

I3_Help

Chad Randall

Copyright © 1995 by Chad Randall. All Rights Reserved.

COLLABORATORS

	<i>TITLE :</i> I3_Help		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Chad Randall	July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	I3_Help	1
1.1	Iconian 3 help file	1
1.2	A special note about my future with the Amiga.	2
1.3	Copyright	2
1.4	Registering	3
1.5	Introduction to Iconian	4
1.6	Author	5
1.7	Icon Frequently Asked Questions.	5
1.8	Bugs, Errors, and Other Stuff.	6
1.9	Gfx Board Notes	6
1.10	Thanks	7
1.11	NewIcons	7
1.12	Menus	7
1.13	ARexx commands	14
1.14	Gadgets	16
1.15	Keyboard commands	19
1.16	Tooltypes and CLI Arguments	20
1.17	Index	21
1.18	Menu: Project	21
1.19	Menu Item:New	21
1.20	Menu Item:Open	21
1.21	Menu Item:Open New	21
1.22	Menu Item:Restore	22
1.23	Menu Item:Save	22
1.24	Menu Item:Save As	22
1.25	Menu Item:Close	22
1.26	Menu Item:Import	22
1.27	Menu Item:Export	23
1.28	Menu Item:Buffer Size	23
1.29	Menu Item:Highlight	23

1.30 Menu Item:Icon Type	24
1.31 Menu Item:Save Mode	24
1.32 Menu Item:Newicons	25
1.33 Menu Item:Iconify	25
1.34 Menu Item:About	26
1.35 Menu Item:Quit	26
1.36 Menu Item:Save Mode	26
1.37 Menu Item:Undo	26
1.38 Menu Item:Redo	26
1.39 Menu Item:Paste	27
1.40 Menu Item:Cut	27
1.41 Menu Item:Copy	27
1.42 Menu Item:Erase	27
1.43 Menu Item:Exchange	27
1.44 Menu Item:Paste from Other	27
1.45 Menu Item:Undo Buffers	28
1.46 Menu Item:Load	28
1.47 Menu Item:Save Picture	29
1.48 Menu Item:Move Image	29
1.49 Menu Item:Flip Image	29
1.50 Menu Item:Scale Image	29
1.51 Menu Item:Recolor	30
1.52 Menu Item:Use Brush	30
1.53 Menu Item:Load Brush	31
1.54 Menu Item:Save Brush	31
1.55 Menu Item:Enter Text	31
1.56 Menu Item:Select Font	31
1.57 Menu Item:Text Mode	32
1.58 Menu Item:Flip Brush	32
1.59 Menu Item:Scale Brush	32
1.60 Menu Item:Brush Mode	32
1.61 Menu Item:Handle Position	33
1.62 Menu Item:Drawing Mode	33
1.63 Menu Item:Bevel Box Type	34
1.64 Menu Item:Air Brush	34
1.65 Menu Item:Dither	34
1.66 Menu Item:Palette	35
1.67 Menu Item:Open	35
1.68 Menu Item:Screenmode	36

1.69 Menu Item:Window Font	36
1.70 Menu Item:Snapshot	36
1.71 Menu Item:Load Preferences	36
1.72 Menu Item:Save Preferences as	37
1.73 Menu: User (Macros)	37
1.74 MENU#0192	37
1.75 MENU#0193	37
1.76 MENU#0194	37
1.77 MENU#0195	37
1.78 MENU#0196	37
1.79 MENU#0197	37
1.80 MENU#0198	37
1.81 MENU#0199	37
1.82 MENU#019A	38
1.83 MENU#019B	38
1.84 MENU#00D9	38
1.85 MENU#00DA	38
1.86 Menu Item: Search	38
1.87 Magnification Area	38
1.88 Using Project Windows	39
1.89 Horizontal scroller	39
1.90 Vertical scroller	39
1.91 Magnification level	39
1.92 Select drawing tool	39
1.93 Select pen color	39
1.94 Select background color	40
1.95 Undo button	40
1.96 Redo button	40
1.97 Icon type	40
1.98 Highlight type	40
1.99 Icon left edge	40
1.100Icon top edge	40
1.101Icon stack size	40
1.102Free icon position	40
1.103Icon window left edge	41
1.104Icon window top edge	41
1.105Icon window width	41
1.106Icon window height	41
1.107Center datatype image	41

1.108Integer scale	41
1.109Remap	41
1.110Use aspect	41
1.111Aspect X	41
1.112Aspect Y	41
1.113Scale	42
1.114Stretch	42
1.115Dither	42
1.116Clear	42
1.117Default icon directory	42
1.118Default icon directory selector	42
1.119Default icon image directory	42
1.120Default icon image directory selector	42
1.121Default picture directory	42
1.122Default picture directory selector	42
1.123Gadget directory	43
1.124Use gadget directory	43
1.125Gadget directory selector	43
1.126Default appicon	43
1.127Default appicon selector	43
1.128Red component value (string)	43
1.129Green component value (string)	43
1.130Blue component value (string)	43
1.131Red component value (slider)	44
1.132Green component value (slider)	44
1.133Blue component value (slider)	44
1.134Copy color	44
1.135Exchange color	44
1.136Spread color	44
1.137Remap color	44
1.138Dither method	44
1.139Quantize method	44
1.140Palette	45
1.141Tooltypes	45
1.142Tooltype entry	45
1.143Add tooltype	45
1.144Remove tooltype	45
1.145Dither type gadget	45
1.146Beveled box gadget	45

1.147Macro #1 selector	45
1.148Macro #2 selector	45
1.149Macro #3 selector	45
1.150Macro #4 selector	46
1.151Macro #5 selector	46
1.152Macro #6 selector	46
1.153Macro #7 selector	46
1.154Macro #8 selector	46
1.155Macro #9 selector	46
1.156Macro #10 selector	46
1.157Macro #1 string entry	46
1.158Macro #2 string entry	46
1.159Macro #3 string entry	46
1.160Macro #4 string entry	47
1.161Macro #5 string entry	47
1.162Macro #6 string entry	47
1.163Macro #7 string entry	47
1.164Macro #8 string entry	47
1.165Macro #9 string entry	47
1.166Macro #10 string entry	47
1.167Auto enlarge	47
1.168Always overwrite icons	47
1.169Filter datatypes	47
1.170Mini sunmouse	48
1.171Center on select	48
1.172Size Window Gadget	48
1.173Drag Window Gadget	48
1.174Window Depth Gadget	48
1.175Zip Window Gadget	48
1.176Close Window Gadget	48
1.177ARexx: ABOUT	49
1.178ARexx: AIRBRUSH	49
1.179ARexx: BUFFERSIZE	50
1.180ARexx: BRUSH	50
1.181ARexx: CHANGESAVEDEPTH	50
1.182ARexx: CHANGESAVEMODE	51
1.183ARexx: CIRCLE	51
1.184ARexx: CLEAR	52
1.185ARexx: CLOSE	52

1.186ARexx: CLOSESTCOLOR	52
1.187ARexx: COLOR	53
1.188ARexx: COPY	53
1.189ARexx: CUT	53
1.190ARexx: DOTISSTANDARD	53
1.191ARexx: DOUBLEBRUSH	54
1.192ARexx: ERASE	54
1.193ARexx: EXCHANGE	54
1.194ARexx: FILL	54
1.195ARexx: FLUSHUNDOS	55
1.196ARexx: FREEHAND	55
1.197ARexx: GETBGCOLOR	55
1.198ARexx: GETBUFFERSIZE	56
1.199ARexx: GETCOORD	56
1.200ARexx: GETFGCOLOR	56
1.201ARexx: GETPIXEL	57
1.202ARexx: GETSCREENDEPTH	57
1.203ARexx: HIGHLIGHT	57
1.204ARexx: ICONTYPE	57
1.205ARexx: LEFTMOUSE	58
1.206ARexx: LOADDATATYPE	59
1.207ARexx: LOCKGUI	59
1.208ARexx: MARKASSTANDARD	59
1.209ARexx: MATTE	60
1.210ARexx: MOVETO	60
1.211ARexx: OPEN	60
1.212ARexx: PASTE	60
1.213ARexx: PLOT	61
1.214ARexx: QUIT	61
1.215ARexx: REDO	61
1.216ARexx: REPLACE	62
1.217ARexx: RESTORE	62
1.218ARexx: RIGHTMOUSE	62
1.219ARexx: RUBTHRU	63
1.220ARexx: SAVE	63
1.221ARexx: SCREENTOFRONT	63
1.222ARexx: SETFGCOLOR	63
1.223ARexx: SETBGCOLOR	64
1.224ARexx: SMOOTH	64

1.225ARexx: UNDO	64
1.226ARexx: UNLOCKGUI	64
1.227ARexx: XOR	65
1.228ARexx: blank	65
1.229Technical Information	65
1.230The Future of Iconian	65
1.231History	66

Chapter 1

I3_Help

1.1 Iconian 3 help file

Iconian 3

Copyright ©1993,94,95,96 Chad Randall

Special Note A sad note about my Amiga.

Copyright Copyright information.

Introduction A brief introduction.

Menus List of menu items.

ARexx List of ARexx commands.

Gadgets List of gadgets.

Keyboard List of keyboard commands.

Tooltips Tooltype and CLI arguments.

Hints Making using Iconian 3 easier.

Icon FAQ Frequently Asked Questions.

NewIcons Everything you wanted to know, and more.

Bugs Errors.

GFX Boards Some notes to gfx board users.

Technical Some technical information.

History Recent history of changes and corrections.

Future Things yet to be done.

Registering Instructions on how to register.

Author How to contact the author.

Thanks Acknowledgements.

THIS AmigaGuide® is UNDER CONSTRUCTION! Sorry, but writing all this stuff, testing the links, and making it look pretty is *MORE* work than writing the actual program!!

1.2 A special note about my future with the Amiga.

As some of your IRC'ers may already know, I have been without a working Amiga for some time now. As it looks, I may be without an Amiga for a while longer. Until that time comes, development on Iconian has been halted.

I have been thinking about what the Amiga means to me. It means alot. I've been with the Amiga since the early 500's went on sale. I've been with Commodore computers since PET's and CBM's were around. I used to defend myself from Apple and Atari users on local BBS's. I can never forget these wonderful moments in my life.

But a time comes when a person must face reality. And the reality is that the Amiga has a very dim future.

In the past month, I've purchased a 120mhz Pentium (INTEL), along with Windows 95 (MICROSOFT), and I play games such as Quake/Doom. These are things that I've said I would never do. But I've changed.

It is hard not to ignore the television ads, the games in the stores, or the magazines on the shelves. I felt I was missing out on all the new technology.

Don't get me wrong. The WINTEL world is far from perfect. There are hundereds of things I miss from my Amiga.

This all brings me to an important point. If someone more loyal than me, that is skilled in AmigaE, would like to take over development of Iconian, that EMail me. I am willing to give the code to someone to work on, granted that they follow a few rules (to be worked out).

To those registered users that have waited over a month for your keyfile, I'm sorry it took so long. I finally found someone I could hook my IDE harddrives to in order to retrieve my software. I'll have limited access in the future to fullfill registrations. But I strongly urge you to hold off on registering this software for now.

1.3 Copyright

Copyrights

The Iconian 3 package is Copyright ©1993-96 by Chad Randall. All rights are Reserved.

The Iconian 3 package MAY NOT be modified, transfered, or used outside the guidelines defined below.

Distribution

Permission is granted to duplicate and re-distribute the original Iconian 3 package.

Documentation may be edited, but must be noted as an alteration, and the original document must be present elsewhere in the package.

All executables must be in their original condition, and may not be packed, encrypted, or modified in any way.

Personal keyfiles must not be duplicated, except for archival purposes.

The Iconian 3 package must not be sold for a fee greater than the sum of media cost, shipping, and a maximum duplication cost of 1 dollar US, but not to exceed 5 US dollars in total.

CD-ROM collections, magazines, and "cover-disks" must obtain permission from the author before including the Iconian 3 package. All Aminet and Fred Fish collections are exempt from this clause.

Disclaimer

The author assumes no responsibility for the actions taken by the content of the Iconian 3 package.

All executables are guaranteed by the author to be free from intentional incompatibilities or destructive tendencies at the time of original release.

The author is not responsible for any alterations that result in lost data, viral infection, or other destructive actions.

Compatibility with various hardware setups other than the setup used for devopment, is not guaranteed, nor implied. The program is released "as-is". Modification to fit specific setups is left to the sole discretion of the author.

All end users of any part of the Iconian package are granted no rights or protection. Therefore, Iconian may not be used in those countries that protect the consumer's rights. By using any part of the Iconian 3 package in such countries, the consumer is waiving all rights guaranteed under the law.

1.4 Registering

Shareware Limitations

The shareware version of Iconian has only a few differences from the registered version. These limitations are not obtrusive, except for the "power" user.

1. An annoyance requester will appear upon startup.
2. Buffer sizes are limited to 128x100.
3. Undo buffer size is limited to 100k.

Please note that these limitations are not guaranteed to remain the same in future releases.

Registration Information

Registration is only 10\$. An additional 5\$ is required for postage and packaging.

Upon registration, you will receive a personalized keyfile. This keyfile will work with all future releases of Iconian.

How to Register

You can send one of the following:

*15\$ US

*10\$ US

A valid EMail address.

*10\$ US

A blank disk

A self-addressed return envelope

1.00\$ in US postage stamps

*10\$ US

A blank disk

A self-addressed return envelope

2 international response coupons (IRC's).

Please:

Write as neatly as you can.

Allow upto 6 weeks for mail delivery.

Allow upto 5-15 days for EMail delivery.

List where you obtained Iconian, and what version you have.

Optionally:

List any features you would like to see.

List any problems you have encountered.

Mail to:

Chad Randall

229 S. Washington St.

Manchester, Michigan

48158-9680

United States of America

(ps.) The Australian Continental Drift BBS is no longer available.

1.5 Introduction to Iconian

Introduction

Iconian 3 is a full featured icon editor. It was designed from the start with AmigaOS 3.0 in mind, and utilizes every OS3.0 enhancement where possible.

Features

Iconian 3 includes all most every IconEdit 3.0 feature, without the annoying limitations and bugs:

- * 15 drawing tools.
- * Can save C, E, and Asm source code.
- * Full clipboard support.
- * Dithering built into all drawing tools.
- * Supports all icon types and highlight methods.
- * Can edit upto 256 colors.
- * Supports the 8 color+8 plane method used by IconEdit.

Iconian 3 has support for basic icon manipulation outside of IconEdit 3.0 features:

- * Tooltype editing.
- * User defined sizes, including an exotic "explode" method.
- * Many preference windows, all co-existing with the main program.
- * A beveled box drawing tool.
- * A realtime brush drawing tool.
- * Support for the right mouse button as a secondary pen.

Iconian 3 also has features unique to other available icon editors:

- * A full featured ARexx port. (Work In Progress)
- * Extensive online help. (Work In Progress)
- * Multiple projects.
- * Multiple undo and redo, limited only by available RAM.
- * Datatype support, with scaling, aspect retention, dithering, and more.
- * Many ways to specify the icon format, size, and appearance.
- * Customizable work environment. Gadgets, fonts and pointers.

Iconian 3 fully supports the NewIcon format when the newicon.library is available:

- * Recognizes NewIcon files
- * Allows each image to contain a custom palette, when Iconian 3 is run on a custom screen.
- * Has the NewIcon palette built in.
- * Can save the recommended size for easy and quick editing.

Iconian 3 is not limited to designing and editing icons. With an extensive set of drawing tools, plus the ability to import and export many file formats, it is also ideal for designing other small, image related things:

- * Workbench friendly sketch program.
- * Image structures for programming, both planar and chunky format.
- * Mouse pointer images.
- * Brushes.
- * Thumbnails.

Iconian 3 has been a leading icon editing environment for over 2 years. There are over 120 registered users worldwide. It is written and maintained by a very devoted Amiga fanatic. You can bet that it will still be around for years to come.

1.6 Author

Contacting the Author

You can send registrations, flames, comments or suggestions to me at:

EMail: crandall@msen.com

SnailMail: Chad Randall

229 S.Washington St.

Manchester, Michigan

48158-9680

U.S.A.

WWW site: <http://www.msen.com/~crandall/>

Beta ftp: <ftp://music.phlab.missouri.edu/pub/amiga/iconian>

Placing the word "Iconian" somewhere in your email subject line will ensure the mail gets attention.

You can also find me on IRC occasionally with the nick Iconian.

Who am I?

I've been playing around with computers for almost 15 years. I've used everything from a VIC=20 and PET, upto today's standards. I currently own an Amiga 1200 decked out to the max. It has 18 megs of RAM, dual 500 megabyte harddrives, and a 50MHZ 68030 cpu. I also own an Amiga 500, but have long since sold off it's parts. I still have several C=64's in my closet.

I've written lots of games and programs in the past. But up until I purchased Internet access two years back, none of my programs got farther than my friends. In the past two years, I've released a few programs onto the Amiga community. Although Iconian is definatly the most popular, it's little brother Picticon is fairly well known also. PlayKiSS has it's spot on a few peoples hard drives. Some lesser known utilities include randomcopy, mwb2ni, and the newer Workbench Palette Injector (WBPI).

With most of my programs skewing towards the Workbench and icon nature, you would think my Workbench is fancy. But the irony is that I rarely use Workbench beyond launching new programs I've just downloaded. Once I decide to keep and use a program, it gets quickly added to my massive ToolsDaemon menu and/or a obscure hotkey. My Workbench is almost always covered up with shell windows. And DirOpus 5 (BUY IT) fills in for file handling.

I almost always write programs to fill my needs. I can also be coaxed into modifying these to suit other's needs. As in the case of Iconian, I wrote it to create 3 icons I wanted at the time. After sending it to a few people, I realized I could help others. Most of Iconian's features rose from requests. So if you have one, LET ME KNOW! I probably won't add anything otherwise. I wouldn't even be writing this guide and this sentence, had it not been for people mailing me, asking me to do so!

Personal Stuff

I like to listen to music while doing heavy programming. I like everything from Hard Rock to Punk and Heavy Metal. Some of my all time favorite albums and bands are Faster Pussycat, G-n-R, Skid Row, Megadeth, and Pennywise. As I am typing this, I'm listening to the new Rancid album, "...and Out Come the Wolves"

I also like to watch TV. I watch a few hours everyday, just like every other normal person. But Sunday night is my favorite. FOX from 7:00 to 10:00! I also like the new ST:Voyager. And I hate to admit it, but I like Saturday morning cartoons too. Animaniacs rule! :)

1.7 Icon Frequently Asked Questions.

Frequently Asked Questions

- "What's the easiest way to get icons with more than 8 colors?"

This is a hard one. The easist way is to stick with 4 or 8 colors. But there are two main ways to get more than 8. The easiest way is to create NewIcon format icons. The drawback with this is that NewIcons take longer to load from disk. And, under the right circumstances, may take up to 2 to 3 times the amount of CHIP memory.

The other method is to use a palette daemon, such as the one that comes with MagicWB, or my own program, shovecolors.

- "Is it possible to remove the border around WB icons?"

No...not legally anyways.

Recently, I stumbled across a patch that does this. But it does not remove the spacing around the icon, just turns off the border rendering. It can disable the space, but at a price... you can not use background patterns.

1.8 Bugs, Errors, and Other Stuff.

Bugs and Errors

Annoyances:

*Iconian still doesn't have support for the shift keys to lock vertical/horizontal movements. Try using a nice commodity to do this until I work a way.

*The brush menu's first subitem (brush 0) will get shifted to the right occasionally. This appears to be a bug in LayoutMenus(). I've tried to work around it, but it persists. At least now it won't move to the right 17 pixels every time the screen is re-opened!

Known bugs:

*About 800 bytes are lost to unknown reasons everytime Iconian is run, according to CodeWatch.

*pop.gadget can start rendering directly to the screen even when the screen is not "locked"... This occurs when you press mouse buttons rapidly.

Known incompatibilities:

*MagicCX's cycle-to-menu gadgets do no send help notifications.

*MagicCX's cycle-to-menu gadgets do not accept italic or boldface fonts. Therefore, don't select italics or boldface under the Select Window Font menu item. (You shouldn't be picking fancy fonts for windows anyways...you're only asking for trouble if you are!)

*Problems still reported with Magic Menus.

See Also:

[GFXBoard Notes](#)

1.9 Gfx Board Notes

Gfx Board Notes

I don't have access to any gfx boards of any kind. I live in Amiga isolation, using an Amiga 1200 for programming. The only information about Iconian 3 gfx board compatibility I have is from end users. Therefore, although I try hard to do everything "by the book", you may experience difficulties.

There are a couple of things you can try:

* Select the "SLOWUNDO=TRUE" tooltype. This uses plain chunky-2-planar conversion, instead of WritePixelFormat8().

* Run Iconian 3 on a custom screen, selecting a standard Amiga display mode. Of course, this is probably a last resort.

* Turn off friend bitmaps by selecting the "FRIENDS=FALSE" tooltype or the "NOFRIENDS" cli switch. Cybergraphics users want friends (system legal), Picasso users won't (old way).

If you have experience in gfx boards, and can give me a few tips on making my program "gfx board friendly", please let me know!

1.10 Thanks

Thank You's

In no particular order:

Everyone who has registered Iconian. This numbers just above 150 people at the moment!

Wouter - If it wasn't for E, I doubt Iconian would exist.

Russ LeBar - He's helped a bit in the past.

Niels Jørgens (Dexion Design) - For the original Iconian icon.

Mike Chou aka "DV8R" - He touched up the Brilliance gadgets.

All the IRC folks: RMerlin, Aesculape, Ashiq, Hammer-D, WiZ-Id, etc.

Everyone else who has mailed me in the past about Iconian.

1.11 NewIcons

NewIcons

NewIcons is a system patch. It intercepts calls to GetDiskObject(). It then extracts chunky-format images from extra tooltypes, and remaps these images to the Workbench screen.

NewIcons was authored by Nicola Salmoria. Nicola recently gave up on Amiga support, but plans are underway to continue updating the NewIcons package. Contact Aesculape or RMerlin on the #amiga IRC channel. I was going to help with the updates, but Nicola did not want me to work on his program--Probably because of my MWB2NI document. >:)

Creating NewIcons can be frustrating. Most NewIcon artists create the images in paint programs and then convert them using CLI based commands. This works best, as you don't have to worry about 3 palettes and such. But don't give up on Iconian yet!

The number one thing you want to do is to run Iconian on it's own screen. This will allow you to edit seperate palettes for every image. Make sure that the "Seperate Palettes?" flag is set under the "Color" menu if you wish to have different palettes for every image.

Create the two NewIcon images anyway you wish, but make sure that they are bigger than 92 pixels across and down. The hardcoded limit to the newicon.library is approximately 94x94, but 92 is safest.

If you wish to attach standard icon image(s), load it now. Make sure that one of the standard image buffers is highlighted, then select "Mark as Standard" under the "Highlight" submenu.

Highlight the NewIcon images, and select "Save As Newicon...". Select a name and you are all set! Double check the icon by looking at it on Workbench. Drop the icon onto Iconian's appicon to double-check that a standard image is there.

1.12 Menus

Menu List

Project

New

Close

Open

Open New

Restore

Save

Save As**Import**

Pointer Prefs

Export

ASM Source Code...

C Source Code...

E Source Code...

Chunky Array...

Pointer Prefs...

Buffer Size

Width=36

Width=64

Width=128

Width=256

Width=640

{\0Width=User

Height=40

Height=50

Height=100

Height=250

Height=512

}\0Height=User

Highlight

-\0Complement

=\0Backfill

;\0Image

Icon Type

!\0Disk

@\0Drawer

#\0Tool

\$\0Project

%\0Trashcan

Device

^\0Kick

AppIcon

Save Mode

Strip Planes

Current Depth

Force Eight

Auto Size

Full Size

Custom Size

NewIcon Standard

Explode Mode

Newicons

Save

Save As...

Seperate Palettes?

Mark as Standard

Dot is Standard

Iconify

About

Quit

Edit

Undo

Redo

Paste

Cut

Copy

Erase

Exchange

Paste from Other

Undo Buffers

Goto Oldest

Goto Newest

Flush All

Flush Current

Set Memory Size...

Images

Load

J\OPicture...

B\OPicture and Palette...

Y\ONormal Image...

U\OSelected Image...

I\OBoth Images...

Default Images

WhatIs Images

Save Picture

Move Image

T\0Auto Topleft

Up

Down

Right

Left

Flip Image

Quick Horizontal

Quick Vertical

Smart Horizontal

Smart Vertical

Scale Image

Half Horizontal

Half Vertical

Half Both

Double Horizontal

Double Vertical

Double Both

Recolor

M\01.3 to 2.0

G\03 to 8 Planes

Use 2 Planes

Recolor Pen2 with Pen1

Swap Pen1 and Pen2

Brushes

Use Brush

Brush 1

Brush 2

Brush 3

Brush 4

Brush 5

Load Brush**Save Brush****Enter Text****Select Font****Text Mode**

Jam1

Jam2

Complement

Inverse Video

Flip Brush

Horizontal

Vertical

Scale Brush

Half Horizontal

Half Vertical

Half Both

Double Horizontal

Double Vertical

Double Both

Brush Mode

Matte

Color

Replace

Complement

Smooth

Handle Position

Center

Top Left

Top Right

Bottom Left

Bottom Right

Custom

Paint

Drawing Mode

Plot

Draw

Line

Beveled Box

Rectangle

Filled Rectangle

Oval

Filled Oval

Fill

Filled Polygon

Air Brush

Define Area

Cut Brush

Paste Brush

Stretch Brush

Paste Text

Bevel Box Type

Single

Button

Ridge

Dropbox

Recessed

Filled

Air Brush

Small

Medium

Large

Focas?

Drip?

Dither

None

94%

88%

75%

63%

50%

38%

25%

13%

7%

Bars

Lines

Transparent?

Palette

Load Palette...

Save Palette...

MagicWB Preset

NewIcon Preset

RomIcon Preset

JBCIcon Preset

Iconographix Preset

Windows

Edit

Program Settings...

Paths...

Icon Information...

Datatype Settings...

Macros...

Mover...

Palette...

Tooltypes...

Screenmode

Workbench

Default Pubscreen

Custom Screen...

Window Font**Snapshot**

Default Settings

Window Positions

Paths

Screenmode

Fonts

Macros

Program Settings

Datatype Settings

Load Preferences**Save Preferences as****User**

1\0Macro #1

2\0Macro #2

3\0Macro #3

4\0Macro #4

5\0Macro #5

6\0Macro #6

7\0Macro #7

8\0Macro #8

9\0Macro #9

0\0Macro #10

Help

Contents**Index****Search**

1.13 ARexx commands

ARexx Command List

ABOUT

AIRBRUSH

BUFFERSIZE

BRUSH

CHANGESAVEMODE

CHANGESAVEDEPTH

CIRCLE

CLEAR

CLOSE

CLOSESTCOLOR

COLOR

COPY

CUT

DOTISSTANDARD

DOUBLEBRUSH

ERASE

EXCHANGE

FLUSHUNDOS

FREEHAND

GETBGCOLOR

GETBUFFERSIZE

GETFGCOLOR

GETPIXEL

GETSCREENDDEPTH

HIGHLIGHT

ICONTYPE

LEFTMOUSE

LOADDATATYPE

LOCKGUI

MARKASSTANDARD

MATTE

MOVETO

OPEN

PASTE

PLOT

QUIT

REDO
REPLACE
RESTORE
RIGHTMOUSE
RUBTHRU
SAVE
SCREENTOFRONT
SETFGCOLOR
SETBGCOLOR
SMOOTH
UNDO
UNLOCKGUI
XOR

" link

ARexx Command Syntax

The following describes what each "Syntax:" line may contain:

string

This is any string.

<variable>

This is a numerical value.

"literalstring"

This string must be specific, and is usually used as a switch.

parameter

This parameter must be present.

(parameter)

This is an optional parameter.

parameter1 | parameter2 | parameter3

Only one of the listed parameters may be used.

ARexx Command Parsing

The template of a command follows standard AmigaOS 2.0 commandline parsing rules. The template is the actual definition of the command parameters. The syntax is how Iconian actually handles the command. How a command is actually parsed is a combination of both the syntax and template.

ARexx Command Example

A sample command entry might look like this:

SAMPLE

Function:

This line describes briefly what "SAMPLE" is used for.

Syntax:

SAMPLE file (file2) "VAR" <var> ("OPTSTR" ("OPT1" | "OPT2") | <var1> <var2>)

Template:

FILE/A,FILE2,VAR/A/K,OPTSTR/S,OPT1/S,OPT2/S,VAR1/N,VAR2/N

Results:

If anything is returned in the ARexx variable RESULT, it is listed and described here.

Options:

FILE file

This is a required string, consisting of anything.

FILE2 file2

This is an optional string.

VAR <var>

This is also a required variable. The keyword VAR must be present.

OPTSTR "optstr"

This is an optional string. It is mutually exclusive of " <var1> <var2> " parameter portion of the command.

OPT1 "opt1", OPT2 "opt2"

These two are mutually exclusive of each other. They are also only used if the OPTSTR portion of the command is fulfilled.

VAR1 <var1>, VAR2 <var2>

Both variables need to be defined. They are only used if the OPTSTR is absent.

Notes:

Any other comments about this command are listed here.

Examples:

```
/*One or more examples may be given.*/
```

```
/*These below follow the syntax for this command.*/
```

```
SAMPLE file.iff VAR=3
```

```
SAMPLE FILE=file.iff other.iff VAR=3
```

```
SAMPLE file.iff VAR=3 OPTSTR OPT1
```

```
SAMPLE file.iff VAR=3 VAR1=25 VAR2=25
```

1.14 Gadgets

Gadget List

Add tooltype

Always overwrite icons

Aspect X

Aspect Y

Auto enlarge

Beveled box gadget

Blue component value (slider)

Blue component value (string)

Center datatype image

Center on select

Clear

Close Window Gadget

Copy color

Default appicon

Default appicon selector

Default icon directory

Default icon directory selector

Default icon image directory

Default icon image directory selector

Default picture directory

Default picture directory selector

Dither

Dither method

Dither type gadget

Drag Window Gadget

Exchange color

Filter datatypes

Free icon position

Gadget directory

Gadget directory selector

Green component value (slider)

Green component value (string)

Highlight type

Horizontal scroller

Icon left edge

Icon stack size

Icon top edge

Icon type

Icon window height

Icon window left edge

Icon window top edge

Icon window width

Integer scale

Magnification level

Mini sunmouse

Palette

Quantize method

Red component value (slider)

Red component value (string)

Redo button

Remap

Remap color

Remove tooltype

Scale

Select background color

Select drawing tool

Select pen color

Size Window Gadget

Spread color

Stretch

Tooltype entry

Tooltypes

Undo button

Use aspect

Use gadget directory

Vertical scroller

Vertical scroller

Window Depth Gadget

Zip Window Gadget

Macro #1 selector

Macro #2 selector

Macro #3 selector

Macro #4 selector

Macro #5 selector

Macro #6 selector

Macro #7 selector

Macro #8 selector

Macro #9 selector

Macro #10 selector

Macro #1 string entry

Macro #2 string entry

Macro #3 string entry

Macro #4 string entry

Macro #5 string entry

Macro #6 string entry

Macro #7 string entry

Macro #8 string entry

Macro #9 string entry

Macro #10 string entry

1.15 Keyboard commands

Keyboard commands

space/tab - Switches between the normal and selected images.

[and] - Changes the current pen number.

{ and } - Changes the erase pen number.

(and) - Changes the dithering density.

< and > - Changes the magnification strength

* - Toggles the dithering transparency.

comma - Turns on color pick mode

a - Selects "air brush" tool.

b - Selects "cut brush" tool.

shift+b - Selects "paste brush" tool.

d - Selects "plot" tool.

shift+d - Selects "filled polygon" tool.

e - Selects "oval" tool.

shift+e - Selects "filled oval" tool.

f - Selects "fill" tool.

ctrl+f - Selects brush font.

h - Halves the current brush.

shift+h - Doubles the current brush.

i - Opens the icon information window.

shift+i - Opens the datatype settings window.

m - Toggles magnification.

shift+m - Opens the mover window.

n - Centers on mouse cursor.

ctrl+n - Same as project->open new... menu item.

p - Opens the palette window.

shift+p - Opens the path window.

ctrl+p - Opens the settings window.

r - Selects "rectangle" tool.

shift+r - Selects "filled rectangle" tool.

s - Selects "draw" tool.

t - Selects "paste text" tool.

shift+t - Brings up the text brush entry window.

u - Undo

shift+u - Redo

v - Selects "line" tool.

shift+v - Selects "beveled box" tool.

ctrl+v - Opens the about window.
x - Flips brush horizontally.
shift+x - Opens the macro window.
y - Flips brush vertically.
shift+y - Opens the tooltype window.
1 - 5 - Selects different brushes.
csr keys - Moves the visible location one pixel.
shift+csr - Moves the visible location to the extreme.
ctrl+csr - Moves the image one pixel.
F1 - F6 - Changes the brush mode.
ctrl+Fx - Changes the buffer size.
alt+Fx - Changes save methods.
DEL - Erases the buffer with color 0.
HELP - Help on menu or gadget.

1.16 Tooltypes and CLI Arguments

Tooltypes and CLI Arguments

APPICONSTRING = \$

This is the text displayed below Iconian 3's drop-appicon. Defaults to "Drop Files Here".

+ ARROWHEIGHT = <value>

Defines the height of arrow gadgets.

+ ARROWWIDTH = <value>

Defines the width of arrow gadgets.

FANCYPOPUPGADGET = BOOLEAN

Defines the style of popupgadget borders.

POPUPTORIGHT = BOOLEAN

Popup windows will appear to the right, instead of the default center on item.

* PREFS = \$

Can be used to specify an alternate preferences file.

+ SCROLLHEIGHT = <value>

Defines the height of the user buffer height.

+ SCROLLWIDTH = <value>

Defines the width of the user buffer width.

SIZEBBOTTOM = BOOLEAN

Places a sizedgadget border on the bottom of the window.

SIZEBRIGHT = BOOLEAN

Places a sizedgadget border on the right side of the window.

* SLOWUNDO = BOOLEAN

Sets the chunky-to-planar conversion method.

* SMARTREFRESH = BOOLEAN

Defines the refresh methods of windows.

* USERHEIGHT = <value>

Defines the height of the user buffer height.

* USERWIDTH = <value>

Defines the width of the user buffer width.

A "*" denotes that this option is available from the CLI.

A "+" denotes that this option is available from an environment variable.

1.17 Index

Index

This node is currently empty.

1.18 Menu: Project

Menu: Project

Contains actions and settings that involve project operations.

1.19 Menu Item:New

Menu: Project

Item: New

Opens a new project window. The default "untitled" icon will be loaded.

Notes

A new project receives all paths reset to the defaults set in the paths win dow .

/@KEYWORDS "Creating a new project window.@"{fg text}

1.20 Menu Item:Open

Menu: Project

Item: Open...

Opens a filerequester to be used to select the icon to load.

/@KEYWORDS "Opening an existing icon."

1.21 Menu Item:Open New

Menu: Project

Item: Open New...

Combines the **new** and **open** functions into one action.

1.22 Menu Item:Restore

Menu: Project

Item: Restore

/@KEYWORDS "Restoring a project to it's original state."

Reloads the icon information from the time it was last saved.

1.23 Menu Item:Save

Menu: Project

Item: Save

/@KEYWORDS "Saving the icon."

Saves the icon using the current filename.

Notes

Be careful if the name is untitled. You may end up overwriting your default icon.

This menu operation ignores the "Always overwrite icons?" setting, and will always overwrite the icon, unlike **save as**.

1.24 Menu Item:Save As

Menu: Project

Item: Save As...

/@KEYWORDS "Saving the icon under a new name."

Saves the icon using a file requester.

Notes

If there is an existing icon, the actual operation is affected by the Always Overwrite Icons? setting.

1.25 Menu Item:Close

Menu: Project

Item: Close

/@KEYWORDS "Closing a project window."

Closes the current project window. If the window is the last project opened, Iconian will quit.

1.26 Menu Item:Import

Menu: Project

Item: Import

/@KEYWORDS "Importing pointer images."

Sub: Pointer Prefs...

Imports a standard OS3.0 pointer preference file. When running on a custom screen, the pointer colors will also be loaded, starting with register 0. The user defined area will be set to the pointer hotspots.

1.27 Menu Item:Export

Menu: Project

Item: Export

/@KEYWORDS "Exporting icon images as source code."

Sub: ASM Source Code...

Sub: C Source Code...

Sub: E Source Code...

Saves the icon as a source code file containing one or two Intuition image structures.

The file is not complete, and will need some hand editing to be useful.

Sub: Chunky Array...

Similar as other source code, but the image is saved as a series of chunky bytes as opposed to a planar structure.

/@KEYWORDS "Exporting pointer images."

Sub: Pointer Prefs...

Saves both images as a standard, system usable pointer preference file.

Both hotspots are taken from the upper-left corner of the user defined area tool.

Two requesters will appear prompting for the resolution types.

Please note that not all resolutions are available for all machines.

The exported files may be used in conjunction with the gadget directory customization option.

1.28 Menu Item:Buffer Size

Menu: Project

Item: Buffer Size

/@KEYWORDS "Changing the graphic buffer dimensions."

Subs: Width=x (multiple entries)

Height=x (multiple entries)

Enlarges or reduces the usable buffer size. This operation is undoable, so use caution.

Subs: Width=User

Height=User

/@KEYWORDS "Using user definable buffer sizes."

Changes the buffer to the user defined buffer sizes. These parameters are available from both the CLI and tooltypes.

1.29 Menu Item:Highlight

Menu: Project

Item: Highlight

/@KEYWORDS "Changing the icon highlight method."

Subs: Complement

Backfill

Image

Changes the icon highlighting method. If necessary, the project window will modify its shape and size.

Notes

Backfill images will not be previewed correctly when selected.

See Also

Project windows

1.30 Menu Item:Icon Type

Menu: Project

Item: Icon Type

/@KEYWORDS "Changing the icon type."

Subs: Disk

Drawer

Tool

Project

Trashcan

Device

Kick

AppIcon

Changes the icon type.

Device, Kickstart, and AppIcon types are really not useful, and are only included to be complete.

Different icon types affect the icon information window.

See Also

Icon information window.

1.31 Menu Item:Save Mode

Menu: Project

Item: Save Mode

/@KEYWORDS "Changing how the icon will be saved."

Sub: Strip Planes

Most efficient way to save an icon. Only the minimum number of planes will be saved.

Sub: Current Depth

The icon will be stored using the exact number of planes of the screen it was designed on.

Sub: Force Eight

Commodore's normal save depth. This format is only useful for eight color icons, designed on an eight color screen. (8 is the magic number here.)

Sub: Auto Size

The dimensions of the icon will be automatically based on the icon imagery.

Sub: Full Size

The icon dimensions will be the same as the buffer size.

Sub: Custom Size

The icon size will be taken from the current "normal" image's user defined area. Use the define area tool to select the dimensions.

Sub: NewIcon Standard

The icon will be saved as a 40x36 rectangle, regardless of the image or buffer size.

Sub: Explode Mode

This special save mode takes advantage of an icon's ability to have an image larger than the gadget structure.

The gadget size will be taken from the user defined area size. Use the define area tool for this.

The image size will be automatically sized according to the visible image size, identical to the auto size method.

To make the icon bevel box appear correctly, you will need to re-create the righthand and bottom line. Make the right hand line 4 pixels to right from the user area. The bottom line is 3 pixels away.

To use this mode, you'll need some patience and practice.

1.32 Menu Item:Newicons

Menu: Project

Item: Newicons

/@KEYWORDS "Newicon support functions."

Subs: Save

Save As...

Saves the current project in Newicon format. The old style icon is determined by the switches below.

Sub: Seperate Palettes?

When selected, each icon image has it's own palette. The palette is only displayed when the image is the current project.

This option is only available when Iconian is run on a custom screen.

Sub: Mark as Standard

Selects the current project as the standard (old style) image. The project's beveled box will turn be highlighted to signal this.

Sub: Dot is Standard

This signals that the Newicon will not have a standard image.

This can decrease the file size, but will cause incompatibilities when Newicon is not present, both on the host and other computers.

1.33 Menu Item:Iconify

Menu: Project

Item: Iconify

Places Iconian in an asleep state.

Notes

The only way to awaken Iconian is by the appicon. Double clicking will simple awaken Iconian. Dropping icons on the appicon will awaken Iconian, and the icons will be opened as new projects.

The ARexx port is only active when Iconian is. ARexx messages sent while Iconian is asleep will be held until Iconian reawakens.

1.34 Menu Item:About

Menu: Project

Item: About

Simply opens an about window. The current ARExx port is displayed, along with other copyright and version information.

1.35 Menu Item:Quit

Menu: Project

Item: Quit

Quits if all projects have been saved. Otherwise, confirmation must be made to each modified project.

1.36 Menu Item:Save Mode

Menu: Project

Item: Quit

Quits if all projects have been saved. Otherwise, confirmation must be made to each modified project.

1.37 Menu Item:Undo

Menu: Edit

Item: Undo

/@KEYWORDS "Undoing graphic operations."

Moves back in the undo chain one spot.

Notes

Any changes made to a buffer will destroy all "redos" from that point on.

See Also

[Redo](#)

[Undo Buffers](#)

1.38 Menu Item:Redo

Menu: Edit

Item: Redo

/@KEYWORDS "Redoing graphic operations."

Moves forward in the undo chain one spot.

Notes

Any changes made to a buffer will destroy all "redos" from that point on.

See Also

[Undo](#)

[Undo Buffers](#)

1.39 Menu Item:Paste

Menu: Edit

Item: Paste

Loads the any current clipboard image as a datatype.

See Also

[Load Picture](#)

/@KEYWORDS "Pasting images from the clipboard."

1.40 Menu Item:Cut

Menu: Edit

Item: Cut

Places the current image into the clipboard as an image. The image will then be cleared.

/@KEYWORDS "Cutting to the clipboard."

1.41 Menu Item:Copy

Menu: Edit

Item: Copy

Places the current image into the clipboard as an image.

/@KEYWORDS "Copying to the clipboard."

1.42 Menu Item:Erase

Menu: Edit

Item: Erase

Simply erases the current image buffer.

1.43 Menu Item:Exchange

Menu: Edit

Item: Exchange

Swaps the normal and selected buffers.

1.44 Menu Item:Paste from Other

Menu: Edit

Item: Paste from Other

Copies the other image into the current buffer.

Notes

This operation is backwards from IconEdit's "Copy to Other".

1.45 Menu Item:Undo Buffers

Menu: Edit

Item: Undo Buffers

Subs: Goto Oldest

Goto Newest

Similar to undo and redo, but will travel the entire possible distance.

Subs: Flush All

Flush Current

Frees the memory occupied by the undo buffers.

"Flush All" will affect all buffers.

"Flush Current" will only free the current image's undo buffers.

These operations are not reversible, so a confirmation must be made.

Sub: Set Memory Size...

This sets the total maximum amount of memory the undo buffers will occupy.

For registered users, this value is basically unlimited.

1.46 Menu Item:Load

Menu: Images

Item: Load

Subs: Picture...

Picture and Palette...

Loads a datatype into the current buffer, according to the current datatype settings.

If an area is defined, the image will be best fitted into the area.

Subs: Normal Image...

Selected Image...

Both Images...

Loads icon images into the buffer.

Normal and Selected load only one image, into the current buffer.

Both images loads the normal image into the normal buffer, etc..

Subs: Default Images

WhatIs Images

Similar to the above three options, but automatically selects what icon to load from.

Default are system default icons.

WhatIs is much better, but only works when the whatis.library is correctly installed.

1.47 Menu Item:Save Picture

Menu: Images

Item: Save Picture...

Saves the current image as an IFF brush file.

Notes

No run-length encoding is done. Rather, a simple zero compression is used.

1.48 Menu Item:Move Image

Menu: Images

Item: Move Image

Sub: Auto Topleft

Automatically moves the current image to the upperleft corner.

Subs: Up

Down

Right

Left

Moves the image one pixel in the specified direction.

Notes

The keyboard equivalents for these are CTRL+Cursor.

1.49 Menu Item:Flip Image

Menu: Images

Item: Flip Image

Subs: Quick Horizontal

Quick Vertical

A simple flip, either left to right, or up and down.

Subs: Smart Horizontal

Smart Vertical

A better flip routine. Only the occupied area will be flipped.

1.50 Menu Item:Scale Image

Menu: Images

Item: Scale Image

Subs: Half Horizontal

Half Vertical

Half Both

Halves the image in the specified direction(s).

Subs: Double Horizontal

Double Vertical

Double Both

Doubles the image in the specified direction(s). /@KEYWORDS "Resizing the image."

1.51 Menu Item:Recolor

Menu: Images

Item: Recolor

Sub: 1.3 to 2.0

Swaps colors 1 and 2. Icons designed before OS2.0 had white in register 1 and black in register 2. With the introduction with OS2.0, black is now in register 1 and white in register 2.

Sub: 3 to 8 Planes

Recolors registers 4 through 7 with the highest 4 colors available.

Sub: Use 2 Planes

Remaps the buffer down to the first four colors, relying on the current palette values of each pixel.

Sub: Recolor Pen2 with Pen1

Remaps all pixels of color 2 with color 1.

Sub: Swap Pen1 and Pen2

Remaps colors 1 and color 2. Has a near equivalent gadget called **remap**. /@KEYWORDS "Recoloring the image." /@KEYWORDS "Remapping pen colors."

1.52 Menu Item:Use Brush

Menu: Brushes

Item: Use Brush

Subs: Brush 1

Brush 2

Brush 3

Brush 4

Brush 5

Selects the current brush to draw with and operate on.

Notes

This menu may consist of graphical representations of the brushes, if the appropriate tooltype/cliswitch is activated. /@KEYWORDS "Selecting a different brush."

1.53 Menu Item:Load Brush

Menu: Brushes

Item: Load Brush...

Loads a datatype into the current brush's buffer.

Notes

The maximum brush size is equal to the largest project buffer in use. Images will be scaled to fit this buffer, if scaling of datatypes is set.

This operation is affected by the current datatype settings.

See Also

Load Picture /@KEYWORDS "Loading a brush."

1.54 Menu Item:Save Brush

Menu: Brushes

Item: Save Brush...

Saves the brush as an IFF file.

Notes

No compression is used when saving the brush.

See Also

Save Picture /@KEYWORDS "Saving a brush."

1.55 Menu Item:Enter Text

Menu: Brushes

Item: Enter Text...

Brings up a text requester to enter a string to be used by the Paste text drawing mode.

Notes

This operation is not available if ReqTools is not installed. /@KEYWORDS "Changing the paste text string."

1.56 Menu Item:Select Font

Menu: Brushes

Item: Select Font

Use the requester to select the font and style for the paste text drawing tool.

Notes

The font name and style will be saved when the window font is snapshotted. /@KEYWORDS "Selecting the text brush font style."

1.57 Menu Item:Text Mode

Menu: Brushes

Item: Text Mode

Subs: Jam1

Jam2

Complement

Inverse Video

Selects how the text will appear when drawn into the buffer. /@KEYWORDS "Changing the brush text mode."

1.58 Menu Item:Flip Brush

Menu: Brushes

Item: Flip Brush

Subs: Horizontal

Vertical

Flips the brush in the specified direction.

1.59 Menu Item:Scale Brush

Menu: Brushes

Item: Scale Brush

Subs: Half Horizontal

Half Vertical

Half Both

Double Horizontal

Double Vertical

Double Both

Scales the brush in the specified direction(s).

1.60 Menu Item:Brush Mode

Menu: Brushes

Item: Brush Mode

Changes the brush mode to one of the following:

Sub: Matte

Color 0 will be transparent.

Sub: Color

All colors other than 0 will appear in the current pen number.

Sub: Replace

Brush will appear exactly as is.

Sub: Complement

Brush will XOR (Complement) into the buffer.

Sub: Smooth

Colors will be smoothed where the brush holds a color other than 0.

1.61 Menu Item:Handle Position

Menu: Brushes

Item: Handle Position

Subs: Center

Top Left

Top Right

Bottom Left

Bottom Right

Custom

Changes where the "handle" of the brush is.

Notes

Custom is meant for future expansion.

1.62 Menu Item:Drawing Mode

Menu: Paint

Item: Drawing Mode

Subs: Plot

Draw

Line

Beveled Box

Rectangle

Filled Rectangle

Oval

Filled Oval

Fill

Filled Polygon

Air Brush

Define Area

Cut Brush

Paste Brush

Stretch Brush

Paste Text

Changes the current drawing mode. There are keyboard and gadget equivalents, also.

1.63 Menu Item:Bevel Box Type

Menu: Paint

Item: Bevel Box Type

Subs: Single

Button

Ridge

Dropbox

Changes the frame style of the draw bevel box tool.

Sub: Recessed

The bevel box will appear recessed.

Sub: Filled

Fills the bevel box with the current draw pen color.

1.64 Menu Item:Air Brush

Menu: Paint

Item: Air Brush

Subs: Small

Medium

Large

Changes the size of the spraying area.

Sub: Focas?

Paint will be focused around the center.

Sub: Drip?

Acts like a spray paint can.

1.65 Menu Item:Dither

Menu: Paint

Item: Dither

Subs: None

94%

88%

75%

63%

50%

38%

25%

13%

7%

Bars

Lines

Changes the dithering appearance.

Notes

There is are keyboard and gadget equivalents.

Sub: Transparent?

Only one color will be used, and the background will show through.

1.66 Menu Item:Palette

Menu: Paint

Item: Palette

Sub: Load Palette...

Loads a palette into either the global colormap, or if possible, into the individual colormap.

Sub: Save Palette...

Saves the current palette to disk.

Subs: MagicWB Preset

NewIcon Preset

RomIcon Preset

JBCIcon Preset

Iconographix Preset

Quickly selects one of several popular palettes.

1.67 Menu Item:Open

Menu: Windows

Item: Open

Subs: Settings...

Paths...

Icon Information...

Datatype Settings...

Macros...

Mover...

Palette...

Tooltypes...

Opens one of the preference windows. If window is already open, then it will be brought to the top.

1.68 Menu Item:Screenmode

Menu: Windows

Item: Screenmode

Subs: Workbench

Default Pubscreen

Places Iconian on either a public screen. Several palette features will be disabled.

Sub: Custom Screen...

Places Iconian on it's own screen. This enables all palette features.

Notes

The custom screen is not a public one for several reasons. Iconian manipulates the palette in many ways and would interfere with visiting windows.

1.69 Menu Item:Window Font

Menu: Windows

Item: Window Font

Selects the font to be used for gadgets and other window text.

Notes

When Iconian is first installed, it defaults to topaz 8. For the love of god, select a better font!

1.70 Menu Item:Snapshot

Menu: Paint

Item: Snapshot

Subs: Windows

Paths

Screenmode

Fonts

Macros

Settings

Datatype Settings

Saves a specific preference file to ENVARC:Iconian.

1.71 Menu Item:Load Preferences

Menu: Paint

Item: Load Preferences

Resets Iconian to use a specified preference file.

Notes

Iconian saves preferences in IFF chunks. A preference file may contain only one setting, or many.

1.72 Menu Item:Save Preferences as

Menu: Paint

Item: Save Preferences

Stores all preferences into one single file. This file may be passed as tooltype/cliswitch upon startup.

1.73 Menu: User (Macros)

1.74 MENU#0192

See: User (Macros)

1.75 MENU#0193

See: User (Macros)

1.76 MENU#0194

See: User (Macros)

1.77 MENU#0195

See: User (Macros)

1.78 MENU#0196

See: User (Macros)

1.79 MENU#0197

See: User (Macros)

1.80 MENU#0198

See: User (Macros)

1.81 MENU#0199

See: User (Macros)

1.82 MENU#019A

See: [User \(Macros\)](#)

1.83 MENU#019B

See: [User \(Macros\)](#)

1.84 MENU#00D9

See: [Contents](#)

1.85 MENU#00DA

See: [Index](#)

1.86 Menu Item: Search

Menu: Help

Item: Search

Enter the keyword search pattern you wish help on. Full AmigaDOS wildcard patterns are supported.

Example

"#?save#?" - Brings up all operations involving saving data to disk.

1.87 Magnification Area

Magnification area

The magnification area is used to edit icon images. To take full advantage of Iconian 3, you must learn how to use and exploit the magnification area.

Along the right and bottom sides are two scrollers. These control where in the icon image you are viewing. The position and size of these scrollers are affected by several factors. Magnification level, icon buffer size, magnification view size, and the current focal point.

On the extreme right side is the magnification level. When this slider is at the far bottom, the magnification window will display the current project's image at real size. When this slider is at the top, every real pixel will be displayed as a 16x16 box.

When the pointer is within the magnification drawing area, it will change to one of several shapes. These shapes reflect the current drawing mode. The current position will also be reflected by either a dot or crosshairs following the pointer.

Each drawing mode has its own method of operation. Clicking in the magnification area will begin the current drawing mode. Releasing the mouse button will usually finish the draw procedure. The magnification area will update what the finished draw procedure will look like in real time.

See Also

[Beveled box gadget](#)

[Dither type gadget](#)

Keyboard commands

Magnification level

Redo button

Select drawing tool

Select pen color

Undo button

1.88 Using Project Windows

Mouse operation in project windows.

There will be one or two boxes in the window, depending upon the highlight method. Clicking in either box will select that icon and image as the "focas" project.

Also, pressing the mouse button in the first box will display what the icon will appear as when highlighted.

Special operations rely on the center on select setting.

Notes

Backfill highlighting preview is unsupported. Instead, the icon will appear complemented when selected.

1.89 Horizontal scroller

Slides the view to the left and to the right.

1.90 Vertical scroller

Slides the view up and down.

1.91 Magnification level

Adjust the magnification level.

Keyboard commands: "<" ">" - Changes magnification level.

1.92 Select drawing tool

Changes the current drawing mode.

See Also: Plot Draw Line Beveled Box Rectangle Filled Rectangle Oval Filled Oval Fill Filled Polygon Air Brush Define Area Cut Brush Paste Brush Stretch Brush Paste Text

1.93 Select pen color

Selects the color to draw with.

1.94 Select background color

Selects the color to erase with.

1.95 Undo button

Undos the previous drawing action.

Keyboard command: "u"

1.96 Redo button

Redos the next drawing action.

Keyboard command: "shift-u"

1.97 Icon type

Selects the icon type.

See Also: Menu Item:Icon Type

1.98 Highlight type

Selects the icon highlight method.

See Also: Menu Item:Highlight

1.99 Icon left edge

Changes where the left edge of the icon will be.

1.100 Icon top edge

Changes where the top edge of the icon will be.

1.101 Icon stack size

Changes the stack size of the icon.

1.102 Free icon position

Changes the freedom state of the icon.

1.103 Icon window left edge

Changes the drawer window left edge.

1.104 Icon window top edge

Changes the drawer window top edge.

1.105 Icon window width

Changes the drawer window width.

1.106 Icon window height

Changes the drawer window height.

1.107 Center datatype image

Determines if the image will be centered in the defined area.

1.108 Integer scale

Integer scaling will be used. This helps eliminate artifacts.

1.109 Remap

Remapping will be done.

1.110 Use aspect

Scaling will retain aspects.

1.111 Aspect X

The screen ratio, determined by X:Y.

1.112 Aspect Y

The screen ratio, determined by X:Y.

1.113 Scale

Scales the image to fit inside the defined area or the image buffer.

1.114 Stretch

Image will be stretched to exactly fit the defined area.

1.115 Dither

Dithering will be used.

1.116 Clear

The buffer will be cleared before rendering.

1.117 Default icon directory

Determines where the default icon directory is.

1.118 Default icon directory selector

Brings up a directory selector. Select the default icon directory.

See Also: Default icon directory

1.119 Default icon image directory

Determines where the default icon image directory is.

1.120 Default icon image directory selector

Brings up a directory selector. Select the default icon image directory.

1.121 Default picture directory

Determines where the default picture directory is.

1.122 Default picture directory selector

Brings up a directory selector. Select the default picture directory.

1.123 Gadget directory

Determines where the user gadget directory is. Tool gadget images, user palette, and mouse pointer images will be used if present.

To create tool gadget images, save the images as icons to the gadget directory. The images are named "Gadget-00" through "Gadget-15". These are numbered from the upper left to lower right.

To change the mouse pointer images, use the "Export...Pointer Prefs" menu item. The pointer files are system 3.0 compatible and may also be edited with the system pointer prefs program. Iconian utilizes the busy image when the select button is pressed.

To specify a palette for custom screen, save a palette file as "Default.pal" in the gadget directory. This file will be loaded upon startup. (May NOT be implemented yet)

It is possible to leave out one or any of the above. Iconian will default back to internal images if need be.

See Also: Export Palette

1.124 Use gadget directory

Determines if the user gadget directory is scanned at startup.

See Also: Gadget directory

1.125 Gadget directory selector

Brings up a directory selector. Select the default gadget directory.

See Also: Gadget directory

1.126 Default appicon

Determines what the appicon will be.

1.127 Default appicon selector

Brings up a directory selector. Select the default appicon file.

See Also: Default appicon

1.128 Red component value (string)

Changes the current pen color's red value.

1.129 Green component value (string)

Changes the current pen color's green value.

1.130 Blue component value (string)

Changes the current pen color's blue value.

1.131 Red component value (slider)

Changes the current pen color's red value.

1.132 Green component value (slider)

Changes the current pen color's green value.

1.133 Blue component value (slider)

Changes the current pen color's blue value.

1.134 Copy color

Copies the current color to the next selected color. The destination color may be selected by picking one from the palette window, or by clicking in the drawing area. To cancel, select the gadget a second time.

Notes: This operation currently cannot be undone, so use caution.

1.135 Exchange color

Exchanges the current color with the next selected color. The destination color may be selected by picking one from the palette window, or by clicking in the drawing area. To cancel, select the gadget a second time.

Notes: This operation currently cannot be undone, so use caution.

1.136 Spread color

Spreads the current color to the next selected color. The destination color may be selected by picking one from the palette window, or by clicking in the drawing area. To cancel, select the gadget a second time.

Notes: This operation currently cannot be undone, so use caution.

1.137 Remap color

Remaps the current color with the next selected color. The destination color may be selected by picking one from the palette window, or by clicking in the drawing area. To cancel, select the gadget a second time.

1.138 Dither method

Selects the dithering method.

1.139 Quantize method

Selects the color reduction method.

1.140 Palette

Selects the current pen color.

1.141 Tooltypes

Lists the tooltypes of the current icon.

1.142 Tooltype entry

Edits the currently selected tooltype.

1.143 Add tooltype

Adds a blank tooltype to the tooltype list.

1.144 Remove tooltype

Removes the currently selected tooltype.

Notes: This operation currently cannot be undone, so use caution.

1.145 Dither type gadget

Selects the dithering draw mode.

1.146 Beveled box gadget

Selects the type of beveled box.

1.147 Macro #1 selector

Selects macro #1.

1.148 Macro #2 selector

Selects macro #2.

1.149 Macro #3 selector

Selects macro #3.

1.150 Macro #4 selector

Selects macro #4.

1.151 Macro #5 selector

Selects macro #5.

1.152 Macro #6 selector

Selects macro #6.

1.153 Macro #7 selector

Selects macro #7.

1.154 Macro #8 selector

Selects macro #8.

1.155 Macro #9 selector

Selects macro #9.

1.156 Macro #10 selector

Selects macro #10.

1.157 Macro #1 string entry

Specifies macro #1.

1.158 Macro #2 string entry

Specifies macro #2.

1.159 Macro #3 string entry

Specifies macro #3.

1.160 Macro #4 string entry

Specifies macro #4.

1.161 Macro #5 string entry

Specifies macro #5.

1.162 Macro #6 string entry

Specifies macro #6.

1.163 Macro #7 string entry

Specifies macro #7.

1.164 Macro #8 string entry

Specifies macro #8.

1.165 Macro #9 string entry

Specifies macro #9.

1.166 Macro #10 string entry

Specifies macro #10.

1.167 Auto enlarge

Always enlarges buffer to fit oversized icons.

1.168 Always overwrite icons

Existing icons will always be overwritten.

1.169 Filter datatypes

Only datatype pictures will be displayed in picture file requesters.

1.170 Mini sunmouse

The main window will be activated when the mouse pointer passes over it. This helps for right mouse button drawing operations.

1.171 Center on select

The magnified view will center when the pointer is clicked in the project windows.

1.172 Size Window Gadget

Size Window Gadget

Sizes the window. Click and hold the mouse button. Move the window to the desired size and release the button.

1.173 Drag Window Gadget

Drag Window Gadget

Moves the window. Click and hold the mouse button. Move the window to the desired position and release the button.

1.174 Window Depth Gadget

Window Depth Gadget

Changes the window's depth position. If it is the top-most window, it will be moved to the very back. Otherwise, the window will be moved to the very front.

NOTES:

Iconian 3 does not support saving window depth positions to preference files. Windows will always be opened in a specific order.

1.175 Zip Window Gadget

Zip Window Gadget

Zips the window size to the alternate position and dimensions.

NOTES:

Iconian 3 does not support saving alternate positions to preference files. Only the current position and dimensions will be remembered.

1.176 Close Window Gadget

Close Window Gadget

Closes the window.

Main Window

Quits the program. Requesters will appear for any unsaved projects.

Project Window

Closes the project. You must confirm the action if the project has been changed. If the window is the last open project, the program will quit.

Preference and Settings Windows.

Hides the window. You can bring it back using Windows->Open->* menu items.

1.177 ARexx: ABOUT

ABOUT

Function:

Display information window.

Syntax:

ABOUT

1.178 ARexx: AIRBRUSH

AIRBRUSH

Function:

Activates and controls airbrush draw tool.

Syntax:

AIRBRUSH ("SMALL" | "MED" | "LARGE")
("DRIP" | "NORMAL")
("FOCAS" | "EVEN")

Template:

SMALL/S,MED/S,LARGE/S,DRIP/S,NORMAL/S,FOCAS/S,EVEN/S

Options:

"SMALL", "MED", "LARGE"

Select airbrush area size.

"DRIP", "NORMAL"

Select spraypaint method.

"FOCAS", "EVEN"

Select aiming method.

Example:

/*Activate airbrush draw tool, sets small and no focusing attributes:*/

AIRBRUSH SMALL EVEN

1.179 ARexx: BUFFERSIZE

BUFFERSIZE

Function:

Change buffer size of current project.

Syntax:

BUFFERSIZE <width> <height> | <width> | "H" <height>

Template:

W/N/A,H/N/A

1.180 ARexx: BRUSH

BRUSH

Function:

Activate and cut brushes.

Syntax:

BRUSH (<left> <top> <width> <height> ("CUT")) (<whichbrush>)

Template:

LEFT/N,TOP/N,WIDTH/N,HEIGHT/N,CUT/S,WHICH/N

Options:

LEFT <left> TOP <top> WIDTH <width> HEIGHT <height>

Defines upperleft corner.

CUT

Area will be cleared after brush is grabbed.

WHICH <whichbrush>

Selects current brush to use. Valid range is 0 thru 4.

Example:

```
/*Cut out a brush and store it in buffer #3*/
```

```
BRUSH WHICH=3 0 0 15 15 CUT
```

1.181 ARexx: CHANGESAVEDEPTH

CHANGESAVEDEPTH

Function:

Changes the save depth method of the current project.

Syntax:

CHANGESAVEDEPTH <savedepth>

Template:

MODE/N/A

Options:

MODE <savedepth>

A value of 0 thru 2, corresponding to the following:

0. Strip Planes
1. Current Depth
2. Force Eight

1.182 ARexx: CHANGESAVEMODE

CHANGESAVEMODE

Function:

Select how an icons dimensions are obtained when saved.

Syntax:

CHANGESAVEMODE <sizemethod>

Template:

MODE/N/A

Options:

MODE <sizemethod>

A value of 0 thru 4, corresponding to the following:

0. Auto Size
1. Full Size
2. Custom Size
3. NewIcon Standard Size
4. Explode Method

1.183 ARexx: CIRCLE

CIRCLE

Function:

Activate circle draw tool, and to draw circles.

Syntax:

CIRCLE (<xcoord> <ycoord> <xradius> (<yradius>) ("FILL"))

Template:

X/N,Y/N,RADX/N,RADY/N,FILL/S

Options:

X <xcoord>, Y <ycoord>

Selects the circle/oval center point.

RADX <xradius>, RADY <yradius>

Adjusts the circle size.

"FILL"

Fills the circle interior.

Example:

```
/*Draw a plain circle.*/
```

```
CIRCLE 50 50 25 25
```

1.184 ARexx: CLEAR

CLEAR

Function:

Clears the graphics buffer of the current project, using the current background pen.

Syntax:

CLEAR

1.185 ARexx: CLOSE

CLOSE

Function:

Closes a project window.

Syntax:

CLOSE (<project>) ("FORCE")

Template:

PROJECT/N,FORCE/S

Options:

PROJECT <project>

Selects which project window to close.

"FORCE"

Closes window, even if project has not been saved.

1.186 ARexx: CLOSESTCOLOR

CLOSESTCOLOR

Function:

Locates the pen nearest the given color values.

Syntax:

CLOSESTCOLOR <red> <green> <blue>

Template:

RED/N/A, GREEN/N/A, BLUE/N/A

Result:

The pen number of the closest color found.

Options:

RED <red>, GREEN <green>, BLUE <blue>

Selects the RGB values to locate.

Example:

```
/*Locate a greyish color pen.*/
```

```
CLOSESTCOLOR 128 128 128
```

```
greypen=result
```

1.187 ARexx: COLOR

COLOR

Function:

Activates the "Color" brush draw mode.

Syntax:

COLOR

1.188 ARexx: COPY

COPY

Function:

Copies the current graphic buffer to the clipboard device.

Syntax:

COPY

1.189 ARexx: CUT

CUT

Function:

Cuts the current graphic buffer to the clipboard device.

Syntax:

CUT

1.190 ARexx: DOTISSTANDARD

DOTISSTANDARD

Function:

Specifies that the Newicon old style image is a dot.

Syntax:

DOTISSTANDARD

Results:

none

1.191 ARexx: DOUBLEBRUSH

DOUBLEBRUSH

Function:

Doubles the brush size in the specified direction(s).

Syntax:

DOUBLEBRUSH ("X") ("Y")

Template:

X/S,Y/S

1.192 ARexx: ERASE

ERASE

Function:

Erases the graphics buffer of the current project, using color 0.

Syntax:

ERASE

1.193 ARexx: EXCHANGE

EXCHANGE

Function:

Swaps the normal and selected graphics buffers.

Syntax:

EXCHANGE

1.194 ARexx: FILL

FILL

Function:

Fills in enclosed areas.

Syntax:

FILL (<xcoord> <ycoord>)

Template:

X/N,Y/N

Options:

X <xcoord>, Y <ycoord>

Selects where to fill from.

1.195 ARexx: FLUSHUNDOS

FLUSHUNDOS

Function:

Frees undo buffer memory.

Syntax:

FLUSHUNDOS ("ALL") ("FORCE")

Template:

ALL/S,FORCE/S

Options:

"ALL"

All undo buffers will be flushed, not only the current image.

"FORCE"

Suppresses verification requester.

1.196 ARexx: FREEHAND

FREEHAND

Function:

Selects freehand drawing tool.

Syntax:

FREEHAND ("FILL")

Template:

FILL/S

Options:

"FILL"

Activates filled polygon method.

1.197 ARexx: GETBGCOLOR

GETBGCOLOR

Function:

Obtains the current background pen value.

Syntax:

GETBGCOLOR

Result:

The current secondary (background) pen.

1.198 ARexx: GETBUFFERSIZE

GETBUFFERSIZE

Function:

Obtains the current buffer width or height.

Syntax:

GETBUFFERSIZE "WIDTH" | "HEIGHT"

Template:

W=WIDTH/S,H=HEIGHT/S

Result:

Either the width or height in pixels.

1.199 ARexx: GETCOORD

GETCOORD

Function:

Obtains the current user mouse location.

Syntax:

GETCOORD "X" | "Y"

Template:

X/S,Y/S

Result:

The mouse location, either horizontally or vertically.

Notes:

The value returned may be outside the bufferspace. Use GETBUFFERSIZE to check if the returned location is valid, and handle appropriately.

Example:

```
/*Get the current mouse location, and store these in mousex and mousey*/
```

OPTIONS RESULTS

GETCOORD X

mousex=RESULT

GETCOORD Y

mousey=RESULT

1.200 ARexx: GETFGCOLOR

GETFGCOLOR

Function:

Obtains the current foreground pen value.

Syntax:

GETFBCOLOR

Result:

The current primary (foreground) pen.

1.201 ARexx: GETPIXEL

GETPIXEL

Function:

Get the color value at a specified coordinate.

Syntax:

GETPIXEL <xcoord> <ycoord>

Template:

X/N/A,Y/N/A

Result:

The value at the specified coordinate.

Notes:

No checking is done to see if this is a legal coordinate. 0 will be returned if it outside the buffer area.

1.202 ARexx: GETSCREENDEPTH

GETSCREENDEPTH

Function:

Returns the number of bitplanes of the screen Iconian is running on.

Syntax:

GETSCREENDEPTH

Result:

The actual number of bitplanes of the screen Iconian is running on.

1.203 ARexx: HIGHLIGHT

HIGHLIGHT

Function:

Changes the highlight method of an icon.

Syntax:

HIGHLIGHT "COMPLEMENT" | "BACKFILL" | "IMAGE"

Template:

C=COMPLEMENT/S,B=BACKFILL/S,I=IMAGE/S

1.204 ARexx: ICONTYPE

ICONTYPE

Function:

Changes the icon type of an icon.

Syntax:

ICONTYPE <type> | "PROJECT" | "TOOL" | "DRAWER"

Template:

TYPE/N,PROJECT/S,TOOL/S,DRAWER/S

Options:

TYPE <type>

Where type is one of the following:

1. Disk
2. Drawer
3. Tool
4. Project
5. Trashcan
6. Device
7. Kickstart
8. AppIcon

1.205 ARexx: LEFTMOUSE

LEFTMOUSE

Function:

Controls the left mouse button state.

Syntax:

LEFTMOUSE "UP" | "DOWN" | "CLICK"

Template:

UP/S,DOWN/S,CLICK/S

Options:

"UP"

Acts as if the user has released the left button.

"DOWN"

Acts as if the user has pressed the left button.

"CLICK"

Acts as if the user has clicked the mouse button rapidly.

Notes:

The command, like all the other mouse commands, only activates drawing states and methods. Other mouse functions like gadgets, windows, and menus are not affected by these commands.

Attempting to start drawing while the macro-cursor position is outside the drawing area will fail. To ensure that a click or mouse down command will take affect, use the GETBUFFERSIZE commands. Check the desired position against 0,0 and width-1,height-1.

1.206 ARexx: LOADDATATYPE

LOADDATATYPE

Function:

Begins the datatype loading procedure.

Syntax:

LOADDATATYPE file ("FILLCMAP")

Template:

FILE/A,FILLCMAP/S

Options:

FILE file

Specifies the picture to load.

FILLCMAP "FILLCMAP"

This switch will load the colormap associated with the picture.

Example:

```
/*Load a picture into the buffer, using current DT settings.*/
```

```
LOADDATATYPE picture.gif FILLCMAP
```

1.207 ARexx: LOCKGUI

LOCKGUI

Function:

Disables user input into Iconian.

Syntax:

LOCKGUI ("FASTDRAW")

Template:

FASTDRAW/S

Options:

FASTDRAW "FASTDRAW"

This switch disables all visual updates. Once an UNLOCKGUI is issued, all visual information is updated. This reduces the overhead of long scripts.

1.208 ARexx: MARKASSTANDARD

MARKASSTANDARD

Function:

Marks the current project as the NewIcon standard image.

Syntax:

MARKASSTANDARD

1.209 ARexx: MATTE

MATTE

Function:

Selects the "matte" brush drawing mode.

Syntax:

MATTE

1.210 ARexx: MOVETO

MOVETO

Function:

Specifies the script mouse position, for use with LEFTMOUSE/RIGHTMOUSE.

Syntax:

MOVETO <xcoord> <ycoord>

Template:

X/N/A,Y/N/A

1.211 ARexx: OPEN

OPEN

Function:

Loads an icon, with option of opening a new project window.

Syntax:

OPEN file ("NEW") | "NEW"

Template:

FILE,NEW/S

Result:

The number of the new window, but only if the "NEW" keyword is used.

Options:

NEW "NEW"

A new window will be opened to accept the icon file.

1.212 ARexx: PASTE

PASTE

Function:

Pastes from the clipboard, if an image is present. Also will copy from the other project buffer.

Syntax:

PASTE ("OTHER")

Template:

OTHER/S

Options:

OTHER "OTHER"

Unrelated to the clipboard. Instead, the image from the other buffer, from the same project, will be copied into the current buffer.

1.213 ARexx: PLOT

PLOT

Function:

Sets a pixel at a specified location.

Syntax:

PLOT <x> <y>

Template:

X/N/A,Y/N/A

1.214 ARexx: QUIT

QUIT

Function:

Closes the Iconian program.

Syntax:

QUIT ("FORCE")

Template:

FORCE/S

Options:

FORCE "FORCE"

Ignores any unsaved, modified projects.

1.215 ARexx: REDO

REDO

Function:

Redoes an operation, if available.

Syntax:

REDO

1.216 ARexx: REPLACE

REPLACE

Function:

Selects the "replace" brush drawing mode.

Syntax:

REPLACE

1.217 ARexx: RESTORE

RESTORE

Function:

Restores an icon to it's original, last saved state.

Syntax:

RESTORE ("FORCE")

Template:

FORCE/S

Options:

FORCE "FORCE"

Ignores any unsaved, modified projects.

1.218 ARexx: RIGHTMOUSE

RIGHTMOUSE

Function:

Controls the right mouse button state.

Syntax:

RIGHTMOUSE "UP" | "DOWN" | "CLICK"

Template:

UP/S,DOWN/S,CLICK/S

Options:

"UP"

Acts as if the user has released the right button.

"DOWN"

Acts as if the user has pressed the right button.

"CLICK"

Acts as if the user has clicked the mouse button rapidly.

Notes:

The command, like all the other mouse commands, only activates drawing states and methods. Other mouse functions like gadgets, windows, and menus are not affected by these commands.

Attempting to start drawing while the macro-cursor position is outside the drawing area will fail. To ensure that a click or mouse down command will take affect, use the GETBUFFERSIZE commands. Check the desired position against 0,0 and width-1,height-1.

1.219 ARexx: RUBTHRU

RUBTHRU

Function:

Selects the "rubthru" brush drawing mode.

Syntax:

RUBTHRU

1.220 ARexx: SAVE

SAVE

Function:

Saves an icon to disk.

Syntax:

SAVE (file) ("NEWICON")

Template:

FILE,NEWICON/S

Options:

file "FILE"

A new name to save under. Otherwise the original name is used.

NEWICON "NEWICON"

Saves the icon in NewIcon format.

1.221 ARexx: SCREENTOFRONT

SCREENTOFRONT

Function:

Brings Iconian's screen to the front position.

Syntax:

SCREENTOFRONT

1.222 ARexx: SETFGCOLOR

SETFGCOLOR

Function:

Sets the color of the primary (foreground) pen.

Syntax:

SETFGCOLOR <pennumber>

Template:

PENNUM/N/A

Options:

PENNUM <pennumber>

The new pen register to use for primary drawing.

1.223 ARexx: SETBGCOLOR

SETBGCOLOR

Function:

Sets the color of the secondary (background) pen.

Syntax:

SETBGCOLOR <pennumber>

Template:

PENNUM/N/A

Options:

PENNUM <pennumber>

The new pen register to use for secondary drawing.

1.224 ARexx: SMOOTH

SMOOTH

Function:

Selects the "smooth" brush drawing mode.

Syntax:

SMOOTH

1.225 ARexx: UNDO

UNDO

Function:

Undoes the last operation, if possible.

Syntax:

UNDO

1.226 ARexx: UNLOCKGUI

UNLOCKGUI

Function:

Allows user input following a LOCKGUI command.

Syntax:

UNLOCKGUI

1.227 ARexx: XOR

XOR

Function:

Selects the "complement" brush drawing mode.

Syntax:

XOR

1.228 ARexx: blank

Function:

Syntax:

Template:

Result:

Options:

Example:

1.229 Technical Information

Technical Information

Iconian 3 was written in Amiga E 3.2a registered. AmigaE is a custom language designed by Wouter van Oortmerssen. Iconian 3 contains over 17,500 lines of code.

Buffers are stored as off-screen rastports. These rastport each have a layer-info to do auto-clipping. Unlike earlier versions of Iconian, rastport now only have as many bitplanes as the screen.

Gadget layout is done the hard way, by hand. No layout libraries or classes are currently used, although this may change in the future, if I write such a class.

Undo information is held in lists. Each buffer image has a personal list. When an undo is created, it is date-stamped. When the total number of undo memory is reached, each list is traced to find the oldest node. Undo nodes are stored and restored used WritePixelArray8(). As this function has had problems in the past, storage may also be accomplished with a series of WritePixel()'s, at the users request.

Right mouse buttons events are trapped using IDCMP_MENUVERIFY. This IDCMP flag is only set when Iconian is in a "ready" state, and therefore does not interfere with other windows. When an IDCMP_MENUVERIFY request is found, it is quickly responded to. If the mouse is in a certain area, a CANCEL code is used, inhibiting menu operations.

All windows share a single IDCMP port. This makes menu, gadget, and key handling simple, as only a few events require knowing exactly which window actually received the event. All menus and keyboard commands are accessible in any window.

1.230 The Future of Iconian

Future

- * Finish .help guide.
 - * Work on template requester.
 - * Keep adding error handling.
 - * Add more rexx commands.
 - * Add undo ability to color/palette changes.
 - * Optimize/reduce code!!!
-

1.231 History

History

2.98 - upto April 14, 1996

New GUI layout system completed.

Tracked down a bug involving incorrect hilight/type when loading an icon.

(May have been around a while. Took me a few hours to track down, too!)

Added ordered dithering method for those that care. :P

2.97b - March 31, 1996

Fixed a few bugs when using "Open...Icon..."

Rebuilt gui layout system in E-OOP. (cleaner, more consistant look)

2.97a - March 26, 1996

Uses WB font as default instead of Topaz 8

(After seeing Iconian on JMi's WB snapshot, using TOPAZ *YUCK*)

2.97 - March 17, 1996

Added MUI bodychunk source support.

Wrote RLE packing routine for images.

2.97 - March 3, 1996

Moved the scrollers to the window borders.

Fixed the OPEN ARexx command to always strip .info.

Other various things added, changed, or fixed...

(I have not had much time to work on Iconian.

Most fixes or changes occur when I'm playing around

with it in my fre spare moments. I forget to list

the changes here...and therefore I forget what I did.)

2.96b - November 26, 1995

Installed pop.gadget support.

Removed confusing Bevel gadget.

Fixed bigbug with copy/exchange when clicking outside drawing area.

2.96 - AMINET release

minor stuff to make ready release.

2.95c - October 7, 1995

A few more error messages.

process.windowptr now always set when possible.

Asks before creating Seperate Palette buffers.

Added NewIcon color 0 transparency support.

2.95b - October 4, 1995

Sped up remapping routines (for next entry:)

Added "Remap To (preset)" item for icons with own palettes. (match up newicons...)

Line drawmode now reports 90degrees, instead of 45-89! :)

Better IFF error handling.

More error messages.

Gave all AllocBitMap()'s a "friend" bitmap, being the screen. Hope this helps a few gfxboards!?!)

Made the Write/ReadPixelArray() always use multiple of 8. (Joop van de Wege)

Made the "Maximum Project.." requester only appear once. (Peter Sandén)

2.95 - October 3, 1995

Added lots of extra-error handling in several routines.

Shifted around some more menu items. NewIcons got integrated into different places...

Started work on Template stuff.

Menubar headers will be abbreviated if it is too long to fit the current screen.

Cleaned up a few gadget routines. (More consistency).

Added pan-button.

Added middlemouse pan mode. Plus, center-on-middle-click in the project windows.

2.94k - October 2, 1995

Added POPUPTORIGHT tooltype. (James Brown)

Continued work on the help guide.

Oops!! catalogs not being loaded since 2.94e!! (I caught this one!)

Moved, changed and added several menus/menuitems.

2.94j - October 1, 1995

Iconian will now supply its address to macros in rexxmsg.commaddr !

The rexxmsg.fileext field is filled in. (Defaults to .icrx)

Will report REXX error messages.

Added Palette path/file gadget. Works only on custom screens.

2.94i - September 30, 1995

Added 2 new magnifier modes...not implemented yet.

Fixed long-standing filled oval refresh bug, finally!

Fixed major Enforcer hits in Smooth and RubThru modes.

Sped up alot of visual updates.

Sped up cpu-intensive routines upon startup. (Uwe Roehm)

Tweaked the window sizing limits. (Riccardo Giangualano)

Added some keyboard commands.

Added a few REXX commands + Added FASTDRAW switch for LOCKGUI

Now strips .info from filelist when run from CLI.

2.94h - September 26, 1995

Added save default menuitem. (Stefan Thomas Pazourek)

Made the FANCYPOPUPGADGET an option (tooltype). "" "" ""

Hopefully finally fixed all the tooltype window errors. (Riccardo Giangualano)

Fixed problem with opening Macro window on startup/iconify. (Trevor Morris)

Removed Enforcer hit when loading an icon without tooltypes. (Uwe Roehm)

Fixed some locale and string errors. (Edmund Vermeulen)

Played around with the about window, again.

Sped up a few CPU intensive routines. Iconian should load a bit faster, now.

Added APPICONSTRING tooltype.

Modified the helpfile search engine a bit. Checks help:language/*.help now, using locale.

Reworked a few cleanup routines.

Disabled ReqTool callback hooks again, until I find out why they fuck up. Grrr...

2.94g - September 25, 1995

Rewrote menu-check handler. Doesn't "assume" anymore.

Added rub-thru.

Fixed smooth brushmode. (didn't calculate handle correctly.)

Sped up some visual updating.

2.94f - September 22, 1995

Correctly recognizes tooltype/program icon on startup. (Stefan Thomas Pazourek)

Dither gadget and menu now co-operate. (Stefan Thomas Pazourek)

Rewrite of help file. (About half done!)

Added keyboard commands to menus.

Rewrote brush-image menu handler, now works with Magic Menus! (appears to anyways)

2.94e - September 20, 1995 (AMINET release)

Played with centering magnification, more like Iconian 1.98

"Center on Select" flag, moves visible area when "selecting" icon images.

Redid "default" window sizes and positions for new users. (NTSC 640x200)

Mucked with buffer sizing, added user defined sizes.

Changed a few catalog strings, mainly "about" window stuff.

Fixed a major tooltype window bug.

Traced down a few memory corruption and un-allocated memory bugs.

Added "slow undo" for GFX board users. (LET ME KNOW)

Added a few rexx commands. Whoopee! :P

Added some more keyboard commands.

Worked on brush-handle code.

2.94d - September 18, 1995

Made most busy states not effect screen colormap. (helpful with REXX scripts...)

Added dither and bevel gadgets.

Removed vertical gradient...it just crashed my machine!

Added remap gadget.

Palette and tooltype window are freed once a busy-state is done.

Changed all remapping modes to affect only current buffer.

2.94c - September 17, 1995

Fixed palette handling in Import->Pointer prefs.

Can now be run without ReqTools present (some features disabled.)

Can be run without AmigaGuide library.

Windows can be SimpleRefresh once again!!!

Added more REXX commands.

2.94b - September 17, 1995

Uploaded to Iconian's BETA ftp site ([music.phlab.missouri.edu:/pub/amiga/iconian](ftp://music.phlab.missouri.edu/pub/amiga/iconian))

2.94a - September 14-16, 1995

Rewrote AmigaGuide handler. Better path checking. Better message handling.

Rewrote REXX routines. Added several REXX commands.

Added Macros window, along with related stuff.

Added "exploding" icons save mode.

2.94 - September 11, 1995

Forth AMINET release.

--older history is, well... ancient history!