

**Iconian\_English**

**COLLABORATORS**

	<i>TITLE :</i> Iconian_English		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 2, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Iconian_English</b>	<b>1</b>
1.1	Iconian_English.guide . . . . .	1
1.2	Disclaimer . . . . .	2
1.3	Copying . . . . .	2
1.4	Introduction . . . . .	2
1.5	Neat things, and why you should use this program. . . . .	3
1.6	Requirements . . . . .	4
1.7	Installation and Running . . . . .	4
1.8	Tooltypes . . . . .	5
1.9	Menu descriptions . . . . .	5
1.10	The Project menu . . . . .	6
1.11	The Edit menu . . . . .	7
1.12	The Images menu . . . . .	7
1.13	Color menu . . . . .	8
1.14	Tools menu . . . . .	8
1.15	Settings menu . . . . .	9
1.16	Gadget descriptions . . . . .	10
1.17	Palette Sharing on public screens . . . . .	11
1.18	Included drawing tools . . . . .	11
1.19	Opps. . . . .	13
1.20	node_betatesters . . . . .	13
1.21	Keyboard equivalentents . . . . .	13
1.22	Loading of datatypes . . . . .	14

---

# Chapter 1

## Iconian\_English

### 1.1 Iconian\_English.guide

Iconian V1.90B beta  
February 26, 1994

This document describes and explains Iconian V1.90 beta, an icon editor that supports OS3.x functions. It also supports AGA display modes, such as palette sharing and 256 colors.

All binaries, executables, source, and documentation is Copyright ©1993,1994 Chad Randall. All Rights Reserved

Disclaimer	
. . . . .	Legal information
Copying	
. . . . .	Distribution and rights
Introduction	
. . . . .	The purpose of this program.
Features	
. . . . .	Features in this program.
Requirements	
. . . . .	Equipment and software required.
Installation/Running	
. . . . .	How to get this program running.
ToolTypes	
. . . . .	List of available tooltype parameters.
Menus	
. . . . .	Description of menu functions.
Gadgets	

---

. . . . . Description of gadget functions.

KeyBoard  
. . . . . Keyboard commands.

Loading Images  
. . . A very important feature.

BETA  
. . . . . README

## 1.2 Disclaimer

### Disclaimer

I am not currently planning any future updates. This may change, depending upon the amount of responses I receive, both E-Mail, and donation wise. I really wish to move to another project, and forget about this one. Like I said, my feelings are subject to change.

Iconian at this point is BETA. This means that absolutly no responsibility is taken by the author. This program is provided AS IS. Please, as this program is beta, please only use unimportant data until such time that you are sure of Iconian's capabilities. You must be ready for this program to crash at ANY time, and therefore, do not use while important data is present. This means other programs, as well.

## 1.3 Copying

### Copying

This program is Shareware. You may not sell this program for a profit. You are allowed to make as many copies as you want, but only when no modifications are done. The archive should retain all original contents, in their original form.

Persons wishing to include this program on a "cover disk" or CD-ROM should contact me first. This excludes the AMINET CD-ROM or Fred Fish CD-ROM collections, which have my permission. (These sources are two of the main reasons why the Amiga is still around!)

## 1.4 Introduction

---

## Introduction

Iconian was written over eleven weeks (so far), in response to the demand for a good 3.x icon editor. As everyone should already know, IconEdit 3.0 is not the most capable of programs.

Iconian should do everything IconEdit can, and more. It is able to use all 256 colors, which is a big request.

IFF brushes are also remapped to the current Workbench screen colors. This is done with the new Palette Sharing functions under KickStart 3.x.

All common drawing tools are present, including Bevel boxes.

In designing this program, I looked at 3.x systems as a target. Knowing that not all displays are 640x200, I wanted sizable windows. This posed a problem with the gadgets, but I believe they came out pretty well.

## 1.5 Neat things, and why you should use this program.

### Features of Iconian v1.85

- Support for sizable windows and custom fonts.
- Multiple undos
- Appwindows
- Custom magnification levels
- Upto 256 colors
- 3.x palette sharing
- RGB color editing
- Integer scaling of oversized brushes
- Optional remapping of IFFs to screen colors
- Can save 8 plane icons which stay "normal" regardless of screen depth.
- Full range of drawing tools including beveled boxes
- Recolors 1.3 icons
- Clipboard support
- Extensive keyboard support
- Ability to cut and paste brushes
- Drawing with text, including selectable font and styles
- 100% system supported, no reqtools/explode/iff/powerpacker required
- Untested locale support

(But feel free to use RTPatch, as it's worth it!)

### Planned for the Future...

- Add an AREXX port
- Editing icon attributes, including tooltypes, window size, x/y coords
- More customization via tooltypes, including icon workspace size
- Nicer GUI, more resilient to resizing.
- Better text and error messages all around.

---

Undecided plans, suggestions welcome

More drawing tools?  
Maybe a MUI version?  
More professional presentation?  
Custom write, in ASM, a faster magnification routine?

Future projects under consideration

Screens/appicons program. (Good possibility)  
Better palette program to edit/lock/free pens and colors.  
Multitasking AGA action/platform game. Need artists.  
Euro style AGA GFX demo. Need musicians/artists.  
Animated icons task and editor? (Yea, right)

^Email crandall@garnet.msen.com with suggestions/help/offers.

## 1.6 Requirements

Requirements

Minimum setup requires Kickstart 3.0 and Workbench 3.0.

A better setup would include an AGA machine, 68020.

Recommended setup would include a 68030 and true FAST memory.

No external libraries are needed.

One preference file located in ENV:, named "Iconian.prefs".

At least 350k of free memory is recommended to edit 16 color icons, with 8 levels of undo. Note this requirement gets larger with bigger icons, more undo buffers, or more colors. The amount of RAM consumed can approach two (2) megs of memory or more. (This is with 16 undo buffers and 320x200x256 images.)

A fast processor is recommended. The program was written and tested on a 68030, and seemed more than fast. It might seem a tad slow on refreshes/magnification on anything less.

## 1.7 Installation and Running

Installation and Running

This program requires no external files and may be placed anywhere.

An optional preference file may be located in ENV: or ENVARC:.

---

To run:

Just double-click the icon, or "run" it from CLI.

You may pass one icon as a file; Click once on Iconian, hold the SHIFT key, and double click on the icon you wish to edit.

## 1.8 Tooltypes

### Tooltype descriptions

There are currently 4 supported tooltype parameters.

#### MAXIWIDTH

This sets the horizontal size of the image workspaces. The minimum is 32 and the maximum is 1024.

#### MAXIHEIGHT

This sets the vertical size of the image workspaces. The minimum is 32 and the maximum is 1024.

#### UNDOLEVEL

This specifies how many undo buffers are to be allocated. These are located in FAST memory.

#### DEFTYPE

Specifies which default type of the untitled icons. Not very useful, I suppose.

CLI arguments are supported, also.

#### FILE/A

This is the icon to be edited. (A bug due to my programming requires that this option be present in order to activate the other arguments.)

#### MAXIWIDTH=MW/N

This is the working images' widths. A filename must be present at this point to allow this to work. (The filename can be anything...)

#### MAXIHEIGHT=MH/N

This is the height for the working images. A filename must also be present at this point. (See MAXIWIDTH)

#### UNDOLEVEL=UL/N

Number of undo buffers allocated. A filename is required to allow this to work.

## 1.9 Menu descriptions

---



## Menu Descriptions

Project  
Major file operations and general program control.

Edit  
Undo, clipboard support, and erase.

Images  
Image loading and tools.

Color  
Palette manipulation.

Tools  
Image flipping, resizing, and moving.

Settings  
Icon settings, and preference access.

## 1.10 The Project menu

Project

New

Asks to erase all edited imagery, and default back to "RAM:untitled.info"

Open

Loads an icon image from disk and makes it the current project.

Save

Saves the icon back to disk, there are no backups made.

Save As...

Saves the icon to disk under a name different from the project name. The current project then becomes this file.

Iconize

Closes all windows and displays an AppIcon on the Workbench screen. To restore the program, double click on this icon. (This feature is VERY buggy when screen modes/public screen changes are made, and should be AVOIDED at this point!)

About

Displays some information about the program. The second window contains memory usage stats, for those interested.

Quit

Asks to leave the program.

---

## 1.11 The Edit menu

Edit

Undo

Undos the last graphic operation. Not all operations are undoable. Some actions cancel the undo, such as selecting a new tool. Also any operation that effects both images is undoable, such as recolor.

Redo

Redos the last undo. Undo and Redo are only available for one image. If a change is made to the other image, all undo buffers are lost. (Make sure everything is kosher with the current image before deciding to edit the other image.)

Paste

Opens the clipboard and attempts to load any image found there. The colors are remapped to the public (Workbench) screen. This may be turned off.

Cut

Saves the current image to the clipboard as a brush. The entire image is saved, no trimming is done. The image is then erased.

Copy

Similar to "Cut" except the image is not erased after saving.

Erase

The current image is erased. This is undoable.

## 1.12 The Images menu

Images

Exchange

This swaps the two images.

Copy from other

This copies the unselected image to the current one. This is undoable. Please note that this option works opposite of IconEdit. This is so you can undo the operation.

Load Default

This attempts to load the default imagery for the selected icon type. This may be from ROM, or a user defined image in ENV:SYS.

Load->Image

This will attempt to load a file as a graphic datatype. Please see special notes on this feature, located in  
Loading Images

Load->Normal Image

This loads the normal imagery from an .info file.

---

Load->Selected Image

Similar to Load->Normal, except the selected image is loaded.

Load->Both Images

Similar also to Load->Normal, except both images, normal and selected images, are loaded.

Save as Brush

Saves the image as an IFF file. The entire image is saved. No clipping is done.

Restore

Asks to restore both images to the last time the project was saved.

## 1.13 Color menu

Color

Load Palette

This will load a palette from a normal IFF file. If a "LOCK" hunk is present, pens will be locked and or freed.

Save Palette

This will save the palette to a IFF palette file. A "LOCK" hunk will be created. This hunk will remember which pens were locked at the time the palette was saved. Remember that only locked pens should be used by icons. Otherwise these might be changed by visiting programs.

Show->Free Pens

Show->Locked Pens

Show->Others' Pens

Try these. I'm getting bored...

Recolor

Swaps colors 1 and 2 for both images. In essence, converts 1.3 icons to 2.x+ icons.

Recolor MagicWB

Recolors the highest 4 colors to pen numbers 4-7. Not a useful option, YET.

## 1.14 Tools menu

Tools

Test Icon

Brings up a window displaying the normal icon image. Click into this window to view the selected imagery. Note that "Backfill" type icons will not appear *exactly* like they will on the WorkBench.

---

Text->Enter Text

Allows you to enter a string of text to be used as a brush.

Text->Select Font

Brings up a font requester. You can choose font type, style, and background pen. There are six ways of drawing the fonts:

Text: This draws the text using only the current pen color.

Text and Field: A bar of text is drawn using pen color and field color.

Complement: This is a simple XOR of bits. The pen color does not affect the appearance.

Outline: The field color will surround the pen color.

Shadow: The font will appear to cast a shadow using the field pen.

Bevel Emboss: The pen color will be used as a highlight, the field color as a darkened edge, and the background as the actual text. Similar in appearance to bevel boxes, but available in any color.

Move->Auto Topleft

Moves the selected image as far up and to the left as possible.

Move->Up

Move->Down

Move->Left

Move->Right

Moves the selected image one pixel in the specified direction. CTRL cursor keys are equivalents.

Flip->Horizontal

Flips the selected image left to right.

Flip->Vertical

Flips the selected image up and down.

Scale->Half X

Scale->Half Y

Scale->Half Both

This halves the image either in horizontal, vertical, or both directions.

Scale->Double X

Scale->Double Y

Scale->Double Both

This doubles the image either in horizontal, vertical, or both directions.

## 1.15 Settings menu

Settings

Highlight->Backfill

Highlight->Complement

Selects the complemented/backfill highlighting method. Note that only the normal image is saved, unlike IconEdit 3.x. This may cause

---

problems when loading an Iconian saved complement icon into IconEdit, as only the normal image will be present. Iconian follows the 2.x RKMs, IconEdit 3.x does not.

Highlight->Image

Selects the alternate image as the highlight method. This is the best way (IMHO) to save any icon, and Iconian will usually default to this method.

Type->Disk

Type->Drawer

Type->Tool

Type->Project

Type->Trashcan

Type->Device

Type->Kick

Type->AppIcon

Selects the type of icon that is saved. "Device" and "AppIcon" will probably NEVER be used by ANYONE. They are present only to be complete.

Brush Handle->Center

Brush Handle->Top Left

Brush Handle->Top Right

Brush Handle->Bottom Left

Brush Handle->Bottom Right

Selects where the brush handle will be located.

Preferences

Brings up the preferences window.

Save Prefs

Saves the current preferences into ENV: and ENVARC:.

## 1.16 Gadget descriptions

### Gadget descriptions

Most gadgets are pretty much self explanatory. Those with questionable functions will be described here.

"Switch" a button located in the Palette window.

This is a color swap button. The first color selected, and the second color selected will be remapped to each other. EX. If you selected color 3, pressed "Switch", then selected color 4. All dots drawn in color 3 will be changed to pen 4 and vice versa. The actual RGB values will not be affected, only the current selected image buffer. This is undoable

"blank" This button is a small square. It's a virtual window scroller type device. (Like MUI). Press and hold the button on it. Now move the mouse around to scroll some in any direction. It lags behind, and may take some time to get used to.

"Other" This button switches the current active icon image. This is

equivalent to pressing the SPACE bar.

## 1.17 Palette Sharing on public screens

### Palette Sharing on public screens

OS3.x contains new semi-retargetable graphics routines. The most important one that affects most users right now is Palette Sharing.

Each pen color may be in one of three states:

#### Free

This means that the pen is open to all screen visitors, and may be changed at any time.

#### Locked and public

This states that a pen color is locked, but any program is free to draw using the pen. The RGB values are owned by one program, and may not be changed. Workbench usually has eight (8) of these colors. The pen numbers are 0-3 and  $(2^{\text{depth}}-4)-(2^{\text{depth}}-1)$ .

#### Locked and private

Any pen in this mode is off limits to programs other than the owner. Iconian will avoid these when remapping IFF's, but you are still free to draw with these. Be warned that the RGB values of these might change at any time.

Iconian allows you to select which state a pen is in.

#### Free

Any program may lock, change, or obtain this pen. The state of this pen can change at any minute.

#### Lock

Iconian has placed a lock on the pen. This is not an exclusive, but sharable. Any program may also place a lock on the pen. Although you should not change the RGB values of a locked pen, it is possible. Be aware that if another program has locked a locked pen, and you attempt to free it, the state of the pen may change to olock.

#### OLock

Another program has locked the pen. It may be exclusive or sharable. You are able to use it to draw, but remember that another program may change the RGB values if it is exclusive. You are not allowed change the RGB values of a OLock pen. Although, this limitation may be overridden by the "Ignore Locks" preference option. Only experience persons should do this.

## 1.18 Included drawing tools

### Drawing Tools

---

There are currently nine (16) drawing tools:

Plot (A series of dots)

This will draw a series of unconnected pixels.

Draw (Curved line)

This will draw a series of lines, connected end to end.

Line (Diagonal line)

This will draw a rubberband line while the left mouse button is pressed down. Once release, a permanent line is placed.

Vertical/horizontal (2 straight lines)

Box (Empty, one color box)

This tool will draw a rubberband frame while the left mouse button is pressed down. Once released, a one color rectangle will be placed.

Filled Box (Filled, one color box)

Used like the Box. Once the mouse button is released, a filled box is drawn instead.

Circle (Circle)

Draws a circle.

Disk (Filled circle shape)

Draws a polygon shaped like a circle. (buggy?)

Bevel Box (Raised, beveled box)

This tool is also drawn similar to the box. Except a two color box is drawn using the Bright Edges and Dark Edges colors specified in the Workbench Palette Preferences.

Fill (Arrow pointing at line)

This will fill the color pointed at by the mouse with the current color. (buggy?)

Filled freehand (Blobby shape)

This draws a freehand polygon. (buggy?)

Cut Brush (4 cornered box)

This tool allows you to cut a rectangular square for use as a brush. Once the image is cut, Paste Brush becomes the active tool. Note that no masking is done to the brush, (yet).

Paste Brush (A black rubber stamper)

This will paste a Cut Brush image once the left mouse button is released. Please note that no masking is done, so only rectangular images may be move. This is a quick and crude implementation, subject to change.

If a text string has been entered, the Cut Brush shape will be lost, and this option will draw text instead.

---

## 1.19 Opps.

KNOWN BUGS as of 5:58pm February 20, 1994 (V1.85 beta)

Hmm, yea there are bugs. Otherwise I'd be selling the stupid thing! :)

After being AppIconized, and the screen depth has changed. When Iconian tries to bring the buffers and windows back up, a message will be displayed. Some of the time, after okaying the message, MAJOR problems will arise, even though they don't appear right away. AVOID THIS.

Some memory loss seems apparent when handling some fatal errors. The leaks are quite small, only 70-250 bytes. I'll look into it.

Any fill type drawing that uses the TmpRas seems to be buggy occasionally. This includes DISKS, FILL, and POLYs. Advvoid, if possible, as the results and appearance occasions are unpredictable.

Not really a bug, but LOCALE support is not complete. As the all the strings started changing on me, I could not keep up. Locale will be finished as soon as I finish the rest of the program! ;)

## 1.20 node\_betatesters

Please, as this program is not tested very well by me (I never could "bang" my own programs.), don't use it for anything real important.

Test everything first, and once it seems ok, then you can use it for real.

As this is my first MAJOR program, and my first MAJOR program in 'E', as well, there are bound to be MAJOR bugs. The code is definatly spaghetti.

## 1.21 Keyboard equivalents

Keyboard commands

'b'	Cut brush mode.
'B'	Paste brush mode.
'r'	Box drawing mode.
'R'	Filled Box drawing mode.
'd'	Plot drawing mode.
'f','p'	Fill drawing mode.
'l','v'	Line drawing mode.
'u','ESC'	Undo
'1','F1'	Selects normal image
'2','F2'	Selects selected image
'SPACE'	Toggles normal image/selected image
','	Decreases magnification
.'	Increases magnification

---



```
'F6'      Selects 1x magnification
'F7'      Selects 2x magnification
'F8'      Selects 4x magnification
'F9'      Selects 8x magnification
'F10'     Selects 16x magnification
 '['      Decreases current pen number
 ']'      Increases current pen number
UP        Scrolls the view area up
DOWN      Scrolls the view area down
LEFT      Scrolls the view area left
RIGHT     Scrolls the view area right
CTRL-UP   Moves the image one pixel up
CTRL-DOWN "
CTRL-LEFT "
CTRL-RIGHT "
```

Note that some keys aren't listed.

## 1.22 Loading of datatypes

This operation will call the `datatype.library` to load a picture class. There are several ways to control how an image is loaded.

No remap

The brush will be loaded straight into the image buffer. If the image uses pen 1, then a dot will be drawn with pen 1. This is not recommended, as datatypes contain their own color maps, which 95% of the time are totally different than the screen's.

Remap with dithering at zero, using "Clip" or "Quick Scale".

Iconian will call `FindColor()` to find a free or sharable pen matching closely to that of the datatype image.

Remap with dithering at zero, using "Scale"

A group of pixels will be quantized to find an average RGB value. This value will be passed to `FindColor()` to find a suitable match. This is a bit slower than normal remapping.

AVOID THE DITHERING FOR NOW. (It's semi-functional, but will change. The current form was hacked one night and is VERY slow.)

\*Author's note: Dithering should be disabled at this point, but not removed.\*

---