

Image Alchemy

Version 1.9.1

Addendum

Handmade Software, Inc.

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 or Image Alchemy PS 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.

Trademarks

Image Alchemy and Image Alchemy PS are trademarks of Handmade Software, Inc.

All other products or services mentioned in this manual are trademarks, registered trademarks, service marks, or registered service marks of their respective companies or organizations.

Copyright

Copyright © 1990-1996 Handmade Software, Inc.,
Fremont, California

Portions © Copyright 1990-1991, Bitstream Inc.
Portions Copyright © 1988-1996 Aladdin Enterprises,
Menlo Park, California

All Rights Reserved

Printed in the United States of America.

Second Printing, April 1996

Image Alchemy v1.9.1 was written by:

Marcos H. Woehrmann
Allan N. Hessenflow
David Kettmann
Paul H. Yoshimune

Handmade Software, Inc.
48860 Milmont Drive, Suite 106
Fremont, CA 94538

1 800 252 0101
+1 510 252 0101
+1 510 252 0909 fax
+1 510 252 0929 BBS

<http://www.handmadesw.com/>
CompuServe: GO HANDMADE

Contents

Chapter 0

Introduction to this Addendum	7
-------------------------------------	---

Chapter 1 Installing Image Alchemy

Overview	9
----------------	---

Chapter 2 Image Alchemy PS

Reading PDF Files	11
Threading	11
Colour Mode	13
Pages	15
Generating PostScript files using Windows 95	17

Chapter 3 Conversion Options

Introduction	25
File Formats	26

Image File Formats

Adobe Acrobat PDF	27
Adobe Photoshop	29
EPS	30

Epson Stylus	33
GIF	35
Intergraph	40
JPEG	41
OS/2 Icon	44
PNG	45
Spaceward Graphics	47
US Patent Image	48

Chapter 4 Other Options

Introduction	49
Multi-Page Input	50
Multi-Page Output	52
Only Scale If Too Large	54
Override input type	55
Preserve palette while scaling	57
Set Horizontal DPI	58
Set Vertical DPI	60
Slide show (MS-DOS Only)	61
Transparency	62
Use input file format for output	63
Use input filename for output	64
Use 3 letter extensions	65

Appendix A Configuring DOS/4GW Pro

.....	67
-------	----

Introduction to this Addendum

Why an Addendum?

This addendum covers changes made to Image Alchemy and Image Alchemy PS since the manual was printed. These changes consist of new file formats, improvements in existing file formats, and additions and changes to other options.

New file formats:

- Adobe Photoshop
- Epson Stylus
- Intergraph
- OS/2 Icon
- PNG
- Spaceward Graphics
- US Patent Image

Changes to existing file formats:

- Adobe Acrobat PDF - added reading (Alchemy PS Only)
- EPS - added output compression
- GIF - Added transparency and multi-page support
- JPEG - Added multiple pass support

New options:

- Multi-page Output
- Only scale if too large
- Preserve palette while scaling
- Set Horizontal/Vertical DPI

Slide show (MS-DOS Only)

Transparency

Use input file format for output

Use input filename for output

Use 3 letter extensions

Changes in Alchemy PS options:

Colour Mode

Pages

Changes to existing options:

Multi-page Input

Override Input Type

Installing Image Alchemy

Overview

The installation instructions found in the manual are still current. The MS-DOS versions of Image Alchemy and Image Alchemy PS are now distributed with compressed executable files. The install.exe program automatically uncompresses them during installation.

If you are updating or upgrading your copy of Image Alchemy the install program will replace the old version of Image Alchemy with this version. It will not remove any other files.

If you are installing the MS-DOS version of Image Alchemy/386 or Image Alchemy PS you should be aware that the environment variable used for configuring virtual memory has changed. If you are using the DOS4GVM environment variable you should read Appendix A.

Image Alchemy PS

Reading PDF Files

In addition to the other file formats it can convert, Image Alchemy PS v1.9.1 now has the ability to read Adobe Acrobat PDF files. The options that you can use when reading PDF files are the same as the options for reading PostScript and EPS files.

Refer to Chapter 3 in the Image Alchemy PS manual for information on these options.

Threading

The two programs which make up Image Alchemy PS, `alchemy` and `alchps`, are now combined into one program. This allows the two programs to be threaded (except for the MS-DOS non-386 version).

Threading combines the PostScript module and the main Alchemy conversion module into one program. This allows the PostScript data to be RIPPed and immediately passed to the rest of Alchemy, as opposed to RIPPing the entire image and then converting it. The new method has several advantages:

- Faster, especially for multi-page PostScript files.
- Makes use of multiple CPUs (UNIX versions).
- Uses less disk space.

- When sending files directly to a device (such as a NovaJet) output will start appearing sooner (only in the `alch386.exe` variant of the MS-DOS version; the UNIX versions also start generating data sooner, but the spoolers used by UNIX systems wait until the data is completely received before sending it to the output device).

Any references to `alchps` in the old manual should be disregarded, as that program is no longer needed (except for the MS-DOS non-386 version which still uses it).

Colour Mode

-Zm

Specify whether to render the image in black and white, grayscale, or colour.

Syntax

-Zm mode

Parameter

mode:

0:Black and White - 1 bit

1:GrayScale - 8 bit

2:RGB Colour - 24 bit

3:CMYK - 4 bit

4:CMYK - 32 bit

The default is Black and White.

Comments

Alchemy defaults to rendering the image in 1 bit black and white.

If you specify grayscale or RGB colour, Alchemy will automatically fall back to grayscale or black and white if the image doesn't use any colour or grayscale, respectively.

However, the memory, disk space requirements, and processing time all increase dramatically when telling Alchemy to render in Mode 1 or Mode 2, so don't specify those modes if it is not necessary.

You may want to specify grayscale output even when converting a PostScript file which will be printed on a black and white device. This way Alchemy can do a better job raster scaling the image and you have control over the dithering type used. If you specify black and white mode the only dither available is the halftone dither.

Using CMYK - 4 bit mode can significantly speed up conversion times, since the amount of data written is less than in the RGB Colour or CMYK - 32 bit modes. However the only dither algorithm available with this mode is a digital halftone.

Using CMYK - 32 bit mode can be useful if you have colour separations in the PostScript file (for example, CMYK EPS files). The CMYK - 32 bit option will preserve these separations so the colour representations will be more accurate and has an advantage over the CMYK - 4 bit mode in allowing any of the dithering types to be used.

Examples

Convert the file poster.ps to a colour RTL file for plotting on a NovaJet plotter:

```
alchemy poster.ps -Zm2 --r10
```

Do the same thing, but go directly to 4-bit CMYK:

```
alchemy poster.ps -Zm3 --r10
```

Do the same thing, but generate a 24-bit CMYK file and then convert it to RTL using dither type 3:

```
alchemy poster.ps -Zm4 -d3 --r10
```

Pages

-Zp

Specify which pages to render.

Syntax

-Zp page

-Zp [startPage [endPage]]

Parameter

page:

Specify page number

The default is page 1.

startPage:

Specify beginning page number.

endPage:

Specify ending page number.

The default is to render all pages.

Comments

If you specify a single page after the *-Zp* option, only that page will be converted. If you specify two page numbers all pages between those two numbers will be converted (inclusive, e.g. *-Zp 2 3* will convert pages 2 and 3).

To convert all pages in a document specify *-Zp* without a following parameter or use the *-U* option.

When converting multiple pages, either multiple files will be written (each containing a single page) or a single multi-page file will be written, depending on the use of the *--U* option. See the *---U* option for more information and the examples section below for an example.

Examples

Convert page 2 of the file *test.ps* to a GIF file called *test.gif*:

```
alchemy test.ps -g -Zp 2
```

Convert pages 2 through 9 of the file test.ps to multiple GIF files:

```
alchemy test.ps -g -Zp 2 9
```

Convert pages 2 through 9 of the file test.ps to a single multi-page GIF file, called pages.gif:

```
alchemy test.ps -gl -Zp 2 9 ---U pages.gif
```

Convert all the pages in the file test.ps to a single multi-page GIF file, called pages.gif:

```
alchemy test.ps -gl -Zp ---U pages.gif
```

Generating a PostScript file using Microsoft Windows 95

Generating a PostScript file which can be read by Image Alchemy PS when running Microsoft Windows 95 requires setting up a PostScript output device that Windows can print to.

Microsoft Windows 95 includes several different PostScript device drivers, including drivers for the Apple LaserWriter, HP LaserJet PostScript, NEC Colormate PS, etc. However most of these do not include colour support, therefore we recommend selecting the QMS ColorScript 100 as the printer driver.

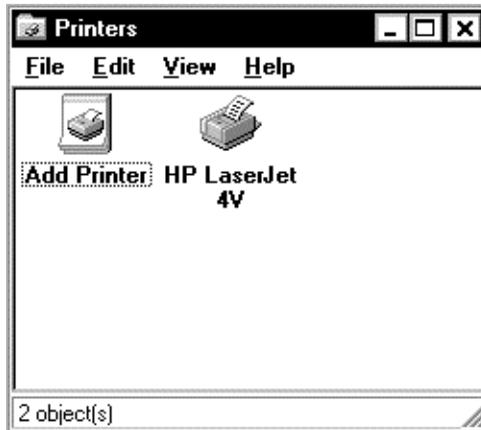
You accomplish this with the following procedure:

Setting up
Microsoft
Windows 95

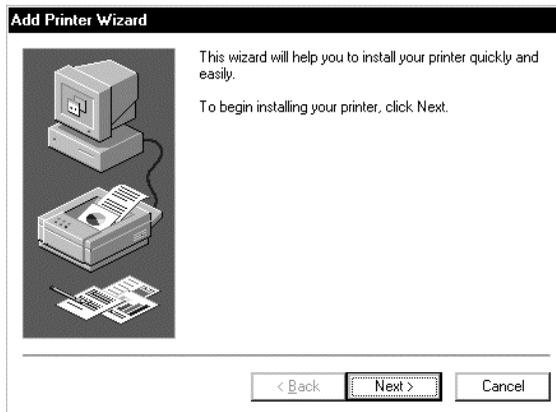
Select **P**rinters from the **S**tart menu under **S**ettings:



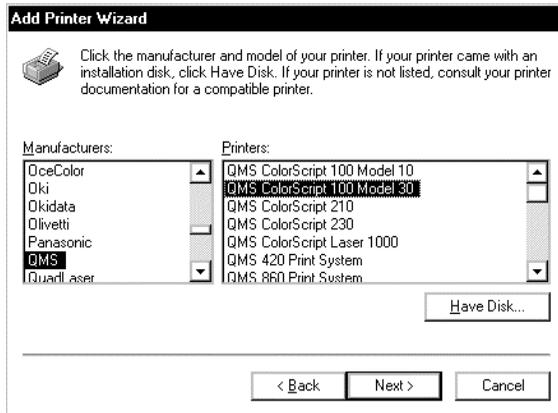
This brings up the Printers window. Double click on the **Add Printer** icon:



This brings up the Add Printer Wizard:

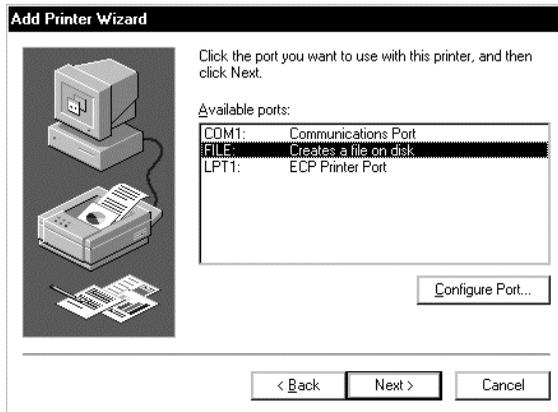


Click **Next >** to continue with the procedure.



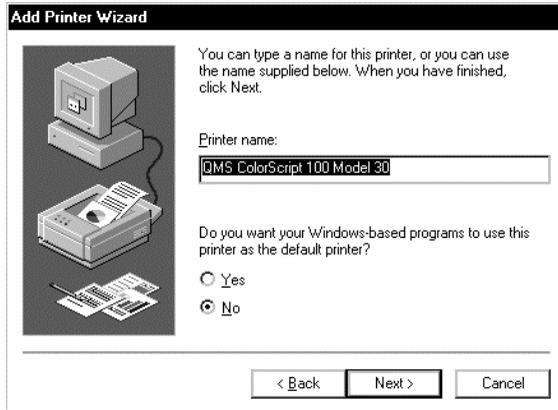
Choose **QMS** from the **Manufacturers:** window, then select **QMS ColorScript 100 Model 30** from the **Printers:** list.

Now click on the **Next >** button to bring up the connect dialog box:

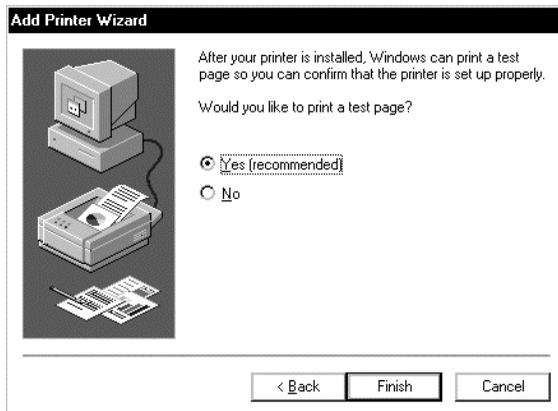


Select the **FILE:** item from the **Available Ports:** list. This indicates to Windows 95 that the device doesn't actually exist and that output sent to that driver should be directed to a file. Click on the **Next >** button to accept this choice.

If you want to make this the default printer you can select **Yes** to the "Do you want ... to use this printer as the default printer?" question. This isn't recommended, since it will make it more difficult to use your normal printer with Windows. You may want to change the name of the printer (perhaps to "Generate PostScript file") to better reflect what this printer choice is used for.



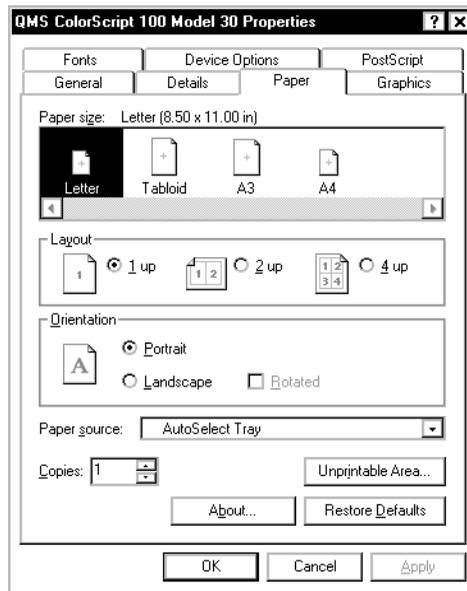
Click **Next >** to finish the installation procedure:



You may be prompted to insert one or more of the Windows 95 distribution diskettes at this point.

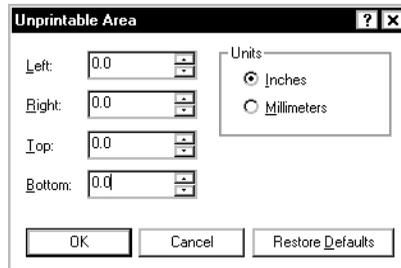
The printer is now available for use. However, there are a couple of settings under the Properties sheet that you may want to change. To bring up the Properties sheet select the printer by single clicking on it and choosing Properties from the File menu.

Under the Paper tab you can select whether you want **P**ortrait or **L**andscape to be the default choice for printing.



The **P**aper **S**ize selection is best left at 8.5 x 11 in, since that is the default size that Alchemy PS expects for PostScript files. The **P**aper **S**ource selection is not used by Alchemy PS.

Click on the **Unprintable Area...** button to bring up the unprintable area dialog:



These can all be set to zero, since Alchemy PS does not have any unimageable area near the edges. If you will routinely be sending files that you convert with Alchemy PS to a hard copy output device, such as a PCL printer, you can set the margins appropriately for that device. Windows will then warn you if you are printing too close to the edge.

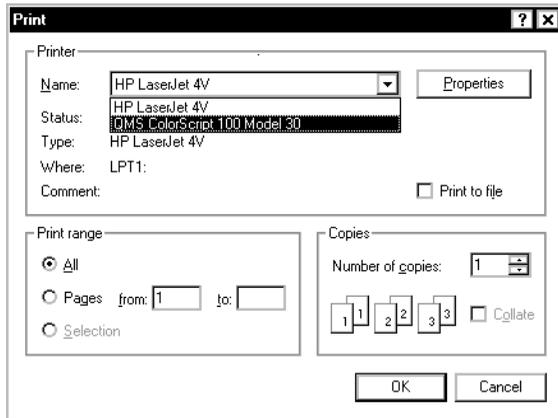
Now click on the various **OK** buttons to close the dialog boxes and accept the choices you have made.

This completes the setup. You are now ready to print a PostScript file.

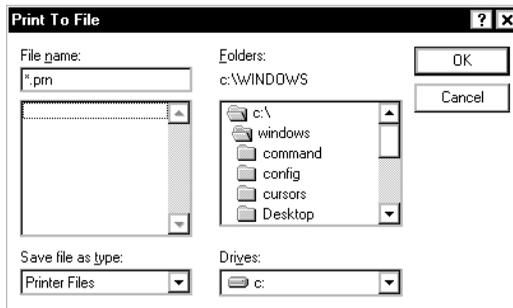
Printing to a PostScript file

To print to a file select **Print...** from the **File** menu. This will bring up a dialog box similar to the one shown; different programs have different dialogs (this example is from the Paint program, distributed with Windows 95).

Select the printer **QMS ColorScript 100 Model 30** as the printer to print to and click **OK**. If you setup this device as the default printer you can skip this step. You can also change the **Orientati on** choice at this point.



Selecting **OK** will bring up the following dialog:



Enter the file name of the PostScript file to be generated. This file will be created and may be converted with Alchemy PS.

Conversion Options

Introduction

This section lists the new and changed conversion options that Image Alchemy v1.9.1 supports.

Name of format

-option

Overview of file format.

Syntax

Description of syntax.

Parameters

Brief description of the parameters. Those parameters which require a detailed explanation are further documented under the comments section below.

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). Some formats use more than one file per image; in that case the extension for each portion of the image is 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 variations supported by Image Alchemy.

Limitations

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

Comments

Miscellaneous things of which you should be aware.

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.

Examples

Sample conversions involving this image format.

Adobe Acrobat PDF

--d

Adobe Acrobat PDF (Portable Document Format) files are used by Adobe Acrobat.

Syntax

--d [*compressionType*]

Parameter

compressionType:

0:None

1:Run Length

2:LZW

3:CCITT Group 3 fax

4:CCITT Group 4 fax

5:JPEG Low Quality

6:JPEG Medium Quality

7:JPEG High Quality

0:ASCII Encoding

10:Binary Encoding

The default is no compression and ASCII Encoding. Options are combined by adding (see below for an example).

Extension

.pdf

Creator

Adobe Systems Incorporated

Used by

Adobe Acrobat

Variations

Reads and writes 1 bit black and white, 8 bit grayscale, 8 bit paletted, and 24 bit colour images.

Limitations

CCITT Group 3 fax and Group 4 fax files are always 1 bit, black and white. Selecting either compression type will cause Alchemy to automatically convert the input image to black and white.

Comments

For more information on reading Adobe Acrobat files see Chapter 3 in this Addendum and in the Image Alchemy PS User's Manual.

Alchemy can write multi-page PDF files when used with the `---U` option. See below for an example.

Related options

`---U` Write multi-page file

Examples

Convert the JPEG file `sample.jpg` to a Run Length compressed PDF file:

```
alchemy sample.jpg --d 1
```

Convert the JPEG file `sample.jpg` to a high-quality JPEG compressed PDF file with binary encoding:

```
alchemy sample.jpg --d 17
```

Convert all the pages in the TIFF file, `doc.tif`, to a multi-page Group 4 compressed PDF file (see the `---U` section below for more information):

```
alchemy doc.tif --d 4 -U ---U
```

Adobe Photoshop

---p

Adobe Photoshop files are used by Adobe Photoshop.

Syntax

---p *compressionType*

Parameter

compressionType:

0:None

1:Compressed

The default is None.

Extension

.psd

Creator

Adobe Systems Incorporated

Used by

Adobe Photoshop

Variations

Writes 1 bit black and white, 8 bit grayscale, 8 bit paletted, and 24 bit colour images.

Comments

Alpha channel data can be read and written by using the -I option, see the appropriate section in the User's Manual for more information.

Related options

-I Include Alpha channel information

Examples

Convert the JPEG file sample.jpg to an uncompressed Photoshop file:

```
alchemy sample.jpg ---p
```

Convert the Targa file alpha.tga to a compressed Photoshop file, with alpha channel:

```
alchemy sample.jpg ---p1 -I
```

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 type

Parameter

type:

0:No preview
1:Device independent preview
2:TIFF preview

0:UNIX newlines
10:Mac newlines
20:MS-DOS newlines

0:Showpage
100:No showpage

0:Uncompressed
1000:LZW
2000:CCITT Group 4 fax
3000:JPEG Low Quality
4000:JPEG Medium Quality
5000:JPEG High Quality

The default is an uncompressed EPS file with a TIFF preview, UNIX newlines, and showpage (output type 2). Options are combined by adding (see below for an example).

Extensions

.epsi
.eps
.epi

Creator	Adobe Systems, Inc.
Used by	PostScript printers
Variations	Black and White, Gray-scale, and RGB.
Comments	<p>For information about reading EPS files see Chapter 3.</p> <p>If the output is black and white or gray-scale, it will work with any PostScript device. If it's colour, then the CMYK extensions or a level 2 device is required.</p> <p>If you are writing an EPS file which you intend to send directly to an EPS output device, such as a printer, you will want to write a type 0 EPS file (No Preview, Unix Newlines, and include the showpage).</p> <p>EPS files are normally written using the UNIX newline convention. To write an EPS file with Macintosh newlines, add 10 to the preview type. To write an EPS file with MS-DOS newlines, add 20 to the preview type. See below for an example.</p> <p>To omit the showpage from the end of the EPS file add 100 to the preview type. Some software which imports EPS files does not correctly handle EPS files which contain the showpage.</p>
Related options	<p>-_ Offset image</p> <p>--_ Center image</p>
Examples	<p>Convert the file input.gif to an uncompressed colour EPS file called input.eps with no preview:</p> <pre>alchemy input.gif -e 0 -24</pre> <p>Do the same think, but use LZW compression:</p> <pre>alchemy input.gif -e 1000 -24</pre>

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

```
alchemy input.gif gray.eps -e 1 -b
```

Convert the file `test.gif` to a black and white EPS file called `test.eps`, with no preview and MS-DOS newlines:

```
alchemy test.gif -e 20 -b -c2
```

Epson Stylus

--K

Epson Stylus files are used by Epson Stylus printers.

Syntax

--K [*outputType*]

Parameter

outputType:

0: Microweave

10: Disable Microweave

0: Uni-directional

20: Bidirectional

The default is Microweave and Uni-directional. Options are combined by adding.

Extensions

.prn

Creator

Epson

Used by

Epson Stylus printers

Variations

1 bit, CMYK.

Limitations

Epson Stylus files can only be 720, 360, or 180 DPI. If you specify any other DPI value the output file will automatically switch to one of those.

Comments

Since the Epson Stylus format is a 1-bit CMYK format you may want to use an undercolour removal file when converting a colour image to this format. See the -C option in the Image Alchemy User's Manual for more information.

Related options

-C Undercolour removal

Examples

Convert the file `page1.tif` to a 360 DPI Epson Stylus file, scaled to 5 inches wide:

```
alchemy page1.tif --K -D 360 360 -Xb5i --
```

Do the same thing, but use a gamma correction value of 2.0:

```
alchemy page1.tif --K -D 360 360 -Xb5i --  
-Gi1.0 -Go2.0
```

GIF

-g

GIF files were developed by CompuServe as a machine-independent image file format. GIF files are the most popular way of storing 8 bit, scanned or digitized images. GIF files are frequently used for storing images on the WWW.

Syntax

`-g [type [delay [repeatCount]]]`

Parameter

type:

0:GIF87A

1:GIF89A

0:non-interleaved

10:interleaved

The default is GIF87A and non-interleaved. Options are combined by adding (see below for examples).

delay:

Specifies the delay between multiple pages in GIF files, in hundredths of seconds (a delay of 250 is 2.5 seconds). The default is 0 (display images with no delay between pages).

repeatCount:

Specifies the number of times the images are to be repeated. Indicating a repeat count of 0 causes the images to repeat continuously. The default is 0. This is a Netscape specific tag.

Extension

.gif

Creator

CompuServe, Incorporated

Used by

CompuServe

WWW

Everyone

Variations

Reads 1 through 8 bit GIF87A and GIF89A interleaved and non-interleaved files, single and multi-page.

Writes 1 through 8 bit GIF87A and GIF89A interleaved and non-interleaved files. Also writes images with transparency information and multi-page GIF images.

Limitations

Any text, overlays, pauses, palette changes, etc. are ignored when reading GIF images.

When writing a multi-page GIF file or one with transparency information, the GIF89A type must be used. Alchemy will automatically change to writing a GIF89A file in these cases.

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

GIF89A files are a newer variation of GIF files that were introduced in 1990. They allow the inclusion of transparency information, text, simple animation, and multiple pages in GIF files.

When writing a simple GIF file you will want to use the GIF87A variation, since the GIF89A extensions aren't necessary to store single images and some software still can't read GIF89A images. The advantages of GIF89A are: aspect ratio information is preserved, transparency information is stored, and multiple pages are allowed.

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).

To write a GIF file with transparency information use the `---t` option (see below). When writing a GIF file with transparency information Alchemy defaults to making the lightest colour the transparency colour; you can override this by using the `---t` option (see below for an example).

Multi-page GIF files are used by certain Web browsers (most notably Netscape Navigator 2.x). See the `---U` command below for more information on writing multi-page files.

When writing a multi-page GIF file you may specify the delay, in hundredths of seconds, between images. If you do not specify a delay it defaults to 0 (which will display the images as quickly as possible). You may also specify a repeat count. This field is a Netscape specific tag that indicates how many times to display the sequence. The default is 0, indicating that the sequence is to repeat indefinitely

Alchemy will write a multi-page GIF file with a global palette if you use either the match to palette (`-f`) or false colour (`-F`) option, otherwise Alchemy will write local palettes. If you have a series of images which contain identical palettes you can force Alchemy to write a global palette by using the `-F` option and give the name of the first file (see below for an example). If you have a series of images which do not contain identical palettes but you still want to write a global palette you can use Alchemy to generate a multi-image palette file and then match each of the images to that palette using the `-f` option (see below for an example).

The LZW compression used in GIF files is patented by Unisys Corporation and used under license. If you write software to read or write GIF files you need to contact Unisys to arrange a license. See Appendix I in the Image Alchemy manual for contact information.

Related options

- t Transparency
- U Write multi-page file
- 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
```

Convert the image logo.pcx to a GIF89A image, using white as the transparent colour (white is the default transparent colour, so we do not need to specify 255 255 255 after the ---t):

```
alchemy logo.pcx -g1 ---t
```

Do the same thing, with red as the transparent colour:

```
alchemy logo.pcx -g1 ---t 255 0 0
```

Write out a multi-image GIF file called output.gif, using the files image00.gif through image99.gif (this example will write out a local palette for each image):

```
alchemy image??.gif -g1 ---U output.gif
```

Do the same thing, but force Alchemy to write out a global palette (this assumes that all of the input gif images contain an identical palette):

```
alchemy image??.gif -gl ---U output.gif -F
image00.gif
```

If the images contained different palettes, but you still wanted a single, global palette, you could use the Multi-Image Palette output option to generate a .pal file and then match each of the images to that palette:

```
alchemy image??.gif -L temp.pal -c256
alchemy image??.gif -gl ---U output.gif -f
temp.pal
```

Write out a multi-image GIF file called output.gif, using the files image00.gif through image99.gif (this example will write out a local palette for each image), specifying a delay of 1 second between images, the loop will repeat indefinitely:

```
alchemy image??.gif -gl 100
---U output.gif
```

Do the same thing, but specify that the loop is to be displayed exactly once:

```
alchemy image??.gif -gl 100 1
---U output.gif
```

Intergraph

---r

The Intergraph file format was developed by Intergraph Corp. and is used by them.

Syntax

---r [*compressionType*]

Parameter

compressionType:

9:Run Length Compressed, black and white

24:Group 4 compressed, black and white

The default is 24.

Extensions

.prn

Creator

Intergraph Corp.

Used by

Intergraph Corp.

Variations

Read and writes 1 bit black and white images.

Limitations

If you need to read or write other type Intergraph files please contact us; we are happy to add support for any of the Intergraph formats.

Example

Convert page.tif to an Intergraph file

```
alchemy page.tif ---r
```

JPEG/JFIF

-j

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

Syntax

`-j[coding] [quality [passes]]`

Parameters

coding:

Specify the type of entropy coding to perform.

none:default Huffman coding

h:optimum Huffman coding

The default is defaultHuffman coding.

quality:

1 through 100 (larger is higher quality)

The default quality is 32.

passes:

1:write one pass JPEG files

4:write four pass JPEG files

The default number of passes is 1.

Extension

.jpg

Creator

Joint Photographic Experts Group (JPEG)

Used by

WWW

Everyone else storing 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

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.

Four pass files are useful when generating JPEG files which are going to be displayed on the WWW. A four pass file allows a rough preview of the image to be displayed quickly when using a browser which supports multi-pass JPEG files.

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 photo.tga -j70
```

Convert the file photo.tga to a JPEG file called photo.jpg, using a low quality setting and generating optimum Huffman tables:

```
alchemy photo.tga -jh10
```

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

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

OS/2 Icon

--O

OS/2 Icon files are used by IBM OS/2.

Syntax

--O *outputType* (Uppercase letter o)

Parameter

outputType :

0:OS/2 2.0 and Warp

1:OS/2 1.2

The default is OS/2 2.0 and Warp.

Extension

.ico

Creator

IBM Corp.

Used by

OS/2

Variations

Reads and writes 1, 4, 8, and 24 bit RGB files.

Comments

OS/2 1.2 Icon files are older version files which are supported because some OS/2 software cannot read current OS/2 Icons.

Examples

Convert the image icon.bmp to an OS/2 Icon file called program.ico:

```
alchemy icon.bmp program.ico --O
```

PNG (Portable Network Graphics)

---n

PNG files are used by the WWW.

Syntax

---n [*outputType*]

Parameter

outputType:

- 0:None
- 1:Sub-filtering
- 2:Up-filtering
- 3:Averaging
- 4:Paeth filtering
- 9:Adaptive filtering

- 0:Standard
- 10:Interlaced

The default is Paeth filtering, non-interlaced. Options are combined by adding.

Extension

.png

Creator

The PNG development group

Used by

WWW

Variations

Reads and writes 1 bit black and white, 2, 4, and 8 bit grayscale, 2, 4, and 8 bit paletted, and 24 bit colour images.

With and without alpha channels.

Writes files with transparency information.

Limitations

Non-image chunks (such as copyright information) are discarded when reading.

Comments

When writing a PNG file with transparency there are several different modes that are used depending on whether the file is paletted, grayscale, or true colour. In the case of true-colour images specifying the transparent colour using the `---t` option (see below) causes that value to be treated as transparent. When writing a paletted or grayscale file the nearest match to the colour specified is made transparent.

Related options

`-I` Include Alpha channel
`---t` Transparency

Examples

Convert the JPEG file `sample.jpg` to a PNG file:

```
alchemy sample.jpg ---n
```

Convert the JPEG file `sample.jpg` to a PNG file, using red as the transparent colour:

```
alchemy sample.jpg ---n ---t 255 0 0
```

Spaceward Graphics

---S

Syntax	<code>---s <i>compressionType</i></code>
Parameter	<i>compressionType</i> : 0:None 1:Compressed The default is None.
Extensions	<code>.r</code> Red channel image data <code>.g</code> Green channel image data <code>.b</code> Blue channel image data <code>.a</code> Alpha channel image data [optional]
Creator	Spaceward Graphics
Used by	Spaceward Graphics
Variations	Writes 1 bit black and white, 8 bit grayscale, 8 bit paletted, and 24 bit colour images.
Comments	Alpha channel data can be read and written by using the <code>-I</code> option; see the appropriate section in the User's Manual for more information.
Related options	<code>-I</code> Include Alpha channel information
Examples	Convert the JPEG file <code>sample.jpg</code> to a Spaceward file: <pre>alchemy sample.jpg ---s</pre> Convert the Targa file <code>apple.tga</code> to a Spaceward file, include the alpha channel information: <pre>alchemy apple.tga -I ---s</pre>

US Patent Image

---P

Used by the US Patent and Trademark Office to store and distribute patent data.

Syntax

---P *compressionType*

Parameter

compressionType:
0:Group 3 compressed
1:Group 4 compressed
The default is Group 4 compressed.

Extensions

.pat

Creator

US Patent and Trademark Office

Used by

US Patent and Trademark Office

Variations

1 bit black and white images.

Limitations

Must be black and white

Comments

Alchemy can write multi-page US Patent files when used with the ---U option; see below for an example

Related options

---U Write multi-page file

Examples

Convert the TIFF file page1.tif to a US Patent file:

```
alchemy page1.tif ---P
```

Convert all the pages in the TIFF file pages.tif to a multi-page US Patent file:

```
alchemy pages.tif ---P -U ---U
```

Other Options

Introduction

This section describes options which are not file format output options.

Multi-Page Input

-U

Purpose

Allow the conversion of multiple pages with a single execution of Alchemy.

Syntax

`-U [firstPage [lastPage]]`

Parameters

firstPage:

Specifies the first page to be converted.

lastPage:

Specifies the last page to be converted.

Comments

The multi-page option allows you to process multiple pages of an image when reading an image file which contains multiple pages. Each page of the image can be written to a separate file or to a single multiple-page output file (if writing to a format which supports multiple pages and using the `---U` option). If writing separate files the output file names will be as specified, with the extension replaced with `.001` for first page, `.002` for the second, and so on.

When using the multi-page option in conjunction with the wildcard option the number of output files generated can be very large.

Limitations

The only file formats which support multi-page reading are: TIFF, DCX (PCX), PCL, and US Patent and in the case of Alchemy PS, PostScript and PDF.

Examples

Convert all the pages in the PCL file `doc.pcl` to TIFF files:

```
alchemy doc.pcl -U -t
```

Convert all the pages of all the TIFF files to PCX files, placing the output files into the directory \images\output:

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

Do the same thing, but write a single multi-page DCX variant PCX file, called doc.pcx:

```
alchemy *.tif -U -p1 ---U doc.pcx
```

Multi-Page Output

---U

Purpose	Allow the output of image files which contain multiple pages.
Syntax	---U [<i>filename</i>]
Parameter	<i>filename</i> : Specifies the output filename when writing a single multi-page file (see the Examples section below for more information).
Comments	<p>The multi-page output option allows you to write multiple images or pages to a single image file.</p> <p>Because it is possible to write a single multi-page file based on multiple single pages files, a single multi-page file, or multiple multi-page files and it is also possible to write multiple multi-page files based on multiple multi-page input files there are four permutations to consider. See the examples section below for details.</p>
Limitations	Only certain image file formats allow multiple images in a single file: Adobe PDF, PCX (the DCX variant), GIF, TIFF, US Patent, and HP PCL.
Examples	<p>Convert all the pages in the TIFF file doc.tif to a multi-page PDF file (note that in this case the -U option is needed to cause Alchemy to read all the pages in the doc.tif file, without it only the first page would be read):</p>

```
alchemy doc.tif -U ---U --d
```

Convert the TIFF files `page1.tif`, `page2.tif`, and `page3.tif` to a multi-page PDF file called `output.pdf` (the `--` option is required to tell Alchemy that there are multiple files being read and the filename after the `---U` option is required to specify the output file name):

```
alchemy -- page1.tif page2.tif page3.tif
      ---U output.pdf --d
```

Convert the multi-page TIFF files `doc1.tif`, `doc2.tif`, and `doc3.tif` each to their own multi-page PDF files (this assumes that each TIFF file is a multi-page file). The `-U` option is required to tell Alchemy to treat the input files as multi-page documents. In this case there is no filename specified after the `---U`, because a single multi-page output file is not being written:

```
alchemy -- doc1.tif doc2.tif doc3.tif -U
      ---U --d
```

Convert the multi-page TIFF files `doc1.tif`, `doc2.tif`, and `doc3.tif` into a single multi-page PDF file (this assumes that each TIFF file is a multi-page file). In this case there is a filename specified after the `---U`, because a single multi-page output file is being written:

```
alchemy -- doc1.tif doc2.tif doc3.tif -U
      ---U docs.pdf --d
```

Only scale if too large

--+

Purpose

Causes Alchemy to only scale images down.

Syntax

--+

Comments

This command can be useful if you have a variety of images and want to scale them all to be no larger than a certain size. If this command is not used all of the images that are smaller will be scaled up.

Example

Scale all the GIF files in the current directory to be no larger than 640 x 480, preserving aspect ratio and placing the output files in the directory called new:

```
alchemy -- *.gif new\ -Xb640 -Yb480 -- --+
```

Override Input Type

-=

Purpose

Force Alchemy to treat the input file as the specified file type. This can be used if Alchemy cannot identify or misidentifies the format of an input image.

Syntax

`-= inputType`

Parameter

The *inputType* must be a valid number identifying a supported format. The *inputTypes* are as follows:

ADEX	24	Macintosh PICT	10
Adobe PDF	65	MacPaint	49
Adobe Photoshop	74	MTV	17
Alias PIX / Vivid IMG	16	OS/2 BitMaP	55
Alpha Microsystems BMP	42	OS/2 Icon	58
Autologic	28	PCPaint/Pictor	29
AVHRR	43	PCX	9
Binary (BIF)	31	PDS	37
Calcomp	50	PhotoCD	56
CALS	41	PNG	75
Core IDC	66	Portable BitMap (PBM)	13
Cubicomp PictureMaker	44	PostScript	14
Dr. Halo CUT	45	Puzzle	51
Epson Stylus	72	Q0	21
ER Mapper Raster	59	QDV	18
Erdas LAN/GIS	19	QRT Raw	20
Fargo Primera	69	RIX	38
First Publisher Art	46	SGI Image	11
Freedom of Press	25	Spaceward Graphics	76
Gem VDI Image	22	Spot Image	39
GIF	1	Stork	32
GOES	40	Sun Icon	52
Hitachi Raster	63	Sun Raster	8
HP PCL	15	Targa	6
HP-48sx Graphic Object	60	TIFF	4
HSI JPEG	30	US Patent Image	73
HSI Palette	3	Utah RLE	23
HSI Raw	5	Verity Image Format	70
IBM Picture Maker	48	VITec	64
IFF/ILBM	7	Windows BitMaP	12
Img Software Set	61	Word Perfect Graphic	27
Intergraph	77	X BitMap (XBM)	35
Jovian VI	36	X PixMap (XPM)	47
JPEG	2	XWD	33
Lumena CEL	62		

Comments

Rarely will Alchemy misidentify a file; the file is usually damaged in some way when this happens. If the file is damaged, or if you specify an input type that does not correspond to the actual type of the image, the results will be unpredictable. If you have a file which Alchemy misidentifies but is otherwise undamaged please contact us.

Example

Convert the file `unknown.xxx` to an OS/2 Bitmap file called `output.bmp`, forcing `unknown.xxx` to be treated as a Sun Raster image:

```
alchemy unknown.xxx output.bmp -O -=8
```

Preserve palette while scaling

---f

Purpose

Keep the original palette when scaling paletted images.

Syntax

---f

Comments

When using type 'b' or better scaling on paletted images Alchemy has to convert the image to true colour as part of the scaling process and then convert the image back to paletted before saving. Ordinarily the best results are obtained if Alchemy is allowed to choose the final palette based on the scaled image content. However there may be times when you wish to preserve the original palette instead of generating a new one. This option does that.

Examples

Scale the gif file flowers.gif to 320 x 200, preserving the original palette:

```
alchemy flowers.gif new.gif -g ---f
-Xb320 -Yb200
```

Scale all the files ending in .gif to 320 x 200, preserving the original palette, placing the new files in the directory new:

```
alchemy *.gif new\ -g ---f -Xb320 -Yb200
```

Set Horizontal DPI

--X

Purpose

Set the horizontal DPI based on a final image size.

Syntax

`--X size[units]`

Parameters

size:

The size of the output image in the horizontal dimension.

units:

The units the size parameter is in:

i:inches

c:centimeters

units is optional; the default is inches. The units value must immediately follow the size parameter.

Comments

This command is similar to the `-D` command, except that the DPI value is set based on the size specified and the number of pixels in the image.

To set the Vertical DPI use the `--Y` command (described below).

If you want to preserve the aspect ratio in the image use the `++` command. Using both the `--Y` and `++` commands will cause the dimensions given to be treated as a bounding box.

Related options

`--Y` Set Vertical DPI

`++` Preserve aspect ratio

Examples

Set the horizontal DPI value so that the image `test.tif` is converted to an EPS file that will be 5 inches across when printed:

```
alchemy test.tif -e --X 5i
```

Do the same thing, but preserve the aspect ratio (so the vertical DPI will be set to the same value as the horizontal DPI):

```
alchemy test.tif -e --X 5i --
```

Do the same thing, but set the vertical size to 6 inches, this will cause the 5 in x 6 in dimension to be treated as a bounding box:

```
alchemy test.tif -e --X 5i --Y 6i --
```

Set Vertical DPI

--Y

Purpose	Set the vertical DPI based on a final image size.
Syntax	<code>--Y size[units]</code>
Parameters	<p><i>size</i>: The size of the output image in the vertical dimension.</p> <p><i>units</i>: The units the size parameter is in: i:inches c:centimeters <i>units</i> is optional; the default is inches. The units value must immediately follow the size parameter.</p>
Comments	See the --X option above.
Related options	<code>--x</code> Set horizontal DPI <code>--+</code> Preserve aspect ratio
Examples	See the --X option above, for examples.

Slide show (MS-DOS only)

--

Purpose	Automatically change from one image to the next when displaying images.
Syntax	-- [<i>delay</i>]
Parameters	<i>delay</i> : Delay between images in seconds (for example, 0.5 is 1/2 second). Negative numbers indicate that the slideshow repeats. The default is 0, no delay between images.
Comments	The slide show mode command does not reset the SVGA card between images, this has been found to work on most SVGA cards. Use --- to force a reset between images (robust slide show mode). Robust slide show mode causes some monitors to have to resync between images.
Examples	Display the images flower.gif, puppy.gif, and kitten.gif in a slideshow, repeating the loop and pausing 5 seconds between images: <pre>alchemy -- flower.gif puppy.gif kitten.gif -v -- -5.0</pre> Do the same thing, but use robust slide show mode: <pre>alchemy -- flower.gif puppy.gif kitten.gif -v --- -5.0</pre>

Transparency

---t

Purpose

Specify which colour in the output image is considered to be transparent. Note that transparency is only supported by certain file formats.

Syntax

`---t [red green blue]`

Parameters

red green blue:

Specifies the colour to use for the transparent colour (0 0 0 is black, 255 255 255 is white). The default is 255 255 255 (white).

Comments

Only supported by GIF89A and PNG. See those file formats for more information.

Examples

Convert the image logo.tif to a GIF file, specifying white as the transparent colour:

```
alchemy logo.tif -g1 ---t
```

Convert the image logo.tif to a GIF file, specifying black as the transparent colour:

```
alchemy logo.tif -g1 ---t 0 0 0
```

Do the same thing, this time using red as the transparent colour:

```
alchemy logo.tif -g1 ---t 255 0 0
```

Use input file format for output

--=

Purpose	Causes Alchemy to write out a file in the same format as the file being read.
Syntax	--= <i>compressionOption</i>
Parameter	<i>compressionOption</i> : Sets the output compression or output type based on the format being written.
Comments	If the output file already has the extension that Alchemy uses for the file being read and the output file is not being written to a different directory this command will fail unless the --o command is used.
Limitations	Only one parameter can be specified (i.e. if writing a multi-page GIF file the delay between images and the repeat count cannot be specified when using this option).
Examples	<p>Convert all the files called image.* to files with the same format, flipping the files and placing the output in a directory called \flipped:</p> <pre>alchemy image.* --= --^ \flipped</pre> <p>Do the same thing, but replace the existing files with the new files instead:</p> <pre>alchemy image.* --= --^ --o</pre> <p>Do the same thing, using type 1 compression (what that actually means depends on the file format being written):</p> <pre>alchemy image.* --= 1 --^ --o</pre>

Use input filename for output

--o

Purpose

Causes Alchemy to write out a file with the same name as the file being read, replacing the input file.

Syntax

--o

Comments

If the input file is read only this command will fail.

Example

Convert the GIF file, test.gif, to a GIF file, scaling it to 640x480 and keeping the same name:

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

Use 3 letter extensions

--3

Purpose

Causes Alchemy to use 3 letter extensions (this is the default operation on MS-DOS).

Syntax

--3

Comments

Ordinarily Alchemy will use the extension specified by the image file format (.GIF, for example), however some file formats, such as TIFF, specify that on systems which allow it the extension should be more than 3 letters (.TIFF, in the case of TIFF). This can be a problem if you are interchanging files with an MS-DOS system. This option causes Alchemy to always use no more than 3 letter extensions.

Example

Convert the file large.gif to a TIFF file with a 3 letter extension:

```
alchemy large.gif -t --3
```


Configuring DOS/4GW Pro

MS-DOS Only

DOS/4GW Pro is the DOS Extender used by Image Alchemy/386 and Image Alchemy PS. The DOS Extender is the software that allows protected mode software to run under MS-DOS. This chapter explains how to control various operations of the DOS Extender to optimize use of your computer.

Updating from previous versions

If you have been using earlier versions of Image Alchemy and have made use of the DOS4GVM environment variable you will have to make changes to that variable. In particular the old style DOS4GVM=1 setting is no longer supported (it isn't needed, since that is now the default operation) and all '#' symbols must be changed to '!'.

Virtual Memory

The Virtual Memory Manager (VMM) uses a swap file on disk to augment RAM. With VMM you can use more memory than your machine actually has. When RAM is not sufficient, a portion of memory is swapped out to disk until it is needed again. The combination of the swap file and available RAM is called *virtual memory*.

Image Alchemy automatically uses virtual memory if you do not have enough physical memory available. However, there may be cases where you wish to change the behavior of virtual memory. This is done through the DOS4GVM environment variable.

To set the DOS4GVM environment variable, use:

```
set DOS4GVM=[option [:value]] ...
```

(A ":" is used with options that take values instead of "=" since the DOS command shell does not allow "=" to appear in environment variables.)

DOS4GVM Options

DeleteSwapFile:[On:Off] Default: On
By default the swap file is deleted when Alchemy exits. Program startup is quicker if the file is not deleted.

PhysMin:n [K|M] bytes Default: 1024K
The minimum amount of RAM managed by VMM.

PhysMax:n [K|M] bytes Default: 65M
The maximum amount of RAM managed by VMM.

SwapFileName:[path][filename]
The default name is "DOS4GVM.SWP". The default path is the executable directory.

SwapInc:n [K|M] bytes Default: 4096K
The size by which the swap file grows.

SwapMin:n [K|M] bytes Default: 0
The minimum or initial size of the swap file.

VirtualSize:n [K|M] bytes Default: 16M
The size of the virtual memory space.

Changing the Defaults

To change the default values set the DOS4GVM environment variable. For example:

```
set DOS4GVM=VirtualSize:64M PhysMax:16M
```

Sets up a swap file up to 64 megabytes in size after using up to 16 megabytes of RAM.

