

Image Engineer

COLLABORATORS

	<i>TITLE :</i> Image Engineer		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 14, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Image Engineer	1
1.1	Image Engineer Documentation	1
1.2	Introduction	2
1.3	Copyrights	4
1.4	Disclaimer	4
1.5	Distribution	5
1.6	Motivation	5
1.7	System Requirements	6
1.8	Installation & Starting Up	6
1.9	Interface	7
1.10	Tutorials	8
1.11	Loading, Rendering and Saving an Image	8
1.12	Scaling and Locking an image to a palette	10
1.13	Filtering and Removing Noise from an Image	11
1.14	Applying a Vignette effect using an Alpha channel	11
1.15	Menus	12
1.16	About	14
1.17	Open Submenu	14
1.18	Open As	15
1.19	Open	15
1.20	Open Clipboard	16
1.21	Open Device	16
1.22	Save Submenu	17
1.23	Save	18
1.24	Save to Clipboard	19
1.25	Save Device	19
1.26	Save Data Submenu	19
1.27	Save Data	20
1.28	Save Data to Clipboard	21
1.29	Save Data to Device	22

1.30 Close Project	22
1.31 Close	22
1.32 Module Window	23
1.33 ARexx Window	23
1.34 Infobar Window	23
1.35 Preferences submenu	24
1.36 Font Preferences	24
1.37 General Preferences	24
1.38 menusprefs	26
1.39 Open Device Preferences	27
1.40 Save Device Preferences	27
1.41 SVObject Preferences	27
1.42 Screen Mode	28
1.43 Recalculate Palette	28
1.44 Quit	29
1.45 Quit	29
1.46 Project Info	30
1.47 Get Project Info	30
1.48 Get Project List	31
1.49 Set Project variables	31
1.50 Project Name	32
1.51 Use as Secondary/Alpha	32
1.52 Mark As	33
1.53 Zoom	33
1.54 Render	34
1.55 Render Control	34
1.56 Render	36
1.57 Render Colours	36
1.58 Render Depth	37
1.59 Render Device	38
1.60 Render Dither	39
1.61 Render Get Options	40
1.62 Render Palette	41
1.63 Render Screen Mode	42
1.64 Set a project's Render Options	43
1.65 Close Render	43
1.66 Render Close	44
1.67 Palette Submenu	44
1.68 Edit Palette	44

1.69 Load Palette	45
1.70 Load Palette	45
1.71 Save Palette	46
1.72 Palette Save	46
1.73 Render to Front	47
1.74 Execute Module	47
1.75 Execute ARexx script	48
1.76 Enter Co-ordinates for the Crop Box	48
1.77 Cut or Crop	48
1.78 Crop	48
1.79 Keyboard short cuts	49
1.80 Image Processing Effects Modules	50
1.81 ARexx Support Module	51
1.82 Input Form	52
1.83 Get Dir	55
1.84 Get File	56
1.85 Get Files	57
1.86 Request	58
1.87 Get Number	58
1.88 Get String	59
1.89 Autocrop	60
1.90 Autocrop	61
1.91 Brightness	61
1.92 Brightness	62
1.93 Bulge	63
1.94 Bulge	64
1.95 Colour Filter	64
1.96 Colour Filter	65
1.97 Composite	66
1.98 Composite	68
1.99 Contrast	69
1.100 Contrast	70
1.101 Contrast Stretch	71
1.102 Contrast Stretch	71
1.103 Convert to Colour	71
1.104 Convert to Colour	72
1.105 Convert to Grey	72
1.106 Convert to Grey	73
1.107 Convolve	73

1.108	What the smeg is a convolve?	74
1.109	Convolve	75
1.110	Displace	76
1.111	Displace	77
1.112	Histogram Equalization	77
1.113	Histogram Equalization	78
1.114	False Colour	78
1.115	False Colour	79
1.116	Gamma	79
1.117	Gamma	80
1.118	Halftone	81
1.119	Halftone	81
1.120	HighBoost	82
1.121	HighBoost	82
1.122	HighPass	83
1.123	HighPass	84
1.124	Histogram	84
1.125	What's a Histogram anyway?	85
1.126	Hue	87
1.127	Hue	87
1.128	Hue Mask	88
1.129	Hue Mask	89
1.130	Impress	89
1.131	Impress	90
1.132	Local Contrast Stretch	91
1.133	Local Contrast Stretch	91
1.134	LowPass	92
1.135	How exactly does it work?	92
1.136	LowPass	93
1.137	Median Filter	93
1.138	Maximum Filter	94
1.139	Median Filter	94
1.140	Minimum Filter	95
1.141	Negative	95
1.142	Negative	96
1.143	Noise	96
1.144	Noise	97
1.145	Posterize	98
1.146	Posterize	99

1.147Reflect X	99
1.148Reflect X	100
1.149Reflect Y	100
1.150Reflect Y	101
1.151Remove Feature	101
1.152Remove Feature	102
1.153Resize	103
1.154Resize	103
1.155Ripple	104
1.156Ripple	105
1.157Rotate	106
1.158Rotate	106
1.159Saturation	107
1.160Saturation	108
1.161Scale	108
1.162Scale	109
1.163Sharpen	110
1.164Sharpen	110
1.165Threshold	111
1.166Threshold	111
1.167Twirl	112
1.168Twirl	113
1.169ARexx	113
1.170Script Writing Tips	114
1.171Supplied Scripts & Macros	116
1.172ARexx Commands	119
1.173Changes from 2.1	124
1.174Get	124
1.175Get File Type	125
1.176Image Engineer to Front	125
1.177Last Error	126
1.178Get Task Priority	127
1.179Set Task Priority	128
1.180WB to Front	128
1.181Use with Martin Apel's VMM	129
1.182Hints, tips & FAQ	129
1.183register	130
1.184Reporting Bugs	133
1.185Thanks and Greetings	133

1.186 On-line Support	134
1.187 The Future	135
1.188 History	135
1.189 Bibliography	142
1.190 Index	143

Chapter 1

Image Engineer

1.1 Image Engineer Documentation

Image Engineer v3.41 (Wislon release 2)

Copyright © 1999 by Wislon
All rights reserved

This program uses the Superview library by Andreas R. Kleinert,
and the BGUI Library by Jan van den Baard and Ian J. Einman and
BGUI Developers Team.

Introduction & Features

Copyrights

Disclaimer

Distribution

Motivation

System Requirements

Installation & Starting Up

User Interface

Tutorials

Menus

Keyboard Short Cuts

Effects Modules

ARexx Commands

Use with VMM

Hints, tips & FAQ

How to Register

Reporting Bugs

Thanks and Greetings

On-line Support

The Future

History

Bibliography

Index

Contact Wision by email <marko.seppanen@wwnet.fi>

1.2 Introduction

Introduction

~~~~~

Image Engineer is a shareware image processing application. Image Engineer can be used for tasks varying from converting images between different file formats, rendering 24 bit images down to standard Amiga screen modes, enhancing badly scanned images, applying special effects even up to advanced image composition. What you can use it for is basically limited by what you can think of.

Features

~~~~~

- * Very fast, coded in 98% 68020 assembler
- * CyberGraphX support
- * Multiple image editing
- * Dithered Colour or grey preview images.
- * Zoom in preview images.
- * Processing done in 8 bit grey or 24 bit colour
- * Extensive ARexx support and many supplied effects scripts
- * Thumbnail previews shown for most effects.
- * Font sensitive, resizable GUI with keyboard shortcuts, using the BGUI Library.
- * User configurable menus.
- * Uses superview.library for loading and saving. superview.library currently supports:-

IFF-ACBM	- Windows, OS/2 BMP	- PNM (PBM, PGM, PPM)
IFF-ILBM	- JPEG (JFIF v6)	- IMG (GEM Graphics)
IFF-DEEP	- PhotoCD	- WPG (WP Bitmaps)
IFF-FAXX	- TIFF	- MacPaint MAC

```

IFF-PBM           - PCX           - EPS (Header/PS)
IFF-RGB8          - Atari Degas    - C64 (Koala, Doodle)
IFF-RGBN          - Spectrum ZX    - TIM
IFF-RGFX          - XWD           - FBM
IFF-YUVN          - Windows Icon   - CDR-Previews
QRT/POV Raytrace - Mac PICT-2     - SGI
SVG               - Targa         - Pictor/PCPaint
SunRaster         - UtahRLE       - PNG
CVP (passphoto)  - Amiga Icons (WB/Magic/New)

```

as well as all OS3-Datatypes
and 24 Bit DataTypes (picture.datatype V43).

Many formats with 24 Bit-Support.
Reads XPK- and PP20-packed data via Unpack.svobject.

Writes C-Sourcecode in chunky/RGB notation.

Together with external third party programs also the
following file formats seamlessly can be embedded:

File Format	needs program	also needs support for
- AMF	MetaView	(ILBM)
- CLP	MetaView	(ILBM)
- DXF/AutoCAD (tm)	MetaView	(ILBM)
- FIG	AmiFIG	(Postscript)
	or: MetaView	(ILBM)
- CGM	GPlot	(Postscript)
	or: MetaView	(ILBM)
- IFF-DR2D	MetaView	(ILBM)
- IFF-DRSD	MetaView	(ILBM)
- Limbo	Limbo 4.0	(PNM)
- Postscript (TM)	Ghostscript	-
- WPG-Vector	MetaView	(ILBM)
- WMF	MetaView	(ILBM)

Devices (via SVOjects):

```

- Scanner (scanner.device of ScanQuix)
- Printer (printer.device/TurboPrint/Studio)

```

- * Can display images using superview.library. superview.library currently supports ECS, AGA, EGS7, OpalVision, Retina, CyberGraphics, PicassoII, Merlin, Picasso96, AGA/RTG Window, CgxOverlay
- * Supports AGA where available.
- * Render images to standard Amiga screen modes including HAM6 and HAM8.
- * Dither images using Floyd-Steinberg, Burkes, Stucki, Sierra, Jarvis and Stevenson-Arce error diffusion algorithms.
- * Support for Petri Nordlund's Executive UNIX like task scheduler.
- * Image processing effect implemented as disk based modules. Modules exist for:-

```

Autocrop           Hue
Brightness         Local Contrast Stretch

```

Bulge	Lowpass filter
Composite (with alpha channel)	Contrast
Median filter	Contrast Stretch
Convert To Colour	Negative
Convert To Grey	Posterize
Convolve (user defineable)	Reflect X
Displace (with alpha channel)	Reflect Y
Histogram Equalisation	Remove Feature
False Colour	Resize
Gamma	Rotate
Halftone	Saturation
HighBoost filter	Scale
Highpass filter	Sharpen
Histogram	Threshold
Twirl	Colour Filter
Ripple	HueMask
Noise	Impress

1.3 Copyrights

Copyrights

~~~~~

Image Engineer is Copyright © 1999 Wision, Copyright © 1995-99 Simon Edwards. All rights reserved. All of the image processing effect modules included in this distribution are Copyright © 1999 Wision, © Copyright 1996 Simon Edwards. All rights reserved.

SuperView Library is © Copyright 1993-99 by Andreas R. Kleinert. All rights reserved. SuperView Library is Freeware (product-specific Licenseware). Any usage from and by other programs of SuperView Library without an explicit license is strictly forbidden. (See SuperView Library documentation for more information).

BGUI library is © Copyright 1996-1999 BGUI Developers Team, © Copyright 1996 Ian J. Einman, © Copyright 1993-1996 Jaba Development/Jan van den Baard. All Rights Reserved.

### 1.4 Disclaimer

#### Disclaimer

~~~~~

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

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

Have a nice day. :)

1.5 Distribution

Distribution

~~~~~

The unregistered version of Image Engineer is freely distributable. It is not public domain, as all copyright remains with the author. This means that you may copy and distribute Image Engineer provided the following conditions are met.

- \* Original archives are unmodified.
- \* No profit is made beyond minimal copy and posting fees.
- \* Image Engineer may not be commercially distributed without the author's written permission.
- \* Including Image Engineer on magazine's coverdisk is allowed, if author is informed about it.

## 1.6 Motivation

Motivation (or "What could drive a person to do such a thing?")

~~~~~

"About half way through last year I read the book "Digital Image Processing" by Gonzalez & Woods. After that I wanted to try out some of the techniques and algorithms (I was also looking for some sort of programming project to do) so I sat down with the intention of writing a small program to do some basic image processing on 8 bit grey scale images. I also had the idea of releasing something as freely distributable planted in my mind. So once I had my little program up and running (even though it couldn't do much), my summer break (December-February, for the benefit of those in the northern hemisphere) was coming up. It was about now that I decided to write a full blown 24 bit colour image processing program, I figured that would keep me busy over the long months stuck at home in the middle of nowhere. The rest

is pretty well history..."

- Image Engineer 1.0 documentation.

Well that was over a year ago, during that time Image Engineer has reached version 2.1 with many more features added and just as many bug fixes. Feedback has generally been very positive with many people giving credit as well as asking for new features and urging me to keep on going with IE, and that's exactly what I've been doing. The most notable change that IE 3.0 brings is an entirely rewritten and redesigned user interface. After reading Alan Cooper's book "About Face, The Essentials of User Interface Design" (which I strongly recommend to anyone involve with software) I decided that IE's interface basicly sucked and needed to be redone. Hopefully I've succeeded. Also you will have noticed that IE is now shareware with the unregistered version limiting the size of images that can be processed. A few people have mailed me pointing out that if it would further future development, they would be willing to pay a fair amount for IE's features. I think you'll find that for the features of IE, the shareware fee is very reasonable.

Simon Edwards
5/5/96

1.7 System Requirements

System Requirements

~~~~~

- \* 68020 or higher processor from 680x0 -serie.
- \* OS v2.04+
- \* 2Mb of RAM bare minimum, at least 6Mb recommended, infinity is best.
- \* Superview Library V15 or higher to be installed. (The latest version can be found inside the archives of SViewIV available on aminet: /gfx/show/SvIV-\*.lha). Download archives numbered 1-2, 3a, 3b, 4-8. Other archives are optional.

Listen me carefully now, Image Engineer needs libraries, not the SviewIV itself. Sublibraries should be in Libs:svobjects/. There IS an installer, which installs ONLY these libraries to the right place. Use it. There is no need to register SViewIV to be allowed to make use of the library.

- \* 640x400 screenmode or higher.

## 1.8 Installation & Starting Up

Installation & Starting Up

~~~~~

Once you have decompressed that archive (which you obviously already have), click on the installer script, the script will then install Image Engineer on to your hard disk, and check your system to make sure that all of the needed files are in the right place.

Users of FastIPrefs, MCP and other patches please read the hints and tips

.

Image Engineer (IE) can be started by clicking on its icon from the workbench or entering "IE" from the Shell. IE currently takes no arguments from the Shell. When starting up IE will try and load it's Prefs from S:IE.config. It will then put up a screen mode requester allowing you to choose the screen mode and dimensions that you want IE to operate in (the screen must be atleast 640 width by 400 high). (IE can be made to open its screen without asking, see Prefs PREFS}. Once it has finished opening it's screen you'll be left looking at a very unexciting screen. The Project menu will be the only menu currently available.

If you're not familiar with IE I suggest that you work through (or at least skim through) the

tutorials
section to

acquaint yourself with how IE does things. Once you've gone through the tutorials, to learn more I recommend that you simply play and experiment. Don't be intimidated by some of the technical sounding menu items. ;)

Website of Visual Engineering also has tutorials for both for novice and professional.

1.9 Interface

About the User Interface

~~~~~

Keyboard short cuts for buttons are denoted by an underscore under letter in the button's text. The tab key is used for switching between panels on multi-panel requesters. The key for the "Ok" button is always return, and cancel is always the escape key.

In the preview windows, the cursor keys and shift key can used to scroll around the image.

When the pointer is over a preview image, the co-ordinates and RGB and HSB value of the pixel under the pointer is shown in the title bar.

The the region used for cropping an image can simply be dragged out in the preview window in the usual way.

The next/previous preview window can be made active by using the [ and ] keys.

All of the requesters and windows remember their positions and contents between invocations.

## 1.10 Tutorials

```

~~~~~
Tutorials
~~~~~

#1 Loading, Rendering and Saving an Image

#2 Scaling and Locking an image to a palette

#3 Filtering and Removing Noise from an Image

#4 Applying a Vignette effect using an Alpha channel
Websites of Image Engineer and Visual Engineering have more ←
tutorials
available for both novice and professional users.

```

## 1.11 Loading, Rendering and Saving an Image

Loading, Rendering and Saving an Image

Loading, rendering and saving are the three operations that you'll be doing with Image Engineer more than anything else.

1. IE distinguishes between two types of images, grey (8 bit) and colour (24 bit) images. Images loaded as 8 bit grey are automatically converted and stored in memory using 8 bits per pixel. Images loaded as 24 bit colour are automatically converted and stored in memory using 24 bits per pixel, 8 bits for each colour component (Red, green, blue). It pays to make sure that you load grey images as 8 bit, as they will take up less memory and all operations on them will be faster.

What images are treated as when loading is set on the Open As submenu on the Project menu.

```

+-----+
| Grey (8bit) | <-- These two options control whether images
| Colour (24 bit)| <-- are loaded as 8 bit grey or 24 bit colour.
+-----+

```

The image we're going to use is in colour, so now is the time to set IE to load images as 24 bit colour. Just select "Colour (24 bit)" so that it now has a tick beside it.

2. Select the "File..." menu item on the Open submenu. The file requester will now appear. Select the file molecules.iff which should be in



the Pics directory. IE will now identify, load, and convert the image to 24 bit colour. A window will now be opened on IE's screen and a grey preview image will be drawn.

This window is referred to as the project window, and the image is referred to as a project. The title bar of the project window shows the name of the project (each project has a unique name), followed by the x and y coordinates that the mouse pointer is currently over followed by the grey level at that point for grey images, or the Red Green Blue (RGB) value at that point for colour images.

Note: The co-ords are only shown if that window is the active one. The active window's title bar is generally darker than an inactive window's.

3. Each project has its own set of information associated with it that describes how it should be rendered.

Select the "Render Control..." menu item from the Image menu. The render control requester will now appear. From here you can everything about how this project should be rendered. What screenmode to use, how many colours to use, how to choose the palette, what dithering to use etc. For this tutorial we want to render it down to 32 colours Low-Res using Jarvis dithering. To do this, set the "Device:" gadget to "Amiga", now click on the "Choose..." button, a screen mode requester will come up, select a Low Res screenmode, (like "NTSC:Low Res" for example) then click on "Ok" to return to the render control requester. Move the "Colours" slider so that it says 32. Set the "Palette Size:" slider to 32 also. And finally, click on the "Palette:" gadget so that it says "Pick Best" and the "Dither" gadget so that says "Jarvis". When you're finished hit the "Ok" button.

Now all of the render options have been set up. Thankfully you can set what default values should be used for new grey and colour projects, so that you don't have set up the render options to something intelligent every time you load in a new image.

4. Select "Render" from the Image menu. This tells IE to actually render the image. An indicator entitled "Rendering..." will appear. When it says that it's half way through rendering, a new screen will open and IE will draw the image.

This screen is referred to as the project's render screen.

If you press the right mouse button, IE's screen will reappear. The project's render screen has not been closed, just moved behind all of the other screens (like the Workbench screen and IE's screen). Select "Render" from the Image menu again, since the project has render screen has already been created, the screen is just moved to the front of the display.

5. Once a project has been rendered it can then be saved to disk.

Select "File" from the "Save Render" submenu on the Project menu. The save File Format requester will appear. This is where you select what save format you would like the image saved as. Select "ILBM CmpByteRun1", this the standard Amiga image format, used by DPaint, Brilliance, and every other Amiga graphics program. Click on "Ok" to continue. Next is

---

a file requester allowing you to enter the file name to save the rendered image as, set it to molecules32col.iff. Click on "Save" now and IE will save the image.

Well, that's that. If you don't want to render the image down to a displayable screen mode, but instead wanted to save the 8 bit grey or 24 bit colour image data. You would save it using items on the "Save Data" submenu. This works the same way as the "Save Render" submenu, except that there is no need to render the image first.

To close the render screen use "Close Render" on the Image menu.

## 1.12 Scaling and Locking an image to a palette

Scaling and Locking an image to a palette  
~~~~~

In this tutorial will be how to scale an image down to the size of an icon and then render it using the Workbench palette.

1. Open up the picture molecules.iff as 24 bit colour.
2. Make the molecules.iff project window the active one. (Just click on the window's title, it's the active one if title has a darker background.)

Now select the "Scale..." menu item which is on the "Edit" menu. The scale requester will now appear. Click on the "Lock Aspect" check box (so that it shows a tick). Move the "% Width:" slider all the way to the left to 25%. The "% Height:" slider will also move to stay at the same level, in order to preserve the aspect ratio. Click on the "Method:" gadget till it shows "Colour Average". This means that the Colour Average method will be used to scale the image, this is slower than the fast method (also known as "Nearest Neighbour"), but produces better results. Now click on "Ok" to let it go to work.

3. Go to the "Render Control" requester for the small image and set the Device to "Amiga" and select a High Res screen mode. Set the "Colours" and "Palette Size" to 4. Set "Palette" to "Lock Palette", this forces to IE to use this project's current palette when rendering. Set the dither to "Floyd-Steinberg" dithering. Press "Ok" when finished.
4. Before we can render the image we need to load in the palette that we want the rendered with. Choose the menu item "Load..." on the "Palette" submenu on the Image menu. A file requester will let you choose the palette file to load. Load the file WB.palette which should be in the IE:Palettes directory.
5. Render the image.

Save the rendered image as molecules.icon.iff.

The saved image could now be imported into an icon editor (like IconEdit supplied with Workbench).

1.13 Filtering and Removing Noise from an Image

Filtering and Removing Noise from an Image

~~~~~

In this I'll demonstrate how Image Engineer's filtering functions can be used to remove noise from a corrupt image.

1. Open up the picture NoiseU2.iff from the Pics directory as "Grey (8 bit)".
2. As you can see the image has been corrupted such that it looks like U2 are on tour in a snow storm. Go to the "Filter" menu and select "Lowpass". This will bring up the "Lowpass" function. Set the "Width:" and "Height:" to 3. In the top left of the "Lowpass" requester is a thumbnail image of our U2 picture. On it is select box that can be moved with the mouse. To the right is shown a preview of the contents of the select box. Notice how all of the white noise has been blurred all over the image. Try different widths and heights. The image looks worse now than in the beginning. There is a better way. Click on "Cancel" when you're finished playing (or press the escape key).
3. Go to the "Filter" menu and select "Median". The "Median filter" requester works basically the same way as the "Lowpass Filter" requester. Set the "Width:" and "Height:" to 3. Notice how the filter has removed about all of the noise instead of just blurring it. Click on "Ok" to apply the filter to the whole image.
4. Go to the "Histograms..." menu item on the "Balance" menu for the filtered image. This shows the histogram of the image. The Histogram is a graph showing the relative frequency of each grey level in the image. The higher the bar the greater the frequency of that grey level in the image. Notice how the grey levels are clustered towards the dark (left) end of the graph and that the full range of the graph isn't being used. Click on the "Ok" to exit.
5. Select "Contrast Stretch" from the "Process" menu for the filtered image. This will increase the contrast of the image. Now go back to the histogram. The histogram will now be stretched to cover the whole range.

Now you could render and save the enhanced image to disk. When trying to enhance, remove noise and bring out detail in an image, it pays to try several approaches and then seeing which one gave the best result, as some images respond well to some filters, while others are simply degraded more. There are no hard rules.

## 1.14 Applying a Vignette effect using an Alpha channel

Applying a Vignette effect using an Alpha channel

~~~~~

Have you ever noticed that effect that photographers sometimes use that makes images appear blurred around the edges. This tutorial shows you how the same effect can be created using IE.

1. Open up the picture Face.HAM8.iff as 24 bit colour.
2. Apply a 7x7 (Width and height set to 7) lowpass filter to the image.
3. Open the picture Spherical2.alpha which should be in the Alpha directory, as grey (8 bit).
4. Make the blurred image active and select info from the Image menu. Now take note of the width and height of the blurred image.
5. Go to the Spherical image and scale it to the same size as the blurred image. (To do this you're going to have to go to the Scale requester and make sure that "Lock Aspect Ratio" is turned off, then enter the width and height into the gadgets at the top of the requester).
6. Now it's time to composite the images. We're going to use the blurred image as the secondary image, and the scaled grey image as the alpha channel. First go to the blurred image and select "Use as Secondary" from the "Image" menu. There should now be an 'S' before the project's title. Now, go to the scaled grey image and select "Use as Alpha" from the "Image" menu. There should be a 'A' in front of it's title. By doing all of this you will have specified which images are going to be used for what when we go to create the composite image.
7. Select "Composite..." from the "Alpha" menu. The "Composite" requester should now pop up. We want to combine the images using the Alpha channel, so click on the radio button for "Alpha Channel". "X:" and "Y:" should be set to zero. Now click on "Ok". It will go to work and produce the composite image.
8. Now Render the image as HAM6 or HAM8 to get a good look at it. Notice how the image is blurred around the edges, since the alpha channel was dark around the edges it meant that more of the blurred image should be used there when creating the final image. While at the parts of the image that correspond to the light parts in the alpha channel used more of the original image.

By using alpha channels to combine images, it's possible to create 1000's of interesting effects that would otherwise not be possible. (Instead of blurring the image, try converting it to grey, false colour etc.)

Also, steps 4 and 5 could have been skipped by using the Stretch Alpha to fit option in the Composite function which will automatically scale the Alpha image to the right size.

1.15 Menus

Menus

~~~~~

IE only has 2 set menus. All other menus are configured in with the Project->Prefs->Menus menu item.

Project

Image

---

```
About...

Info...          <A>I
-----
Project Name...

Open             »

    Use Secondary <A>2

Open As         »

    Use Alpha    <A>3

Save Render     »
-----

Save Data       »

Zoom            »
-----
Zoom In        <A>+

Close

Zoom Out       <A>-
-----

ARexx Window... <A>A

Render         <A>R

Module Window...<A>M

Render Control...<A>P

Infobar Window...

Palette        »
-----
Close Render   <A>T

Prefs         »
-----

Re-calc Palette <A>L

Enter Co-ords... <A>E
-----
Cut           <A>X

Quit         <A>Q
```

---

## 1.16 About

About

~~~~~

This just brings up a small requester giving some information like the name of the program, it's version, who the author is, etc.

1.17 Open Submenu

Open Submenu

~~~~~

```
+-----+
|File...  <A>O|
|Clipboard <A>V|
|Device...  |
+-----+
| <Most      |
|  Recently  |
|  Used      |
|  Directories> |
+-----+
```

The "File..." item lets you open an image from disk. A file requester will appear, letting you choose the file to open. Image Engineer will automatically identify the image format, load it and convert it to either 8 bit grey or 24 bit colour.

The "Clipboard" item opens an image from the system clipboard if it currently contains graphics.

The "Device" item opens an image from an external device. A list of available devices is displayed to choose from. The only available device to open images from is the Scanner device at the moment. See the Superview Library documentation for more information.

In the four spaces below the "File..." and "Clipboard" items, the names of the last four most recently used directories will be placed. Selecting one of these works in the same way as the "File..." item except that the file requester is automatically placed in the corresponding directory. This basically saves you the time and hassle of having to manually change the file requester between the directories containing your images.

ARexx Equivalent:

```
OPEN
,
OPEN_CLIPBOARD
,
OPEN_DEVICE
```

## 1.18 Open As

Open As Submenu

~~~~~

```
+-----+
|_/_ Grey (8 bit) |          "_/" is meant to be a checkmark. BTW.
|  Colour (24 bit)|
+-----+
```

"Grey (8 bit)" and "Colour (24 bit)" determine whether images will be loaded, converted and stored internally as either 8 bit grey or 24 bit full colour images. The check mark indicates what images will currently be loaded as. If you're working with grey or black and white images, you should have this set to Grey (8 bit), this way IE will use less memory and operations will be faster.

1.19 Open

Open

~~~~~

Synopsis:

```
OPEN <FileName> [<GREY/8/8BIT | COLOR/COLOUR/24/24BIT>]
```

Function:

Open an image from disk, and convert to 8 bit grey or 24 bit colour.

Inputs:

```
FileName          - Complete path and filename of the image to
                   load.
GREY/8/8BIT       - Specifies that the image should be loaded
                   as 8 bit grey.
COLOR/COLOUR/24/24BIT - Specifies that the image should be loaded
                   as 24 bit colour.
If no colour depth is specified, it defaults to whatever the user
has specified in the "Open As" menu.
```

Result:

Returns the name of the newly created project in RESULT.

Example:

```
/* Ask the user what image they want to process */
'GET_FILE "Select an Image to process" "Go!!!" /* Get a file */
if RC=5 then exit /* Exit if we were cancelled */
MyFile=RESULT

'OPEN "' || MyFile || '" COLOUR'
MyProject=RESULT
```

Menu equivalent:

Project/Open

## 1.20 Open Clipboard

Open Clipboard

~~~~~

Synopsis:

OPEN_CLIPBOARD [<GREY/8/8BIT | COLOR/COLOUR/24/24BIT>]

Function:

Open an image from the clipboard, and convert to 8 bit grey or 24 bit colour.

Inputs:

FileName - Complete path and filename of the image to load.
 GREY/8/8BIT - Specifies that the image should be loaded as 8 bit grey.
 COLOR/COLOUR/24/24BIT - Specifies that the image should be loaded as 24 bit colour.
 If no colour depth is specified, it defaults to whatever the user has specified in the "Open As" menu.

Result:

Returns the name of the newly created project in RESULT.

Menu equivalent:

Project/Open

1.21 Open Device

Open Device

~~~~~

### Synopsis:

OPEN\_DEVICE <FileType> [<GREY/8/8BIT | COLOR/COLOUR/24/24BIT>]

### Function:

Opens an image from the given device SVObject, and convert to 8 bit grey or 24 bit colour.

### Inputs:

FileType - The name of the device SVObject to read from. Currently the only available one is 'Scanner'.  
 GREY/8/8BIT - Specifies that the image should be loaded as 8 bit grey.  
 COLOR/COLOUR/24/24BIT - Specifies that the image should be loaded as 24 bit colour.



If no colour depth is specified, it defaults to whatever the user has specified in the "Open As" menu.

Result:

Returns the name of the newly created project in RESULT.

Menu equivalent:

Project/Open

## 1.22 Save Submenu

Save Render Submenu

~~~~~

```
+-----+
|File...  <A>S|
|Clipboard  |
|Device...  |
+-----+
| <Most     |
| Recently  |
| Used      |
| Directories>|
+-----+
```

Provided the current project has been rendered, this submenu will let you save it to a file or the system clipboard. When saving to a file you will be presented with a file requester from where you can enter the file name to save the project as. Next a small requester entitled "Select Save File Type" will let you select the file format that you want the project to be saved as. (If you're unsure as to what file format you should use, use "ILBM CmpByteRun1", the standard Amiga image format). The "Options..." button lets you edit any available options associated with a SVOBJECT/file type.

The "Device..." item writes the rendered image to a device SVOBJECT. A list of available device SVOBJECTS is displayed from which to choose from. Currently the only available device SVOBJECT is the preferences printer. See the Superview Library documentation for more information.

In the four spaces below the "Device..." item, the names of the last four most recently used directories will be placed. Selecting one of these works in the same way as "File..." item except that the file requester is automatically placed in the corresponding directory. This basically saves you the time and hassle of having to manually change the file requester between the directories containing your images. This list is also repeated on the "Save Data" submenu.

Note: If the rendered image is in HAM or HAM8, only use ILBM. As most other formats do not support the saving of Amiga HAM modes, even if they do appear to work.

ARexx Equivalent:

```
SAVE
,
SAVE_CLIP
,
SAVE_DEVICE
```

1.23 Save

Save

~~~~

Synopsis:

```
SAVE <Project> <FileName> <FileType>
```

Function:

Save a project's render screen to disk.

Inputs:

```
Project - Name of the project.
FileName - Complete path and file name to save as.
FileType - The name of the file format to save as.
```

Result:

None.

Example:

```
...
    /* The image has been render, now */
    /* Ask the user for a save file name */
'GET_FILE "What do you want to save it as?"'
DestName=RESULT

    /* Ask the user for what format they want to save as */
'GET_FILE_TYPE "What format do you want?"'
FileType=RESULT

    /* Save it! */
'SAVE' MyProject ' " || DestName || " " || FileType || "'
...

```

Menu equivalent:

```
Project/Save/File
See also:

GET_FILE_TYPE
,
SAVE_DATA
```

## 1.24 Save to Clipboard

Save to Clipboard

~~~~~

Synopsis:

SAVE_CLIP <Project>

Function:

Save a project's render to the clipboards.

Inputs:

Project - Name of the project.

Result:

None.

Menu equivalent:

Project/Save/Clipboard

See also:

OPEN_CLIPBOARD

1.25 Save Device

Save Device

~~~~~

Synopsis:

SAVE\_DEVICE <Project> <FileType>

Function:

Save a project's render screen to the given device SVOject. Currently the only available device SVOject to write to is the 'Printer'.

Inputs:

Project - Name of the project.

FileType - The name of the device SVOject to save to.

Result:

None.

Menu equivalent:

Project/Save/Device

## 1.26 Save Data Submenu

---

## Save Data Submenu

```

~~~~~
+-----+
|File... <A>D|
|Clipboard <A>C|
|Device... |
+-----+
| <Most |
| Recently |
| Used |
| Directories> |
+-----+

```

This lets you save the an image's 8 bit or 24 bit data to a file or the system clipboard directly without having to render it to a displayable screen mode. It works in the same way as the menu items on the "Save Render" submenu.

The "Device..." item writes the rendered image to a device SVOBJECT. A list of available device SVOBJECTs is displayed from which to choose from. Currently the only available device SVOBJECT is the preferences printer. See the Superview Library documentation for more information.

In the four spaces below the "Device..." item, the names of the last four most recently used directories will be placed. Selecting one of these works in the same way as "File..." item except that the file requester is automatically placed in the corresponding directory. This basicly saves you the time and hassle of having to manually change the file requester between the directories containing your images. This list is also repeated on the "Save Render" submenu.

Note: Not all formats will support the writing of 8 bit or 24 bit data. (ie the C64 format Doodle doesn't support 24 bit images, (surprise, surprise) :)

ARexx Equivalent:

```

SAVE_DATA
,
SAVE_DATA_CLIP
,
SAVE_DATA_DEVICE

```

## 1.27 Save Data

## Save Data

```

~~~~~

```

Synopsis:

```
SAVE_DATA <Project> <FileName> <FileType>
```

Function:

Save a project's image data to disk as 8 bit grey or 24 bit colour.

---

## Inputs:

Project - Name of the project.  
 FileName - Complete path and file name to save as.  
 FileType - The name of the file format to save as.

## Result:

None.

## Example:

```

...
        /* The image has been render, now */
        /* Ask the user for a save file name */
'GET_FILE "What do you want to save the data as?"'
DestName=RESULT

        /* Ask the user for what format they want to save as */
'GET_FILE_TYPE "What format do you want?"'
FileType=RESULT

        /* Save it! */
'SAVE' MyProject ' " || DestName || ' " " || FileType || ' ' '
...

```

## Menu equivalent:

Project/Save/File

See also:

GET\_FILE\_TYPE

,

SAVE

## 1.28 Save Data to Clipboard

Save Data to Clipboard

~~~~~

Synopsis:

SAVE_DATA_CLIP <Project>

Function:

Save a project's data to the clipboards.

Inputs:

Project - Name of the project.

Result:

None.

Menu equivalent:

Project/Save Data/Clipboard

See also:

OPEN_CLIPBOARD

1.29 Save Data to Device

Save Data to Device

~~~~~

Synopsis:

SAVE\_DATA\_DEVICE <Project> <FileType>

Function:

Save a project's image data to the given device SVOBJECT. Currently the only available device SVOBJECT to write to is the 'Printer'.

Inputs:

Project - Name of the project.  
FileType - The name of the device SVOBJECT to save to.

Result:

None.

Menu equivalent:

Project/Save Data/Device

## 1.30 Close Project

Close Project

~~~~~

This closes a project and removes it from memory completely. Like the "Quit" menu item it doesn't check to see if the project has been saved or not. Does the same as thing as clicking on a project's window's close gadget.

ARexx Equivalent:

CLOSE

1.31 Close

Close

~~~~~

Synopsis:

CLOSE <ProjectName>

Function:

Close a project. Same as clicking on the project's close box.

Inputs:

ProjectName - Name of the project.

Result:

None.

Menu equivalent:

Edit/Close Project

See also:

## 1.32 Module Window

### Module Window

~~~~~

This will open the Module window. The module window contains a list of all of the available modules. The 'Execute' button will run the currently selected module on the current image. The 'About...' button will open a small window containing a short text description of the currently selected module. The window can be left open on the screen or its close gadget can be used to close it again.

1.33 ARexx Window

ARexx Window

~~~~~

This opens the ARexx window. The ARexx window contains a list of ARexx scripts that can be run by clicking on 'Execute'. The directory gadget can be used to change to a different directory of ARexx scripts for IE. Clicking on the icon will bring up a directory requester from which the new directory can be selected.

## 1.34 Infobar Window

### Infobar Window

~~~~~

This opens the Infobar Window. The Infobar window shows the name of the current project in an editable gadget, as well as letting you change the project's current zoom level using the "Zoom" gadget. The "Alpha" and "Second" gadgets let you mark the current image as being the Alpha image or the Secondary image respectively. To the right of the Infobar is a compact progress indicator which is used instead of constantly opening and closing the main one.

The title of the Infobar windows shows the coordinates of the pointer in the current project as well as the RGB values and HSB values of the pixel that is currently under the pointer. Followed by the width and height of the

current project and the size of the image in kilobytes.

1.35 Preferences submenu

```

                Preferences submenu
~~~~~

                Font...

                General...

                Menus...

                Open Device...

                Save Device...

                Screen Mode...

                SVObject...

```

1.36 Font Preferences

```

Font Preferences
~~~~~

```

This lets you select which font IE should use for text in its windows and requesters. The font is selected using the standard system ASL Font requester which will appear.

1.37 General Preferences

```

General Preferences
~~~~~

```

This lets you setup defaults for Image Engineer. The options are as follows.

Render Control Defaults:

This lets you set what the default render options 24 bit colour and 8 bit grey images should use once they've been loaded. Clicking on "Grey..." or "Colour..." will bring up the "Render Control" requester. This basically saves you the hassle of having to set up the render options for every image that you open.

Flush Memory:

If this is checked IE will flush the memory just after startup (flushing all sobjects) and just before exiting. Same as the AmigaDOS command "Avail flush".

Modules Load:

This lets you specify how loading and removing from memory of the disk based effects modules should be handled. It can be done in one of three ways:-

At Start up - All of the modules are loaded into memory when IE is started. Starting IE therefore takes more time and uses more memory, but all of the modules are in memory ready for use.

When Needed - IE loads modules into memory once they are needed. From then on they stay in memory until IE is exited.

Everytime - IE loads modules into memory once they are needed and removes them from memory once it has finished with them. This uses the least amount of memory, but it is a bit slower as IE has to load modules from disk each time they are used.

Windows Refresh:

This let you set how the preview windows should be refreshed using either SMART fresh or SIMPLE. SMART refresh is faster although it uses more memory. While SIMPLE is slower but uses little memory. If you're running out of chip memory you may want to switch over to SIMPLE refresh.

Initial Project Zoom:

This lets you specify what the zoom level for images that are loaded in should be. This lets you stop images that are bigger than the screen from opening at 1:1 zoom and covering everything. This only takes effect if the "Adjust zoom level when opening images" checkbox is checked. If it is now checked all opened projects will start with thier zoom level at 1:1. The "% of screen to use" slider lets you set how much of the screen newly opened projects are allowed to use. For example, if you set it to 50%, any newly opened image will be zoomed out so that its width/height isn't greater than 50% of the screen's width/height.

Execute ARexx Script:

At Startup - This is the full path and file name of an ARexx script that should be executed as soon as IE has loaded. The button to the right of the string gadget lets you select an ARexx script using a file requester.

At Exit - This is the full path and file name of an ARexx script that should be executed once the user tries to quit IE. The script is executed just before IE shuts down and exits.

Executive:

This is only useful for people who are using Petri Nordlund's Executive UNIX like task scheduler on thier system.

Nice process when working - If this is checked then IE's nice value will be changed to the nice value below when IE is busy working.

Nice Value - This is the nice value that IE should use when working. Ranges from 20 to -20. See the Executive documentation for more information on how nice values work.

Clicking on "Ok" will use the new Prefs, they are saved to disk when IE is exited. "Cancel" will close the Preferences requester without changing the Prefs.

1.38 menusprefs

Menus Preferences

~~~~~

All of the menus other than the "Project" and "Image" menus are user configurable. This makes it easy to combine image processing effects and ARexx scripts together on one set of menus.

The requester is broken up into three sections, the first is for the menu titles, next menu items, and then sub-menu items.

The Menu titles section shows the list of menu titles and has buttons for inserting and deleting Menu titles and for editing the menu's title. The Menu titles in the list can be rearranged by using the mouse to drag them into place, this also works for the other lists.

The Menu section of the requester lets you edit the menu items for the currently selected Menu title. Like the Menu titles, it also has buttons for editing, inserting and deleting items for the list. Clicking on the "Edit" button brings up the "Edit Menu Item" requester. This lets you set up the menu item as one of five things:-

#### Sub Menu Title:

This menu item is just a sub menu title which can have sub menu items hanging off it.

#### Bar:

The menu item just shows a horizontal bar on the menu, used for visually separating and grouping related menu items.

#### Module:

This menu item runs an effects module. The name of the module to run can be entered in the "Command or Path" gadget, or can be chosen from the list of currently installed modules by using the "Choose..." button.

#### ARexx Command:

This menu item simply makes IE execute a single ARexx command. The command is given in the "Command or Path" gadget, by using [] in the command, the current project's name will be substituted. For example, to apply a 3x3 lowpass filter to the current image the Command would be:-

```
LOWPASS
[] 3 3
```

The main use of this is for adding presets to the menus (like a lowpass 3x3 and lowpass 5x5 menu item), without having to resort to writing ARexx scripts all of the time that only contain a single command.

ARexx Script:

This menu item makes IE execute a given ARexx script. The ARexx script to be executed can be chosen from disk using the "Choose..." button.

The "Text:" field lets you enter the text which is used on the menu for this item. The "Key" list lets you select a key to be used as a short cut for this item. The "Load..." and "Save..." buttons let you load/save the current menu preferences to a file.

Clicking on "Ok" will use the new menu Prefs, they are saved to disk when IE is exited. "Cancel" will close the Menu Preferences requester without changing the Menu Prefs.

### 1.39 Open Device Preferences

Open Device Prefs

~~~~~

This allows you to change the options of the device SVOBJECT that are used for opening images from. A list is displayed containing the available devices. The "Options..." button brings up any available options for the current item.

1.40 Save Device Preferences

Save Device Prefs

~~~~~

This allows you to change the options of the device SVOBJECT that are used for saving images to. A list is displayed containing the available devices. The "Options..." button brings up any available options for the current item.

### 1.41 SVOBJECT Preferences

SVOBJECT Prefs

~~~~~

This lets you view the list of SVOBJECTS that are used to load images into IE. The "Options..." button lets you edit any available options associated with a SVOBJECT. When editing a SVOBJECT's prefs, click on "Ok" to use the new settings or click on "Save" to use and save the settings to disk.

Normally to edit a SVOBJECT's prefs you need to edit a text file (known as a ControlPad file by superview.library) called <objectname>.controlpad stored in the ENV: directory. This allows you to edit a SVOBJECT's controlpad file

in a more user friendly manner.

Note:- For 'SVObject Prefs' to work you need to be using at least version 11.7 of the superview library. If your not, then you won't be able to edit anything here.

Also, the list of file formats can be rearranged using the mouse to drag and drop items into any given order. The order of the list is remembered between sessions and is used when selecting a save file format. This makes it possible to put the file formats that you use the most in easy reach at the top of the list.

1.42 Screen Mode

Screen Mode

~~~~~

This lets you change the screen mode and dimensions of IE's screen. IE's screen will be closed and a screen mode requester will appear on the Workbench screen. If you change your mind and select Cancel, IE will reopen it's screen using the previous screen mode.

The top half of the requester offers all of the standard controls for selecting the screen mode for IE. IE needs to be run in a screen mode that is atleast 640 pixels width by 400 high. In the lower half are these additional settings:-

Colour/Grey:

These radio buttons let you choose between IE drawing the preview images in shades of grey or colour. A colour screen uses more memory than a grey one and requires a few moments to initialise when opening the work screen, but is just as fast in use as a grey screen.

Dither:

If this is checked, IE will use dithering to render the preview images. Resulting in vastly improved image quality. When using colour previews, having dithering turned on uses more memory and takes a few moments to initialise when opening the work screen, but once it's been initialised preview drawing is just as fast.

Set screen mode at start up:

If this option is checked then IE will put up a screen mode requester at start up for you to choose what screen mode you would like it to operate in. If this option is not checked then IE will open its screen using the current screen mode.

IE's screen is a public screen with the name IMAGEENGINEER.

## 1.43 Recalculate Palette

---

Recalculate Palette  
~~~~~

This makes IE calculate a new palette for the preview screen based on the images that are currently open. This results in the preview screen using a palette that's best suited for the current images, giving better preview image quality. This is available only if you're using a colour screen.

1.44 Quit

Quit
~~~~~

This closes all render screens and projects and exits the program completely. It does *\*not\** ask if you're sure, and does *\*not\** check to see if you have unsaved work. Make sure that you've saved all of your work before using this menu item. You will not get a 2nd chance. So make sure you mean it.

ARexx Equivalent:  
QUIT

## 1.45 Quit

Quit  
~~~~~

Synopsis:
QUIT

Function:
This closes all projects and render screens and quits Image Engineer completely.

Inputs:
None.

Result:
None.

Example:
...
QUIT /* Finished work, exit IE */
say 'See you later.'

Menu equivalent:
Project/Quit
See also:

1.46 Project Info

Project Info
~~~~~

This displays a requester detailing the Width and Height of the image, the project's name and the name of the file that the image was loaded from originally and the image's memory usage.

## 1.47 Get Project Info

Get Project Info  
~~~~~

Synopsis:

```
PROJECT_INFO <Project> <W/WIDTH|H/HEIGHT|TYPE|BOX|ZOOM|OFFSETS|WINDOW>
```

Function:

Get information about a project.

Inputs:

Project - Name of the project.
W/WIDTH - Get project's width.
H/HEIGHT - Get project's height.
TYPE - Get project's type.
BOX - Get the dimensions of the project's crop box.
ZOOM - Get the current zoom level for the project.
OFFSETS - Get the current offsets of the image into the project window. This basically the state of the scroll bars on the project window.
WINDOW - Get the position of the project's window.

Result:

WIDTH - Returns the project's width.
HEIGHT - Returns the project's height.
TYPE - Returns the project's type. "GREY" or "COLOUR" will be returned.
BOX - Returns a string of 4 numbers will be returned in the form "x1 y1 x2 y2", suitable for passing to CROP.
ZOOM - Returns the zoom level as a ratio of the window pixels to image pixels. A string like "1:16" through to "16:1" is returned.
OFFSETS - Returns two integers, one for the x offset and the other the y offset.
WINDOW - Returns the x and y position of the upper left corner of the window followed by the width and height of the window.

Example:

```
...
/* Get MyProjects width */
'PROJECT_INFO' MyProject 'WIDTH'
```

```

ProjectWidth=RESULT

/* Get MyProjects height */
'PROJECT_INFO' MyProject 'HEIGHT'
ProjectHeight=RESULT
...
/* Crop the project */
'PROJECT_INFO' MyProject 'BOX'
BoxSize=RESULT
CROP MyProject BoxSize
CroppedProject=RESULT
...

```

Menu equivalent:

```

Edit/Info
See also:

PROJECT_SET

```

1.48 Get Project List

```

Get Project List
~~~~~

```

Synopsis:

```
PROJECT_LIST
```

Function:

Get a list of the currently open projects.

Result:

Returns a list of the names of the currently open projects. Each name is terminated by a space.

1.49 Set Project variables

```

Set Project Variables
~~~~~

```

Synopsis:

```
PROJECT_SET <Project> <ZOOM <level>>
                |<OFFSETS <xoff> <yoff>>
                |<WINDOW <x> <y> <width> <height>>
```

Function:

Set a variable for the given project.

Inputs:

ZOOM - Set the current zoom level for the project.

- level - This is only used when ZOOM is set. It is a string representing the new zoom level. "1:16" through to "16:1"
- OFFSETS - Set the current offsets of the image into the project window. This basicly the state of the scroll bars on the project window.
- xoff - This is only used for OFFSETS. It is the x offset of the image into the window.
- yoff - This is only used for OFFSETS. It is the y offset of the image into the window.
- WINDOW - Set the positon of the project's window.
- x - This is only used for WINDOW. It is the x position of the top left hand cornder of the window.
- y - This is only used for WINDOW. It is the y position of the top left hand cornder of the window.
- width - This is only used for WINDOW. It is the width of the window.
- height - This is only used for WINDOW. It is the height of the window.

Result:

none.

See also:

PROJECT_INFO

1.50 Project Name

Project Name

~~~~~

This lets you edit the name of a project to something more meaningful. Project names mustn't contain spaces, if it does the spaces will be changed to underscores. Project names must also be unique, if it's not unique a number will be prepended to it.

## 1.51 Use as Secondary/Alpha

Use as Secondary/Alpha

~~~~~

Some image processing effects need a secondary image and/or an alpha channel image to operate on in association with the source image. To mark an image for use as the secondary or alpha image, you simply check these menu items. A marked image will have anyone of the letters S & A in front of its title depending on what it has been marked as. The letters simply indicate that the image has been marked as secondary, alpha or both.

ARexx Equivalent:

MARK

1.52 Mark As

Mark As

~~~~~

### Synopsis:

```
MARK <ProjectName> <PRIMARY|SECONDARY|ALPHA>
```

### Function:

This marks a project for use with the Composite command.

### Inputs:

```
ProjectName - Name of the project.
PRIMARY     - This keyword specifies that this project should be
              used as the Primary image when compositing.
SECONDARY   - This keyword specifies that this project should be
              used as the Secondary image when compositing.
ALPHA       - This keyword specifies that this project should be
              used as an Alpha channel when compositing.
```

### Result:

None.

### Example:

```
/* Combine the BackgroundProject with LogoProject using */
/* LogoAlphaProject as an alpha channel */
MARK BackgroundProject SECONDARY
MARK LogoProject PRIMARY
MARK LogoAlphaProject ALPHA
COMPOSITE 0 0 ALPHA
...
```

### Menu equivalent:

Alpha/Primary,Secondary,Alpha  
See also:

COMPOSITE

## 1.53 Zoom

Zoom Sub Menu

~~~~~

```
+-----+
|  1:16 |
|  1:8  |
|  1:4  |
|  1:2  |
```

```
|_ / 1:1 |
|  2:1 |
|  4:1 |
|  8:1 |
| 16:1 |
+-----+
```

This sub menu shows and lets you set the current zoom level for this project's preview image. For example, a zoom factor of 1:16 means that for every pixel of the preview image represents 16 pixels in the real image.

Zoom In/Out Item

```
~~~~~
```

This pair of menu items simply zoom in or out of the image one zoom level.

1.54 Render

```
Render
```

```
~~~~~
```

This renders the project using the settings defined on the Render Control requester.

A 'R' is shown in the project's window title to indicate that the project has been rendered.

ARexx Equivalent:

```
RENDER
```

1.55 Render Control

```
Render Control
```

```
~~~~~
```

This allows you to set up how you want the given project rendered. Choosing this menu item brings up the "Render Control" requester. The main parts to this requester are as follows.

Device:

The "Device" cycle gadget at the top of the requester, determines how the image should be rendered. It may be one of the following.

- Amiga - Renders the image using the standard Amiga colour mapped screen modes.
- SVDriver - Renders the image using the default SVDriver. This allows you to render images using 24bit graphics boards provided you have a SVDriver that supports it. (See the superview.library documentation for more information about SVDrivers).
- HAM8 - Renders the image using Amiga HAM8 mode (where available). This is only available for 24 bit colour images.

HAM6 - Renders the image using Amiga HAM6 mode.

Resolution:

This shows the name of the currently selected render screen mode. To select a new screen mode click on the "Choose" button, this will bring up a screen mode requester from where you can choose.

Colours:

The "Colours" sliders determines how many colours the render screen should have (not to be confused with the number of colours that should be used). This slider is only relevant when using the Amiga as the display device

Overscan:

This sets how big the overscan used for the render screen should be.

Palette Size:

This determines how many of the render screen's colours should be used to render the image.

Palette:

This determines what palette should be used to render the image. Set to "Pick Best", IE will choose a suitable palette to use. "Lock Palette" will force IE to use the images current palette.

Dither:

The "Dither" cycle gadget lets you specify a dithering algorithm to be used when rendering the image. The dithering algorithms available in order of complexity are Floyd-Steinberg, Burkes, Stucki, Sierra, Jarvis and Stevenson-Arce. Floyd-Steinberg is generally good for most things, the effectiveness of each is a subjective thing, I recommend that you try all of them and see what you think of each.

The "Load..." and "Save..." buttons let you load/save the current settings to/from disk.

The "Ok" button accepts the changes, while the "Cancel" button forgets the whole thing.

ARexx Equivalent:

```
RENDER
    RENDER_CLOSE

RENDER_COLOURS

RENDER_DEPTH

RENDER_DEVICE

RENDER_DITHER

RENDER_GET

RENDER_PALETTE
```

```

RENDER_SCREENMODE

RENDER_SET

RENDER_TO_FRONT

```

1.56 Render

Render

~~~~~

### Synopsis:

```
RENDER <Project> [QUIET]
```

### Function:

Render a project.

### Inputs:

```

Project - Name of the project to render.
QUIET   - This keyword forces IE to open the screen at the back of
          the display and not to interrupt the user by stealing the
          input focus. Making it possible to run a script in the
          background while the user works elsewhere.

```

### Result:

None.

### Example:

```

        /* Find out how the user would like us to render the image */
'GET_RENDER COLOUR "How would like it rendered?"
RenderOptions=RESULT

...

/* Set up the project's render options */
'SET_RENDER' MyProject RenderOptions

/* Render the image quietly */
'RENDER' MyProject 'QUIET'

```

### Menu equivalent:

```

Image/Render
See also:
RENDER_CLOSE

```

## 1.57 Render Colours

## Render Colours

~~~~~

Synopsis:

RENDER_COLOURS <Project> <Colours>

Function:

Set the number of colours a project's render screen should use.
Corresponds to the "Palette Size" value in the Render Control requester.

Inputs:

Project - Name of the project.
Colours - The number of colours to use.

Result:

None.

Menu equivalent:

Image/Render Control

See also:

RENDER
RENDER_CLOSE

RENDER_DEPTH

RENDER_DEVICE

RENDER_DITHER

RENDER_GET

RENDER_PALETTE

RENDER_SCREENMODE

RENDER_SET

RENDER_TO_FRONT

1.58 Render Depth

Render Depth

~~~~~

## Synopsis:

RENDER\_DEPTH &lt;Project&gt; &lt;Depth&gt;

## Function:

Set a whether the number of colours a project's render screen should use. Corresponds to the "Colours" value in the "Render Control" requester.

## Inputs:

Project - Name of the project.  
 Depth - The maximum number of colour the screen should have.  
           ie 2, 4, 8, ... 2^n, ...256.

## Result:

None.

## Menu equivalent:

Image/Render Control

See also:

RENDER

    RENDER\_CLOSE

RENDER\_COLOURS

RENDER\_DEVICE

RENDER\_DITHER

RENDER\_GET

RENDER\_PALETTE

RENDER\_SCREENMODE

RENDER\_SET

RENDER\_TO\_FRONT

## 1.59 Render Device

Render Device

~~~~~

Synopsis:

RENDER_DEVICE <Project> <AMIGA|SVDRIIVER|HAM8|HAM6>

Function:

Set a what device should be used to render a project. Corresponds to the "Device" gadget in "Render Control" requester.

Inputs:

Project - Name of the project.
 AMIGA - Use the standard Amiga register based display.
 SVDRIIVER - Use the current SVDriver.
 HAM8 - Use Amiga HAM8 (AGA only of course!).
 HAM6 - Use Amiga HAM6.

Result:

None.

Menu equivalent:

Image/Render Control

See also:

RENDER

RENDER_CLOSE

RENDER_COLOURS

RENDER_DEPTH

RENDER_DITHER

RENDER_GET

RENDER_PALETTE

RENDER_SCREENMODE

RENDER_SET

RENDER_TO_FRONT

1.60 Render Dither

Render Dither

~~~~~

Synopsis:

RENDER\_DITHER <Project> <NONE|FLOYD|BURKES|STUCKI|SIERRA|JARVIS|STEVENSON>

Function:

Set a what dithering should be used to render a project.

Inputs:

Project - Name of the project.  
NONE - none.  
FLOYD - Use Floyd-Steinberg dithering.  
BURKES - Use Burkes dithering.  
STUCKI - Use Stucki dithering.  
SIERRA - Use Sierra dithering.  
JARVIS - Use Jarvis dithering.  
STEVENSON - Use Stevenson-Arce dithering.

Result:

None.

Menu equivalent:

Image/Render Control

See also:

```

RENDER
    RENDER_CLOSE

RENDER_COLOURS

RENDER_DEPTH

RENDER_DEVICE

RENDER_GET

RENDER_PALETTE

RENDER_SCREENMODE

RENDER_SET

RENDER_TO_FRONT

```

## 1.61 Render Get Options

Render Get Options

~~~~~

Synopsis:

```
RENDER_GET <title> [<Render Options>]
```

Function:

Get render settings from the usr.

Inputs:

```
Title           - Title to use for the requester.
Render Options - Optional string describing the initial
                  render settings.
```

Result:

Returns a string of numbers describing render setting to use, suitable for passing to SET_RENDER. If the user cancels the requester, 5 will be returned in RC.

Example:

```

    /* Find out how the user would like us to render the image */
    'RENDER_GET "How would like it rendered?'"
    RenderOptions=RESULT

    'OPEN "bono.iff" 24'    /* Open the image */
    BonoProject=RESULT

    'RENDER_SET' BonoProject RenderOptions
    'RENDER' BonoProject

```

Menu equivalent:

Image/Render Control
See also:

RENDER
 RENDER_CLOSE

RENDER_COLOURS

RENDER_DEPTH

RENDER_DEVICE

RENDER_DITHER

RENDER_PALETTE

RENDER_SCREENMODE

RENDER_SET

RENDER_TO_FRONT

1.62 Render Palette

Render Palette

~~~~~

Synopsis:

```
RENDER_PALETTE <Project> <LOCK|PICK>
```

Function:

Set a what palette the project should use when rendered.

Inputs:

Project     - Name of the project.  
LOCK        - Lock the palette, use the project's current palette.  
PICK        - Choose a palette.

Result:

None.

Menu equivalent:

Image/Render Control  
See also:

RENDER  
    RENDER\_CLOSE

RENDER\_COLOURS

RENDER\_DEPTH

RENDER\_DEVICE

```
RENDER_DITHER
RENDER_GET
RENDER_SCREENMODE
RENDER_SET
RENDER_TO_FRONT
```

## 1.63 Render Screen Mode

```
Render Screen Mode
~~~~~
```

### Synopsis:

```
RENDER_SCREENMODE <Project> <ModeID>
```

### Function:

Set a how a project should be quantized when rendering.

### Inputs:

Project - Name of the project.  
ModeID - Screen mode ID.

### Result:

None.

### Menu equivalent:

Image/Render Control

See also:

```
RENDER
RENDER_CLOSE
```

```
RENDER_COLOURS
```

```
RENDER_DEPTH
```

```
RENDER_DEVICE
```

```
RENDER_DITHER
```

```
RENDER_GET
```

```
RENDER_PALETTE
```

```
RENDER_SET
```

```
RENDER_TO_FRONT
```

## 1.64 Set a project's Render Options

Set a project's Render Options

### Synopsis:

```
RENDER_SET <ProjectName> <Render Options>
```

### Function:

Set a project's render options.

### Inputs:

ProjectName - Name of the project to set.  
Render Options - String of numbers describing the project's new render settings.

### Result:

None.

### Example:

```
/* Find out how the user would like us to render the image */
'RENDER_GET "How would like it rendered?"'
RenderOptions=RESULT

'OPEN "bono.iff" 24' /* Open the image */
BonoProject=RESULT

'RENDER_SET' BonoProject RenderOptions
'RENDER' BonoProject
```

### Menu equivalent:

Image/Render Control

See also:

RENDER\_GET

## 1.65 Close Render

Close Render

This closes a project's render screen. To open the screen again it has to be rendered again.

### ARexx Equivalent:

RENDER\_CLOSE

## 1.66 Render Close

~~~~~  
Render Close  
~~~~~

### Synopsis:

RENDER\_CLOSE <ProjectName>

### Function:

Close a project's render screen.

### Inputs:

ProjectName - Name of the project.

### Result:

None.

### Menu equivalent:

Image/Close Render

See also:

RENDER

## 1.67 Palette Submenu

~~~~~  
Palette Submenu  
~~~~~

Edit...

Load...

Save...

## 1.68 Edit Palette

~~~~~  
Edit Palette  
~~~~~

The "Edit..." menu options opens up a palette requester. If you have a AGA Amiga, you will be able to edit 256 colours, else you'll be limited to 32. The current palette is shown at the top of the screen. Below the palette on the left hand side the currently selected colour is shown. To right are sliders for RGB (red, green, blue), CMY (cyan, magenta and yellow) and HSB (hue, saturation and brightness) colour models.

Below are the following gadgets.

---

**Copy:**

This lets you copy the values of the current colour to another colour in the palette. First click on the button and then select the colour to copy to.

**Spread:**

This lets you create a smooth range of colour from the current colour to another colour. Click on the button and then select the colour to spread to.

**Swap:**

This lets you swap the current colour with another in the palette.

**Restore:**

This restores to the palette to how it was when the palette editor was opened.

**Load:**

This lets you load a palette from disk. (BTW, you can also load in a palette from any IFF ILBM picture, not just palette files).

**Save:**

This lets you save the current palette to disk.

Clicking on "Ok" accepts the new palette, "Cancel" cancels the requester completely.

**ARexx Equivalent:**

PALETTE\_LOAD

PALETTE\_SAVE

**1.69 Load Palette**

Load Palette

~~~~~

This lets you load a palette in from disk. (Same as the "Load..." button on the palette requester).

**ARexx Equivalent:**

PALETTE\_LOAD

**1.70 Load Palette**

Palette Load

~~~~~

**Synopsis:**

```
PALETTE_LOAD <Project> <FileName>
```

Function:

Load a palette into a project.

Inputs:

```
Project - Name of the project.
FileName - File name of the palette to load.
```

Result:

None.

Example:

```
 /* Ask the user for a new palette */
 'GET_FILE "Select New palette"'
 NewPalette=RESULT

 /* Load the new palette for MyProject */
 'PALETTE_LOAD' MyProject ' ' || NewPalette || ' '

 ...
```

Menu equivalent:

```
 Image/Palette
 See also:
```

## 1.71 Save Palette

```
 Save Palette
```

```
~~~~~
```

This lets you save the project's palette to disk. (Same as the "Save..." button on the palette requester).

ARexx Equivalent:

```
      SAVE_PALETTE
```

## 1.72 Palette Save

```
      Palette Save
```

```
~~~~~
```

Synopsis:

```
PALETTE_SAVE <Project> <FileName>
```

Function:

Save project's palette to disk.

---

**Inputs:**

Project - Name of the project.  
 FileName - File name to save palette as.

**Result:**

None.

**Example:**

```

 /* Ask the user what they want the palette saved as */
 'GET_FILE "Select File to save palette as"'
 PaletteName=RESULT

 /* Save MyProject's palette */
 'PALETTE_SAVE' MyProject ' '||PaletteName||' '
 ...

```

**Menu equivalent:**

Image/Palette  
 See also:

## 1.73 Render to Front

**Render to Front**

~~~~~

**Synopsis:**

```
RENDER_TO_FRONT <Project>
```

**Function:**

Move a project's render screen to the front of the display.

**Inputs:**

Project - Name of the project.

**Result:**

None.

**Menu equivalent:****See also:**

## 1.74 Execute Module

**Execute Module**

~~~~~

This lets you execute an image processing effects module to work on the current image. The requester will appear with the list of available modules on the left and a short text description of the currently selected module on the right.

---

## 1.75 Execute ARexx script

Execute ARexx script

~~~~~

This menu item lets you select an ARexx script from disk to execute. A file requester will appear letting you select a script.

## 1.76 Enter Co-ordinates for the Crop Box

Select Box Co-ords

~~~~~

This lets you enter co-ords for a project's crop box, instead of having to drag one out in the project's window. The X and Y gadgets specify where the top left corner of the box is. The Width and Height gadgets let you enter the width and height of the box (surprise, surprise). Initial values for the X, Y, width and height are taken from where the current crop box is. Towards the bottom are X and Y gadgets for the bottom right hand corner of select box should you prefer to enter the bottom right co-ords instead of the width and height.

"Ok" accepts the new values, and "Cancel" returns you to where you were before you opened the requester.

## 1.77 Cut or Crop

Cut or Crop

~~~~~

This cuts out the boxed part of the current project to create a new project. Before you can use this menu item, the current project needs to have a crop box marked out. You can do this by just holding down the left mouse button and dragging out a box in the project's window, or by entering co-ords by hand. The new project will have the same render options as the project it was cut from.

ARexx Equivalent:

CROP

## 1.78 Crop

Crop

~~~~

Synopsis:

CROP <ProjectName> <x1> <y1> <x2> <y2>

Function:

Crop the given rectangle out of the given project



**Inputs:**

ProjectName - Name of the project to crop.  
 x1 - X co-ord of the top left corner of the crop rectangle.  
 y1 - Y co-ord of the top left corner of the crop rectangle.  
 x2 - X co-ord of the bottom right corner of the crop rectangle.  
 y2 - Y co-ord of the bottom right corner of the crop rectangle.

**Result:**

Returns the name of the newly created project in RESULT.

**Example:**

```

/* Cut out the rectangle from (10,10) to (20,20) from the image */
/* MyProject */
...
CROP MyProject 10 10 20 20
...

```

**Menu equivalent:**

Image/Cut  
 See also:

## 1.79 Keyboard short cuts

**Keyboard Short Cuts**

~~~~~

|                               |           |
|-------------------------------|-----------|
| Previous project              | [         |
| Next project                  | ]         |
| Project->Open->File           | Amiga + O |
| Project->Open->Clipboard      | Amiga + V |
| Project->Save->Render         | Amiga + S |
| Project->Save Data->File      | Amiga + D |
| Project->Save Data->Clipboard | Amiga + C |
| Project->Close                | Amiga + K |
| Project->ARexx Window         | Amiga + A |
| Project->Module Window        | Amiga + M |
| Project->Re-calc Palette      | Amiga + L |
| Project->Quit                 | Amiga + Q |
| Image->Info                   | Amiga + I |
| Image->Project Name           | Amiga + N |
| Image->Use as Secondary       | Amiga + 2 |
| Image->Use as Alpha           | Amiga + 3 |
| Image->Zoom In                | Amiga + = |
| Image->Zoom Out               | Amiga + - |
| Image->Render                 | Amiga + R |
| Image->Render Control         | Amiga + P |
| Image->Close Render           | Amiga + T |

|                           |           |
|---------------------------|-----------|
| Image->Enter Co-ords      | Amiga + E |
| Image->Cut                | Amiga + X |
| Close Window or Requester | Esc       |
| Ok button                 | Return    |

## 1.80 Image Processing Effects Modules

### Effects Modules

~~~~~

All of the image processing functions in IE are implemented as disk based modules. This makes it easy to add new functions to IE and to upgrade older ones. The modules currently available are:

- ARexxSupport
- Autocrop
- Brightness
- Bulge
- Colour Filter
- Composite
- Contrast
- ContrastStretch
- ConvertToColour
- ConvertToGrey
- Convolve
- Displace
- Equalisation
- FalseColour
- Gamma
- Halftone
- HighBoost
- Highpass
- Histogram
- Hue

---

HueMask  
Impress  
LocalStretch  
Lowpass  
Median  
Negative  
Noise  
Posterize  
ReflectX  
ReflectY  
RemoveFeature  
Resize  
Ripple  
Rotate  
Saturation  
Scale  
Sharpen  
Threshold  
Twirl

Each entry contains information on the module's name, who holds the ↵  
copyright, description of its function, documentation of the its user  
interface (if any), documentation on the ARexx commands it has, and a  
version history.

## 1.81 ARexx Support Module

ARexxSupport

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.1

Description

~~~~~

This module can not be executed from the user interface. It's purpose is to  
add some utility ARexx commands for use from ARexx scripts. Several commands

---

are provided for getting different kinds of input from the user in a user friendly manner. The FORM command makes it possible to get numerous inputs at once by creating simple to relatively complex input windows.

#### User Interface

~~~~~

Has none.

#### ARexx Commands

~~~~~

##### FORM

- Command for get multiply inputs at once.

##### GET\_DIR

- Command for requesting a directory name from the user.

##### GET\_FILE

- Command for requesting a file from the user.

##### GET\_FILES

- Command for requesting a list of files from the user.

##### GET\_NUMBER

- Command for requesting a number from the user.

##### GET\_STRING

- Command for requesting a string from the user.

##### REQUEST

- Command for requesting a choice from the user.

#### History

~~~~~

- 1.0 5/5/96 Initial Release.
- 1.1 1/7/96 The GET\_NUMBER ARexx command can now have a slider attached to the integer gadget. ->Patrick Nydensten
- 2.0 20/11/96 Written in C, windows and file/dir requesters now remember thier positions.
- 2.1 18/1/97 Several bugs removed and the previously broken REQUEST command is now fixed again. FORM command has been added.

## 1.82 Input Form

#### Form

~~~~~

#### Synopsis:

FORM <Title> <Buttons> <Gadgets...>

#### Function:

This displays on screen a caller defined input form containing numerous gadgets. This makes it possible to get numerous pieces of data from the user without having to resort to popping up hundreds of

individual requesters. This is a lot more friendly from a user point of view.

#### Inputs:

Title - Title of the window.  
 Buttons - Text to use for the terminating buttons. Each option is separated by a | character. By default you should use "Ok|Cancel".

Gadgets... - One or more gadget specifications separated by whitespace. The available gadgets are:-

#### Checkbox:

This specifies a checkbox gadget. It is specified this way:-

```
CHECKBOX,label,default
```

CHECKBOX - This keyword must be given to identify this as being a checkbox gadget.

label - A text label to display beside the checkbox.

default - Boolean value, 1 for checked, 0 if not.

#### Cycle:

This specifies a cycle gadget. It is specified this way:-

```
CYCLE,label,labels,default
```

CYCLE - This keyword must be given to identify this as being a cycle gadget.

label - A text label to display beside the cycle gadget.

labels - A string containing the | separated labels that the user can choose from.

default - The ordinal number of the default selected label, starting from 0.

#### Dir:

This specifies a directory gadget. A string gadget is displayed on the window containing the chosen directory. To the right of the string gadget is a directory button which brings up a directory requester that the user can use to select a dir instead of manually entering the directory. It is specified this way:-

```
DIR,label,default
```

DIR - This keyword must be given to identify this as being a directory gadget.

label - Text label that is displayed beside the gadget.

default - The default directory.

#### File:

This specifies a file gadget. A string gadget is displayed on window containing the chosen file complete with path. To the right of the string gadget is a file button which brings up a file requester from which the user can select the file instead of entering it manually. It is specified this way:-

FILE,label,default

FILE - This keyword must be given to identify this as being a file gadget.  
label - Text label that is displayed beside the gadget.  
default - The default file and path.

#### Integer:

This specifies an integer gadget. It is specified this way:-

INTEGER,label,min,max,default[,SLIDER]

INTEGER - This keyword must be given to identify this as being an integer gadget.  
label - Text label that is displayed beside the gadget.  
min - The lowest value that the gadget accepts.  
max - The highest value that the gadget accepts.  
default - The default value for the gadget.  
SLIDER - If this keyword is given then the integer gadget is also accompanied by a slider gadget.

#### Radio:

This specifies a radio gadget. It is specified this way:-

RADIO,label,labels,default

RADIO - This keyword must be given to identify this as being a radio gadget.  
label - A text label to display above the radio gadget.  
labels - A string containing the | separated labels that the user can choose from.  
default - The ordinal number of the default selected label, starting from 0.

#### String:

This specifies a string gadget. It is specified this way:-

STRING,label,default,length

STRING - This keyword must be given to identify this as being a string gadget.  
label - Text label that is displayed beside the gadget.

---

default - The default string.  
length - The maximum length of the string. The limit is 256.

**Text:**

This specifies a read only text gadget. It is specified this way:-

TEXT,text

TEXT - This keyword must be given to identify this as being a text gadget.

text - A string of text that the text gadget should display. This string is allowed to contain newline (ASCII 10) characters to span multiple lines.

**Result:**

Returns a complex string containing all of the data. The first part string is a number representing the button that the user selected to end the form. Like the REQUEST ARExx command it returns 1 for the left most button and 0 for the rightmost. Buttons in the middle will return one more than the button to it's left.

(ie Buttons: Brilliance, Ok, Average, Crap.

return 1, 2, 3, 0.)

The rest of the string contains the values for each of the gadgets in the same order that they were given to command, separated by a space.

Checkbox - Returns 1 if the checkbox was checked, else 0.

Cycle - Returns the ordinal number of the selected item counting from 0.

Dir - Returns the selected dir surrounded by double quotes.

File - Returns the selected file surrounded by double quotes.

Integer - Returns the entered integer.

Radio - Returns the ordinal number of the selected item counting from 0.

String - Returns the entered string surrounded by double quotes.

Text - Returns absolutely nothing.

**Example:**

See the supplied ARExx scripts for real life examples.

## 1.83 Get Dir

Get Dir

~~~~~

**Synopsis:**

GET\_DIR <Title> [<Ok Text> [<Initial Dir>] ]

**Function:**

Get a directory from the user.

**Inputs:**

Title - Title of the directory requester.  
 Ok Text - Optional text to use for the Ok button.  
 Initial Dir - Optional directory that the requester should start in.

**Result:**

Returns name of the chosen directory in RESULT.  
 If the user cancels the requester, 5 will be returned in RC.

**Example:**

```
/* This example shows how to get a directory from the user and */
/* how to fix it to make it ready to append filename to. */
/* ie 'sys:t' needs to become 'sys:t/' before we can use it to */
/* build complete pathnames. */

Options Results /* We want to receive results */

'GET_DIR "Select Destination Dir" "Go!!!"' /* Get destination */
if RC=5 then exit /* Exit if we were cancelled */
destdir=RESULT

endpart=right(destdir,1) /* Fix it so that it ends in ':' or '/' */
if endpart~=":" & endpart~/="/" then destdir=destdir||"/"
/* Now destdir is ready to append file names to it */
...
```

**See also:**

```
GET_FILE
,
GET_FILES
```

## 1.84 Get File

Get File

~~~~~

**Synopsis:**

```
GET_FILE <Title> [<Ok Text> [<Initial Dir>]]
```

**Function:**

Get a path and filename from the user.

**Inputs:**

Title - Title of the file requester.  
 Ok Text - Optional text to use for the Ok button.  
 Initial Dir - Optional directory that the requester should start in.

**Result:**

Returns the complete path and file name in RESULT.  
 If the user cancels the requester, 5 will be returned in RC.

**Example:**



```
'GET_FILE "Select an Image to process" "Go!!!"' /* Get a file */
if RC=5 then exit /* Exit if we were cancelled */
MyFile=RESULT
...
```

See also:

```
GET_DIR
'
GET_FILES
```

## 1.85 Get Files

Get Files

~~~~~

Synopsis:

```
GET_FILES <Title> [<Ok Text> [<Initial Dir>]]
```

Function:

Get multiple path and filenames from the user.

Inputs:

```
Title - Title of the file requester.
Ok Text - Optional text to use for the Ok button.
Initial Dir - Optional directory that the requester should start in.
```

Result:

Returns a list of the path and file names separated by a ';' in RESULT. If the user cancels the requester, 5 will be returned in RC.

Example:

```
'GET_FILES "Select Images to process" "Go!!!"' /* Get a files */
if RC=5 then exit /* Exit if we were cancelled */
MyFileList=RESULT /* MyFileList now contains the list of files */
/* in the form "sys:pics/File1;sys:pics/File2" */

do while MyFileList~="" /* Keep going while we still have files left */
parse var MyFileList FileName ';' MyFileList

... /* Do something with the image in file FileName */

end

/* See the script batch_convert.rexx for a full example */
```

See also:

```
GET_DIR
'
GET_FILE
```

## 1.86 Request

Request

~~~~~

Synopsis:

```
REQUEST <Message> [<Buttons>]
```

Function:

Put up a requester to the user and get a response.

Inputs:

Message - Message to put in the requester. The message may contain newline characters (ASCII 10).

Buttons - String of button labels separated by a |.

Result:

Returns 1 for the left most button and 0 for the rightmost. Buttons in the middle will return one more than the button to it's left.

(ie Buttons: Brilliance, Ok, Average, Crap.

return           1,           2, 3,           0.)

Example:

```

/* Tell the user that we've finished */
'REQUEST "All done"
...

/* Tell the user what this script does */
'REQUEST "This shows how to put up' D2C(10),
'requesters that may be' D2C(10),
'split over several lines." "I understand"

... /* Find out what the user thinks */
'REQUEST "What do think about this script" "Great|Ok|Crap"
reply=RESULT
if reply=1 then 'REQUEST "I am glad you like it!"'
if reply=2 then 'REQUEST "That is good to hear."'
if reply=0 then 'REQUEST "oh well, you gets what you pays for."'
...

```

See also:

## 1.87 Get Number

Get Number

~~~~~

Synopsis:

```
GET_NUMBER <Title> <Min> <Max> [<Ok Text>] [<Initial>] [SLIDER]
```

Function:

Get a number in the range of Min to Max from the user.

**Inputs:**

Title - Title of the requester.  
 Min - Smallest number to accept.  
 Max - Greatest number to accept.  
 Ok Text - Optional text to use for the buttons. The text for the ok and cancel button should be separated by a | . ie "Ok|Cancel" If only text for one button is supplied, it will be used as the cancel button text, and there will be no Ok button.  
 Initial - Optional number to initially place in the requester.  
 SLIDER - If this keyword is given a slider will be added to the integer gadget.

**Result:**

Returns the entered number in RESULT.  
 If the user cancels the requester, 5 will be returned in RC.

**Example:**

```

/* Ask the user how many frames they want generated */
'GET_NUMBER "Enter the number of Frames" 1 100 "Ok|Cancel" 10'
if RC=5 then exit
NumberOfFrames=RESULT

say 'Number to frames to generate:' NumberOfFrames
...

```

**See also:**

GET\_STRING

## 1.88 Get String

Get String

~~~~~

**Synopsis:**

GET\_STRING <Title> [<Ok Text>] [<Initial>]

**Function:**

Get a string from the user.

**Inputs:**

Title - Title of the requester.  
 Ok Text - Optional text to use for the buttons. The text for the ok and cancel button should be separated by a | . ie "Ok|Cancel" If only text for one button is supplied, it will be used as the cancel button text, and there will be no ok button.  
 Initial - Optional string to initially place in the requester.

**Result:**

Returns the string in RESULT.  
 If the user cancels the requester, 5 will be returned in RC.

Example:

```

/* Ask the user what file extension should be added */
'GET_STRING "Enter file extension to add" "Ok|Cancel" ".new"'
if RC=5 then exit
FileExt=RESULT

say 'File extension set to' FileExt
...

```

See also:

```

GET_NUMBER
,
GET_STRING

```

## 1.89 Autocrop

Autocrop

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.0

Description

~~~~~

This crops out all of the background surrounding an image.

ie. 

|                                  |                 |                                  |
|----------------------------------|-----------------|----------------------------------|
| <pre> **** * *** * * **** </pre> | would become... | <pre> **** * *** * * **** </pre> |
|----------------------------------|-----------------|----------------------------------|

~~~~~

User Interface

~~~~~

Just autocrops the current image without any additional interface.

ARexx Command

~~~~~

AUTOCROP

History

~~~~~

V1.0 Initial Release

## 1.90 Autocrop

Autocrop  
~~~~~

Synopsis:

```
AUTOCROP <ProjectName>
```

Function:

Crop out all of the background surrounding an image.

Inputs:

ProjectName - Name of the project to autocrop.

Result:

Returns the name of the newly created project in RESULT.

Example:

```
options results
...
MyProject='bono.iff'
...
AUTOCROP MyProject /* Autocrop MyProject */
say 'Autocropped project name is' RESULT
```

See also:

## 1.91 Brightness

Brightness

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.0

Description

~~~~~

This adjusts the brightness of the given image.

User Interface

~~~~~

The brightness requester is split over two panels (which can be switched between by using the tabs at the top of the requester). At the bottom of the requester is a slider and integer gadget for changing the number of grey levels the brightness should be changed by.

When editing a colour image there will also be a collection of checkboxes at the bottom of the requester which allow you to set which colour components that should be affected. This is useful in situations where you have to correct an image (perhaps a scan) that has one (or more) of its RGB components too bright with respect to the other components.

The first panel contains a preview image to show the new brightness.

The second panel shows a preview of the new histogram for the image, along

with a collection of statistics for the histogram.

Clicking on "Ok" makes the changes, "Cancel" cancels the whole requester.

ARexx Command

~~~~~

BRIGHTNESS

History

~~~~~

V1.0 Initial Release.

## 1.92 Brightness

Brightness

~~~~~

Synopsis:

BRIGHTNESS <ProjectName> <Value> [INTENSITY | [RED] [GREEN] [BLUE] ]

Function:

Change the brightness of an image.

Inputs:

ProjectName - Name of the project.  
 Value - Value to change brightness by.  
 INTENSITY - Specifies that the intensity component should be affected (default).  
 RED - Specifies that the red colour component should be affected.  
 GREEN - Specifies that the green colour component should be affected.  
 BLUE - Specifies that the blue colour component should be affected.

Result:

Returns the name of the newly created project in RESULT.

Example:

```
BRIGHTNESS MyProject 50 /* Increase brightness by 50 */
BRIGHTNESS MyProject 50 INTENSITY /* as above */
...
BRIGHTNESS MyProject -50 GREEN /* Reduce the brightness of the */
/* green component by 50 */
...
BRIGHTNESS MyProject 10 RED BLUE /* Slightly increase the */
/* brightness of the red and blue components */
```

See also:

## 1.93 Bulge

### Bulge

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.01

#### Description

~~~~~

This applies a bulge effect to the image.

#### User Interface

~~~~~

There will be a short delay before the bulge requester is opened. This is to create the small preview image that's shown on the right of the requester.

The gadgets are as follows:

#### Radius:

The radius of the area to be 'bulged'.

#### Bulge:

Determines how much the area should be affected, ranging from -100 to make the image appear like it's been reflected on the inside of a spoon, to 100 to make it appear like it's been reflected on the outside.

#### X:

The x position of the center of the bulge effect in the image.

#### Y:

The y position of the center of the bulge effect in the image.

#### Method:

Determines whether the twirl should be done using a fast "Nearest Neighbour" algorithm or the better (but slower) "Colour Average" algorithm.

To the right of the controls is a thumbnail image showing which part of the image is to be affected. The center of the bulge can be changed by clicking on the thumbnail image. To right of this image is the thumbnail preview image.

Click on "Ok" to apply the effect, else "Cancel" to cancel the requester completely.

#### ARexx Command

~~~~~

BULGE

History

~~~~~

V1.0 Initial Release.

V1.01 19/8/96 Fixed the ARexx command, was broken.

## 1.94 Bulge

Bulge

~~~~~

Synopsis:

```
BULGE <Project> <X> <Y> <Radius> <amount> <FAST | BEST/COLOUR_AVERAGE>
```

Function:

Bulge part of an image.

Inputs:

```
Project - Name of the project.
X - X position of the center of the bulge.
Y - Y position of the center of the bulge.
Radius - Radius of the bulge.
Amount - Amount to bulge by.
FAST - Use fast nearest neighbour algorithm.
BEST/COLOUR_AVERAGE - Use colour average algorithm.
```

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.95 Colour Filter

Colour Filter

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.0

Description

~~~~~

This applies a user defined colour filter to the image.

User Interface

~~~~~

In the top left hand corner is a group of check boxes for specifying which colour components of the image should be affected.

To the right is a set of pages, Preview, Intensity, Red, Green and Blue. The preview page shows a thumbnail preview of the resulting image. The other pages show previews of the histograms of the resulting image.

The gadgets are as follows:-

Low:

This slider controls where the bottom end of the intensity component range should be moved to.

High:

This slider controls where the top end of the intensity component



range should be moved to.

Red Low:

Same as the Low slider, except for the red component.

Red High:

Same as the High slider, except for the red component.

Green Low:

Same as the Low slider, except for the green component.

Green High:

Same as the High slider, except for the green component.

Blue Low:

Same as the Low slider, except for the blue component.

Blue High:

Same as the High slider, except for the blue component.

Click on "Ok" to apply the effect, else "Cancel" to cancel the requester completely.

ARexx Command

~~~~~

COLOUR\_FILTER

History

~~~~~

V1.0 30/5/96 Initial Release.

## 1.96 Colour Filter

Colour Filter

~~~~~

Synopsis:

```
COLOUR_FILTER <Project> <INTENSITY <low> <high> |
 RGB <rl> <rh> <gl> <gh> <bl> <bh> >
```

Function:

Applies a colour filter to the given image.

Inputs:

```
Project - Name of the project.
INTENSITY - Specifies that the intensity component should be affected.
RGB - Specifies that individual red, green and blue components
 should be affected.
<low> - Specifies how far the low end of the intensity component
 should be moved. In range [-255,255].
<high> - Specifies how far the high end of the intensity component
 should be moved. In range [-255,255].
<rl> - Same as <low> except for the red component.
<rh> - Same as <high> except for the red component.
```

```

<gl> - Same as <low> except for the green component.
<gh> - Same as <high> except for the green component.
<bl> - Same as <low> except for the blue component.
<bh> - Same as <high> except for the blue component.

```

**Result:**

Returns the name of the newly created project in RESULT.

**Example:**

```

/* Apply a colour filter to the intensity component of MyProject. */
/* Make the intensity component use the middle half of the range.*/
COLOUR_FILTER MyProject INTENSITY 64 -64
...

/* Apply a colour filter to each RGB component of MyProject. */
/* This will effectively tint it an off yellow 'antique' colour.*/
COLOUR_FILTER MyProject RGB 0 -20 0 -42 0 -87

```

See also:

## 1.97 Composite

### Composite

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.1

#### Description

~~~~~

Combines source, secondary and alpha images to create composite images using various methods.

#### User Interface

~~~~~

Before the user interface can be invoked, there must be an image marked for use as the secondary image. (See {"Image->Use as Secondary" LINK MARKAS}).

#### X:

This is the x offset of the source (primary) image into the secondary image.

#### Y:

This is the y offset of the source (primary) image into the secondary image.

#### Method:

This controls how the images should be combined. There are currently 8 ways:

Add - The values of the primary and secondary images are simply added together and clipped into the range [0,255].

Alpha Channel - The images are combined using an image as an alpha channel. An alpha channel is simply an image

that is used to determine how the primary and secondary image should be combined for each pixel. If a pixel in the alpha channel is 100% white, then it means that the pixel that it corresponds to should be 100% of the primary image and 0% of the secondary. If the pixel in the alpha channel was black (ie 0% white) then it means that 0% of the primary and 100% of the secondary should be used. While a value of 50% white in the alpha channel means that the corresponding pixel should be made up of 50% primary and 50% secondary. This method is only available if an image has been marked for use as an Alpha channel.

Difference - The difference of the primary and secondary images is used to create the image.

Maximum - The maximum value of the primary and secondary image is used to create the image.

Minimum - The minimum value of the primary and secondary image is used to create the image.

Mix - The primary and secondary images are simply to be mixed (or blended) together using the given percentage of the primary image with the secondary image.

Multiply - The primary and secondary values are simply multiplied together and scaled into the range [0,255].

Subtract - The primary image is subtracted from the secondary image, then clipped into the range [0,255].

Stretch Alpha to fit:

This is only available when doing an Alpha channel composite. If this is checked then the alpha image will be stretched to the same size as the current image that is going over the background.

%: This determines what percent mix should be used when using the Mix method to combine the images.

Black is Transparent:

When using the Mix method this treats all of the black pixels in the primary as transparent, (in much the same way as a genlock). This is very useful if you're trying to place a title over a background image.

To the left of the requester is a thumbnail preview image, with a select box on it. To the right of this is a zoomed in preview of the area selected in the thumbnail preview image.

As always, hitting the "Ok" button will set it into operation, while cancel will forget the whole thing.

ARexx Command

~~~~~

COMPOSITE  
History

---

```

~~~~~
V1.0      Initial Release
V1.01 27/5/96 Fixed bug that would sometimes cause memory to be read pass
          the buffer in the previews.
V1.02 3/6/96 Fixed a bug that stopped the Mix with Black as transparent
          method from working with a grey primary and colour secondary
          image.
V1.03 19/8/96 Using the ARexx command with Mix and genlock mix% would end
          up being set to 0.
V1.04 26/8/96 Small Alpha composite bug that showed up when the alpha image
          is not as wide as the primary image. And another one that
          caused the alpha image to be placed incorrectly when y offset
          is negative.
V1.1 27/8/96 Stretch alpha option added to the interface and ARexx command.

```

## 1.98 Composite

### Composite

#### Synopsis:

```

COMPOSITE <X Offset> <Y Offset> <ADD|ALPHA [STRETCH] |DIFFERENCE
          |MAX|MIN|MULTIPLY|SUBTRACT
          |<MIX <%> [GENLOCK]>>

```

#### Function:

Create a composite image from the marked Primary and Secondary images by mixing the images or using an alpha to combine them.

#### Inputs:

```

ProjectName - Name of the project.
X Offset    - X Offset of the Primary image with respect to the
              Secondary image.
Y Offset    - Y Offset of the Primary image with respect to the
              Secondary image.

ADD
ALPHA      - This keyword specifies that the two images should be
              combined using an alpha channel.
STRETCH    - If this keyword is present the alpha image will be
              stretched so that it is the same size as the primary
              image.
DIFFERENCE - This keyword specifies that the two images should be
              combined by taking the difference of two images.
MAX        - This keyword specifies that the two images should be
              combined by taking the greater pixel value.
MIN        - This keyword specifies that the two images should be
              combined by taking the smaller pixel value.
MIX        - This keyword specifies that the images should be mixed
              together at <%> percent.
MULTIPLY   - This keyword specifies that the two images should be
              combined by multiplying pixel values.
SUBTRACT   - This keyword specifies that the primary image is
              subtracted from the secondary image, then clipped into
              the range [0,255].

```

```

%           - Percentage of the Primary image to use when mixing.
GENLOCK    - This keyword specifies that all black in the primary
              image should be made transparent.

```

Result:

Returns the name of the newly created project in RESULT.

Example:

```

/* Mix the BackgroundProject with the LogoProject*/
/* using 50% mix with black being transparent */
MARK BackgroundProject SECONDARY
MARK LogoProject PRIMARY
COMPOSITE 0 0 MIX 50 GENLOCK
...

/* Combine the BackgroundProject with LogoProject using */
/* LogoAlphaProject as an alpha channel */
MARK BackgroundProject SECONDARY
MARK LogoProject PRIMARY
MARK LogoAlphaProject ALPHA STRETCH
COMPOSITE 0 0 ALPHA
...

```

See also:

MARK

## 1.99 Contrast

Contrast

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.0

Description

~~~~~

This adjusts the contrast of the given image.

User Interface

~~~~~

The contrast requester is split over two panels (which can be switched between by using the tabs at the top of the requester). At the bottom of the requester is a slider and integer gadget for changing the contrast of the image.

When editing a colour image there will also be a collection of checkboxes at the bottom of the requester which allow you set which colour components should be affected. This is useful in situations where you have to correct an image (perhaps a scan) that has one (or more) of its RGB components too bright with respect to the other components.

The first panel contains a preview image to show the new contrast.

The second panel shows a preview of the new histogram for the image, along

with a collection of statistics for the histogram.

Clicking on "Ok" makes the changes, "Cancel" cancels the whole requester.

ARexx Command

~~~~~

CONTRAST

History

~~~~~

V1.0 Initial Release

## 1.100 Contrast

Contrast

~~~~~

Synopsis:

```
CONTRAST <ProjectName> <Value> [INTENSITY | [RED] [GREEN] [BLUE] ]
```

Function:

Change the contrast of an image.

Inputs:

```
ProjectName - Name of the project.
Value       - Value to change contrast by.
INTENSITY   - Specifies that the intensity component should be
              affected (default).
RED         - Specifies that the red colour component should be
              affected.
GREEN       - Specifies that the green colour component should be
              affected.
BLUE        - Specifies that the blue colour component should be
              affected.
```

Result:

Returns the name of the newly created project in RESULT.

Examples:

```
CONTRAST MyProject 50          /* Increase contrast */
CONTRAST MyProject 50 INTENSITY /* same as above */
...

CONTRAST MyProject -50 GREEN   /* Reduce the contrast of the */
...                          /* green component by 50 */

CONTRAST MyProject 10 RED BLUE /* Slightly increase the */
                          /* contrast of the red and blue components */
```

See also:

1.101 Contrast Stretch

Contrast Stretch

© Copyright 1996 Simon Edwards. All rights reserved.
Version: 1.01

Description

~~~~~

This increases an image's contrast so that it uses the full dynamic range available. This generally improves the appearance of an image and helps bring out detail.

User Interface

~~~~~

Applies the contrast stretch to the current image without any additional interface.

ARexx Command

~~~~~

CONTRAST\_STRETCH

History

~~~~~

V1.0 Initial Release

V1.01 4/7/96 Fixed a div by 0 bug that only occurs when using degenerate images that have only 1 unique value.

1.102 Contrast Stretch

Contrast Stretch

~~~~~

Synopsis:

CONTRAST\_STRETCH <ProjectName>

Function:

Contrast stretch an image.

Inputs:

ProjectName - Name of the project.

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.103 Convert to Colour

Convert to Colour

---

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.0

#### Description

~~~~~

This converts a grey (8 bit) image to colour (24 bit).

User Interface

~~~~~

None, just converts the given image to colour. Will only work on grey images.

#### ARexx Command

~~~~~

CONVERT_TO_COLOUR

History

~~~~~

V1.0 Initial Release

## 1.104 Convert to Colour

### Convert to Colour

~~~~~

Synopsis:

CONVERT_TO_COLOUR <ProjectName>

Function:

Convert a 8 bit grey image to 24 bit colour.

Inputs:

ProjectName - Name of the project.

Result:

Returns the name of the newly created project in RESULT.

See also:

1.105 Convert to Grey

Convert to Grey

© Copyright 1996 Simon Edwards. All rights reserved.
Version: 1.0

Description

~~~~~

This merely converts an image to grey. Converts RGB to grey using the weighted average with the (computationally fast) 2:4:1 weights.

---



User Interface

~~~~~

None, just converts the given image to grey. Will only work on colour images.

ARexx Command

~~~~~

CONVERT\_TO\_GREY

History

~~~~~

V1.0 Initial Release

1.106 Convert to Grey

Convert to Grey

~~~~~

Synopsis:

CONVERT\_TO\_GREY <ProjectName>

Function:

Convert a 24 bit colour image to 8 bit grey

Inputs:

ProjectName - Name of the project.

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.107 Convolve

Convolve

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.02

Description

~~~~~

Applies a user defined
convolve
to an image.

User Interface

~~~~~

The "Load..." button opens a file requester, allowing you load in a convolve from disk.

The name gadget holds the name given to this convolve.

The convolve requester is split across two panels.

The first shows the preview of the convolve. On the left is a thumbnail image of the image with a select box that can be moved with the mouse. To the right is a zoomed in preview of the selected part of the image.

The second panel lets you edit the convolve. The number gadgets below the "Convolve Matrix:" hold the values of the matrix elements. (BTW, to get to the next one you don't have to click in the next one with the mouse, you can just press tab). The divisor and bias gadgets hold the values for divisor and bias respectively. The "New" button will clear the matrix and the divisor and bias values. The "Save..." button will let you save the current convolve to disk.

Finally "Ok" applies the convolve, while "Cancel" cancels the entire operation.

ARexx Command  
 ~~~~~

CONVOLVE
 History

~~~~~

V1.0 Initial Release  
 V1.01 29/5/96 Put in the tab key cycle loopy thing.  
 V1.02 25/6/96 Moved the New, Save, Divisor and Bias to below the matrix.

## 1.108 What the smeg is a convolve?

What the smeg is a convolve?  
 ~~~~~

A convolve is just a matrix of elements, a number for the divisor, and one for the bias.

That's great, what do you do with it?
 ~~~~~

A convolve works like this. To calculate the new value of each pixel, the matrix is centered over the source pixel and each matrix element is multiplied by the value beneath it, the values summed then divided by the divisor and finally the bias is added.

For example.

If the convolve matrix looks like this.

```
0 -1 0
-1 4 -1
0 -1 0
```

Divisor = 8  
 Bias = 0

---

Applying it to pixel e in the diagram below.

```
a b c
d e f
g h i
```

The new value for pixel e would be.

$$(0xa + -1xb + 0xc + -1xd + 4xe + -1xf + 0xg + -1xh + 0xi) / \text{Divisor} + \text{Bias}$$

Simple, eh? Convolves can be used to create 100s of different effects.

Examples of convolves

~~~~~

A 3x3 lowpass filter merely averages the pixel values in a 3x3 neighbourhood, therefore it's convolve matrix looks like this.

```
1 1 1
1 1 1
1 1 1
```

Divisor = 9

Bias = 0

A blur that's not as heavy as a 3x3 lowpass filter would be done using a convolve like this.

```
0 1 0
1 4 1
0 1 0
```

Divisor = 8

Bias = 0

1.109 Convolve

Convolve

~~~~~

Synopsis:

```
CONVOLVE <ProjectName> <ConvolveFilename>
```

Function:

Apply a convolve to an image.

Inputs:

ProjectName - Name of the project.  
ConvolveFileName - Path and filename of the convolve to use.

Result:

Returns the name of the newly created project in RESULT.

Example:

```
options results
```

```

...
MyProject='bono.iff'
...
CONVOLVE MyProject "Convolve/Raise_Low" /* Convolve MyProject*/
/* with Raise_Low */
say 'Convolved project name is' RESULT

```

See also:

## 1.110 Displace

Displace

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.02

Description

~~~~~

This displaces the pixels in the current image using the values of the pixels in the alpha image.

It works like this, you specify the number of pixels to displace in the x and y direction first. Each pixel in the resulting image is determined by taking its corresponding value from the alpha channel. If the value from the alpha is white then the pixel is displaced by given amount. If the value from the alpha channel is black then the pixel displaced by by the given amount in the negative direction. While if the alpha value is mid grey then the pixel doesn't move at all.

Example. For when the value from the alpha channel is white. The resulting pixel is displaced by (dx,dy).

```

|-dx-|
Pixel -> .
         \
          \
           \
            \
             \
              \
               \
                \
                 \
                  \
                   \
                    \
                     \
                      \
                       \
                        \
                         \
                          \
                           \
                            \
                             \
                              \
                               \
                                \
                                 \
                                  \
                                   \
                                    \
                                     \
                                      \
                                       \
                                        \
                                         \
                                          \
                                           \
                                            \
                                             \
                                              \
                                               \
                                                \
                                                 \
                                                  \
                                                   \
                                                    \
                                                     \
                                                      \
                                                       \
                                                        \
                                                         \
                                                          \
                                                           \
                                                            \
                                                             \
                                                              \
                                                               \
                                                                \
                                                                 \
                                                                  \
                                                                   \
                                                                    \
                                                                     \
                                                                      \
                                                                       \
                                                                        \
                                                                         \
                                                                          \
                                                                           \
                                                                            \
                                                                           \

```

User Interface

~~~~~

This can only be invoked if an image is marked for use as an alpha image.

The preview shows the a thumbnail version of the image, and a zoomed in preview on the right. The area shown in the preview image can be selected on the thumbnail image.

The "X:" and "Y:" gadgets control how great and in what direction the displacement will be.

The "Method:" gadget controls whether the displacement should be done using a "Fast" algorithm or a better (but slower) "Colour Average" algorithm.

The "Ok" buttons applies the effect, "Cancel" cancels the whole operation.

ARexx Command

~~~~~

DISPLACE

History

~~~~~

V1.0 Initial Release

V1.01 13/6/96 The ARexx function completely stuffed up its handling of the Primary and Alpha projects, i.e. it was broken.

->Patrick Nydensten

V1.02 24/6/96 Removed some old test code that was causing it to Init and Expunge everytime the GUI was called. Not a major problem, but still...

V1.03 27/8/96 The preview had problems when the alpha image was grey.

->Patrick Nydensten

## 1.111 Displace

Displace

~~~~~

Synopsis:

DISPLACE <x displacement> <y displacement>

Function:

Displaces the pixels in the primary image using the alpha channel.

Inputs:

x displacement - Number of pixels to displace by on the x axis.

y displacement - Number of pixels to displace by on the y axis.

Result:

Returns the name of the newly created project in RESULT.

See also:

MARK

1.112 Histogram Equalization

Histogram Equalization

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.0

Description

~~~~~

This applies a histogram equalization to the image. It tries to change an

image so that each grey level is represents an equal number of pixels in the image. Or to put it in other words, it tries to make the histogram of the image flat. Due the discrete nature of the whole operation the resulting histogram will not be flat, but only an approximation. This has the effect of increasing the contrast of an image and brings out detail.

User Interface

~~~~~

Simply applies a histogram equalization to current image.

ARexx Command

~~~~~

HISTOGRAM\_EQUALIZATION

History

~~~~~

V1.0 Initial Release

1.113 Histogram Equalization

Histogram Equalization

~~~~~

Synopsis:

HISTOGRAM\_EQUALIZATION <Project>

Function:

Apply histogram equalization to an image.

Inputs:

Project - Name of the project.

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.114 False Colour

False Colour

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.0

Description

~~~~~

This applies a false colour effect to the current image.

User Interface

~~~~~

Applies a false colour to the current image. If the image is grey then the

resulting project will be colour.

ARexx Command

~~~~~

FALSE_COLOUR

History

~~~~~

V1.0 Initial Release

## 1.115 False Colour

False Colour

Synopsis:

FALSE\_COLOUR <ProjectName>

Function:

Apply a false colour effect to an image.

Inputs:

ProjectName - Name of the project.

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.116 Gamma

Gamma

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.01

Description

~~~~~

Adjusts the gamma content of the image.

User Interface

~~~~~

The gamma requester is split over two panels (which can be switched between by using the tabs at the top of the requester). At the bottom of the requester is a slider and integer gadget for changing the gamma of the image.

When editing a colour images there will also be a collection of checkboxes at the bottom of the requester which allow you set which colour components should be affected.

The first panel contains a preview image to show the new gamma.

---

The second panel shows a preview of the new histogram for the image, along with a collection of statistics for the histogram.

Clicking on "Ok" makes the changes, "Cancel" cancels the whole requester.

Increasing the gamma content of an image has the effect of darkening the mid-grey values. While decreasing the gamma does the opposite. The main use for changing the gamma (or Gamma Correction as it is known) is to correct effects caused by the non-linear way in which monitors respond to inputs.

The best way to see this is open the image Gradient.alpha which is in the Alpha directory. Now render it using as many shades of grey as possible. Notice how the perceived intensity decreases slowly from left to right until it gets to the right where it drops off quickly to black. Now apply gamma correction of +47. Render the new image. Notice how the intensity decreases more uniformly across the image.

ARexx Command

~~~~~

GAMMA
History

~~~~~

V1.0 Initial Release  
V1.01 29/5/96 Fixed it so that it is more in line with other image processing software with positive values making the image brighter.

## 1.117 Gamma

Gamma

~~~~~

Synopsis:

GAMMA <ProjectName> <Value> [RED] [GREEN] [BLUE]

Function:

Change the Gamma content of an image.

Inputs:

ProjectName - Name of the project.
Value - Value to change gamma by.
RED - Specifies that the red colour component should be affected.
GREEN - Specifies that the green colour component should be affected.
BLUE - Specifies that the blue colour component should be affected.

Result:

Returns the name of the newly created project in RESULT.

See also:

1.118 Halftone

Halftone

© Copyright 1996 Simon Edwards. All rights reserved.
Version: 1.0

Description

~~~~~

This halftones the image. The alpha channel image is used to specify the halftone pattern to be used. It works by using each pixel in the alpha channel to threshold the corresponding pixel in the current image. If the alpha channel image isn't as large as the current image then it's simply tiled to cover.

User Interface

~~~~~

Simply applies the halftone effect. An image must be marked for use as an alpha image.

ARexx Command

~~~~~

HALFTONE

See also the supplied halftone Arexx script which provides a simple interface to many common halftone patterns. ↵

History

~~~~~

V1.0 Initial Release

1.119 Halftone

Halftone

~~~~~

Synopsis:

HALFTONE

Function:

Halftones the primary image using the alpha image as the halftone pattern.

Inputs:

None.

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.120 HighBoost

HighBoost

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.02

### Description

~~~~~

This applies a highboost filter to the image. This is effectively the same as a highpass filter except that a fraction of the original image is added back in to the highpass result. This can be useful in that the resulting image looks more like the original except that the high frequencies (edges, lines) have been highlighted.

User Interface

~~~~~

In the top half of the HighBoost requester is the usual thumbnail and zoomed in preview image. The other components are:-

#### Percent:

This controls what fraction of the original image is added into the highpass result. The greater, the more the result looks like the original image.

#### Width:

This is the width of the highpass filter used.

#### Height:

This is the height of the highpass filter used.

The "Ok" button applies the filter, and the "Cancel" button cancels the requester.

### ARexx Command

~~~~~

HIGHBOOST

History

~~~~~

V1.0 Initial Release

V1.01

V1.02 28/7/96 Fixed a bug that broke the slider<->int link thingy.

## 1.121 HighBoost

HighBoost

~~~~~

Synopsis:

HIGHBOOST <Project> <%>

Function:

Apply highboost to an image.

Inputs:

Project - Name of the project.
% - Percentage to the original to add to the highpass result.

Result:

Returns the name of the newly created project in RESULT.

See also:

1.122 HighPass

HighPass

© Copyright 1996 Simon Edwards. All rights reserved.
Version: 1.0

Description

~~~~~

This applies a highpass filter to the image. The highpass filter removes the low frequencies in an image, leaving the high ones that correspond to edges and sharp intensity transitions etc.

#### User Interface

~~~~~

On the left side of the highpass requester is the thumbnail image and zoomed preview image to the right of that. The other gadgets are as follows:-

Normalise:-

Since we can have negative transitions in an image the result from the highpass filter will be in the range of [-255,255] and therefore has to be converted back into the range [0,255]. There are four different ways that can be used to "normalize" it.

- Absolute - The absolute value is used. In the result the edges will be surrounded by a double white line.
- Scale - It's scaled into the range [0,255]. The result doesn't look entirely unlike some sort emboss effect.
- Clip Negative - All negative values simply become zero.
- Clip Positive - All positive values become zero and the negative values are negated.

Width:

This is the width of the highpass filter used.

Height:

This is the height of the highpass filter used.

ARexx Command

~~~~~

HIGHPASS  
History

~~~~~  
V1.0 Initial Release

1.123 HighPass

HighPass
~~~~~

### Synopsis:

HIGHPASS <Project> <Width> <Height> [ABSOLUTE|SCALE|CLIPNEG|CLIPPOS]

### Function:

Apply a highpass filter to an image.

### Inputs:

Project - Name of the project.  
Width - Width of the mask to use. Currently must be an odd number.  
(ie 1,3,5...) An even number will be rounded down to an odd number.  
Height - Height of the mask to use. Currently must be an odd number.  
(ie 1,3,5...) An even number will be rounded down to an odd number.  
ABSOLUTE - Normalize by using the absolute value.  
SCALE - Normalize by scaling the values into range.  
CLIPNEG - Normalize by clipping negative values.  
CLIPPOS - Normalize by clipping positive values.

### Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.124 Histogram

Histogram

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.01

### Description

~~~~~

This displays the image's histogram, along with the number of unique values in the histogram, the mode value (ie the most common value), what the lowest value is and what the highest value is. For colour projects you can view the intensity, hue or saturation histograms for the image or the histograms for each of the three primary colour components (Red, Green, Blue).

User Interface

~~~~~

---

For grey images the requester just show the Intensity histogram, the statistics are along the bottom. For the colour images, the requester is split across 6 panels, one for each of intensity, hue, saturation, red, green and blue.

ARexx Command

~~~~~

None.

History

~~~~~

V1.0 Initial Release

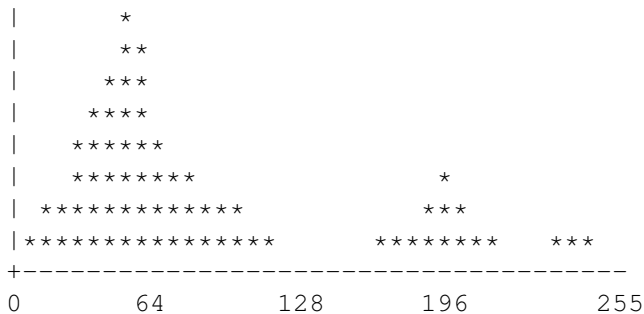
V1.01 26/5/96 Fixed possible bug that may cause it to display an error message for no reason.

## 1.125 What's a Histogram anyway?

What's a Histogram anyway?

~~~~~

A histogram is just a graph showing the relative frequency of each grey level in an image. It typically looks something like this.



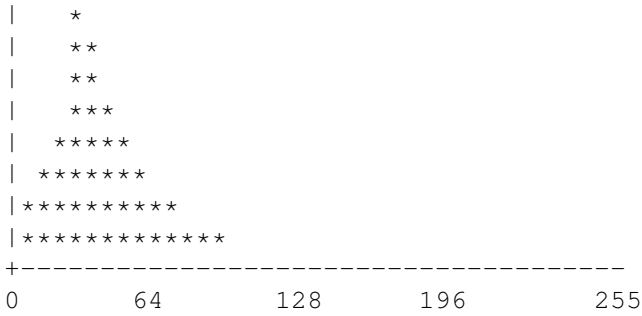
The grey levels are in the range of 0 to 255 inclusive. The axis along the bottom is the grey value axis. The y axis shows the relative frequency. The higher the column graph for a given grey level, the greater frequency it has in the image relative to the other grey levels (In the graph above we can see the most frequent grey level is approximately 60, meaning that the grey level of 60 occurs more often than any other in the image).

In colour images each colour is converted to a grey value and that value is used to form the graph. It's also possible to create the histogram using the values of a single component (ie one of red, green or blue).

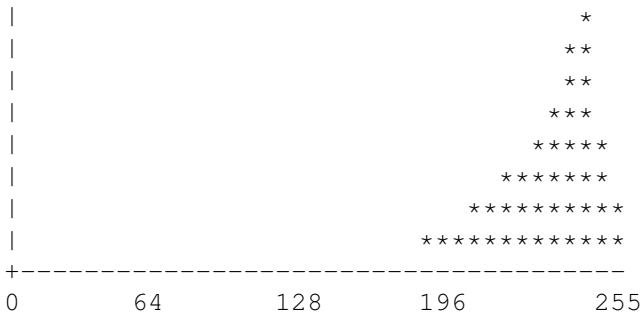
What's the good in it?

~~~~~

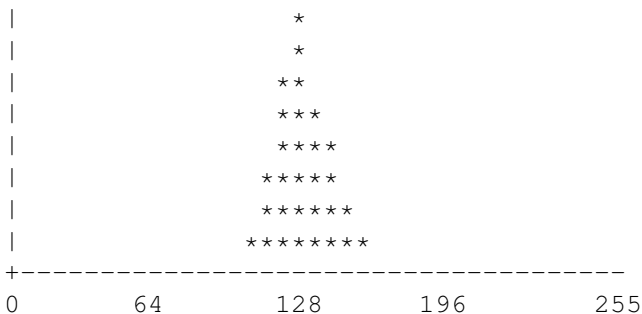
Although it says nothing about the content of an image, the histogram does give important information about the global characteristics of an image. Information from the histogram comes in very useful when adjusting the brightness and contrast of an image, and when affecting the overall balance of an image.



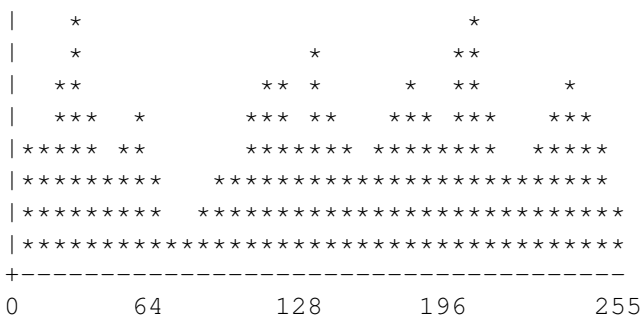
The histogram above shows that the grey values in this image are clustered towards the lower end. This corresponds to a dark image. While the histogram below corresponds to a bright image.



Since the grey levels are clustered together in the histogram below, it therefore corresponds to an image with low contrast. This image would appear a murky grey.



The histogram below corresponds to an image with high contrast, as the grey levels are well spread.



The histogram below corresponds to an image of a white object with a black background. In this case the histogram would be useful in choosing a

threshold to use to turn the image into a real black and white image. (BTW, you would choose the threshold to be at about 150).

```

|      *
|      *
|     ***
|    ****
|   *****
|  ********
|  *********
|  *********
|  *********
+-----+
0         64       128       196       255

```

## 1.126 Hue

Hue

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.0

### Description

~~~~~

Adjusts the hue of the image.

User Interface

~~~~~

The hue requester is split over two panels (which can be switched between by using the tabs at the top of the requester). At the bottom of the requester is a slider and integer gadget for shifting the hue of the image.

The first panel contains a preview image to show the new hue.

The second panel shows a preview of the new histogram for the image, along with a collection of statistics for the histogram.

Clicking on "Ok" makes the changes, "Cancel" cancels the whole requester.

### ARexx Command

~~~~~

```

HUE
History

```

~~~~~

V1.0 Initial Release

## 1.127 Hue

Hue

~~~

Synopsis:

HUE <ProjectName> <Value>

Function:

Change the hue of an image.

Inputs:

ProjectName - Name of the project.
Value - Value to change hue by.

Result:

Returns the name of the newly created project in RESULT.

See also:

1.128 Hue Mask

Hue Mask

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.0

Description

~~~~~

Creates a black and white image from the source image based on a selected range of hue values. All of the pixels whose hue is in the defined range are shown as white in the resulting image, else black.

**User Interface**

~~~~~

The interface is split over two panels. At the bottom of the window are sliders for setting the left and right values of the hue range. The hue component is circular, in that the left value can be greater than the right value, in which case the range will spread over the edge of the histogram.

The first shows a preview image. The second panels shows the histogram of the image's hue component. On the histogram the range of hue values that will be shown in white in the resulting image is enclosed by a box of 'marching ants'. The hue range can be selected by dragging a box out on the histogram.

Clicking on "Ok" creates the new image, "Cancel" cancels the whole requester.

ARexx Command

~~~~~

HUE\_MASK

History

~~~~~

V1.0 8/6/96 Initial Release

V1.01 14/11/96 Small change in the way that it creates the new project.

1.129 Hue Mask

Hue Mask

~~~~~

Synopsis:

HUE\_MASK <ProjectName> <Left> <Right>

Function:

Create a binary image from the given image based on a range of hues. Pixels that have a hue range that is in the given range are set to else they are set to black.

Inputs:

ProjectName - Name of the project.  
Left - Left value of the hue range.  
Right - Right value of the hue range.

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.130 Impress

Impress

© Copyright 1996-97 Simon Edwards. All rights reserved.

Version: 1.13

Description

~~~~~

This creates an impression of the alpha image onto the current image. The lighter parts of the alpha channel will make the corresponding pixels in the result lighter, while the dark parts of the alpha image will make the result darker.

This effect works best if you use a tiled picture (like a workbench tile background) for the alpha image. Also try applying a Emboss_High convolve using the convolve function to an image and use the result as the alpha image to Impress for good effect.

User Interface

~~~~~

This can only be invoked if an image is marked for use as an alpha image.

The preview shows the a thumbnail version of the image, and a zoomed in preview on the right. The area shown in the preview image can be selected on the thumbnail image.

The "Light:" and "Dark:" gadgets control how strong an effect of the lighter and darker parts of the alpha image should be. Both gadgets are in the range of 0 to 256. 0 means that it will have no effect, while 256 means maximum

effect.

If the "Stretch Alpha to fit" check box is checked then the alpha image will be stretched to fit the size of the current image, otherwise it will just be tiled in order to make it fit.

The "Ok" buttons applies the effect, "Cancel" cancels the whole operation.

ARexx Command

~~~~~

IMPRESS

History

~~~~~

V1.0 14/8/96 Initial Release  
 V1.01 21/8/96 Bug that caused a vertical line clitch in the output just after the first tile part.  
 V1.1 23/8/96 Stretch alpha to fit feature added.  
 V1.11 1/11/96 Fixed bug that caused a crash when a grey image is used for both primary and alpha.  
 V1.12 13/1/97 Fixed a bug that caused a the alpha channel pointer to be incorrectly calculated for the first part of a scan line. This may also have caused some crashes on some machines. (A3000s namely).  
 V1.13 14/1/97 Added WINDOW\_NoBufferRP tag to window. It should fix the problems that show up under CyberGraphX screenmodes.

## 1.131 Impress

Impress

~~~~~

Synopsis:

IMPRESS <light> <dark> [STRETCH]

Function:

Creates an impression of the alpha image on the primary image. An image needs to be marked for use a the Primary, and another for the Alpha image.

Inputs:

light - Controls how strong the effect that the lighter parts of the alpha channel should have, in the range [0,256] same as on the GUI.
 dark - Controls how strong the effect that the darker parts of the alpha channel should have, in the range [0,256] same as on the GUI.
 STRETCH - If this keyword is given the alpha image will be stretched to fit the size of the primary image.

Result:

Returns the name of the newly created project in RESULT.

See also:

MARK

1.132 Local Contrast Stretch

Local Contrast Stretch

© Copyright 1996 Simon Edwards. All rights reserved.
Version: 1.0

Description

~~~~~

This applies a Local Contrast Stretch to an image. This is like a normal contrast stretch except that instead of using the whole image to calculate what the new pixel value should be, only a small neighbourhood around the pixel is used. This effectively brings out local detail.

### User Interface

~~~~~

The usual thumbnail image is in the top half of the requester, accompanied by the zoomed in preview image.

The "Width:" and "Height:" gadgets control the size of the mask used.

"Ok" applies the filter, "Cancel" cancels the whole operation.

ARexx Command

~~~~~

LOCAL\_CONTRAST\_STRETCH  
History

~~~~~

V1.0 Initial Release

1.133 Local Contrast Stretch

Local Contrast Stretch

~~~~~

### Synopsis:

LOCAL\_CONTRAST\_STRETCH <Project> <Width> <Height>

### Function:

Apply a local contrast stretch to an image.

### Inputs:

Project - Name of the project.  
Width - Width of the mask to use. Currently must be an odd number.  
(ie 1,3,5...) An even number will be rounded down to an odd number.

Height - Height of the mask to use. Currently must be an odd number. (ie 1,3,5...) An even number will be rounded down to an odd number.

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.134 LowPass

LowPass

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.0

Description

~~~~~

This applies a

lowpass

filter to an image. A lowpass filter removes the high frequencies (edges, sharp intensity transitions) in an image, thus effectively blurring it. This very useful for removing noise from an image.

User Interface

~~~~~

The usual thumbnail image is in the top half of the requester, accompanied by the zoomed in preview image.

The "Width:" and "Height:" gadgets control the size of the mask used. The larger the mask, the more blurred the result.

"Ok" applies the filter, "Cancel" cancels the whole operation.

ARexx Command

~~~~~

LOWPASS

History

~~~~~

V1.0 Initial Release

## 1.135 How exactly does it work?

How exactly does it work?

~~~~~

The values for each new pixel is calculated by averaging the values of its surrounding neighbours. So for a 3x3 lowpass filter the new value of each pixel is just the average of the nine surrounding neighbours.

```
a b c
d e f
g h i
```

In this diagram the new value of pixel e will be $(a+b+c+d+e+f+g+h+i)/9$.

The larger the filter the "heavier" and more blurred it is.

Also, by using a long thin filter, you can sort of fake a motion blur.

1.136 LowPass

```
LowPass
~~~~~
```

Synopsis:

```
LOWPASS <Project> <Width> <Height>
```

Function:

```
Apply a lowpass filter to an image.
```

Inputs:

```
Project - Name of the project.
Width   - Width of the mask to use. Currently must be an odd
          number. (ie 1,3,5...) An even number will be rounded down
          to an odd number.
Height  - Height of the mask to use. Currently must be an odd
          number. (ie 1,3,5...) An even number will be rounded down
          to an odd number.
```

Result:

```
Returns the name of the newly created project in RESULT.
```

See also:

1.137 Median Filter

Median Filter

```
© Copyright 1996-97 Simon Edwards. All rights reserved.
```

```
Version: 2.0
```

Description

```
~~~~~
```

```
This filter is extremely good at removing noise and preserving edges in
an image. It works by replacing each pixel in an image with the median
value of its neighbouring pixels.
```

User Interface

```
~~~~~
```

```
The usual thumbnail image is in the top half of the requester, accompanied by
```

the zoomed in preview image.

The "Width:" and "Height:" gadgets control the size of the mask used. The larger the mask, the more blurred the result.

The "Method:" gadget determines what algorithm should be used. Minimum uses the smallest pixel value while maximum uses the greatest.

"Ok" applies the filter, "Cancel" cancels the whole operation.

ARexx Commands

~~~~~

```

        MEDIAN
        ,
        MAXIMUM
        ,
        MINIMUM
        History

```

~~~~~

V1.0 Initial Release

V2.0 Now incorporates the maximum and minimum modules.

1.138 Maximum Filter

Maximum Filter

~~~~~

Synopsis:

```
MAXIMUM <Project> <Width> <Height>
```

Function:

Apply a maximum filter to an image.

Inputs:

```

Project - Name of the project.
Width   - Width of the mask to use. Currently must be an odd
          number. (ie 1,3,5...) An even number will be rounded down
          to an odd number.
Height  - Height of the mask to use. Currently must be an odd
          number. (ie 1,3,5...) An even number will be rounded down
          to an odd number.

```

Result:

Returns the name of the newly created project in RESULT.

## 1.139 Median Filter

Median Filter

~~~~~

Synopsis:

```
MEDIAN <Project> <Width> <Height>
```

Function:

Apply a median filter to an image.

Inputs:

Project - Name of the project.
Width - Width of the mask to use. Currently must be an odd number. (ie 1,3,5...) An even number will be rounded down to an odd number.
Height - Height of the mask to use. Currently must be an odd number. (ie 1,3,5...) An even number will be rounded down to an odd number.

Result:

Returns the name of the newly created project in RESULT.

1.140 Minimum Filter

Minimum Filter

```
~~~~~
```

Synopsis:

```
MINIMUM <Project> <Width> <Height>
```

Function:

Apply a minimum filter to an image.

Inputs:

Project - Name of the project.
Width - Width of the mask to use. Currently must be an odd number. (ie 1,3,5...) An even number will be rounded down to an odd number.
Height - Height of the mask to use. Currently must be an odd number. (ie 1,3,5...) An even number will be rounded down to an odd number.

Result:

Returns the name of the newly created project in RESULT.

1.141 Negative

Negative

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.0

Description

```
~~~~~
```

This merely takes the negative of an image.

User Interface
 ~~~~~  
 Just takes the negative of the current image.

ARexx Command  
 ~~~~~

NEGATIVE
 History

~~~~~  
 V1.0 Initial Release

## 1.142 Negative

Negative  
 ~~~~~

Synopsis:
 NEGATIVE <Project>

Function:
 Negate an image.

Inputs:
 Project - Name of the project.

Result:
 Returns the name of the newly created project in RESULT.

See also:

1.143 Noise

Noise

© Copyright 1996 Simon Edwards. All rights reserved.
 Version: 1.01

Description
 ~~~~~

This adds noise to selected components of the image.

User Interface  
 ~~~~~

In the top left is a a thumbnail with the zoomed in preview image shown to the right.

The parts of the GUI are as follows:-

Components:
 These checkboxes specify which components in the image should be

affected. Intensity, Hue, Saturation or Red, Green, Blue components may be selected.

Affect %:

This specifies the percentage of pixels that should be affected by the noise.

Type:

This lets you choose between Random pixel noise or Additive. Random noise simply replaces pixels with a new entirely random value. Additive noise affects pixels by adding a random value to the original pixel value.

Distribution:

This is only used when Additive noise is selected. This determines what distribution the noise should have. Either a simple distribution where each value is equally likely, or a gaussian distribution using the familiar bell shaped curve.

Amount %:

This is only used when Additive noise is selected. It determines the 'strength' of the additive noise.

"Ok" applies the effect, "Cancel" cancels the whole requester.

ARexx Command

~~~~~

NOISE  
History

~~~~~

V1.0 Initial Release

V1.01 4/10/96 Fixed bug in the ARexx implementation. The intensity keyword was not the first component keyword checked.

1.144 Noise

Noise

~~~~~

**Synopsis:**

NOISE <ProjectName> <affect> <<[INTENSITY] [HUE] [SATURATION]> |  
<[RED] [GEE] [BLUE]> > <RANDOM|ADDITIVE> [amount] <SIMPLE|GAUSSIAN>

**Function:**

Adds noise to the image.

**Inputs:**

ProjectName - Name of the project.  
affect - Percentage of pixels to affect.  
INTENSITY - Specifies that the intensity component should be affected.  
HUE - Specifies that the hue component should be affected.

- SATURATION - Specifies that the saturation component should be affected.
- RED - Specifies that the red component should be affected.
- GREEN - Specifies that the green component should be affected.
- BLUE - Specifies that the blue component should be affected.
- RANDOM - Specifies that Random noise should be used.
- ADDITIVE - Specifies that Additive noise should be used.
- amount - 'Strength' of the additive noise, not used if Random is specified.
- SIMPLE - Specifies that the additive noise should use a simple distribution. This keyword should only be used for Additive noise.
- GAUSSIAN - Specifies that the additive noise should use a gaussian distribution. This keyword should only be used for Additive noise.

**Result:**

Returns the name of the newly created project in RESULT.

**Examples:**

```

/* Apply Random noise to the Hue componet of 50% of the pixels */
NOISE MyProject 50 HUE RANDOM
...

/* Apply Random noise to all of the components and pixels */
NOISE MyProject 100 RED GREEN BLUE RANDOM
..

/* Apply additive noise to 50% of the pixels at strength 25% to */
/* the intensity component using a gaussian distribution. */
NOISE MyProject 25 INTENSITY ADDITIVE 25 GAUSSIAN

```

See also:

## 1.145 Posterize

### Posterize

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.0

#### Description

~~~~~

This applies a posterize effect to the image by reducing the number of bits that each of the colour components use.

User Interface

~~~~~

A preview thumbnail image is shown in the top half of the requester. The "Bits" gadget controls the number of bits that should be used for the affected

colour components in the resulting image.

The checkboxes let you specify which colour componets should be affected.

"Ok" applies the effect, "Cancel" cancels the whole requester.

ARexx Command

~~~~~

POSTERIZE

History

~~~~~

V1.0 Initial Release

## 1.146 Posterize

Posterize

~~~~~

Synopsis:

POSTERIZE <ProjectName> <Bits> [RED] [GREEN] [BLUE]

Function:

Applies a posterize effect to the image.

Inputs:

ProjectName - Name of the project.
 Bits - Number of bits each of the affected colour components should use.
 RED - Specifies that the red colour component should be affected.
 GREEN - Specifies that the green colour component should be affected.
 BLUE - Specifies that the blue colour component should be affected.

Result:

Returns the name of the newly created project in RESULT.

See also:

1.147 Reflect X

Reflect X

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.01

Description

~~~~~

This effectively "flips" an image around the y axis (left-right).

User Interface

~~~~~

Just reflects the current image.

ARexx Command

~~~~~

REFLECT\_X

History

~~~~~

V1.0 Initial Release

V1.01 8/11/96 Fixed small bug that stopped the ARexx command from returning the new project's name.

1.148 Reflect X

Reflect X

~~~~~

Synopsis:

REFLECT\_X <ProjectName>

Function:

Reflect a project on the x axis (ie left-right).

Inputs:

ProjectName - Name of the project.

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.149 Reflect Y

Reflect Y

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.01

Description

~~~~~

This effectively "flips" an image around the x axis (up-down).

User Interface

~~~~~

Just reflects the current image.

ARexx Command

~~~~~

REFLECT_Y
History

~~~~~

V1.0 Initial Release  
V1.01 8/11/96 Fixed small bug that stopped the ARexx command from returning  
the new project's name.

## 1.150 Reflect Y

Reflect Y

~~~~~

Synopsis:

REFLECT_Y <Project>

Function:

Reflect a project on the y axis (ie up-down).

Inputs:

Project - Name of the project.

Result:

Returns the name of the newly created project in RESULT.

See also:

1.151 Remove Feature

Remove Feature

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.0

Description

~~~~~

This allows you to remove a circular feature from the image.

User Interface

~~~~~

There will be a short delay before the Remove Feature requester is opened. This is to create the small preview image that's shown to the right of the requester. On the thumbnail image, the area that is to be affected is diagrammatically shown. The gadgets are:-

Radius:

The radius of the circle of pixels that will be stretched to fill the hole.

Remove:

The radius of the feature to be removed.

X:

The x position of the center of the feature to be removed.

Y:

The y position of the center of the feature to be removed.

Method:

Determines whether the remove should be done using a fast "Nearest Neighbour" algorithm or the better (but slower) "Colour Average" algorithm.

The center can be changed by clicking on the thumbnail image.

"Ok" applies the effect, while "Cancel" cancels the whole operation.

ARexx Command

~~~~~

REMOVE\_FEATURE

History

~~~~~

V1.0 Initial Release

1.152 Remove Feature

Remove Feature

~~~~~

Synopsis:

REMOVE\_FEATURE <Project> <X> <Y> <Radius> <Remove> <FAST | BEST/ ↔  
COLOUR\_AVERAGE>

Function:

Remove part of an image.

Inputs:

|                     |                                         |
|---------------------|-----------------------------------------|
| Project             | - Name of the project.                  |
| X                   | - X position of the center of the area. |
| Y                   | - Y position of the center of the area. |
| Radius              | - Radius of the area to be stretched.   |
| Remove              | - Radius of the area to be removed.     |
| FAST                | - Use fast nearest neighbour algorithm. |
| BEST/COLOUR_AVERAGE | - Use colour average algorithm.         |

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.153 Resize

Resize

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.02

Description

~~~~~

This lets you change the size of an image by padding it out with black (ie by **not** scaling it).

User Interface

~~~~~

The "Width" and "Height" gadgets allow you to specify the dimensions of the new image.

The "Presets" allow easy access to some commonly used sizes. The bottom four buttons contain the last four used sizes.

The "Center" checkbox controls whether in the image should be centered in the new image.

The "Tile" checkbox controls whether the image should be tiled to fill resulting image instead of filling it with black.

Resize is particularly useful if you're trying to import an image into a program that's very picky about what size images it will take.

"Ok" proceeds to resize the image, while "Cancel" cancels the requester.

ARexx Command

~~~~~

RESIZE

History

~~~~~

V1.0 Initial Release

V1.01 9/7/96 Small bug that caused CENTER keyword not to work properly from ARexx. ->Patrik Nydensten

V1.02 9/8/96 Stopped the interface from accepting values <1.

V1.03 3/11/96 Fixed bug that causes it to crash when resizing an image to a smaller size. ->Patrik Nydensten

## 1.154 Resize

Resize

~~~~~

Synopsis:

RESIZE <Project> <Width> <Height> [CENTER] [TILE]

Function:

Resize an image.

Inputs:

Project - Name of the project to resize.
Width - New width of image.
Height - New height of image.
CENTER - If this keyword is given, the will centered.
TILE - If this keyword is given, the image is tiled to fill the new image.

Result:

Returns the name of the newly created project in RESULT.

See also:

1.155 Ripple

Ripple

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.0

Description

~~~~~

This a applies a ripple affect to the image. The ripples can go either around the centre or out from the centre.

User Interface

~~~~~

The area to be affected and the centre of the effect is diagramaticly shown on the thumbnail image to the right. The center of the effect can be changed by clicking on the thumbnail image.

The thumbnail preview of the result is shown on the far right of the requester.

The gadgets are:

X:

The x position of the center of the effect in the image.

Y:

The x position of the center of the effect in the image.

Amplitude:

The amplitude of the ripples, if the type of the ripples is set to "Around Centre" then this is the number of degrees that the ripples will move around the center. Otherwise the amplitude determines the the "strength" of the ripples when in "Out from Centre".

Period:

The period in pixels of one ripple to the next.

Phase:

The phase of the ripples. This is control is more useful when

creating animations.

Radius:

If the ripples are attenuated, this determines the cut-off radius of the ripples.

Attenuated:

If this checkbox is set then the ripples will be attenuated, that is they will "die" or "fade" out with distance from the centre.

Type:

This determines whether the ripples will go "Around Centre" or "Out from Centre".

Method:

Determines whether the twirl should be done using a fast "Nearest Neighbour" algorithm or the better (but slower) "Colour Average" algorithm.

"Ok" applies the effect, while "Cancel" cancels the requester.

ARexx Command

~~~~~

RIPPLE  
History

~~~~~

V1.0 22/6/96 Initial Release

1.156 Ripple

Ripple

~~~~~

Synopsis:

RIPPLE <Project> <X> <Y> <Amplitude> <Period> <Phase> <AROUND|OUT>  
<FAST | BEST/COLOUR\_AVERAGE> [Radius]

Function:

Apply a ripple effect to the image.

Inputs:

|                     |                                                             |
|---------------------|-------------------------------------------------------------|
| Project             | - Name of the project to resize.                            |
| X                   | - x co-ord of the centre of the effect.                     |
| Y                   | - y co-ord of the centre of the effect.                     |
| Amplitude           | - The amplitude of the ripples, [0-359]                     |
| Period              | - The period of the ripples.                                |
| Phase               | - The phase of the ripples. [0-359]                         |
| AROUND              | - Specifies that the ripples should go around the centre.   |
| OUT                 | - Specifies that the ripples should go out from the centre. |
| FAST                | - Use fast nearest neighbour algorithm.                     |
| BEST/COLOUR_AVERAGE | - Use colour average algorithm.                             |
| Radius              | - If given this specifies that the ripples should           |

be attenuated and that this is the radius to use.

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.157 Rotate

Rotate

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.01

Description

~~~~~

This lets you rotate the image.

User Interface

~~~~~

A preview thumbnail image is in the top half of the requester.

The degree gadget lets you enter the degrees to rotate the image clockwise by. The slider lets you change the number of degrees from 0 to 359.

The "Method" gadget controls whether the image should be rotated using a "Fast" algorithm or the better (but slower) "Colour Average" algorithm.

"Ok" proceeds to rotate the whole image, "Cancel" cancels the whole operation.

ARexx Command

~~~~~

ROTATE

History

~~~~~

V1.0 Initial Release

V1.01 26/5/96 Fixed bugs in the rotate code that caused black edges in the resulting images.

## 1.158 Rotate

Rotate

~~~~~

Synopsis:

ROTATE <Project> <Degrees> <FAST | BEST/COLOUR_AVERAGE>

Function:

Scale an image to a new size.

Inputs:

Project - Name of the project to scale.
 Degrees - Degrees to rotate clockwise by.
 FAST - Use fast nearest neighbour algorithm.
 BEST/COLOUR_AVERAGE - Use colour average algorithm.

Result:

Returns the name of the newly created project in RESULT.

Example:

```
/* Rotate image 45 degrees fast*/
'ROTATE' MyProject '45 FAST'
MyProject45=RESULT
```

See also:

1.159 Saturation

Saturation

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.01

Description

~~~~~

This increases or decreases the saturation of the given image.

#### User Interface

~~~~~

The saturation requester is split over two panels (which can be switched between by using the tabs at the top of the requester). At the bottom of the requester is a slider and integer gadget for changing the saturation of the image.

The first panel contains a preview image to show the new saturation.

The second panel shows a preview of the new histogram for the image, along with a collection of statistics for the histogram.

Clicking on "Ok" makes the changes, "Cancel" cancels the whole requester.

ARexx Command

~~~~~

#### SATURATION

History

~~~~~

V1.0 Initial Release

V1.01 Fixed small bugs that caused the window to not open and complain about not having enough memory.

1.160 Saturation

Saturation
~~~~~

### Synopsis:

```
SATURATION <ProjectName> <Value>
```

### Function:

Change the saturation of an image.

### Inputs:

```
ProjectName - Name of the project.  
Value       - Value to change saturation by.
```

### Result:

Returns the name of the newly created project in RESULT.

### See also:

## 1.161 Scale

Scale

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.02

Description  
~~~~~

This lets you scale an image to a new size.

User Interface
~~~~~

The gadgets on the scale requester are as follows:-

### Absolute Width:

The width to scale to in pixels.

### Absolute Height:

The height to scale to in pixels.

### Percent Width:

Percentage to scale the width by. The slider will allow values from 25% to 200%, greater or less values can be entered into the integer gadget.

### Percent Height:

Percentage to scale the height by. The slider will allow values from 25% to 200%, greater or less values can be entered into the integer gadget.

### 50%, 100%, 200%:

These just set the percent width and height to 50%, 100% and 200% respectively.

---

**Lock Aspect Ratio:**

This forces the Width and Height to be scaled by the same amount. This preserves the aspect ratio and stops images from becoming stretched or shrunk too much in any direction.

**Method:**

This controls whether the image should be scaled using a "Fast" algorithm or the better (but slower) "Colour Average" algorithm. The "Colour Average" algorithm helps prevent images from becoming too "blocky" when scaled up.

Clicking on "Ok" scales the image, "Cancel" cancels the requester.

**ARexx Command**

```
~~~~~
```

```
SCALE
History
```

```
~~~~~
```

```
V1.0 Initial Release
V1.02 14/11/96 Increased the range of the sliders to 10%-400%.
```

## 1.162 Scale

**Scale**

```
~~~~~
```

**Synopsis:**

```
SCALE <Project> <Width> <Height> <FAST | BEST/COLOUR_AVERAGE>
```

**Function:**

Scale an image to a new size.

**Inputs:**

```
Project - Name of the project to scale.
Width - New width to scale to.
Height - New height to scale to.
FAST - Use fast nearest neighbour algorithm.
BEST/COLOUR_AVERAGE - Use colour average algorithm.
```

**Result:**

Returns the name of the newly created project in RESULT.

**Example:**

```
/* Scale image down to postage stamp size */
'SCALE' MyProject '80 50 FAST'
PostageStamp=RESULT
```

**See also:**

## 1.163 Sharpen

Sharpen

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.0

Description

~~~~~

This sharpens the image. This is extremely useful for bring out detail in a bad blurry image.

User Interface

~~~~~

The thumbnail image and preview image are in the top half of the requester.

The percentage gadget controls how much sharper the image should be made. The greater the percentage the greater the effect. (BTW, the sharpened image is calculated by taking the original and adding a fraction of the highpass result to it. The percentage actually refers to the fraction of the highpass result to add in.)

The "Width" and "Height" gadgets control the size of the mask used. The smaller the mask, the finer the detail that is sharpened.

Clicking on "Ok" applies the filter, whie "Cancel" cancels the whole requester.

I find that this used in conjunction with a lowpass filter is very good at removing high frequency noise, by first applying a small (3x3) sized lowpass filter to the image and then sharpening it to bring out the detail again.

ARexx Command

~~~~~

SHARPEN

History

~~~~~

V1.0 Initial Release

## 1.164 Sharpen

Sharpen

~~~~~

Synopsis:

SHARPEN <Project> <%>

Function:

Apply a sharpening filter to an image.

Inputs:

Project - Name of the project to sharpen.

% - Percentage to sharpen by.

Result:

Returns the name of the newly created project in RESULT.

See also:

## 1.165 Threshold

Threshold

© Copyright 1996 Simon Edwards. All rights reserved.

Version: 1.0

Description

~~~~~

This thresholds the image, turning it into a black and white image.

User Interface

~~~~~

The threshold requester is split across two panels. The first shows a thumbnail of the image, and a zoomed in preview image.

The "Threshold" gadget controls what the threshold levels is.

The second panel shows the histogram of the image and what the threshold level is. The threshold can be set by clicking on the histogram.

Clicking on "Ok" applies the threshold, everything that's below will be come black, else it'll become white. "Cancel" cancels the whole requester.

ARexx Command

~~~~~

THRESHOLD

History

~~~~~

V1.0 Initial Release

V1.01 14/11/96 Changed the way it created the new project.

## 1.166 Threshold

Threshold

~~~~~

Synopsis:

THRESHOLD <Project> <Level>

Function:

Apply a threshold to an image.

**Inputs:**

Project - Name of the project to threshold.  
Level - Grey level threshold in the range of 0 to 255 inclusive.

**Result:**

Returns the name of the newly created project in RESULT.

**See also:**

## 1.167 Twirl

### Twirl

© Copyright 1996 Simon Edwards. All rights reserved.  
Version: 1.0

**Description**

~~~~~

This lets you twirl or twist if you like, part of an image.

**User Interface**

~~~~~

There will be a short delay before the twirl requester is opened. This is to create the small preview image that's shown on the right of the requester.

The area to be affected is diagramatically shown on the thumbnail image. The center of the twirl can be changed by clicking on the thumbnail image.

The gadgets are:

**Radius:**

The radius of the twirl effect.

**Degrees:**

The number of degrees clockwise that the image should be twirled, ranging from 360 anticlockwise to 360 degrees clockwise.

**X:**

The x position of the center of the twirl effect in the image.

**Y:**

The y position of the center of the twirl effect in the image.

**Method:**

Determines whether the twirl should be done using a fast "Nearest Neighbour" algorithm or the better (but slower) "Colour Average" algorithm.

Click on "Ok" to apply the effect, else "Cancel" to cancel the requester completely.

**ARexx Command**

~~~~~

TWIRL

---



## History

~~~~~

V1.0 Initial Release

**1.168 Twirl**

Twirl

~~~~~

## Synopsis:

TWIRL &lt;Project&gt; &lt;X&gt; &lt;Y&gt; &lt;Radius&gt; &lt;Degrees&gt; &lt;FAST | BEST/COLOUR\_AVERAGE&gt;

## Function:

Twirl part of an image.

## Inputs:

Project - Name of the project to twirl.  
 X - X position of the center of the twirl.  
 Y - Y position of the center of the twirl.  
 Radius - Radius of the twirl.  
 Degrees - Number of degrees to twirl clockwise by.  
 FAST - Use fast nearest neighbour algorithm.  
 BEST/COLOUR\_AVERAGE - Use colour average algorithm.

## Result:

Returns the name of the newly created project in RESULT.

## See also:

**1.169 ARexx**

## ARexx

~~~~~

Image Engineer has an ARexx port called IMAGEENGINEER and a public screen with the name IMAGEENGINEER. Most menu items have a corresponding ARexx command. You'll find that function and input descriptions are usually quite brief, read the corresponding menu description for a more in-depth view of how a particular operation works and what the inputs are.

Script Writing Tips

&lt;- Please Read

Supplied Scripts

Command List

Note for Users upgrading from 2.1 or lower

## 1.170 Script Writing Tips

### Script Writing Tips

~~~~~

- \* Options Results should be turned on to get results from commands. (Use the line "Options Results" at the start of your scripts).
- \* Image Engineer's commands return 0 in RC on success, 5 when the user aborts or cancels a command (by clicking on the "Abort" button or selecting cancel if it's a requester). If the commands fails due to bad parameters, out of memory etc 10 is returned. Use the command  

```
LAST_ERROR
```

to get a string describing the error.
- \* ARexx scripts can be invoked from inside by using the "Execute..." menu item on the ARexx menu, or by choosing a user defined menu item from the ARexx menu. Scripts started this way automatically have their command host set to IMAGEENGINEER. They're also called with argument one containing the name of the currently active project, or nothing if the script was invoked with no project active (ie before the user has even opened an image).

It's worth checking to see if your script has been passed a project name to operate on, with a line like this:-

```
if arg()==0 then
do
/* If we need a project to operate on we could now */
/* open a file requester and let the user choose an */
/* image from disk, or put up a message saying that we */
/* need a project, or we could just quietly exit now. */
end
```

- \* Be aware of how ARexx interprets quotes in command clauses. For example.

```
CLOSE 'bono.iff' & 'CLOSE bono.iff'
```

will send the string

```
CLOSE bono.iff
```

to IE, as ARexx interprets the outside quotes. This is important to keep in mind when you're trying to send a command with a file name that contains a space.

For example.

```
'OPEN "sys:pics/bono pic"'
```

which would send the string

```
OPEN "sys:pics/bono pic"
```

to IE.

If you've got a file name stored in a string (that you may have requested from the user) and you want to use it with a command, you should make sure that the name is enclosed in quotes when it's send to IE, as you can't be sure that it contains no spaces.  
For example.

```
MyFile='sys:pics/bono pic'
```

```
'OPEN "||MyFile||"'
```

this would send

```
OPEN "sys:pics/bono pic"
```

to IE. Doing it like this

```
'OPEN '||MyFile
```

would not work if the file name contains a space.

\* IE also has several special commands for use in ARexx scripts.

Screen Commands:-

IE\_TO\_FRONT

WB\_TO\_FRONT

Error Command:-

LAST\_ERROR

User Input Commands:-

FORM

GET\_DIR

GET\_FILE

GET\_FILES

GET\_FILE\_TYPE

GET\_NUMBER

GET\_STRING

REQUEST

Task Priority Commands:-

GET\_PRI

SET\_PRI

GET

PROJECT\_LIST

---

PROJECT\_SET

## 1.171 Supplied Scripts & Macros

Supplied Scripts & Macros

~~~~~

To start any of the scripts RexxMast needs to be running as well as Image Engineer.

The following scripts and macros can be found in the IE's ARexx directory and can be invoked by using the Execute menu item on the ARexx menu.

AddSpace.rexx

This script adds working space around picture.

AddSpaceB.rexx

This script adds working space to one side of picture.

AddTexture.rexx

This script adds tileable texture as a frame around picture.

Adjust by Patrik Nydensten

Cuts the image into either vertical or horizontal sections and 'slides' every second one a given amount.

Anitique

Changes the colours in the image so that it looks like one of those old yellow photos that you usually find at your grandparent's place. IYKWIM ;- ) Looks best if applied to a grey image.

BatchProcess by Patrick Nydensten

See IE\_Batch\_System.guide in Extra/docs directory.

Bleed by Patrik Nydensten

Simulates TV colour bleeding.

CloseAll.rexx

This script closes all picture-windows at once, except active one marked as ALPHA.

CropEdge.rexx

This script crops a part of picture by excluding 1 pixel row/column from every side. Autocrop is done before actual cropping.

ContStretchRegion

Drag out a box on a project, and call this script to contrast stretch the marked out region.

DeInterlace.rexx

This script will separate image from interlaced image line by line, producing 2 new images.

DisplacePixels

---

Randomly moves the pixels in the image by up to a given number of pixels.

#### EdgeEffect

This tries to create an alpha channel image for the current image that can be later used for compositing. It asks you for an image that can be used for the edge pattern. Try using a background 'wallpaper' image for this. I get good results using cement and granite images.

#### EdgeMerge by Patrik Nydensten

Merge image's top and left edges with bottom and right edges to form a seamless pattern image.

#### ExtractComponent

Extracts a primary colour component from a colour image.

#### FadeInAnim

This script takes an image of a title over a black background and composites it over another image, fading it in over a given number of frames. By compiling the resulting frames you can create an animation of your title fading in.

#### FilterPepper

Filters black pixels from the image. Only affects black pixels, does a much better job than any one filter.

#### FilterSalt

Filters white pixels from the image. Only affects white pixels.

#### FitSelectAlpha

Creates an alpha channel image the same size as the current image. Lets you select the kind of alpha image (horizontal gradient, highlight etc)

#### Fresco by Patrik Nydensten

Builds a fresco version of primary image. Similar to the ADPro fresco script.

#### Halftone

This lets you painlessly apply one of several halftone patterns to an image.

#### ImageFX\_Export

This exports the current image to ImageFX. ImageFX must be running at the same time.

#### ImageFX\_Export\_Render

This exports the current rendered image to ImageFX. ImageFX must be running at the same time.

#### ImageFX\_Import

This imports the current image in ImageFX into IE. ImageFX must be running at the same time.

#### Interlace.rexx

This script will combine 2 images line by line. The lines are interleaved.

---

Merge.rexx

This script merges two pictures together.

MotionBlur.rexx

Applies a horizontal or vertical motion blur effect to an image.

Neon by Patrik Nydensten

Creates neon light lines out of image's color edges.

OilPaint

This fakes a sort of oil painting effect.

OpenFiles.rexx

Lets you select and open multiple images in one go.

PageStream\_Export by Volkmar Funke

PageStream\_Export\_E

Exports the current picture into PageStream3.

PixelizeRegion

Pixelizes (makes blocky) the region marked out on the current image.

Preserve.rexx by Patrik Nydensten

Converts an image to grey while preserving a selected basecolour.

PrimaryColors.rexx

Makes a colorful picture gray and then tints it with a colour.

Roll.rexx by Marko Seppänen

Rolls whole image when there's no cropbox at all or  
cropbox's width or height is smaller than 5 pixels.

When cropbox is large enough, then area of cropbox is rolled

Rotate.rexx

This script rotates picture and uses white background instead of  
black, which is build in Rotate-operator.

RotateBlur.rexx

Makes the image appear as though it's spinning.

SaveAll.rexx

This script saves all picture-windows to a selected dir, using  
selected rendering options.

SaveIT.rexx}

This script saves activate picture window to a selected dir in  
selected 24 bit fileformat. It will never overwrite files.

Scale50.rexx

Scales an image by 50%.

Scale200.rexx

Scales an image by 200%.

ScanImage by Volkmar Funke

ScanImage\_E

---

Scans in an image using the ScanQuix software which must be running on the Workbench.

Shear by Patrik Nydensten

Shears the image along the the X or/and Y-axis.

ShiftRGB by Patrik Nydensten

Swaps/copies brightness data in the R, G and B channels.

SlideShowIE by Don Cox

Recursively displays on screen a whole directory of images.

Solarize.rexx improved by Patrick Nydensten

Applies one of 3 different solarize effects to the given image, same as in photography.

Sparkle.rexx

Adds little 'spark-thingys' to the spark edges of a image.

Thicken.rexx

This script is for use with black and white images. It makes thicker the black or white pixels in the image.

Vigette.rexx

This applies a blurred Vigette effect to an image, making it appear blurred around the edges.

VigetteGrey.rexx

This is the same as above except the image appears in grey around the edges.

WetInk.rexx

This takes a black on white image, and makes it appears that the black ink has run, by using a convolve.

Zoom14.rexx

This script sets zoom of activate window to 1:4.

ZoomBlur.rexx

Makes an image appear as though you're moving towards the image at speed. (Try it on a space image using MAX as the mix method, for that jump-to-lightspeed effect (for the Star Wars fans)).

## 1.172 ARexx Commands

ARexx Commands

~~~~~

Command

Location

AUTCROP

Autocrop module

BRIGHTNESS

Brightness module

---

BULGE		Bulge module
CLOSE		IE
CLOSE_RENDER	IE	
COLOUR_FILTER		Colour Filter module
COMPOSITE		Composite module
CONTRAST		Contrast module
CONTRAST_STRETCH		Contrast Stretch module
CONVERT_TO_COLOUR		Convert to Colour module
CONVERT_TO_GREY		Convert to grey module
CONVOLVE		Convolve module
CROP		IE
DISPLACE		Displace module
FALSE_COLOUR		False colour module
FORM		ARexx Support module
GAMMA		Gamma module
GET		IE
GET_DIR		ARexx Support module
GET_FILE		ARexx Support module
GET_FILES		ARexx Support module
GET_FILE_TYPE		

---



AREXX Support module

GET\_NUMBER  
AREXX Support module

GET\_STRING  
AREXX Support module

HALFTONE  
Halftone module

HIGHBOOST  
Highboost module

HIGHPASS  
Highpass module

HISTOGRAM\_EQUALIZATION  
Equalisation module

HUE  
Hue module

HUE\_MASK  
Hue Mask module

IE\_TO\_FRONT  
IE

IMPRESS  
Impress module

LAST\_ERROR  
IE

LOCAL\_CONTRAST\_STRETCH  
Local Contrast Stretch module

LOWPASS  
Lowpass module

MARK  
IE

MAXIMUM  
Median module

MEDIAN  
Median module

MINIMUM  
Median module

NEGATIVE  
Negative module

NOISE

---

## Noise module

OPEN  
IE

OPEN\_CLIPBOARD  
IE

OPEN\_DEVICE  
IE

PALETTE\_LOAD  
IE

PALETTE\_SAVE  
IE

POSTERIZE  
Posterize module

PRIORITY\_GET  
IE

PRIORITY\_SET  
IE

PROJECT\_INFO  
IE

PROJECT\_LIST  
IE

PROJECT\_SET  
IE

QUIT  
IE

REMOVE\_FEATURE  
Remove feature module

RENDER  
IE

RENDER\_COLOURS  
IE

RENDER\_DEPTH  
IE

RENDER\_DEVICE  
IE

RENDER\_DITHER  
IE

RENDER\_GET

---

	IE
RENDER_PALETTE	IE
RENDER_SCREENMODE	IE
RENDER_SET	IE
RENDER_TO_FRONT	IE
REFLECT_X	Reflect X module
REFLECT_Y	Reflect Y module
REQUEST	ARexx Support module
RESIZE	Resize module
RIPPLE	Ripple module
ROTATE	Rotate module
SATURATION	Saturation module
SAVE	IE
SAVE_CLIP	IE
SAVE_DEVICE	IE
SAVE_DATA	IE
SAVE_DATA_CLIP	IE
SAVE_DATA_DEVICE	IE
SCALE	Scale module
SHARPEN	

---

```

 Sharpen module
THRESHOLD Threshold module
TWIRL Twirl module
WB_TO_FRONT IE

```

## 1.173 Changes from 2.1

Changes to the ARexx commands from 2.1

~~~~~

In the transition to 3.0 many of the ARexx commands have been changed to make the command set more logical and easier to use.

| Built in Command  | Notes                                |
|-------------------|--------------------------------------|
| CLOSE_RENDER      | Renamed to RENDER_CLOSE              |
| GET_PRI           | Renamed to PRIORITY_GET              |
| GET_RENDER        | Renamed to RENDER_GET, changed       |
| LOAD_PALETTE      | Renamed to PALETTE_LOAD              |
| OPEN_CLIPBOARD    | Changed and extended                 |
| OPEN              | Changed and extended                 |
| RENDER_AUTOSCROLL | Removed                              |
| RENDER_QUANTIZE   | Renamed to RENDER_PALETTE            |
| SAVE_PALETTE      | Renamed to PALETTE_SAVE              |
| SET_PRI           | Renamed to PRIORITY_SET              |
| SET_RENDER        | Renamed to RENDER_SET, changed       |
| TYPE              | Removed, see OPEN and OPEN_CLIPBOARD |

| Module Command | Notes   |
|----------------|---------|
| RESIZE         | Changed |

## 1.174 Get

Get  
~~~

Synopsis:

```
GET <PRIMARY|SECONDARY|ALPHA>
```

Function:

Get the name of the image currently marked as being primary, secondary or alpha.

Input:

```
PRIMARY - Get the name of the primary image.
SECONDARY - Get the name of the secondary image.
```

ALPHA - Get the name of the alpha image.

Result:

Returns the name of the requested image in RESULT. If no image as been marked as the given image type, then a null string will be returned in RESULT.

Menu equivalent:

None.

See also:

## 1.175 Get File Type

Get File Type

~~~~~

Synopsis:

GET\_FILE\_TYPE [<Title>]

Function:

Get a save file format name from the user.

Inputs:

Title - Optional title of the requester.

Result:

Returns the name of a valid save file format in RESULT.  
If the user cancels the requester, 5 will be returned in RC.

Example:

```
'GET_FILE_TYPE "Select File Format"' /* Get a save file format */
SaveFormat=RESULT

/* Save MyProject using SaveFormat */
'SAVE' MyProjectName ' '||SaveFileName||" " '||SaveFormat||' "'
```

Menu equivalent:

None.

See also:

```
GET_FILE
'
GET_FILES
'
SAVE
```

## 1.176 Image Engineer to Front

## Image Engineer to Front

~~~~~

## Synopsis:

IE\_TO\_FRONT

## Function:

Move IE's screen to the front of the display.

## Inputs:

None.

## Result:

None.

## Menu equivalent:

None.

## See also:

WB\_TO\_FRONT

**1.177 Last Error**

## Last Error

~~~~~

## Synopsis:

LAST\_ERROR

## Function:

Get a string describing why the last error occurred. This is useful in error handler routines.

## Inputs:

None.

## Result:

Returns an error string in RESULT.

## Example:

```

/* Example showing use of LAST_ERROR */
/* I recommend that you use this code in your scripts */

Options Results
Signal On Error /* Tell Rexx that we have an error routine */

... /* Do work here... */

exit

/*****
/* This is our error handling routine. When an error occurs */

```

```

/* program execution goes here. This routine gets the error */
/* string from IE and displays a message saying that an */
/* error has occurred, what the error message is and on what */
/* line of the script it occurred on. */
/*****/
Error:

IE_TO_FRONT /* Move IE to the front of the display */
LAST_ERROR /* Get the error string */
 /* Display an error message */
'REQUEST "Error detected!!!'||D2C(10)||'Error message is as follows'||D2C ←
(10)||result||D2C(10)||'Script failed on line '||SIGL||'" 'Doh!'
Exit

```

Menu equivalent:  
None.

See also:

## 1.178 Get Task Priority

Get Task Priority

~~~~~

Synopsis:

PRIORITY\_GET

Function:

Get Image Engineer's task priority.

Inputs:

None.

Result:

Returns IE's task priority in RESULT.

Example:

```

/* Ask the user how many frames they want generated */
'PRIORITY_GET'
TaskPri=RESULT

say 'Image Engineers task priority currently is ' TaskPri
...

```

Menu equivalent:

None.

See also:

PRIORITY\_SET

## 1.179 Set Task Priority

Set Task Priority

~~~~~

### Synopsis:

```
PRIORITY_SET <Priority>
```

### Function:

Set Image Engineer's task priority.

### Inputs:

Priority - IE's new task priority. In the range of -20 to 20.  
(Same as the AmigaDOS command ChangeTaskPri, except that the range has been limited, instead of the usual -128 to 127 range).

### Result:

Returns the old task priority in RESULT.

### Example:

```

 /* Reduce our Task Priority so that the user */
 /* can get on with other work in the foreground.
'PRIORITY_SET -1'
OldPri=RESULT

... /* Do some serious number crunching here */

'PRIORITY_SET ' || OldPri /*Finished work! restore old priority */

```

### Menu equivalent:

None.

### See also:

PRIORITY\_GET

## 1.180 WB to Front

WB to Front

~~~~~

### Synopsis:

```
WB_TO_FRONT
```

### Function:

Move the Workbench screen to the front of the display.

### Inputs:

None.

### Result:



None.

Menu equivalent:

None.

See also:

IE\_TO\_FRONT

## 1.181 Use with Martin Apel's VMM

Use with Martin Apel's VMM

~~~~~

Image Engineer allocates memory for image data with the MEMF\_PUBLIC flag clear. The only problem is that superview.library's sobjects usually don't allocate memory with the MEMF\_PUBLIC flags (from what I can tell), which means that you can still run out of memory when loading and saving even if you have heaps of VM still left. All I did to get around this was to use the advanced options with 10240 set for both public and non-public allocations. Works great.

## 1.182 Hints, tips & FAQ

What should I be aware of when using FastIPrefs, MCP and other patches?

Users of FastIPrefs beware that on at least some machines running IE with FastIPrefs has been known to cause crashes. If you have trouble running IE, try removing FastIPrefs and try again.

The same goes for MCP. If you are having trouble with IE, try removing MCP and try again. If it starts working fine you can either get rid of MCP or try and find out which feature of MCP is causing the problem by turning them back on one by one and testing it against IE each time. If you identify a particular feature of MCP that seems to be causing problems email my and let me know.

My 24bit IFF images are being loaded and corrupted with this ugly fringing.

This is due to Superview's FastILBM24 sobject which is converting your 24bit IFF image into a HAM6 or HAM8 image along with the fringing inherent to HAM images. To stop this from happening go to the Project->Prefs->SObject menu item and bring up the options for the FastILBM24 sobject. Now make sure that PASSTHROUGH is selected.

When I use my scanner, the scanner window appears on the WB screen.

Go to the Project->Prefs->Open Device menu item, open the options for the scanner and set the name of the public screen for it to use to IMAGEENGINEER and save it. From now on the scanner window will now open on IE's screen.

## 1.183 register

How to Register

~~~~~

IE costs only 25 USD (145 FIM, 25 EUR, 49 DM) and so it is the most powerful and cheapest piece of graphics manipulation software available for Amiga platform. Registered users will get a personal key file, which will unlock the limit on image size. Remember that it's a personal keyfile, so you can NOT copy or give it to a friend or everybody at the office. Companies, which need more than one keyfile should contact Marko Seppänen <marko.seppanen@wwnet.fi> and tell him about your needs.

Registration methods:

CREDIT CARD (service of RegNet)

Place credit card orders using URL, phonenumber or faxnumber below:

URL - <http://www.reg.net/product.asp?ID=5117>

Phone - 1 800 WWW2REG (1 800 999-2734) or (661) 288-1827

Fax - 1 (661) 288-1867

Please refer to the RegNet main page for general information on transactions or if you are unsure about security of paying using a credit card:

<http://www.reg.net>

Credit card ordering is very fast method. Normally you will receive your personal keyfile in couple of hours by email. ~

CASH, CHECK AND MONEY ORDER (service of RegNet)

In addition to credit cards, RegNet can accept cash, check or money order. Be sure to include all of your contact information with your payment. Make checks (drawn on a US bank) payable to RegNet and send to the following address:

RegNet (Wintronix, Inc.)  
24303 Walnut Street, Suite 200  
Newhall, CA 91321-2900  
U.S.A.

When your check arrives, they will process your registration and fulfill the order. Place the order using URL below.

<http://www.reg.net/product.asp?ID=5117>

---

## PAYMENT ORDER: SWIFT AND EUROGIRO

As a registration method, both methods are secure and fast (4 days max from foreign account to account here in Finland). SWIFT-network is world wide, but EUROGIRO is cheaper method for people living in countries, which have joined to EUROGIRO-network. Currently (02.10.1999) these countries have joined: Belgium, Spain, Netherlands, Island, Great Britain, Italy, Austria, Japan, Luxemburg, Norway, France, Sweden, Germany, Finland, Switzerland and Denmark.

1.

Fill up the Registration form and print it.

You can also write it all by hand, if you don't have a printer.

2.

Tell your bank to pay the registration fee in EUR amounts via EUROGIRO or SWIFT to account number 800018-70698769 with Leonia Bank plc (former Postipankki Ltd.), Helsinki, Finland.

Swift address: PSPBFIHH; Telex 121 698 pgiro sf

3.

Send filled registration form together with copy of receipt to this address:

Marko Seppänen  
Maininkikatu 8 as 8  
53920 Lappeenranta  
Finland / Europe

You will receive a personal keyfile by email or by mail on diskette, which unlocks the limit on imagesize. If email isn't possible, add 15 FIM or 3 EUR for postal costs.

## TILISIIRTO (for people living in Finland only)

Tilisiirto on Suomessa kätevin ja halvin registeröintitapa.

1.

Täytä Registeröintikaavake ja tulosta se. Voit myös kopioida tiedot paperille itse, jos sinulla ei ole printteriä käytössäsi.

2.

Siirrä tililtäsi registeröintisumma tilille 800021-16142208.

3.

Lähetä täytetty registeröintikaavake yhdessä kuitin kanssa osoitteeseen:

Marko Seppänen  
Maininkikatu 8 as 8  
53920 Lappeenranta

Tulet saamaan sähköpostissa tai normaalissa postissa disketillä,

---

henkilökohtaisen avaimen, joka poistaa kuvakoon rajoituksen. Jos sähköposti ei ole käytettävissäsi, lisää 5 FIM postimaksuihin.

#### CASH

1.  
Fill up the Registration form and print it.  
You can also write it all by hand, if you don't have a printer.
2.  
Exchange amount of the registration cost to FIM or EUR.
3.  
Send filled registration form together with cash to this address:

Marko Seppänen  
Maininkikatu 8 as 8  
53920 Lappeenranta  
Finland / Europe

You will receive a personal keyfile by email or by mail on diskette, which unlocks the limit on imagesize. If email isn't possible, add 15 FIM or 3 EUR for postal costs.

#### GERMAN registration SITE

Andreas Kleinert is handling registrations of Image Engineer for people living in Germany.

1.  
Fill up the Registration form and print it.  
You can also write it all by hand, if you don't have a printer.
2.  
Acceptable registering methods are:

- Cash-money in DM
- Bank transfer to his account:  
Deutsche Bank Siegen, BLZ 46070090 Kto. 0298174

3.  
Send filled registration form together with copy of receipt or registration fee to this address:

Andreas R. Kleinert  
Am Kornberg 48  
D-57076 Siegen  
Germany / Europe

You will receive a personal keyfile by email or by mail on diskette, which

---

unlocks the limit on imagesize. If email isn't possible, add 2 DM for postal costs.

## 1.184 Reporting Bugs

Reporting Bugs

~~~~~

It'll happen one day that you're using IE and you'll ask it to do something quite reasonable and it'll hang, crash, or do something equally \*evil\*.

Before you reach for your mailer, make sure that you've read all of the documentation \*and\* the superview.library documentation, (especially the "Known Bugs and program limits:". Chances are that it's already been documented.

If you still haven't found a solution or answer, then it's probably time to let me know about it.

When reporting bugs give a detailed report of what the bug is, what conditions it failed under, what exactly you were doing and what options you had set. If it's reproducible, explain how to reproduce it. Also specify what version of IE, OS (Workbench version \*and\* Kickstart version, this information is very important), superview.library, superviewsupport.library you're using. And don't forget to specify what your system consists of and what expansions, hard drives, accelerators, extra ram, graphics boards, etc you also have, along with what other software you may have been running at the time (including commodities, patches, background utilities).

Report bugs to Marko Seppänen <marko.seppanen@wwnet.fi>

## 1.185 Thanks and Greetings

Thanks and Greetings

~~~~~

Thanks and greetings from Simon go to the following people in no particular order...

- \* All programmers of excellent Amiga software.
  - \* Everyone who has 'kept the faith' and stuck with the Amiga.  
(There is really nowhere to go.)
  - \* Andreas R. Kleinert for the Superview library, and for handling German registrations.
-

- \* Jan van den Baard and Ian J. Einman, for the great BGUI library which was used to create the GUI.
- \* All people who have registered and supported shareware.
- \* All of the people on the BGUI mailing list.
- \* Nico François for ReqTools, which I had used in early versions of IE.
- \* Scott Tribbey, for suggestions, help with uploading, bug reporting, beta testing...
- \* Patrick Nydensten, (thanks for the ARexx scripts and convolves, not to mention all of the bug reports and suggestions.. :)
- \* David Putzier for the IE logo/icon.
- \* Volkmar Funke for another icon for IE and PageStream and ScanImage scripts.
- \* Petri Nordlund for creating Executive.
- \* The following people for beta testing and suggestions and other help.
  - Terrence Chun
  - Jerry Fleetwood
  - Joseph Hebert
  - Jon Peterson
  - Stephen Dowdy
  - William Maddock
  - RayDar
- \* Nigel Steward, Fiona Smith and my other friends where ever in the world they may be. (Hi Deb + Friends!)
- \* Micheal Haigh, for the technical support
- \* Residents at the Village.
- \* North West Amiga User group in Melbourne and David Myer for the HD drive.
- \* U2, the greatest rock band in the world.
- \* Everyone who reported bugs and/or made suggestions etc.
- \* to everyone I forgot...

## 1.186 On-line Support

On-line Support  
~~~~~

Image Engineer support page at:

<http://www.amigaworld.com/support/imageengineer>

---

Visual Engineering is a powerful and easy to use scriptpack for Image Engineer. Tutorials and lots of example graphics and downloads area available at:

<http://wwnet.fi/users/smarko/visual/>

## 1.187 The Future

The Future

~~~~~

Future will bring a new version of Image Engineer. Visual Engineering -scriptpack will be glued to Image Engineer. More effects will appear. If you miss a feature in Image Engineer, tell us about it. We are having some technical problems reading your mind. ;)

"Drink when you are thirsty, rest when you are fatigued,  
program when the moment is right."

"You must attempt the absurd to accomplish the impossible."

"Imitation is the most sincere form of copyright infringement."

## 1.188 History

History

~~~~~

V0.0 First the universe formed...

V1.0 21/3/95 Initial Release

V1.1 11/4/95

Bug fixes:-

- \* Couple of minor bugs in the start up code.
- \* Nasty bug when opening 24bit files as 8bit grey.
- \* Fixed problem with the menus under OS 2.0. (->Cavalier)
- \* Bug in Prefs requester.
- \* Numerous other small bug fixes. (I've lost track ;-)

New Features:-

- \* ARexx macros from within IE and user configurable ARexx menu.

V2.0 27/7/95

Bug fixes:-

- \* Nasty bug when rendering narrow (<64) images.
  - \* Bug in the contrast requester.
  - \* Bug when enclosing project names in quotes (AREXX).
  - \* Small bug in Threshold history entries.
  - \* Bug in the Resize Arexx command.
-

- \* Mega-Bug that stopped all gadgets from working under 2.04 (->Jerry Fleetwood, Terrence Chun, Joseph Herbert)
- \* Bug in the Arexx prefs requester
- \* Bug in the CROP Arexx command
- \* Couple of bugs in the Edit palette requester

## New Features:-

- \* BOX added to ARexx command PROJECT\_INFO.
- \* Rotate function.
- \* Halftone function.
- \* Editing SVObject ControlPad files using a GUI.
- \* Maximum and Minimum composite modes.
- \* Threshold now works for colour images.
- \* TYPE Arexx command can now return the current setting.
- \* GET Arexx command.
- \* Twirl function
- \* Bulge function
- \* View Saturation histogram
- \* Change Saturation function
- \* View Hue
- \* Change Hue
- \* Displace Pixels
- \* Tile
- \* Remove Feature
- \* Maximum Filter
- \* Minimum Filter

## Other:-

- \* Lock to palette best is now faster.
- \* WWW Support

V2.1 2/10/95

## Bug fixes:-

- \* Intermediate bug with the gadgets in most windows.
- \* Bug in the Gamma function

## New Features:-

- \* Colour preview images
- \* Dithered previews, colour and grey.

## Other:-

- \* Palette requester will now let you edit 32 colours on non-AGA machines, instead of 0 colours.
- \* RMB in render screens also brings the corresponding project preview window to the front.
- \* can be made to flush memory, on start up/exit.
- \* Licensed to use V11.8+ of SuperView Library.

V3.0 5/5/96

## Bug fixes:-

- \* Tiny bug that caused a problem with the dithering pattern.
- \* Bug in GET\_FILE\_TYPE, wasn't returning the type name if it is the first one on the list. (-> Fred Calendini)
- \* Crash when opening IE in colour on a 2 or 4 colour screen.
- \* Bug in the scale routines that caused bus errors under VMM (->Norman Iscove, Jon Amsler)
- \* Bug that often crashes the machine when saving rendered screens. The code that causes this has been worked around for now. (-> numerous people).
- \* Fixed the progress indicator so that it is only updated only once per second at the most. Everything now runs



faster, the indicator was a \*major\* bottleneck. I can't believe I didn't notice it sooner. Someone kick me!

New Features:-

- \* GUI rewritten and redesigned using the BGUI library. Is now font sensitive, resizable and has keyboard short cuts.
- \* All windows and requesters remember their position and current values between sessions.
- \* Image processing effects now implemented as disk based modules.
- \* Zoom in/out in preview windows.
- \* Preview images are provided for most operations.
- \* Menus are user configurable.
- \* Lists of the Most Recently Used directories are provided on the Open and Save menus, simplifying and speeding up file operations.
- \* Posterize effect.
- \* Image Rendering options are now simpler.
- \* Composite functions has had add, subtract, difference and multiply composite methods added to it.

Other:-

- \* Drag box changed to the more trendy 'marching ants'
- \* Preview window scroller gadgets changed to Multiview style BOOPSI ones.
- \* Dragging out a box, the image now scrolls when you reach the edges.
- \* Mouse pointer colours are preserved and taken from the WB.

V3.1 24/7/96

Bug fixes:-

- \* Bugs that stopped the integer gadgets in SVObject Prefs windows from accepting input.
  - \* Bug in the Saturation module that would sometime stop the window from opening due to lack of memory, even when there is heaps.
  - \* Bug in the Rotate module that caused black edges to appear in the resulting images.
  - \* Bug in the Histogram module that may cause it display an error message for no reason.
  - \* Bug in the MARK ARexx command that sometimes caused the menu items to go out of sync with the project.
  - \* Fixed bug in the Composite module that would sometimes cause memory to be read pass the image buffer in the previews.
  - \* SVObject prefs options windows can now handle cpinfo file that use the \_FILE type.
  - \* Fixed problem that made it possible for window borders to get trashed if the window is resized small half way through a refresh.
  - \* Fixed the tab key cycle loopy thing for the Convolve module.
  - \* Was possible to use the Execute Module function to try and execute a module that didn't have a GUI function, resulting in a crash.
  - \* Highpass would try to divide by 0 if the mask is set to 1x1 and the image is in colour. ->Reflex
  - \* Typo in the Highboost interface.
  - \* Bugs in the Displace module had broken the Displace ARexx
-

- function. ->Patrik Nydensten
- \* Fixed a div by 0 bug that only occurs in the Contrast Stretch module when using degenerate images that have only 1 unique value. ->Don Cox
  - \* After drag-n-drop'ing subitem titles in the Menu Prefs requester, sometimes causes some items to suddenly move on to a submenu.
  - \* Small bug in the Resize module that caused the CENTER keyword not to work properly from ARExx. ->Patrik Nydensten

New Features:-

- \* CyberGraphX support for preview windows. IE can now be opened in any CyberGraphX screen mode.
- \* Colour Filter module.
- \* Hue Mask module.
- \* Ripple module.
- \* Noise module.

Other:-

- \* Moved the Amiga+C shortcut to the Save Data->Clipboard menu item.
- \* Fixed up the Gamma function so that it is more in line with other image processing software with positive values making the image brighter.
- \* The Execute Module function now disables the Ok button if it's not possible to execute the current module.
- \* The size of the preview windows are now limited to the size of the preview image being shown in them.
- \* [ and ] keys now bring the next/previous window to the front and makes it active.
- \* An 'R' is now shown in a project's title bar once it has been rendered.
- \* Pressing the LMB on a render screen acts like the screen to back gadget.
- \* Numerous tweeks to the GUI.
- \* Added gadgets to the "Enter Select Co-ords" requester for co-ords for the bottom right hand corner.  
->Patrik Nydensten
- \* Window refresh method can be switched between SMART/SIMPLE.
- \* The GET\_NUMBER ARExx command can now have a slider attached to the integer gadget. ->Patrik Nydensten
- \* When using <=8 bit screens the initialisation calculations will be cached on disk between sessions in IE:ScreenCache, speeding up future initialisations. ->numerous people
- \* Heapsa typos in the documentation fixed. ->David Putzier
- \* Solarize ARExx script.
- \* ADPro and DOpus5 support scripts from Patrik Nydensten added to the distribution.
- \* Crayon, Emboss\_Ultra, NightVision, RoughNSharp, RoughSurface\_High, RoughSurface\_Low, RoughSurface\_Medium, Smooth\_High, Smooth\_Normal convoles from Patrik Nydensten added to the distribution.
- \* Adjust, Bleed, Preserve and RollRegion ARExx scripts from Patrick Nydensten added.
- \* IE now has a good logo/icon thanks to David Putzier.

V3.2 4/9/96

Bug fixes:-

- \* The SVObject Prefs can now handle the Limbo controlpad.

- \* The GET ARexx command was broken. ->Patrik Nydensten
- \* The slider<->int link thingy in HighBoost was broken.
- \* Fixed the Bulge ARexx command, was broken. ->Patrik Nydensten
- \* Fixed bug in the Composite ARexx command that would cause mix percent to be set to 0 if genlock was used. ->Patrik Nydensten.
- \* The gamma function had problems with very large and small values. ->Patrik Nydensten.
- \* Small alpha Composite bug that showed up when the alpha image is not as wide as the primary image. And another one that caused the alpha image to be placed incorrectly when y offset is negative.
- \* The Displace module preview had problems when the alpha image was grey. ->Patrik Nydensten
- \* The DisplacePixels ARexx script was not included in the archive.
- \* Some of the install scripts in the Extra directory were faulty. ->Patrik Nydensten

New Features:-

- \* Impress module.
- \* The Modules and ARexx windows can be left out on the working screen.
- \* Composite now has a "Stretch Alpha to fit" option.
- \* PROJECT\_LIST ARexx command added. ->Patrik Nydensten

Other:-

- \* Shear and BatchProcess ARexx scripts from Patrik Nydensten now included.
- \* Material for developing modules is now available for the Web page.

V3.3 6/12/96

Bug fixes:-

- \* The font size wasn't remembered between sessions.
- \* Reflect\_X and Reflect\_Y ARexx commands didn't return the name of the new project. ->Patrik Nydensten
- \* Fixed bug in the NOISE ARexx implementation. The intensity keyword was not the first component keyword checked.
- \* Bug in the Impress module that was causing a crash when ever a grey image is used to the primary and alpha image. ->Patrik Nydensten
- \* Bug in the Resize module that caused it to crash sometimes when resizing an image to something smaller. ->Patrik Nydensten
- \* The Scale module crashed if it tried to scale an image to a one pixel square. ->Patrik Nydensten

New Features:-

- \* On screen Infobar (which will be evolving into a Toolbar in the future).
- \* Support for reading and writing to Superview's Device SVOjects. (ScanQuix scanners, Prefs Printer)
- \* Possible to have an ARexx script executed automatically at startup and exit.
- \* PROJECT\_INFO ARexx command has been extended with ZOOM, OFFSETS and WINDOW.
- \* PROJECT\_SET ARexx command added.
- \* Image Engineer's screen is now a public screen.

- \* Supports Petri Nordlund's Executive UNIX like task scheduler.

Other:-

- \* The close project menu item has been moved to the Project menu.
- \* DPaint5, DOpus4 and Mainactor support scripts from Patrik Nydensten now included.
- \* Updated and improved BatchProcess collection of scripts.
- \* When a new project is created it gets the same zoom level as the project that it was devired from.
- \* Now requires at least V15 of Superview library and V7 of the SVSupport library.
- \* Pattern gadgets have been added to the file requesters.
- \* Automatically allocates a bigger stack at startup if needed.
- \* You can now set it to automatically zoom out loaded images so that they fit on the screen.
- \* Increased the range of the sliders in the scale module to 10%-400%.
- \* ARexxSupport module now remembers window positions and file/dir requester positions.
- \* EdgeMerge, Fresco, Neon, ShiftRGB, ImageFX\_Export, ImageFX\_Export\_Render, ImageFX\_Import, OilPaint scripts now included.
- \* Web pages have moved.

V3.4 22/1/97

Bug fixes:-

- \* The Request ARexx command was broken.
- \* Bug in the Impress module that may have caused some crashes on some machines. (A3000s namely).
- \* ToolTips have been disabled for the InfoBar to avoid a possible bug in the BGUI library.
- \* The Impress module should work fine under CyberGraphX screenmodes now.
- \* Crash that occurs if you try and use a grey work screen with no dithering. ->Stephen Dowdy

New Features:-

- \* FORM ARexx command added, making it possible to add complex input windows/control panels to scripts.
- \* Most ARexx scripts have been rewritten and improved to use FORM user interfaces.

Other:-

- \* The Maximum and Minimum modules have been incorporated into the Median module.
- \* Updated and improved BatchProcess collection of scripts as always. :)

V3.41 7/2/97

Bug fixes:-

- \* Numerous little bugs that caused problems when used with the later beta versions of BGUI.
  - \* Noise preview window should work fine now on CyberGfx screens.
  - \* Bug in the Median module that caused the preview box thingy to be stuck in the upper left left corner of the image. ->Don Cox
-

- \* Bug that causes the alpha composite with grey primary, colour secondary and grey alpha to not work.  
->Patrik Nydensten
- \* Small bug that stops you from running modules that don't need an image (none at the moment) from the modules window.

Other:-

- \* Numerous small changes to the GUI of some modules.
- \* Another icon for IE and the PageStream and ScanImage scripts from Volkmar Funke added.
- \* Neon script has been changed again.
- \* Batch process scripts updated.
- \* SliderShowIE.rexx script by Don Cox added.

V3.41 02/10/1999 (Wisision release)

Made AmigaGuide-document up-to-date.  
Converted all .gif -files as ILBMs.  
Slideshow.rexx update by Don Cox.  
Removed AddIcon.rexx.  
Removed RollRegion.rexx

Updates scripts:

Adjust v1.21  
- now saves last used values

EdgeEffect v1.00  
- external pictures now use zoomlevel of original picture

EdgeMerge v1.31  
- alphaimage now uses zoomlevel of original picture

FadeInAnim  
- external pictures now use zoomlevel of original picture  
- was using command set of old Image Engineer. Fixed.

FitSelectAlpha  
- alphaimage now uses zoomlevel of original picture

Halftone v2.11  
- alphaimage now uses zoomlevel of original picture

MotionBlur v2.01  
- now saves last used values

Preserve v2.21  
- now saves last used values

Shear v2.12  
- now saves last used values  
- alphaimage now uses zoomlevel of original picture

Vignette v1.12  
- alphaimage now uses zoomlevel of original picture

VignetteGrey v1.12

---

- alphaimage now uses zoomlevel of original picture

New script: Roll v1.00

- Rolls whole image when there's no cropbox at all or cropbox's width or height is smaller than 5 pixels
- When cropbox is large enough, then area of cropbox is rolled

V3.41 16/10/1999 (Wision release 2)

Following  
scripts  
were originally part of Visual Engineering.  
Now they are officially part of Image Engineer.

AddSpace.rexx v1.63  
AddSpaceB.rexx v1.07  
AddTexture.rexx v1.00  
CloseAll.rexx v1.05  
CropEdge.rexx v1.03  
Merge.rexx v2.02  
PrimaryColors.rexx v1.84  
Rotate.rexx v1.00  
SaveAll.rexx v1.30  
SaveIT.rexx v1.00  
Zoom14.rexx v1.00

New scripts: DeInterlace.rexx v1.00, Interlace.rexx v1.00

New Convolves: Bump#?, Carve#?, DeepPress, Emboss\_High\_0, Pencil#?,  
Reflect#?, SmoothEdge#?, StripEdge#?

"If at first you don't succeed, call it a Beta version."

"Hydrogen and stupidity are the most abundant elements in the universe.  
But not necessarily in that order."

## 1.189 Bibliography

Bibliography

~~~~~

Gonzalez, R. C., and R. Woods. 1992. Digital Image Processing.  
Reading, Massachusetts.: Addison Wesley.

Heckbert, P. 1982. "Color Image Quantization for Frame Buffer Display".  
Computer Graphics, 16: pp 297-307.

Hill, F. S. Jr. 1990. Computer Graphics. New York.: Macmillan.

Kruger, Anton. 1994. "Median-Cut Color Quantization." Dr. Dobb's Journal, September 1994, volume 19, issue 10.

Gosselink, Pieter. 1994. "The Gosselink Ditherer." Dr. Dobb's Journal, December 1994, volume 19, issue 15.

Cooper, Alan, 1995. About Face, The Essentials of User Interface Design. IDG Books WorldWide.

## 1.190 Index

About

About the User Interface

Applying a Vignette effect using an Alpha chan

ARexx

Arexx command

ARexx Commands

ARexx Window

Arexx command

ARexxSupport

Arexx command

Autocrop

Arexx command

Autocrop

Module

Bibliography

Brightness

Arexx command

Brightness

Module

Bulge

Arexx command

Bulge

Module

Changes to the ARexx commands from 2.1

Arexx command

Close

Arexx command

---

---

Close Project

Close Render

Colour Filter  
Arexx command

Colour Filter  
Module

Composite  
Arexx command

Composite  
Module

Contrast  
Arexx command

Contrast  
Module

Contrast Stretch  
Arexx command

Contrast Stretch  
Module

Convert to Colour  
Arexx command

Convert to Colour  
Module

Convert to Grey  
Arexx command

Convert to Grey  
Module

Convolve  
Arexx command

Convolve  
Module

Copyrights

Crop  
Arexx command

Cut or Crop

Disclaimer

Displace

---



---

- Arexx command
- Displace
  - Module
- Distribution
- Edit Palette
- Effects Modules
- Execute ARExx script
  - Arexx command
- Execute Module
- False Colour
  - Arexx command
- False Colour
  - Module
- Filtering and Removing Noise from an Image
- Font Preferences
- Form
  - Arexx command
- Gamma
  - Arexx command
- Gamma
  - Module
- General Preferences
- Get
  - Arexx command
- Get Dir
  - Arexx command
- Get File
  - Arexx command
- Get File Type
  - Arexx command
- Get Files
  - Arexx command
- Get Number
  - Arexx command
- Get Project Info
  - Arexx command

---

Get Project List  
Arexx command

Get String  
Arexx command

Get Task Priority  
Arexx command

Halftone  
Arexx command

Halftone  
Module

HighBoost  
Arexx command

HighBoost  
Module

HighPass  
Arexx command

HighPass  
Module

Histogram  
Module

Histogram Equalization  
Arexx command

Histogram Equalization  
Module

History

How exactly does it work?

How to Register

Hue  
Arexx command

Hue  
Module

Hue Mask  
Arexx command

Hue Mask  
Module

Image Engineer to Front  
Arexx command

---

---

Image Engineer V3.4

Impress  
Arexx command

Impress  
Module

Infobar Window

Installation & Starting Up

Introduction

Keyboard Short Cuts

Last Error  
Arexx command

Load Palette

Loading, Rendering and Saving an Image

Local Contrast Stretch  
Arexx command

Local Contrast Stretch  
Module

LowPass  
Arexx command

LowPass  
Module

Mark As  
Arexx command

Maximum Filter  
Arexx command

Median Filter  
Arexx command

Median Filter  
Module

Menus

Menus Preferences

Minimum Filter  
Arexx command

Module Window

---

Motivation (or What could drive a person to d

My 24bit IFF images are being loaded and corr

Negative  
Module

Negative  
Arexx command

Noise  
Module

Noise  
Arexx command

On-line Support

Open  
Arexx command

Open As Submenu

Open Clipboard  
Arexx command

Open Device  
Arexx command

Open Device Prefs

Open Submenu

Palette Load  
Arexx command

Palette Save  
Arexx command

Palette Submenu

Posterize  
Module

Posterize  
Arexx command

Preferences submenu

Project Info

Project Name

Quit

Quit  
Arexx command

---

Recalculate Palette

Reflect X  
Module

Reflect X  
Arexx command

Reflect Y  
Module

Reflect Y  
Arexx command

Remove Feature  
Module

Remove Feature  
Arexx command

Render

Render  
Arexx command

Render Close  
Arexx command

Render Colours  
Arexx command

Render Control

Render Depth  
Arexx command

Render Device  
Arexx command

Render Dither  
Arexx command

Render Get Options  
Arexx command

Render Palette  
Arexx command

Render Screen Mode  
Arexx command

Render to Front  
Arexx command

Reporting Bugs

---

---

Request  
Arexx command

Resize  
Module

Resize  
Arexx command

Ripple  
Module

Ripple  
Arexx command

Rotate  
Module

Rotate  
Arexx command

Saturation  
Module

Saturation  
Arexx command

Save  
Arexx command

Save Data  
Arexx command

Save Data Submenu

Save Data to Clipboard  
Arexx command

Save Data to Device  
Arexx command

Save Device  
Arexx command

Save Device Prefs

Save Palette

Save Render Submenu

Save to Clipboard  
Arexx command

Scale  
Module

Scale

---

---

Arexx command

Scaling and Locking an image to a palette

Screen Mode

Script Writing Tips

Select Box Co-ords

Set a project's Render Options  
Arexx command

Set Project Variables  
Arexx command

Set Task Priority  
Arexx command

Sharpen  
Module

Sharpen  
Arexx command

Supplied Scripts & Macros

SVObject Prefs

System Requirements

Thanks and Greetings

The Future

Threshold  
Module

Threshold  
Arexx command

Tutorials

Twirl  
Module

Twirl  
Arexx command

Use as Secondary/Alpha

Use with Martin Apel's VMM

WB to Front  
Arexx command

What the smeg is a convolve?

---

What's a Histogram anyway?

Zoom Sub Menu

---