intended
 application.

Comments Microsoft supplied Windows 3.0 utilities cannot
 read RLE4 or RLE8 files.

 If you are converting an image to use as
 wallpaper on a 16 colour Windows display you
 will want to match the palette of the output
 image to one of the existing 16 colour BMP
 images supplied with Windows (chess.bmp, for
 example). If you do not do this the wallpaper
 will not be loaded correctly. See the example
 section below for an example.

Related options -f Match to existing palette

Examples Convert the image test.gif to a 16 colour
 Windows BMP file to be used as wallpaper (the
 file chess.bmp is supplied with Windows; this
 example assumes that it is in the current
 directory):

 alchemy test.gif -f chess.bmp -w

WordPerfect Graphic File

-W

WordPerfect files are images which can be imported into WordPerfect and various other word processors and desktop publishing programs.

Syntax -W

Extension .wpg

Creator WordPerfect Corp.

Used by WordPerfect

Variations 1 through 8 bits per pixel are supported.

Limitations Alchemy does not write optimally compressed files.

Files with default palettes are not read correctly.

Comments In addition to raster images WordPerfect files may contain vectors and text information. Such information is lost when reading WordPerfect files.

Example Convert the image, newpict.pcx, to a black and white WPG file:

```
alchemy newpict.pcx -b -c2 -W
```

XBM

--b

XBM files are used by the X Windowing System. XBM files are C source code files which can be read and written by various X utilities and are designed to be included in C source code for use as icons and other bit-mapped graphic images.

Syntax --b

Extensions .xbm
 .bm

Creator MIT

Used by The X Windowing system

Variations 1 bit per pixel

Limitations Because .xbm files are actually C source code files there can be many variations of .xbm files. Since adding a C preprocessor to Alchemy to handle all the theoretically allowable .xbm files is impractical we have instead designed Alchemy to interchange .xbm files with the PBM utilities and the X supplied utilities, and to read the sample .xbm files from Sun Microsystems. If you run across any .xbm files which Alchemy can not read please contact us.

The hotspot field is ignored when reading .xbm files.

Comments Most of the X supplied utilities (bitmap, for example) are designed to edit small .xbm images.

Example Convert the file picture.im32 to an XBM file using high quality scaling and preserving the aspect ratio:

```
alchemy --b -Xb64 -+ picture.im32
```

XWD

--w

XWD is the file format used by xwd, the X window dumping utility.

Syntax --w

Extension .xwd

Creator MIT

Used by The X Windowing System

Variations 1, 8, and 24 bits per pixel.

Limitations XY format files are not supported.

Example Convert the XBM file, icon.xbm, to an XWD file:

```
alchemy icon.xbm --w
```

General Options

Introduction General options are options which do not affect the conversion of the image. They control such things as the overwriting of existing files and the way that memory is used.

Conserve Memory

-\$

Purpose Use as little memory as possible when converting images.

Syntax -\$ (dollar sign)

Comments Normally Alchemy tries to work on chunks of the image several lines long to improve performance. Use of the -\$ option will cause it to use the smallest size chunks possible for the conversion being performed. On MS-DOS based systems this will usually allow conversion of larger images than would otherwise be possible. On Unix systems this may reduce paging when converting very large images.

Example Convert the image giant.tga to a 16 colour TIFF file conserving memory:

```
alchemy giant.tga -$ -t -c16
```

Display Image Stats -X

Purpose	Display image statistics.
---------	---------------------------

Syntax -X

Comments	Displays image type, size, number of colours, aspect ratio, resolution, and compression ratio.
----------	--

Limitations	Can not be combined with other options.
-------------	---

Example Find out about the image called image.tga:

```
alchemy -x image.tga
```


Help -h

Purpose Give you information on how to use Image Alchemy.

Syntax -h option

Parameter option
0: general help
1: general options
2: output formats a-k
3: output formats l-z
4: colour options
5: scaling options
6: display options (MS-DOS Only)
Default is 0, general help

Comments The help information given by this command is only a summary.

Limitations The help option cannot be combined with any other options.

Related options -? support and update information

Example Get help on the colour options:

alchemy -h4

Overwrite -o

Purpose Force Alchemy to overwrite existing files on the disk.

Syntax -o

Comments Image Alchemy will not overwrite an existing file unless the -o option is specified.

Limitations The input file name and the output file name can not be the same.

Example Convert the file input.tga to a GIF file called output.gif, overwriting the existing file called output.gif:

```
alchemy input.tga -g output.gif -o
```

Program information -?

Purpose Give you information on how to get support for Image Alchemy or inquire about update information.

Syntax -?

Comments Sun-3 and Sun-4 users have to escape the question mark with a back-slash (instead of -? use -\?). This is because the UNIX shell will attempt to perform wildcard expansion on the question mark.

Limitations The information option cannot be combined with any other options.

Related options -h help with commands

Example Get support information:

alchemy -?

Quiet -Q

Purpose Suppress all status messages (but not error messages).

Syntax -Q

Comments This is useful when running Alchemy in the background on Unix systems or in batch files on MS-DOS systems (and you don't want the output of Alchemy scrolling important messages off of the screen).

Limitations There is no way to suppress error messages.

Example Convert the file dummy.gif to a PCX file but don't report any status messages:

```
alchemy -Q dummy.gif -p
```

Wildcard

--

Purpose Allow the conversion of multiple files with a single execution of Alchemy.

Syntax -- (dash)

Comments The wildcard option allows you to specify multiple file names and file names which include wild card characters. Alchemy will perform the same conversion for each input file name that it finds.

On MS-DOS systems the use of the wildcard option (--) is not required if the first file name specified includes a wildcard character (* or ?); however to reduce confusion it is still recommended.

Limitations The wildcard option (--) must be specified before any file names.

If you are using the wildcard option you may not specify an output file name; the file names are automatically generated by substituting an appropriate extension to the input file names. If you do specify an output file name it will be misinterpreted as another input file. An output path name may still be specified.

Any error will terminate the execution of Alchemy; any images which appear in the filename list after the one causing the error will not be processed. This includes attempting to overwrite an already existing file without specifying the -o option.

Alchemy does not intelligently retain information between files. For example, if you are matching a group of files to an existing palette, the inverse palette generation step only needs to be performed once, but it is in fact done for each file. This only affects the speed of conversions, not the quality.

Examples Convert all the GIF files in the current directory to JPEG files:

```
alchemy -- *.gif -j
```

Convert all the TIFF files in the directory
\\tiff to PCX files in the directory
\\images\\output:

```
alchemy -- \\tiff\\*.tif -p \\images\\output
```

Convert the files madonna.gif, bay4.gif,
everest.tga, and basil.tif to JPEG files,
overwriting any existing files:

```
alchemy -- -o madonna.gif bay4.gif -j  
everest.tga basil.tif
```

Convert the files test1.tif, test2.tif, and
new*.gif to ILBM files, matching them to the
palette from the file output.pal:

```
alchemy -- test1.tif test2.tif new*.gif -  
f output.pal -i
```

Colour and Palette Options

Introduction Colour and Palette options are options which affect the appearance of the output image. They control such things as the number of colours in the output image and the dithering techniques used.

Black and White

-b

Purpose Convert the image to black and white.

Syntax -b

Comments In conjunction with the -b option the -c option is used to specify the number of shades of gray in the image. The default is 256 shades of gray when converting from a true colour image. When converting from a paletted image the number of shades of gray defaults to the number of colours in the original image.

The shades of gray are uniformly distributed from 0 to 255.

When converting from true colour the image will be changed to a paletted image unless the -24 option is used.

Related options -8 Paletted output

-24 True colour output

-c Specify number of colours

Examples Convert the file sample.jpg into a 256 shades of gray raw file:

```
alchemy sample.jpg -b -r
```

Convert the file madonna.jpg into a 4 shades of gray gif file called gray.gif:

```
alchemy -b -c4 -g madonna.gif gray.gif
```

Colours

-c

Purpose Specify the number of colours for the output file.

Syntax -c colours [reserveColours]

Parameters colours

Specifies the number of colours in the output image. May be between 2 and 256.

reserveColours

Specifies the number of colours to reserve in the output image. May be between 0 and 255.

Comments If the input file has a larger number of colours than specified for the output file, the image will be quantized using Heckbert's median cut algorithm and dithered. For further information on Heckbert's median cut algorithm see Appendix B, Colour and Dithering.

The number of colours to reserve is an optional parameter. If it is present it causes the specified number of colours to be reserved from the beginning of the palette. The output image will not contain any of those colour indices. This can be useful if you have menus or other information you wish to display at the same time as the images and they use colours at the beginning of the palette. The menu colours will then not interfere with the image. The first indices are set to black unless 16 is specified, in which case they are set to the standard VGA colour palette.

Limitations Specifying the number of colours only has an effect if you are writing a paletted file (using the -8 option) or if the output file type is always paletted.

Converting an image with a large number of colours to a small number of colours (less than 8) will usually give poor results.

The reserved colours will be set to black unless 16 colours are reserved. In that case they will be set to the standard VGA colours.

Related options -8 Convert to paletted image

-d Specify dither type

-u Use uniform palette

Examples Convert the image colours.gif into a 16 colour
PCX file called colour16.pcx

```
alchemy colours.gif -p -c16 colour16.pcx
```

Convert the image colours.tga into a 256 colour
GIF file called output.gif, reserving the first
16 colours.

```
alchemy colours.tga -g -c256 16 output.gif
```

Dither -d

Purpose Specifies the type of dithering to apply to the image.

Syntax -d[s] ditherType [perturbation]

Parameters If the -d is immediately followed by an 's', then a serpentine raster is used.

ditherType can be:

0:None

1:Floyd-Steinberg

2:Stucki

3:Jarvis, Judice, & Ninke

The default is Floyd-Steinberg.

perturbation

0 through 127

The default is 0.

Comments Dithering is used to reduce colour banding in an image caused by the palette not having a perfect match for every colour in the image.

Floyd-Steinberg is the fastest supported dithering algorithm and usually produces the best results. Stucki and Jarvis, Judice, & Ninke both tend to cause an image to appear more grainy on low resolution output devices (such as CRTs). However they produce better results than Floyd-Steinberg on high-resolution, low colour output devices such as laser printers.

The -d option only has an effect if the number of colours is being reduced or the image is being re-mapped to a new palette.

Specifying a perturbation adds noise to the threshold used for dithering, which can help break up visible patterns introduced by dithering. The parameter specifies the magnitude of the noise.

Using a serpentine raster can also help to reduce visible patterns introduced by dithering.

For further information on dithering see Appendix B, Colour and Dithering.

Examples Convert the 256 colour file image.gif to a 16 colour PCX file using a uniform palette and no dithering:

```
alchemy image.gif -p -c16 -d0 -u
```

Convert the true colour image sample.jpg into a 256 colour GIF file called sample.gif, using Stucki dithering:

```
alchemy -g -d2 sample.jpg
```

Convert the 256 colour image sample.gif into a one bit black and white PCL file called sample.pcl, using Jarvis, Judice, and Ninke dithering, a serpentine raster, and a little noise:

```
alchemy -P -b -c2 -ds3 20 sample.gif
```

EGA Palette

-E

Purpose Optimize the image quality for display on an EGA board and monitor.

Syntax -E

Comments If you are converting images to display on an EGA board and monitor this option will optimize the image quality.

This option reduces the palette resolution to two bits and automatically specifies the following: -8 -c16 -z0 2 0.

Limitations The number of colours in an EGA palette must be less than or equal to 16; the number of colours defaults to 16 but can be reduced by using the -c option.

Related options -c specify number of colours

Example Convert the image dave1.tga into dave1.pcx, a PCX file with a palette optimized for EGA use:

```
alchemy -E -p dave1.tga
```

False Colour

-F

Purpose False colour an image using the palette from a file. The input image will be changed to use the palette found in the specified filename but no attempt at picking the best match will be done.

Syntax -F filename

Parameter filename
Any valid image type which contains a palette

Comments This feature can be used to add false colour to monochrome images. The output file is not dithered. This is only applicable to paletted input files.

Limitations Can not be combined with spiff -S or match to palette -f.

Example False colour the file scan.gif using the palette from the file colorful.pcx, creating the GIF file new.gif:

```
alchemy -F colorful.pcx -g scan.gif new.gif
```

Match Palette -f

Purpose Match the output to a palette read from a file.
The input image will be re-mapped to use the palette found in the specified file.

Syntax -f filename

Parameter filename
Any valid image type which contains a palette

Comments Using the -f option will cause the output image to be dithered (unless you specify no dithering by using the -d0 option).

The -f option can be useful if you are combining several images into a collage or want to match an image to a pre-existing palette. You can also create a custom palette from scratch by using a text editor and creating a .PAL file.

Limitations Cannot be combined with spiff -S or false colour -F.

The number of colours in the final image will be equal to the number of colours in the palette being read in.

The specified file must contain a palette (i.e. cannot be true colour).

Related options -l Generate palette file
-F False colour
-d Dither

Examples Convert the image bigimage.tif to a pcx file using the palette from the file standard.pal:

```
alchemy bigimage.tif -p -f standard.pal
```

Convert the image colour.gif to a gif file called colour2.gif using the palette from the file newpal.gif:

```
alchemy -fnewpal.gif -g colour.gif colour2
```


Palette -8

Purpose Force the output image to be paletted.

Syntax -8

Comments This option is -8 because paletted images are typically 8 bits per pixel.

Alchemy defaults to the -8 option if the input file is paletted or gray-scale.

Some file formats require files to be paletted; for those formats the -8 option is assumed. Some file formats do not have a paletted variation; in those cases the -8 option will be ignored if specified. Some file formats only allow gray-scale files to be 8 bit; in those cases Alchemy will ignore the -8 option if the image being written is not gray-scale.

The actual number of bits per pixel is determined by the -c option (below).

If the input file is true colour the output file will be quantized and dithered (see the -c and -d options below).

Related options -15 True colour output
-16 True colour output
-24 True colour output
-32 True colour output
-c specify number of colours in image
-d dither

Examples Convert the JPEG file bigimage.jpg into a paletted TIFF file with 256 colours:

```
alchemy -8 -t bigimage.jpg
```

Convert the Targa file madonna.tga to a 16 colour PCX file (note that the -8 option is implied by the use of the -c16 option):

```
alchemy -c16 -p madonna.tga
```

Purpose Control how the palette is generated. These options only have an effect if the palette is being generated by Alchemy using Heckbert's median cut algorithm.

Syntax -z sortType [selectionType [swapType]]

Parameters Sort the image palette. SortType can be:

- 0:None
- 1:popularity
- 2:luminance (white to black)
- 3:rgb
- 4:luminance (black to white)

The default is None.

SelectionType can be:

- 0:mean
- 1:median
- 2:corner

The default is mean.

See Appendix B, Colour and Dithering, for an explanation of these choices.

SwapType can be:

- 0:None
- 1:IBM (colour 0 is black, 7 is white)
- 2:Macintosh (colour 0 is white, 255 is black)
- 3:Sun (colour 0 is white, 1 is black)

The default is based on the file type being written out (IBM for GIF, Macintosh for Mac PICT, Sun for Sun Raster, and None for all others).

Comments The most common use for this option is to sort the palette; in this case only a single parameter is needed.

This option only affects palettes that are generated by Alchemy. To sort an existing palette you can save the image as a true colour file (such as HSI Raw, by using the -24 -r options) and then convert that back to a paletted file, specifying the desired sort type. In most cases this will not change the image (other than the palette order); however if the palette had entries representing colours that are nearly identical then the image may be modified.

Limitations Note that it is not possible to specify a swapType without first specifying both a sortType and a selectionType. See Appendix B, Colour and Dithering, for more information.

Example Convert the file input.tga to a gif file called output.gif sorting the colours by luminance, using the mean of the Heckbert box for the colour, and moving the colours around so that the lightest colour is colour 0 and the darkest colour is colour 1.

```
alchemy input.tga -g output.gif -z4 0 3
```

Spiff -S

Purpose Enhance the image contrast by stretching the pixel colour values to the full 0 to 255 range.

Syntax -S

Comments This command can be used if the image you are converting is shifted in brightness or squished in contrast. This can happen if you scan or digitize a very dark or very bright image.

Limitations The -S option can not be used at the same time as the -b option when converting from a true colour image. A work around is to do the operation in two steps, converting it to black and white first and then spiffing the resulting image.

Using the spiff option at the same time as the match palette, -f, or false colour, -F, options is not allowed. This is because the spiff option would be performed before the palette is changed, which would nullify the effects. A work around is to do the matching or false colouring first, and then spiff the resultant image.

Related options -b Black and White
-f Match palette
-F False colour image

Example Convert the file gloomy.pcx into a PCX file called better.pcx:

```
alchemy gloomy.pcx -S -p better.pcx
```

True Colour (15 bits) -15

Purpose Force the output image to be true colour, 15 bits (5 bits per component).

Syntax -15

Comments See the True Colour (24 bits) section, below.

Related options -8 Paletted output
-16 True colour output
-24 True colour output
-32 True colour output

Example Convert the GIF file test.gif into an uncompressed, true colour 15 bit Targa file called test.tga:

alchemy test.gif -a0 -15

True Colour (16 bits) -16

Purpose Force the output image to be true colour, 16 bits (5 bits each for red and blue, 6 for green).

Syntax -16

Comments See the True Colour (24 bits) section, below.

Related options -8 Paletted output
-15 True colour output
-24 True colour output
-32 True colour output

Example Convert the GIF file test.gif into an uncompressed, true colour 16 bit Targa file called test.tga:

alchemy test.gif -a0 -16

True Colour (24 bits)

-24

Purpose Force the output image to be true colour (not paletted).

Syntax -24

Comments This option is -24 because true colour images are typically 24 bits per pixel.

Some file formats require files to be true colour; for those formats the -24 option is assumed. Some file formats only have a paletted variation; in those cases the -24 option will be ignored if specified.

The file formats which may be either true colour or paletted default to true colour if the input file is true colour.

Certain file formats may only be paletted if the images are gray-scale, in those cases Alchemy will automatically switch to true colour if the output image is colour.

Converting a paletted image to true colour will not improve its quality or change its appearance. The primary use of this option is to force an image to be true colour when converting to a format which allows either paletted or true colour, but where the paletted variation is not well supported (like the Targa image format).

Related options -8 Paletted output

Example Convert the GIF file test.gif into an uncompressed, true colour Targa file called test.tga:

```
alchemy test.gif -a0 -24
```

True Colour (32 bits) -32

Purpose Force the output image to be true colour, 32 bits (8 bits per component, 8 bits for alpha).

Syntax -32

Comments See the True Colour (24 bits) section, above.

Related options -8 Paletted output
-15 True colour output
-16 True colour output
-24 True colour output

Example Convert the GIF file test.gif into an uncompressed, true colour 32 bit Targa file called test.tga (the alpha channel will be empty):

```
alchemy test.gif -a0 -32
```


Undercolour Removal

-C

Purpose Control the undercolour removal process, colour correction, and density correction for output formats which use the CMYK colour space.

Syntax -C filename

Parameter filename
 The name of the file which contains the undercolour removal information

Comments The undercolour removal portion of the file is compatible with the format used by Stork Colorproofing B.V. The format of this file is described in Appendix G, Undercolour Removal Files.

Sample undercolour removal files can be found in the samples directory on the Alchemy distribution disk or tape.

Example Convert the file image.tga to an HP RTL file called image.rtl using the undercolour removal file sample.ucr:

```
alchemy image.tga --r4 -Csample.ucr
```

Uniform Palette

-u

Purpose Use a Uniform Palette.

Syntax -u

Comments Instead of using the Heckbert median cut algorithm to generate a custom palette for the image, use a palette with entries which are evenly distributed in the RGB colour cube.

The advantage of using a uniform palette is that it's faster than generating a custom palette. However, this is at the expense of image quality since the palette isn't generated based on image content.

When just viewing a true colour image on a paletted display a uniform palette is used.

The -c option can be used in conjunction with -u to specify the size of the uniform palette; in that case Alchemy will generate a palette with not more than the specified number of colours (but not less than 8).

Limitations The palette size will not necessarily match the specified size, as the actual size must be the product of three integers. Alchemy picks integers that roughly correspond to the sensitivity of the human eye to red, green, and blue (30%, 59%, and 11%).

Related options -c Specify number of colours
-d Dither type

Examples Convert the file many.tga to a gif file using a 256 colour uniform palette:

```
alchemy many.tga -g -u
```

Convert the file many.tga to a gif file with up to 128 colours in a uniform palette:

```
alchemy many.tga -g -u -c128
```

Scaling Options

Introduction These options are all related to image scaling.
Note that the -D option does not actually change the size of the image, it specifies an aspect ratio or image resolution to be placed in the header of the output image.

Scale Image in Horizontal Direction -X

Purpose Scale the horizontal dimension of the image to the specified number of pixels.

Syntax -X[scaleType] pixels

Parameters scaleType
 The type of scaling to use:
 a: nearest neighbor
 b: averaging/linear interpolation
 c: lanczos2
 d: lanczos3
 scaleType is optional; the default is nearest neighbor. The higher the scale type the higher the quality (and the longer the processing time).

 pixels
 The number of pixels in the output image in the horizontal dimension.

Comments Nearest neighbor type scaling is faster than the other types but introduces aliasing (which reduces image quality). The highest quality scaling supported is lanczos3, but it takes much longer than averaging/linear interpolation and generally doesn't produce significantly better results.

Limitations All of the scale types other than nearest neighbor give much better results than nearest neighbor scaling, but they are slower and require a new palette to be generated for paletted output files (you can force alchemy to use the original palette by using the -f option and specifying the original image as the palette file).

Related options -Y Scale in vertical dimension
 -+ Preserve aspect ratio

Examples Scale the input image, test.gif, to 640 by 480 using good quality scaling, calling the output file test2.gif:

```
alchemy test.gif -Xb640 -Yb480 -g test2.gif
```

Scale the input image, big.tga, using fast scaling to an image which is 320 pixels across and the same aspect ratio as the input image, calling the output file out.tga:

```
alchemy big.tga -X320 -+ -a out
```

Scale the input image, oddsize.gif, using the highest quality scaling, to an image which is no larger than 640x480, but has the same aspect ratio as the original image, calling the output image new.gif:

```
alchemy oddsize.gif -Yd480 -Xd640 -+  
new.gif -g
```

Do the same thing as the previous example, but retain the same palette:

```
alchemy oddsize.gif -Yd480 -Xd640 -+  
new.gif -g -f oddsize.gif
```

Scale Image in Vertical Direction -Y

Purpose Scale the vertical dimension of the image to the specified number of pixels.

Syntax -Y[scaleType] pixels

Parameters ScaleType
The type of scaling to use:
a: nearest neighbor
b: averaging/linear interpolation
c: lanczos2
d: lanczos3
The default is nearest neighbor.

pixels
The number of pixels in the vertical dimension.

Comments Nearest neighbor type scaling is faster than the other types but introduces aliasing. The highest quality scaling supported is lanczos3, but it takes much longer than averaging/linear interpolation and generally doesn't produce significantly better results.

Limitations All of the scale types other than nearest neighbor give much better results than nearest neighbor scaling, but they are slower and require a new palette to be generated for paletted output files (you can force alchemy to use the original palette by using the -f option and specifying the original file name).

Related options -X Scale in vertical dimension
-+ Preserve aspect ratio

Examples See the -X option, Scale Image in Horizontal Direction, for examples.

Specify Image Aspect Ratio

-D

Purpose Specify aspect ratio for the output image.

Syntax -D aspectRatio

Parameter aspectRatio
The percentage of the width of a pixel to its height.

Comments This option does not actually change the aspect ratio of the image, it just adds the aspect ratio value to the output file. This is important when trying to export the image to software which expects this information.

The aspect ratio of an image is the ratio of the width of a single pixel to the height of a single pixel. (So to specify an aspect ratio of 5:6 use -D 83, since $(5/6)*100$ is 83).

Alchemy attempts to preserve the aspect ratio value when converting images whenever one is found in the input image, but since so few file formats have aspect ratio information this hardly ever happens.

This option can also be used when displaying an image on an IBM PC.

Limitations It is not possible to specify both an aspect ratio and a dots per inch value for an image. This is because specifying a dots per inch value automatically implies an aspect ratio.

Many file types do not have an aspect ratio value, so specifying one will have no effect.

Related options -D Specify resolution

Examples You are converting a 640x350 IBM EGA PCX image called ega.pcx (which has an aspect ratio of 35:48) to a TIFF image and you want the TIFF image to have the correct aspect ratio value (so that an intelligent TIFF reader will correctly interpret the image) (Note that the value of 73 is $(35/48)*100$):

```
alchemy ega.pcx -D 73 -t
```


The resulting image will still be 640x350, but the TIFF file now contains the information that the pixels are not square (and in fact are 35:48).

If you had instead wanted to convert the image to a 640 by 480 image (with square pixels) you could have used:

```
alchemy ega.pcx -Y480 -D100 -t
```

The -D option isn't really needed here, since any software reading the TIFF file will assume that if there is no aspect ratio specified the pixels are square.

Specify Image Resolution

-D

Purpose Specify image resolution in dots per inch for the output image.

Syntax -D dotsPerInchX dotsPerInchY

Parameters dotsPerInchX
The resolution of the image in the X direction in dots per inch.

dotsPerInchY
The resolution of the image in the Y direction in dots per inch.

Comments You must specify both dotsPerInchX and dotsPerInchY, even if they are the same.

This command does not actually change the resolution of the image, it just adds the resolution fields to the output image. This is important when trying to import the image into software which expects this information. For example, Microsoft Word is much more likely to give the expected results when importing a TIFF image for printing on a laser printer if the image has a resolution of 300 dpi.

Reasonable values to use for dotsPerInch include 72 (the resolution of a 13 inch monitor displaying 640x480) and 300 (the resolution of a laser printer).

Alchemy will preserve this information when converting files whenever possible.

Limitations It is not possible to specify both an aspect ratio and a dots per inch value for an image. This is because specifying a dots per inch value automatically implies an aspect ratio.

This option is ignored when writing a file format which does not have image resolution.

Related options -D Specify aspect ratio

Example Convert the Targa file input.tga to a TIFF file called output.tif, specifying that the resolution of the image in the TIFF file is 300 dpi by 300 dpi:

```
alchemy -t input.tga -D 300 300 output
```

Convert the file scan.tif to a DCX variation of a PCX file, scaling the output image to 1500 by 750 (preserving the images aspect ratio) and setting the resolution to 200dpi by 100dpi (this is useful if you will be faxing the image using a fax card):

```
alchemy scan.tif -p1 -X1500 -Y750 -+  
-D 200 100
```

Viewing Options

MS-DOS Only

Display hardware On properly equipped MS-DOS based computers
Image Alchemy can display images.

Depending on the hardware installed Alchemy supports 320x200x256, 360x480x256, 640x400x256, 640x480x256, 800x600x256, 1024x768x256, 1280x1024x256, 640x480x32768, and 800x600x32768 display resolutions (the exact resolutions supported depend on the model board installed).

Alchemy automatically detects which type of display board you have installed. If there are multiple display boards installed in one computer then Alchemy will display images on the first board it finds, searching in the following order:

Western Digital based 8514/A board

AI compatible 8514/A board

VESA compatible SVGA board

Other SVGA board.

Western Digital 8514/A boards which are equipped with the 8514/A Western Digital chipset are automatically recognized by Alchemy. Depending on the model board and the amount of memory installed, 640x480x256, 1024x768x256, and 1280x1024x256 modes are available.

AI 8514/A Alchemy requires AI to be installed to use 8514/A displays which aren't based on the Western Digital chipset. In addition to 8514/A boards Alchemy should also be able to display on other AI compatible boards, such as 340x0 based boards; however this has not been tested. For AI based boards the only resolution available is 1024x768x256

VESA The best support for SVGA boards is available for VESA compatible SVGA boards. VESA is a SVGA standard which allows applications software, such as Image Alchemy, to interrogate the SVGA board to determine which display modes are available. Some SVGA boards have VESA support built directly into the BIOS found on the board; in this case Alchemy will automatically detect the VESA driver and use it. Other SVGA boards require a software driver to be installed; these drivers are usually found on the floppy disks which came with your SVGA board (typically the driver is called VESA.EXE). If you can't find a driver on the diskettes and the documentation does not explicitly mention that VESA support is built into the BIOS you might call the manufacturer to see if a VESA driver is available. VESA drivers are currently available for VGA boards using chipsets from Cirrus Logic, ATI Technologies, Chips and Technologies, Everex Systems, Genoa Systems, Paradise Logic, Sigma Designs, STB Systems, Tecmar, Headland Technology (Video 7), Orchid Technology, Appian Technology, Trident Microsystems, and Oak Technology.

Other SVGA If Alchemy cannot find a VESA SVGA board it attempts to determine what kind of SVGA board is present.

The 320x200x256 mode is a standard IBM VGA mode and will work on all VGA boards. The 360x480x256 is a non-standard VGA mode which should also work on all VGA boards.

Because of various incompatibilities between different VGA boards 640x400x256 and/or 640x480x256 modes are not supported on some VGA boards. VGA boards which have been tested for the higher resolution modes include Paradise, Tseng Labs 3000 & 4000, Video 7, Trident, and Everex chipset based VGA Boards.

General Unless you explicitly specify a resolution
Alchemy automatically uses the lowest resolution
mode which will display the entire picture.

The image will be positioned so that its center
coincides with that of the display.

View Image -v

Purpose View file.

Syntax -v horizontalResolution

Parameter horizontalResolution
320:Use 320x200 mode
360:Use 360x480 mode
640:Use 640x480 mode
800:Use 800x600 mode
1024:Use 1024x768 mode
1280:Use 1280x1024 mode

Comments If displaying on a Western Digital chipset 8514/A or VESA compatible VGA, an optional parameter may follow the -v command. This parameter specifies horizontal resolution and may be 320, 360, 640, 800, 1024, or 1280. The default is to use the lowest resolution which can fit the entire image.

If the image is true colour, a uniform palette will be used and the image will be dithered (dithering may be disabled by use of the -d option, see above). See Appendix B, Colour and Dithering, for more information.

Related options -V Reduce image to fit display

Example View the image madonna.gif:

alchemy madonna.gif -v

View Image in 15 Bit Mode

--v

Purpose View file using 15 bits/pixel mode. This allows true colour images to be viewed without dithering to a uniform palette.

Syntax --v horizontalResolution

Parameter horizontalResolution
640:Use 640x480 mode
800:Use 800x600 mode

Comments Only works on a Tseng 4000 equipped SVGA board with a Sierra DAC and 1 Megabyte of memory on the SVGA board.

Example View madonna.tga:

alchemy madonna.tga --v

View Scaled Image

-V

Purpose View image while scaling image to fit on monitor and correcting aspect ratio.

Syntax -V horizontalResolution

Parameter horizontalResolution
320:Use 320x200 mode
360:Use 360x480 mode
640:Use 640x480 mode
800:Use 800x600 mode
1024:Use 1024x768 mode
1280:Use 1280x1024 mode

Comments This command will scale the image and correct the aspect ratio of the image by removing rows and/or columns from the image.

Note that this option can also be useful for displaying images which are not larger than the screen but which have an aspect ratio different than the display.

Limitations Alchemy assumes that the aspect ratio of a display pixel is 1:1 when in 640x480, 800x600, 1024x768, and 1280x1024 modes, 5:6 when in 640x400 mode and 320x200 modes, and 16:9 in 360x480 mode.

If not otherwise specified by using the -D option or in the file, Alchemy assumes that the aspect ratio of pixels in 640x400 images and 320x200 images is 5:6 and the aspect ratio of pixels in 640x350 images is 35:48. You can override any of these assumptions with the -D option.

Don't worry if this is confusing; in practice Alchemy deals with everything automatically if you use the -V option. However, there is a problem with displaying 320x400 IFF files; see Appendix A, Answers to Frequently Asked Questions, for more information.

Related options -D Specify image resolution
-v View image

Example View madonna.gif:

alchemy madonna.gif -V

A

Answers to Frequently Asked Questions

Question When I view a JPEG compressed image on my VGA board it looks much worse than when I first convert it to a GIF file and then view it. Why is this?

Answer To save time Alchemy automatically uses a uniform palette when you are just viewing a true colour image. When converting to a different file format Alchemy uses Heckbert quantization to generate a palette. The difference in image quality is the difference between using a uniform palette and an optimum palette. See Appendix B, Colour and Dithering, for more information on palette generation.

Question Why can't my paint package read the Targa file I wrote with Image Alchemy?

Answer Some software which reads Targa files can not handle compressed files. In addition, some software can read true colour Targa files, but can not read paletted or gray-scale files. Image Alchemy can be forced to write out a true colour file by using the -24 option.

Question I told Alchemy to convert a PCX file to an 8 bit GIF file (using the -8 option). Yet when I get statistics on the file (using -x) Alchemy reports the file only has 16 colours.

Answer Alchemy will always store the file using the smallest bits-per-pixel allowable for the given image (this results in the smallest possible file). In this case the input file only had 16 colours in it.

Things get more unpredictable with formats such as Sun Raster (which requires 1 bit files to be black and white) and SGI (which requires 8 bit files to be gray-scale). In these cases Alchemy will always do the best it can (giving you a warning message if it does something which may surprise you later).

Question Why is decompressing or compressing a JPEG image so slow?

Answer There are a large number of calculations that have to be done during JPEG compression. This is an inherent limitation of JPEG compression. Image Alchemy has been optimized quite a bit to reduce the number of calculations, and we are working to further reduce the number of calculations. If you are transferring files over modems or storing them on slow media (tape) the compression times are usually more than made up for by the decrease in transmission or retrieval times.

Question Why can't my favorite desktop publishing package read the TIFF file I wrote with Image Alchemy?

Answer TIFF is an extremely versatile standard; it can handle anything from 1 bit images to full colour images with an alpha channel. Also, TIFF allows many different types of compression. Unfortunately this versatility means that it's difficult for a single piece of software to be able to read in every valid TIFF file.

If the software specifies the classes of TIFF it can read, you can force Alchemy to write out a specific TIFF class by using the following options:

- class B: -8 -b -c2 -t2
- class G: -8 -b -t1
- class P: -8 -t1
- class R: -24 -t1

Class B is black and white, Class G is gray-scale, Class P is paletted, and Class R is true colour.

If the supported classes are not specified experiment with various combinations of -24, -8, -b, and -c. In this case it is usually best to use no compression (-t0) while experimenting with the other options, as many TIFF readers have difficulty with compressed files. When you find a set of options that work, then you can try various compression modes to save space. Be aware that using the -b option will force the output file to be gray-scale and you will lose the colour information in the file (most DTP programs only have support for gray-scale TIFF files).

You may also have to use the -Dn n option to specify the resolution of the image (this is especially true when converting from a file format which does not have a value for image resolution). You can generally tell if this is necessary because the program you are using to read in the TIFF file will claim that the file is unreasonably large or small. Generally, if you are using a 300 DPI Laser Printer you want to make the TIFF file 300DPI x 300DPI (-D 300 300).

If you would like further information specific to using Image Alchemy with your word processor or desktop publishing program please contact us; we will be maintaining a list of how to make Alchemy work with other software packages. Similarly if you figure out how to import files into a specific package let us know and we will add your tips to our documentation.

Question I've converted a Mac PICT file to a GIF file, but the GIF file is missing some or all of the information that was in the PICT file. What happened to it?

Answer PICT files are a combination of drawing commands (such as lines, rectangles, and circles) and raster areas (called pixMaps). Alchemy can only read the raster portions of the files. Programs such as MacDraw and MacDraft write out files with drawing commands, programs such as MacPaint write out files which are entirely raster areas (pixMaps), and some programs, such as SuperPaint can write out files which are either or a combination of both. If you are using such a program check the documentation on how to write out files in "paint" mode.

Question When I convert a GIF file to a JPEG file and then back to a GIF file the final GIF file is twice the size of the original. Why is this?

Answer There are two things which might cause this to happen:

JPEG compression doesn't really work well for images which have large areas which are all the same colour. The reason for this is that JPEG is a lossy compression technique. Therefore you are not going to get back exactly the same values for each pixel in an area that was one solid colour before being JPEG compressed. But GIF compression works much better on areas which are one solid colour, so, when you GIF compress these areas, they are quite a bit larger than they were before. The solution to this problem is to use HSI JPEG compression, which automatically detects large areas of solid colours and does not JPEG compress them. The problem with HSI JPEG compression is that it isn't compatible with JPEG or JFIF.

The other possibility is that the input GIF file didn't have very many different colours. When you converted it to a JPEG file the number of colours in the file was lost (JPEG gray-scale files always use 256 shades, and JPEG colour files are always true colour). When the JPEG file was converted back to a GIF file Alchemy assumed you wanted 256 colours in the file, and a 256 colour GIF file is bigger than a 16 colour GIF file. To prevent this you can use a -c32 (or however many colours the original had) option in the command line; this forces Image Alchemy to use that many colours for the output file.

Question I keep getting "Out of Memory trying to ..." messages. Help!

Answer Image Alchemy is running out of memory. First try to do the conversion again with the -\$ (conserve memory) option. Next, if that doesn't help, attempt to maximize the amount of memory available by removing as many memory resident programs as you can. If this still doesn't help please contact us with the following information: your computer configuration (amount of available memory, size of hard disk), operating system version, and what you are trying to do (input file information (size of image and type of file) and options specified). Alchemy can generally convert images as larger than 2000 pixels wide and a virtually unlimited number of pixels tall. However there are certain conversions which require more memory than others.

Question I am using Alchemy to display a 320x400 IFF image created by an Amiga. When I use just the -v option the image comes out tall and skinny. When I use the -V option, which is supposed to correct the aspect ratio, things get worse instead of better (the image is even skinnier). What's going on?

Answer As near as we can tell, some Amiga software has a different idea of what aspect ratio is than the rest of the world.

For displays aspect ratio is defined as the ratio of the width of a single pixel to the height of a single pixel. So if you have square pixels (which you do on a standard monitor in 640x480 mode) the aspect ratio is 1 to 1 (commonly written as 1:1). When you change display modes the height and width of the total display area does not change; what is changing is the width and height of each pixel, which means that the aspect ratio changes. For example, a 640x400 display has an aspect ratio of 1:1.2 (that means each pixel is 1.2 times as tall as it is wide (which makes sense since $480/400$ equals 1.2)). A 640x200 display (remember IBM CGA graphics mode?) has an aspect ratio of 1:2.4.

Now this is where it gets interesting in terms of IFF files. The aspect ratio number stored in Amiga IFF files for 320x400 images is 1:1.1, meaning pixels are 1.1 times as tall as they are wide, so therefore the actual image should be the equivalent size of a 320x440 image with square pixels. And this is what Alchemy will attempt to display when you use the -V option (Alchemy never makes any dimension larger, so the actual image Alchemy displays is 291x400, which is the same ratio as 320x440). However this is obviously wrong, as you can tell when you examine an image. As near as we can tell the correct aspect ratio of these images is 5:3 (the math we used to come up with this number is $640/320:480/400$). And if you tell Alchemy to override the aspect ratio by using a -D 167 option (167 because $5/3*100$ is 166.6666) the image displays correctly. Why Amigas create images which claim they are 1:1.1 remains a mystery.

Question When I convert a 32 bit Targa file to a GIF file and then to a JPEG file it doesn't look nearly as good as if I convert the Targa File directly to the JPEG file. What can I do to maintain high quality in JPEG compressed files?

Answer When the Targa file was converted to the GIF file Image Alchemy had to reduce the number of colours in the file (the original Targa File had up to 16 million colours, GIF files are limited to 256 colours). This step is known as colour quantization (Image Alchemy uses the Heckbert Median Cut method for quantization, see Appendix B, Colour and Dithering, for more information). The difficulty with colour quantization is that it leaves artifacts known as colour banding. To reduce this phenomenon Image Alchemy dithers the image (you can see the effect of colour banding by turning off dithering by using the -d0 option). Unfortunately a dithered image does not JPEG compress very well (dithering adds a lot of high-frequency information to an image; JPEG compression attempts to remove much of that information). In addition JPEG images are always continuous colour images, so when the JPEG file is decompressed it has to be colour quantized and dithered again. Dithering a previously dithered image reduces the quality even more. The solution is to use the best starting quality you can for JPEG compression, ideally a continuous tone image. The compressed image size will be smaller than if you had started with a paletted image and the quality will be better.

Question I've converted an HP PCL file to a GIF file, but the GIF file is missing some or all of the information that was in the PCL file. What happened to it?

Answer PCL files have the same problem as PICT files (see above); they are a combination of drawing commands (such as lines and rectangles) and raster areas (called rasters) and Alchemy can only convert the raster areas in PCL files. PCL Files also contain font and text information, which is also lost. Unfortunately there isn't any general way to preserve this data with Alchemy.

One thing which you can do if running Microsoft Windows 3.0 is to install Adobe Type Manager (ATM). ATM automatically intercepts any text commands and converts them to rasters. In addition, the standard Windows 3.0 HP PCL driver only generates rasters, not vectors. So the file will appear in its entirety when converted by Alchemy. Contact us if you want further information on using Alchemy with Windows 3.0.

Question Why can't Image Alchemy read in JPEG files produced by Kodak's ColorSqueeze (or Sun's VFCtool)?

Answer The JPEG standard is still in draft form. Until it is a mature standard, various manufacturer's will implement different versions of it. As of March 1, 1991 Image Alchemy supports the JFIF format and should work with any other JPEG software which also claims JFIF compatibility. If other software you are using claims to support the JFIF format and you are having trouble please contact us. If the other software does not support JFIF, contact the manufacturer and tell them they should send you an update which does (you can tell them to contact us if they need a copy of the JFIF standard).

Question I converted a PCX file with 16 colours to a 16 shades of gray TIFF file using the -b and -t options. The 16 colour PCX file had some shades of gray in it which were changed in the TIFF file. How can I prevent this?

Answer The problem is that gray-scale TIFF files have a uniformly spaced gray palette. If you create a TIFF file with 16 shades of gray it will have the following shades in it: 0, 17, 34, 51, 68, 85, 102, 119, 136, 153, 170, 187, 204, 221, 238, and 255. However the 16 colour PCX file you started with probably didn't have those exact colours in it (for example, PCX files written out by Windows 3.0 Paint have shades of gray which correspond to 0, 128, 192, and 255). So Alchemy did the best it could and matched the input colours to the output colours (and depending on the other options that you specified may also have dithered the image).

The solution is to tell Alchemy to write out a 256 colour gray-scale TIFF file (which you do by adding a -c256 to the -b and -t options). This file still has a uniform gray palette; but that palette now contains every colour: 0, 1, 2, 3, ..., 255. Therefore Alchemy can map, for example, the colours 128 and 192 to their exact match. This does have the disadvantage of making the resulting 256 colour TIFF file twice as large as the 16 colour TIFF file, but this is the only way to guarantee that Alchemy can find an exact match for all the shades of gray in the input file.

Question Why do you only allow specifying image resolution in Dots Per Inch? Don't you realize that most of the world is metric?

Answer Yes, we do realize that the entire world, with the exception of the United States and Great Britain, claims to use the metric system exclusively (and Great Britain will presumably change in 1992). However, this isn't actually true. A laser printer manufactured in Japan is still 300 dots per inch (not 11.811... dots per mm) and a 19 inch monitor sold in Europe is called a 19 inch monitor (actually a 19 inch monitor is called a 20 inch monitor in Europe, which is a measure of the total picture tube diagonal, not just the viewing area).

B

Colour and Dithering

Paletted vs. true colour Colour images are normally stored in one of two ways: as an array of direct colour values (usually red, green, and blue) (referred to as a true colour file in this document) or as an array of indices into a colour-map which contains red, green, and blue colour values (referred to as a paletted file in this document).

The reason for the existence of paletted images is that they take less memory, so the hardware to display them is less expensive. The dominance of paletted hardware is changing as the price of memory and the processing power it takes to update large amounts of memory at a reasonable speed drops (a Targa 32 board is an example of a true colour board, a VGA board is an example of a paletted board).

Until true colour graphics devices become the norm there's a need to convert images from true colour to paletted. This conversion is done in two steps: the first is to generate a palette for use by the image; the second is to map the image to the new palette.

Colour cube The colour model generally used by computers is a cube with red, green, and blue as the axes (this is known as a colour cube or RGB cube). Each point inside the cube is a different colour, depending on the amount of red, green, and blue used. In nature each of the three axes is nearly continuous, therefore there are a nearly infinite number of colours available. Computer hardware and software represent colours in a discrete fashion.

For true colour displays or file formats the number of discrete positions along each axis of the colour cube gives the colour resolution of the output device. For example, a Targa 24 board for an IBM PC has 8 bits per red, green, and blue channel for a total of 24 bits (or 256 discrete shades of each colour, for a total of 16 million colours (256x256x256)). This is also the colour resolution of most true colour file formats.

The new 15 bit SVGA boards have 5 bits per channel, for a total of 32x32x32 different colours (32,768). This is the same colour resolution as a Targa 15 file.

A paletted display or image file has the same colour resolution limit as a true colour display or image file, but in addition there is a limit on how many points inside the cube can be used at the same time. An 8 bit file format, such as GIF, allows 256 different colours out of 16 million. A normal SVGA board also only allows 256 different colours at one time.

So, converting a true colour file to a paletted file involves reducing the number of occupied points in the colour cube. There are several ways this can be done.

Generating a palette Image Alchemy supports two methods of generating a palette:

Uniform The simplest and fastest method is to use a palette containing colours which are uniformly distributed in the RGB cube, referred to as a uniform palette. This has the advantage that it's fast and the same palette can be used for any image; the primary disadvantage is that most images don't contain colours from everywhere in the RGB cube, so palette entries are wasted representing colours that aren't needed for the particular image being converted.

Optimal To generate a palette which is better for palettes representing a particular image, Image Alchemy supports Heckbert's median cut algorithm. This algorithm first builds a three dimensional table (a histogram cube) indicating how popular any given colour in the RGB cube is in the image being converted. It then proceeds to subdivide this histogram cube (by dividing boxes in half) until it has created as many boxes as there are palette entries. The decision as to where to divide a box is based on the distribution of colours within the box. This algorithm attempts to create boxes which have approximately equal popularity in the image.

Palette entries are then assigned to represent each box. There are other methods of generating a palette from an image, but Heckbert's algorithm is generally regarded as the best tradeoff between speed and quality.

Modifying the You can change the method used to select a palette colour to represent each box by use of the -z selection options.
process

The default method is to use the mean of all the colours in the box. However for some images slightly better results can be obtained by using the center of the box (without regard to where the pixels are in the box).

For images being reduced to a very small number of colours (less than 16) better results can be obtained by using a corner of the box (the boxes tend to be large when reducing an image to a small number of colours; therefore picking colours near the centers of the boxes will give you muddy colours, while using corners of the boxes will give you more saturated colours).

Mapping the The next step is to map the image to the new image to the palette; this is where dithering becomes palette important.

No dithering The simplest approach is to map every colour in the original image to the palette entry which is closest to it (this is what Image Alchemy does if you specify no dithering).

However, since the palette entries generally represent several different colours in the original image, this results in colour banding where areas of smooth colour changes in the original become areas of one solid colour in the paletted version.

Advantages of This can be alleviated by dithering the image dithering data such that any given pixel might not be mapped to its closest palette entry, but the average over some area of the image will be closer to the correct colour than it would otherwise be. Image Alchemy uses a class of algorithms called "error-diffusion" to do dithering.

Error diffusion These algorithms work by using the closest dithering palette entry to a colour and then distributing the error (the difference between the desired colour and the chosen palette entry) to the nearby pixels. This process is repeated for every pixel in the image, using the colour values which have been modified due to the error from previous pixels. The different dithering algorithms spread the error over a different area or use a different weighting within the same area.

Serpentine Error diffusion can be done as a normal raster raster (left to right, top to bottom) or as a serpentine raster (alternating left to right and right to left, top to bottom). A serpentine raster tends to break up visible patterns introduced by dithering.

Noise Random noise can also be added to help break up visible patterns in the resulting image.

Further For more information on Heckbert's median cut information and dithering see the appropriate reference listed in the References section below.

C

What is JPEG Compression?

Who are those JPEG guys? JPEG stands for the "Joint Photographic Experts Group". This is a group of experts who defined a standard compression scheme for still images, commonly called JPEG Compression. Currently the standard is still in draft form. The standard should be finalized in 1991.

Overview JPEG Compression consists of a series of reasonably complex mathematical operations. These include: colour space conversion, discrete cosine transforms, quantization, and entropy coding. After these steps you end up with an image which takes fewer bits to store than you started out with.

However, when you decompress a JPEG compressed image you end up with an image that is not quite the same as the original (which is why JPEG Compression is referred to as "lossy").

Is lossy compression bad? You might well ask why anyone would want to compress an image using a lossy technique. Compression ratios for lossy compression are much better than for lossless compression and the loss is generally very small. And, in fact, every operation of converting an image is lossy (the original photographic or electronic process which captured the image was lossy, scanning or digitizing the image was lossy, displaying the image on a monitor is lossy, and printing the image is lossy).

Details JPEG compression involves the following steps:

Step 1 The image is converted to a colour space with separate luminance and chrominance channels. This is done because the human eye is far more sensitive to the luminance information (Y) than it is to the chrominance information (Cb and Cr); by separating them, it's possible to compress the chrominance information more than the luminance before the perceived image quality suffers.

This step isn't specified in the JPEG draft (it doesn't discuss colour space at all), but is standard practice. Image Alchemy uses CCIR-601 YCbCr, which is the colour space specified by the JFIF standard.

Step 2 The luminance and chrominance information are separately transformed to the frequency domain using a discrete cosine transform acting on 8x8 pixel blocks.

To reduce the amount of data which needs to be compressed the chrominance information may be sub-sampled first. Alchemy uses 2h:1v:1h:1v:1h:1v sub-sampling when writing JPEG files, which means that the first component (luminance) has twice as many samples horizontally as the other two components (chrominance), and the same number of samples vertically. Alchemy can read JPEG files with any sub-sampling allowed by the draft standard.

Step 3 The transformed data is quantized (so some information is thrown away). The samples representing higher frequencies are generally quantized using larger steps than those representing low frequencies.

The quality level you specify is used to scale a set of quantization values which have been found to cause the quantized data to all have approximately equal importance visually. A lower quality number will cause larger quantization steps to be used, and hence increase the compression ratio and decrease the image quality.

Step 4 The quantized data is compressed using an entropy coder. Huffman and Arithmetic coding are allowed by the draft JPEG standard; only Huffman coding is allowed by the JFIF standard. Huffman coding can either be done with a set of fixed tables or custom tables can be generated for an image. Alchemy, by default, uses a fixed set of tables, but can also generate custom tables which usually produce 5-20% (depending on the image and quality setting) better compression. However, producing custom tables requires an additional pass over the image data and therefore takes a little longer.

JPEG Interchange Format This data corresponds to the JPEG Interchange Format and is ready to be stored in a file. Unfortunately the JPEG Interchange Format does not include enough information to actually be able to convert the file back to an image. Specifically the colour space used and the aspect ratio or resolution of the image are not included. Until recently there was no standard way of putting this information in a JPEG file.

JFIF On March 1, 1991 representatives of several JPEG hardware and software developers (including C-Cube, Radius, NeXT, Storm Tech., the PD JPEG group, Sun, and Handmade Software) met at C-Cube and established the JPEG File Interchange Format (JFIF). If you would like more information on the JFIF standard please contact us.

D

Customer Support

Why might We have made every effort to insure that Image Alchemy mess up? Alchemy can read all files in its supported formats. However, because of poorly written standards and non-adherence to standards there are undoubtedly certain files that Image Alchemy does not read correctly.

What we need to If you come across any files which Image Alchemy fix the problem has trouble with please contact us with as much of the following information as you have:
version of Image Alchemy you are using, type of file, type of computer which generated it, name and version of software which wrote the file, size of image, and the number of colours in image. We may ask you to send us the file so that we can figure out what went wrong. If you send us a file we will attempt to modify Image Alchemy so that it can read the file. And if by some miracle we manage to get it to work, we will send you an updated copy of Image Alchemy.

Similarly, if any files that Image Alchemy writes can not be read by other software we want to know that also. Since we do not own a copy of every software package we may ask you to send us a copy of a file that can be read by that package; we will then compare that to a file written out by Image Alchemy to determine what the differences are.

Please contact us even if you are just using a demo copy of Alchemy. In addition to helping fix a potential bug, we feel the best way to get you to purchase a copy of Alchemy is to demonstrate how committed we are to customer support.

How to contact us The best way to contact us is by e-mail; this is especially true if you can send us a sample file which demonstrates the problem. Our e-mail addresses are:

Internet: hsi@netcom.COM
CompuServe: 71330, 3136

We also have a 24 hour bulletin board where you can upload and download files. It speaks 2400 baud, 9600 baud (v32, v42, and v42.bis), and PEP and its number is:

+1 408 356-3297 (BBS)

If you want to contact us through more pedestrian means our address and phone numbers are:

Handmade Software, Inc.
15951 Los Gatos Blvd., Suite 17
Los Gatos, CA 95032

+1 408 358-1292 (Voice)
+1 408 356-4143 (Fax)

E

Binary Information Files (BIF)

Overview Binary files are image files which are just data. In other words, they do not contain any information other than the actual pixels in the image. In order to read these files you must create a file using a text editor which describes to Alchemy the format of the file you are trying to read in. This is called a BIF file (and normally has the extension .bif).

Required information At the minimum a BIF file needs to contain the filename of the image file and either the height or the width of the image. Alchemy will make assumptions about the other characteristics of the image based on the information that it is given and the total length of the image file.

BIF file format The first line contains the letters BIF, which identifies the file as a BIF file.

Each of the rest of the lines in the BIF file consist of an information tag followed by the information. The spelling of the tags must be exact or Alchemy will report an unknown tag error.

Tags

Tag	Description
-----	-------------

filename	The name of the file containing the binary data.
----------	--

width	The width of the image data, in pixels.
-------	---

height	The height of the image data, in pixels.
--------	--

planes	The number of planes of image data (1, 2, 3, or 4).
--------	---

A 1 plane image is assumed to be gray-scale, a 2 plane image is a gray-scale image with an alpha channel, a 3 plane image is a RGB image, and a 4 plane image is a RGB image with an alpha channel.

header	The size of the header, in bytes. This many bytes will be skipped when reading the file.
--------	--

leftpadding	The number of bytes to remove from the beginning of each scan line.
-------------	---

rightpadding	The number of bytes to remove from the end of each scan line.
--------------	---

order	The order of the pixels.
-------	--------------------------

For 3 channel images, this can be any sequence of r, g, and b: rgb, rbg, grb, gbr, brg, or bgr (r=red, g=green, b=blue).

For 4 channel images, this can be any sequence of a, r, g, and b (a=alpha).

Either ga or ag for 2 channel images (g=gray, a=alpha).

The defaults are g, ga, rgb, and rgba, depending on the number of planes.

upsideupdown	The presence of this tag indicates that the data in the file is recorded from the bottom of the screen up to the top of the screen.
--------------	---

Comments	Lines beginning with a # are treated as comments. Comments and blank lines are ignored when processing the file.
----------	--

Palette files If the binary file has a palette available, you can use that palette by writing custom software to convert it to a .PAL file and using the -F option while reading the BIF file.

Example This is an example BIF file which can be used to read a 640 pixel wide, true colour HSI Raw file. Note that HSI raw files have a 32 byte header which is being skipped. Of course you could read the Raw file directly using Alchemy, but this is after all an example of a BIF file.

```
BIF
width 640
#skip past header
header 32

filename sample.raw
planes 3

#the tag below isn't actually needed,
#since rgb is the default, but is
#include here to give an example of
#what an order tag looks like

order rgb
```

Using the Assuming the BIF file is called sample.bif, the example BIF file following Alchemy command can be used to convert the image to a GIF file:

```
alchemy sample.bif -g
```

The height of the image will be automatically calculated from the length of the file and the width, header, and planes tags.

A BIF file is treated as an ordinary file, so all the standard Alchemy commands may be used.

HSI Raw Files

History The HSI Raw format was originally intended as an internal format to Image Alchemy. Because of user demand the format has been documented to allow others to read and write HSI Raw files.

Overview HSI Raw files are completely uncompressed, unpacked, and unpadding image data files. Therefore they tend to be larger than almost any compressed file format. However, they have the advantage, as far as Alchemy is concerned, that they are very fast to read and write and the location of any pixel in the image may be found by simple calculations.

If you need to convert custom files to a format that Alchemy can read we recommend using a Raw file; it is the simplest format to write and the fastest for Alchemy to read.

Variations There are two types of HSI Raw Files: paletted and true colour. Paletted images are stored one byte per pixel with a palette at the beginning of the file. True colour files are stored three bytes per pixel.

Gray-scale Gray scale files are stored as paletted files with a palette that contains all gray values. Alchemy automatically recognizes such files during reading and will treat them appropriately.

Black and white Black and White files are stored as paletted files with a palette that contains two values, black and white. Alchemy automatically recognizes such files during reading and will treat them appropriately.

Warning Note that Handmade Software, Inc. reserves the right to make changes to this format at any time and without notice. And while it is unlikely, it is possible that future versions of Image Alchemy will not support this format.

Old version files This appendix describes version 4 Raw files. This is the version that Image Alchemy has written since March 1991. Before this Alchemy wrote version 2 and 3 raw files (version 2 were 8 bit files, version 3 were 24 bit files). Those raw files can be read by current versions of Image Alchemy but are not otherwise supported. If you run across any of these raw files the easiest thing to do is to use a current copy of Alchemy to convert them to a version 4 raw file.

Details

Word size All values which are not otherwise identified are two byte integers (16 bits). This is the native integer size of most IBM PC C-compilers but not for Macintosh and Sun C-compilers.

Byte order All integers are stored high byte first (big-endian order). This is the native mode for Macintosh's and Sun's but not the native mode for IBM PC's.

See below for a CPU independent method to read and write 2-byte integers.

Pixel format Paletted files are stored one byte per pixel.

True colour files are stored as three bytes per pixel in red, green, blue order.

Padding Neither the palette information nor the pixel data is padded to anything other than a byte boundary. This means that if you store a file which is 13 by 11 pixels it will occupy 429 bytes if stored as a true colour file (not including the header), or 143 bytes if stored as a paletted file (not including the header and palette data).

Hex Numbers including a 0x prefix are hex; all other numbers are decimal.

File format The header for a paletted file is 32 bytes plus the size of the palette. The header for a true colour file is exactly 32 bytes (a true colour file contains no palette).

Magic number Six bytes used to identify the file as a HSI Raw file:

0x6d 0x68 0x77 0x61 0x6e 0x68

Version An integer used to identify the version HSI file:

0x0004

Width An integer indicating the width of the image (in pixels).

Height An integer indicating the height of the image (in pixels).

Palette size An integer indicating the number of entries in the palette. Range is 2 to 256. A 0 indicates a true colour image (which has no palette data).

Horizontal DPI An integer indicating the horizontal resolution of the image, in dots per inch. A zero indicates that the resolution is unknown. A negative number is used if only the aspect ratio is known.

Vertical DPI An integer indicating the vertical resolution of the image, in dots per inch. A zero indicates that the resolution is unknown. A negative number is used if only the aspect ratio is known.

Gamma An integer indicating the gamma of the image, scaled by 100 (a gamma of 2.2 is stored as 220). A zero indicates that the gamma is not known.

Reserved Twelve bytes reserved for future use. Should be set to zero when writing.

Palette The palette data is stored as 3 bytes per palette entry. The bytes are in red, green, blue order; 0 is black, 0xff is full intensity.

True colour raw files have no palette.

Image data The image data.

Example files

8 bit paletted, 6D 68 77 61 6E 68 00 04 01 40 00 C8 01 00 00 00
320 x 200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
49 24 24 24 00 00 00 00 00 DB 6D 6D FF 92 92 FF
B6 B6 92 49 49 FF DB DB FF B6 92 FF FF DB FF DB
B6 FF FF FF B6 6D 6D 6D 24 24 DB 92 6D 6D 49 49
...

24 bit true colour, 6D 68 77 61 6E 68 00 04 01 40 00 C8 00 00 00 00
320 x 200: 49 24 24 49 24 24 49 24 24 49 24 24 49 24 24 49
24 49 24 24 49 24 24 49 24 24 49 24 24 49 24 24
24 49 24 24 49 24 24 49 24 24 49 24 24 49 24 24
...

Reading a two
byte integer

```
int putWord(int i, FILE *stream) {  
    putc(i>>8, stream);  
    return(putc(i&0xff, stream));  
}
```

Writing a two
byte integer

```
int getWord(int i, FILE *stream) {  
    register int temp;  
    temp=getc(stream)<<8;  
    return(getc(stream) | temp);  
}
```

G

Undercolour Removal Files

Summary Undercolour removal files are text files which control the conversion from RGB to CMYK colour space.

This conversion consists of four steps. The first is to convert an RGB value to an ideal CMY value; this simply involves negating the RGB values. The next step is to determine how much black is in that colour; this is done by finding the minimum of the CMY values and using that as an index into the black removal tables documented below. These tables have independent values for how much black to use for that pixel and how much black to subtract from the CMY values. Next, a linear transform is optionally applied to the CMY portion of the CMYK pixel. Finally the CMYK values are optionally translated, independently, through the CMYK density correction tables (this last step is only used if Alchemy is going to dither the image for output on a 1 bit per pixel per component device).

File format Any line beginning with ';' is a comment and is ignored.

Black removal The first 256 non-comment lines contain tables undercolour removal values corresponding to computed black values of 0 (white) to 255 (black). Each of these lines has two numbers; the first indicates how much black to use in place of the computed black value corresponding to the line, and the second indicates how much black to subtract from the cyan, magenta, and yellow components (this value must not be greater than the corresponding computed black value).

After the black removal block the remaining blocks may appear in any order:

CMY linear If there is a line which says only "HSI CMY transform matrix" then the next 3 non-comment lines contain a matrix representing a linear transform which is applied to the cyan, magenta, and yellow components after black removal and before applying the density map. The entries are normalized around 256. The first row and column represent cyan, the second magenta, and the third yellow. The rows are multiplied by the input cyan, magenta, and yellow values to create the corrected values. A matrix of

```
256 0 0
0 256 0
0 0 256
```

is equivalent to omitting the matrix and causes no correction to take place. In this case it would be preferable to omit the matrix as the conversion will run slightly faster without it.

CMYK density If there is a line which says only "HSI CMYK correction density map" then the next 256 non-comment lines contain density correction tables, corresponding to cyan, magenta, yellow, and black values of 0 (white) to 255. Each of these lines has four numbers representing, in order, the amount of cyan, magenta, yellow, and black to use in place of the corresponding computed values. These tables are only applied during dithering; they will not be used for those CMYK output formats which are continuous tone, as devices which take continuous tone input data should be doing their own correction.

Example The following undercolour removal file has undercolour removal tables, CMYK density correction tables, and a CMY colour correction matrix.

```
; Undercolour removal file
;
0 0
1 1
1 1
2 2
3 3
... (256 entries total)
169 169
169 169
170 170
;
```

```

HSI CMY matrix
;the following matrix leaves the
; Cyan and Yellow planes alone, and
; subtracts a bit from the Magenta
; plane when there's Cyan present.
;
256 0 0
-32 256 0
0 0 256
;
HSI CMYK density map
;
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
... (256 entries total)
246 246 246 246
248 248 248 248
251 251 251 251
253 253 253 253
255 255 255 255

```

H

PAL Files

Overview PAL files are text files which contain a palette in an ASCII form. Alchemy can extract palettes from other file formats and write PAL files. Alchemy can also use PAL files when converting images.

File format The first line contains the letters "PAL"; this identifies the file as a palette file.

The next line contains an integer indicating the number of palette entries. Valid values are 2 through 256.

The rest of the file consists of lines of 3 numbers each (separated by spaces) representing the red, green, and blue values for each of the colours. These have a range of 0 (black) to 255 (full intensity).

Example

```
PAL
8      ;# colours
0 0 0 ;black
255 0 0 ;bright red
0 128 0 ;dark green
255 255 0 ;yellow
0 0 255 ;blue
255 0 255 ;magenta
63 63 63 ;gray
255 255 255 ;white
```

I

Version History

Version 1.5 Released for Sun SPARC, Sun-3, and MS-DOS.
10/91

Format Disabled JPEG arithmetic coding pending resolution of patent issues.
Added optimum Huffman table generation for JPEG compression.
Increase in JPEG compression and decompression speed.
JFIF compatibility verified with Radius, Xing Technology, and the PDJPEG group.
Reduced memory requirements when reading GIF files.
The background colour value in GIF files is now set to the darkest colour in the image.
Added 24 bit PCX support.
Added support for many additional types of incorrect TIFF files.
Added numerous new file formats, including Vivid, MTV, DCX, QDV, Erdas, QRT, GEM, Utah RLE, ADEX, RTL, WPG, Pictor, Autologic, q0, BIF, Stork, XWD, Scodl, and XBM.

Display Added support for VESA compatible SVGA boards.
Added support for Sierra HiColor DAC.

Misc Improved documentation.
Added high quality image scaling.
Added aspect ratio preservation flag for scaling.
Added wildcard support.
Added support for 15 bits per pixel true colour, including dithering 24->15 bpp.
Added serpentine raster and noise to dithering.
Added quiet flag.
Added option to not add extension to file names.
Added option to force use of minimum memory.
Added Spiff contrast enhancement.
Added undercolour removal control file for CMYK formats.

Bugs fixed Reading and writing Group III and IV TIFF files.

- Reading JFIF files with thumbnails.

- Reading 24 bit BMP files.

- Writing 24 bit PICT files.

- Reading uncompressed PICTs.

- Reading PICTs with DPI!=72.

- Writing 24 bit EPS files.

- Reading and writing RLE BMP files.

- Reading and writing very large images on Sun-3s and Sun-4s.

- Writing DPI values correctly in ILBM files.

- E option wasn't forcing 16 colours.

Version 1.4 Released for Sun SPARC, Sun-3, and MS-DOS.
03/18/91

Format Added Arithmetic coding and decoding for JPEG compression.

- Added 1 bit PCX file support.

- Added new compression types to Targa.

- Added PBM/PGM/PPM support.

- Added GIF89A support for reading.

- Added SGI support.

- Added Windows BMP support.

- Added writing Group III, Group IV, PICT, and SGI RLE Tiff compression types.

- Added Encapsulated PostScript output.

- Added HP PCL support.

Display Added support for Video 7, Trident, Tseng 4000, and IBM 8514/A display boards.

- Added 320x200x256 display mode.

- Added image scaling during display (which allows display of the entire image on the screen and preserves aspect ratio).

- Images are centered on screen during display.

- Before image display the screen is cleared to the darkest colour.

Misc Added image scaling.
Added conversion to monochrome images.
Preserve and allowing specifying of image aspect ratio or resolution.
Added false colour palette conversion.
Added optimized EGA image generation support.
Added support for very large images.
Added palette sorting, paletted swapping, and palette selection parameters.
Added Stucki and Jarvis, Judice, & Ninke Dithering.

Bugs fixed Writing certain 1 bit Sun Raster files.
Reading incorrectly written PCX files.
Reading 24 bit RLE compressed Sun Raster files.
Reading incorrectly written ILBM files.

Version 1.3 Released for Sun SPARC and MS-DOS.
02/05/91

Formats Reading Sun Run-Length-Encoded (RLE) files.
Added reading support for interleaved GIF files.
Added 8 bit RAW files.
Added optional smoothing on JPEG reading.
Added support for any colour component sub-sampling on JPEG reading.
Targa files are now written bottom-up to make other Targa software happy.
Added palette matching and PAL files.

Display Added checking for supported VGA boards, warning if board is recognized but not supported.

Misc Made memory use more efficient (requiring less virtual memory on Suns and less overlay swapping on PCs).
Numerous miscellaneous performance improvements, (dramatic for some conversions; hardly noticeable for others).

Bugs fixed Reading and writing 8 bit Sun Raster files which have an odd width.
Temporary files are now removed when program exits because of an error.

Version 1.2 First release for Sun SPARC systems.
01/16/91

Bugs fixed Fixed several minor bugs (primarily aesthetic).
Made PCX file identifying and reading more
robust.

Version 1.1 First release for MS-DOS.
01/14/91

J

Acknowledgments

Summary Almost all the software which comprises Image Alchemy was written in house. However the TIFF and 640x400 SVGA display modules are modifications of software originally written by other people.

Both of these modules are free for anyone's use as long as proper credit is given as to the origin of the software.

TIFF Image Alchemy's TIFF I/O is based on libtiff which is copyright by Sam Leffler and is used with his permission. If you are interested in reading or writing TIFF files we strongly suggest that you start with libtiff.

Libtiff is available by anonymous ftp as [ucbvax.berkeley.edu:pub/tiff/*.tar.Z](ftp://ucbvax.berkeley.edu/pub/tiff/*.tar.Z) or [uunet.uu.net:graphics/tiff.tar.Z](ftp://uunet.uu.net:graphics/tiff.tar.Z).

If you can not get a copy of libtiff via anonymous ftp please contact us for a free copy.

VGA display Image Alchemy's 640x400 SVGA display routines are based on VGAKIT, written by John Bridges.

VGAKIT is available free of charge from a variety of bulletin boards

If you can not find VGAKIT locally please contact us for a free copy.

K

Other Useful Software

Summary There are several image processing packages available for little or no cost.

Please be aware that we mention these software packages only as a service to Image Alchemy users. We are not endorsing or recommending any particular package. Many of the packages are not supported by their authors.

If you have trouble finding any of the listed software please send us a blank tape or diskette and we will send you a copy free of charge (please be aware that the software may be quite large; contact us first if you have any questions).

If you know of any other software which would be appropriate to add to this list please let us know.

If you are the author of any of these packages and you would rather not be on this list please let us know.

IBM PC

PicLab A public-domain image file conversion and printing tool.
Written by Lee Crocker and the Stone Soup Group.
Available via CompuServe.

Cshow A shareware image viewing program.
Written by Bob Berry.
Available from:
Canyon State Systems and Software
PO Box 86
Sedona, AZ 86336

Vivid A shareware ray-tracing program.

Written by Stephen B. Coy

Available from:

Stephen Coy

15205 NE 13th Pl., #2904

Bellevue, WA 98007

Workstations These programs are only available as source code and generally require a workstation running UNIX or one of its variants.

Utah Raster Toolkit (URT) Written by Spencer W. Thomas, Rod G. Bogart, and James Painter.

Available via anonymous FTP as pub/urt-3.0.tar.Z

via anonymous ftp from cs.utah.edu,

weedeater.math.yale.edu, or

freebie.engin.umich.edu.

Fuzzy Bitmap Manipulation Written by Michael Mauldin

Available by anonymous ftp as

(FBM) nl.cs.cmu.edu:/usr/mlm/ftp/fbm.tar.Z,

uunet.uu.net:pub/fbm.tar.Z, or

ucsd.edu:graphics/fbm.tar.Z.

Portable BitMap Written by Jef Poskanzer

(PBMPLUS) Available by anonymous ftp as

expo.lcs.mit.edu:contrib/pbmplus.tar.Z or

ftp.ee.lbl.gov:pbmplus.tar.Z.

Img Software Written by Paul Raveling

Set Available by anonymous ftp as

expo.lcs.mit.edu:contrib/img_1.3.tar.Z or

venera.isi.edu:pub/img_1.3.tar.Z.

Glossary

Anonymous FTP An easy way to transfer files via the Internet.

If you don't have Internet access you can't use anonymous FTP; if you do have Internet access you probably already know about it (if you don't, ask your system administrator or local network guru).

Black and white An image which contains just two colours, black and white. Many file formats, such as TIFF and Sun Raster, have special variations for black and white images. You can force Alchemy to write a black and white image by specifying -b -c2 as options.

Dithering A technique for reducing the amount of colour banding in an image when converting from a large number of different colours to a small number of different colours. Different dithering techniques are usually named after the person or persons who first invented them. Alchemy supports Floyd-Steinberg, Stucki, and JJN dithering; these are further described in "Digital Halftoning", by Robert Ulichnet, MIT Press.

Gray-scale An image which contains just shades of gray. Many file formats, such as TIFF and Silicon Graphics, have special variations for gray-scale images. You can force Alchemy to write a gray-scale image by specifying -b -8 as options.

Header The portion of an image file that is not the actual image data. The data in a header generally includes the image size (in pixels), the image depth (in number of bits per pixel or number of colours), and the palette (if the image has a palette). Some file formats include quite a bit of additional data in the header, such as: the name of the image, the date and time the image was created, and the latitude and longitude of the image (primarily used by satellite image data). The header is called the header because it usually appears at the head of the file. Some file formats store information which is usually found in the header in a separate file.

Heckbert colour quantization A technique for reducing the number of colours needed by an image, typically used to convert a true colour image to a paletted image. Named after Paul Heckbert who originally described the technique in "Color Image Quantization for Frame Buffer Display", SIGGRAPH '82 Proceedings, p. 297.

Magic Number A number or sequence of numbers that is found at or near the start of an image file so that software may determine what type of format the file is. Most formats have a well defined magic number; some formats do not, in which case Alchemy examines various parameters in the header of the file and guesses what format the image is.

Paletted An image which isn't true colour. Each pixel in the image is an index into a table of values (typically red, green, and blue) which describe the colour of that pixel. Most paletted images are limited to 8 bits of information, which allows 256 unique colours. Most display adapters only allow the display of paletted images (Alchemy can display true colour images on those display adapters by using a uniform palette).

True colour An image which does not contain a palette. Each pixel in the image is represented by at least three values, typically red, green, and blue. True colour images are generally produced by scanners and digitizers and are better quality and much larger than paletted images. Most display systems can not display true colour images.

References

General Computer Graphics - Principles and Practice,
Graphics Second Edition

(Commonly referred to as Foley and van Dam)

J.D. Foley, A. van Dam, S.K. Feiner, and J.F.

Hughes

Addison-Wesley

ISBN 0-201-12110-7

Principles of Interactive Computer Graphics

(Commonly referred to as Newman and Sproull)

W.M. Newman and R.F. Sproull

McGraw-Hill

ISBN 0-07-046338-7

Algorithms for Graphics and Image Processing

Theo Pavlidis

Computer Science Press

ISBN 0-914894-65-X

Graphics Gems

Andrew S. Glassner

Academic Press

ISBN 0-12-286165-5

Graphics Gems II

James Arvo

Academic Press

ISBN 0-12-064480-0

Bit-Mapped Graphics

Steve Rimmer

Windcrest

ISBN 0-8306-3558-0

Specific Topics

Colour The Reproduction of Colour in Photography,
Printing & Television
R.W.G. Hunt
Fountain Press
ISBN 0 85242 356 X

Dithering Digital Halftoning
Robert Ulichnet
MIT Press.
ISBN 0-262-21009-6

Image Scaling Digital Image Warping
George Wolberg
IEEE Computer Society Press Monograph
ISBN 0-8186-8944-7

VGA Programming Programmer's Guide to the EGA and VGA Cards,
Second Edition
Richard F. Ferraro
Addison-Wesley
ISBN 0-201-57025-4

Image Alchemy 1.5 Order Form

Qty	Description	Price	Total
_____	Image Alchemy for MS-DOS (Dual Media)	79.95	_____
_____	Image Alchemy/SPARC for Sun SparcStations (On 3.5" Disk)	\$199.95	_____
_____	Image Alchemy/SPARC for Sun SparcStations (On 1/4" Tape)	\$199.95	_____
_____	Image Alchemy/68000 for Sun-3 (On 1/4" Tape)	\$199.95	_____

Sub-Total: _____

California Residents add applicable sales tax: _____

Shipping and handling	UPS Ground	\$4.00
(per copy)	UPS Second Day	\$8.00
	FedEx (US only)	\$20.00
	International AirMail	\$9.00 _____

Total: _____

Circle Payment form: Check MC Visa Amex

Card #: _____ Exp. Date: _____

Signature: _____

If ordering by American Express, the Bill To information below must correspond to the address to which your American Express bill is sent.

Ship to: Bill to:
Name: _____ Name: _____

Company: _____ Company: _____

Address: _____ Address: _____

City: _____ City: _____

State: _____ Zip: _____ State: _____ Zip: _____

Phone: (_____) _____ Fax: (_____) _____

Mail to:	Call: +1 408 358 1292
Handmade Software, Inc.	Fax: +1 408 356 4143
15951 Los Gatos Blvd., Ste. 17	Email: hsi@netcom.COM
Los Gatos, CA 95032	Email: 71330,3136 (Compuserve)

