

ColorSaver

Dan Fish

Copyright © CopyrightÂ©1994 Dan Fish

COLLABORATORS

	TITLE : ColorSaver		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Dan Fish	December 7, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ColorSaver	1
1.1	Colorsaver Info	1
1.2	ColorSaver Features	1
1.3	Disclaimer	2
1.4	Installing ColorSaver	2
1.5	distribution	2
1.6	ColorSaver ToolTypes	2
1.7	gadgets	3
1.8	Loading colormaps...	3
1.9	Saving colormaps...	4
1.10	Patching Binaries...	4
1.11	Scanning a binary for a color palette	4
1.12	Selecting a file offset...	4
1.13	Copying palette colors...	4
1.14	Swapping palette colors...	5
1.15	Spreading palette colors...	5
1.16	Shifting the entire palette left...	5
1.17	Shifting the entire palette right...	5
1.18	Complementing the currently selected color...	5
1.19	Changing palette slider functions...	5
1.20	Display program info...	5
1.21	Cancelling changes...	5
1.22	Undo'ing changes...	6
1.23	Accepting changes...	6
1.24	Implementing changes...	6
1.25	ARexx Commands	6
1.26	Patching Binaries	7
1.27	Well.... SOMETIMES it works!	8
1.28	ColorSaver History...	8
1.29	acknowledgements	9
1.30	quicktest	10
1.31	Notes (and other Bugs...;(.	10
1.32	Future plans...	11

Chapter 1

ColorSaver

1.1 Colorsaver Info

+ ColorSaver +

ColorSaver is a general purpose color palette tool designed to be available at any time through the use of a hotkey. It contains Features not generally found in other palette tools

Disclaimer	Distribution	Acknowledgements
Installing	Quick Test	Tool-Types
Gadgets	ARexx Commands	Patching Binaries
Author	History	Bugs
Future		

1.2 ColorSaver Features

ColorSaver has the following features:

- * Implemented as a Commodity, Popup w/Hotkey
- * Load/Save color palettes
- * Sliders selectable between RGB/HSV
- * Copy, Swap, Range, Complement Functions
- * Left/right shifting of the entire palette
- * Ability to permanently alter (patch) the color tables of executables with a statically allocated color table.
- * "Remembers" load/save file path's through the life of the program
- * "Remembers" screen position it was last popped up to
- * Runs from CLI or WorkBench
- * ARexx Port with 15 commands
- * PostCardWare, Includes source
- * Tested with Enforcer

See Also: Mis-Features ToDo

1.3 Disclaimer

```

~~~~~
+                               WARNING!!!!                               +
+
+ *ANY* Program that directly patches executables is inherently dangerous!!
+ (especially those that do so without making a temporary backup until the
+ write completes, such as this one!!). It is STRONGLY advised that when
+ using the "WRITE" function, you do so ONLY on a backup (better yet, a backup
+ of your backup!). The end-user accepts any and all liability for results
+ obtained through the use or misuse of this program, be it direct, indirect,
+ consequential or incidental.
+
~~~~~

```

1.4 Installing ColorSaver

Not much to it... If you always want ColorSaver available, drag its icon into the SYS:WBStartup drawer (Might want to set the CX_POPUP=NO tooltype) Also, if you want to save all your color palettes in the same location, then do one of the following:

1. Set the "PATH=" ToolType (I.E. PATH=SYS:Prefs/Colormaps)
2. Set an environment variable for CS_PALETTES
(I.E. SetEnv CS_PALETTES SYS:Prefs/Colormaps)

(Otherwise ColorSaver will default to "SYS:Prefs" for load/save operations)

1.5 distribution

Every programmer likes to know that Somebody at least Somewhere uses their program... (plus they're also curious about how far from home the thing has gotten). This program is PostCardWare.

If you use it, all I ask is that you let me know, by postcard or by Email

1.6 ColorSaver ToolTypes

ColorSaver has the following command-line/tooltype template:

```
ColorSaver [CX_PRIORITY] [CX_POPUP] [CX_POPKEY] [CX_QUIT] [SELECT]
           [PATH] [POP_X] [POP_Y] [HSV]
```

```
CX_PRIORITY/K/N, CX_POPUP/K, CX_POPKEY/K, CX_QUIT/K, SELECT/K/N, PATH/K,
POP_X/K/N, POP_Y/K/N, HSV/S
```

```
CX_PRIORITY Sets ColorSaver's priority in the commodities queue.
```

Default : 0

CX_POPUP When you specify "CX_POPUP=NO" window will not be opened when you initially start ColorSaver
Default : YES

CX_POPKEY Specifies the "HotKey" to Popup the ColorSaver window.
Default : lalt ralt c

CX_QUIT Specifies the "HotKey" to terminate ColorSaver
Default : lalt ralt q

SELECT Specifies the active Palette color when ColorSaver is started
Default : 0

PATH Specify the path for load/save operations
Default : SYS:Prefs/Presets

POP_X Specify the popup X position
Default : 175

POP_Y Specify the popup Y position
Default : 50

HSV If specified, ColorSaver will open with sliders set to HSV Mode.
Default : Sliders set to "RGB" mode

1.7 gadgets

Gadget Descriptions:

~~~~~

Most of ColorSaver should be fairly Intuitive (no pun intended!) but for the sake of completeness, here's a description of the gadgetry.

|             |            |         |
|-------------|------------|---------|
| Load        | Save       | Write   |
| Scan Binary | Cycle      | Copy    |
| Swap        | Range      | «««·    |
| ·»»         | Complement | RGB/HSV |
| ©           | Okay       | Do      |
| UnDO        | Cancel     |         |

## 1.8 Loading colormaps...

Load    Load a saved colormap. Load path/file is retained through the life of the program. Load directory initially defaults to the "PATH=" tooltype (if specified) or the environment variable "CS\_PALETTES" (if it exists) or "SYS:Prefs" if the previous two are undefined. If the number of colors in the file do not match the depth of the current screen, you will be notified via requester and given a chance to "Cancel" or

"Load Anyway".

Keyboard ShortCut: Right-Amiga-L

## 1.9 Saving colormaps...

**Save** Save the current screen colors to a file for subsequent loading. Save path/file is the same as that specified for LOAD. Always saves number of colors associated with depth of the current screen.

Keyboard ShortCut: Right-Amiga-S

## 1.10 Patching Binaries...

**Write** Patch a binary file. This gadget is disabled until a valid file offset (or offsets) is found using the Scan Binary function. A requester will appear to confirm the file name and offset prior to the actual write. If this is performed successfully, the color palette for the executable will be permanently modified. See Patching Binaries

## 1.11 Scanning a binary for a color palette

**Scan Binary** Scan a file for a sequence of bytes corresponding to the values of the Reset Palette. The Reset Palette is that which exists at the time ColorSaver is first popped up, or after selecting DO . See Patching Binaries

## 1.12 Selecting a file offset...

**Cycle Gadget** Displays valid file offsets. This gadget is disabled until a valid file offset (or offsets) is found using the Scan Binary function. In the event more than one match is found, multiple file offsets will be listed. The selected offset will be that used for a WRITE operation. See Patching Binaries

## 1.13 Copying palette colors...

**Copy** Copy the currently selected color to the color next selected on the color palette.

---



## 1.14 Swapping palette colors...

`Swap` Swaps the currently selected color with the color next selected on the color palette.

## 1.15 Spreading palette colors...

`Range` Spread the colors between the currently selected color and the color next selected on the color palette. A least one intervening color must exist between the two colors or this selection will have no effect.

## 1.16 Shifting the entire palette left...

`««·` Shift the entire color palette left. (I.E. Color 1 becomes color 0, color 2 becomes color 1, ...etc)

## 1.17 Shifting the entire palette right...

`·»»` Shift the entire color palette right. (I.E. Color 1 becomes color 2, color 0 becomes color 1, ...etc)

## 1.18 Complementing the currently selected color...

`Complement` Complements the currently selected color.

## 1.19 Changing palette slider functions...

`RGB/HSV` Change slider functions between RGB and HSV. This will be "remembered" between subsequent ColorSaver "popup's".

## 1.20 Display program info...

`©` Displays some info about the program.

## 1.21 Cancelling changes...

`Cancel` Cancels all color changes and restores palette in effect when ColorSaver was popped up. Closes the ColorSaver window.

Keyboard ShortCut: Right-Amiga-C

---

## 1.22 Undo'ing changes...

UnDO Exactly the same as CANCEL except window is left open for "further diddling".

Keyboard ShortCut: Right-Amiga-U

## 1.23 Accepting changes...

Okay Accepts the current color settings and closes the ColorSaver window.

Keyboard ShortCut: Right-Amiga-O

## 1.24 Implementing changes...

Do Exactly the same as OKAY except ColorSaver window is left open. Used mainly to force the current palette into the reset palette without having to close and re-open the window.

## 1.25 ARexx Commands

ARexx Commands:

~~~~~

The ARexx command set is still minimal, error checking is near nil, both should improve with a future release. Some example scripts have been supplied that show most of the trickier functions. If you come up with some really innovative ARexx uses of ColorSaver, I'd be interested in hearing about them and will ↵ include them in the next release (Along with credits of course;).

Note: All screen references refer to IntuitionBase->FirstScreen. At present, there is no way to specify the screen name.

SHOW -- Popup the ColorSaver window
HIDE -- Hides the ColorSaver window
CANCEL -- Cancels any color changes since the window was popped up or since a STOREPALETTE operation.
QUIT -- Kills ColorSaver

LOAD <Filename> -- Loads a palette. If "Filename" does not contain a path, the path will default to that specified in an environment variable or tooltype. See the Load gadget

Note: Unlike with the "Load" gadget, the ARexx version of LOAD will give you no notification if the colors in the file do not match the screen depth. It will simply do a "Load Anyway".

```

SAVE <Filename> -- Saves the screen palette. Always saves the number
                  of colors associated with the depth of the screen. If "Filename ←
                  "
                  does not contain a path, the path will default to that specified
                  in an environment variable or tooltype. See the Save gadget

STOREPALETTE -- Saves the current screen palette which is restored with
                a CANCEL operation.

SHIFTL -- Shift the entire color palette left. (I.E. Color 1 becomes
         color 0, color 2 becomes color 1, ...etc) See the ««· gadget
SHIFTR -- Shift the entire color palette left. (I.E. Color 1 becomes
         color 2, color 2 becomes color 3, ...etc) See the ·»» gadget

RGBMODE -- Set sliders to RGB mode. See RGB/HSV
HSVMODE -- Set sliders to HSV mode. See RGB/HSV
          Note: at present there is no way to actually set specific HSV
          values from ARexx. All values must be specified in
          RGB using the SETRGB command. If HSVMODE is selected
          however, and the ColorSaver window is open, the HSV
          sliders will be set to the correct values.

NUMCOLORS -- Returns in the ARexx RESULT variable, the number of colors
            (corresponding to the depth of) the screen.

SELECT <#> -- Selects color specified by <#> as the active color. Where:
            <#> = 0 to "NUMCOLORS-1"

GETRGB -- Returns in the ARexx RESULT variable, 3 numbers specifying
         the current R G B Values of the active color for the screen

SETRGB [C=#] [R=#] [G=#] [B=#] -- Set the RGB values for a specific color
                                register, where:

        C = (Color) 0 to "NUMCOLORS-1"
        R = (Red)   0 to 15
        G = (Green) 0 to 15
        B = (Blue)  0 to 15

Parameters not specified will retain their current values.

```

1.26 Patching Binaries

Ever have this happen to you? You start this brand new program and say: "Yeah, this is nice, but the person's choice of colors is really awful!" Everybody has different tastes in colors, so what do you do? Well, if you're lucky, the programmer was thoughtful enough to put in a palette for you to change them with, and if you're *REAL* lucky he was even kind enough to allow you to save them somewhere as the default! (I must at admit, many programs are at least getting much better about including a palette tool.. :-)

So... no palette tool, Huh? Well, that's where ColorSaver can sometimes give you a hand... Just run your program, popup ColorSaver, Click on the "Scan Binary" gadget and select the filename for your program. ColorSaver will attempt to locate the "palette" inside the binary. If it finds one (or more)

it will let you know, and will activate the "Write" and "Offset" gadgets.

Now all you need to do is change the colors to your liking and click on "Write". Poof! For ever-after when you run that program, it will have the colors you selected.

As proof that this really does work, I've included a small "test" program called "ColorTest". Simply run ColorTest, popup ColorSaver, use it to scan ColorTest and it should come back with (1) color table match at offset 4196. Change the colors to your liking, "Write" them, select Cancel or Okay from ColorSaver, (Doesn't matter which, you've already written the colors) and then exit ColorTest. Now when you run ColorTest again, it will have the colors you selected.

Magic, Huh? Naw, not really. I've been doing the same thing for years with a hex editor, ColorSaver just makes it much easier. Note that I said earlier that "ColorSaver can sometimes help..."

1.27 Well.... SOMETIMES it works!

And now the bad news... this won't work with all programs...

Generally, the table must be statically allocated in the source. I.E. something like:

```
UWORD colortable[] = { 0x000, 0xf44, 0x4f4, 0x44f };
```

When you run your program and pop up ColorSaver, it gets the current palette and when you do a "Scan Binary" it looks for that sequence of bytes in the file and will replace them with the colors you choose.

If a program uses individual calls to SetRGB4(), dynamic allocation, does color cycling or other strange things, this won't work. I'd estimate I've been able to successfully alter the palettes of somewhere between 1/3 and 1/2 the binaries I've tried it on.

It is possible that there could be more than one sequence of matching bytes (particularly with 1 and 2 bitplane programs. If this is the case, then (for now anyway) you'll have to "roll the dice" and write to one of the locations. BE ABSOLUTELY, POSITIVELY SURE YOU DO THIS ONLY ON A BACKUP OF A BACKUP OF A BACKUP OF THE PROGRAM!!! (hope you read the Disclaimer) If you then run your program and get a free trip to India, you'll know you selected the wrong one! (Perserverance is the key here!).

Personally I've only run across this a couple of times, and I think they were both 1 bitplane black & white screens.

Note that you are not really limited to scanning and writing to executables, you can also scan binary data and/or configuration files. (or even the color palettes saved by ColorSaver!)

1.28 ColorSaver History...

- 10-Dec-94 Version 1.19
- made \$VER: string style-guide compliant
(noted by Klaus Alexander Seistrup)
 - Minor code changes not affecting operation
- 31-Nov-94 Version 1.18
- ARexx improvements
 - Added custom pointers
 - Added Hide/Quit Requester upon clicking the close gadget
- 25-Aug-94 Version 1.10
- Code getting rather large for one file, split into several smaller files
 - Converted/updated documentation to AmigaGuide format
- 18-Feb-94 Version 1.02
- Minor code changes not affecting operation
- 19-Sep-93 Version 1.0
- (Very few bug reports/suggestions, so found my own...)
(Does anybody but me ACTUALLY use this thing??)
- Fixed irritation that caused "Complement" not to work under OS3.0
 - Cleaned up the source a bit.
 - Added the following tooltypes:
 - SELECT -- Specify the active color on startup
 - PATH -- Specify the load/save file path
 - POP_X -- Specify the popup X position
 - POP_Y -- Specify the popup Y position
 - Added rudimentary ARexx port
- 06-Jun-93 Version 0.86
- Bug Fixed, Called UnlockPubScreen() even though I never called LockPubScreen()
 - Bug Fixed, If "Load Anyway" selected when loading colors of < depth of frontscreen, will now only change the actual number of colors loaded.
- 05-Jan-93 Version 0.84
- Alpha Release (First Public Release)

1.29 acknowledgements

Acknowledgements:

~~~~~

- \* Many thanks to Jan van den Baard for his excellent "GadToolsBox" which makes the task of writing programs such as this so much easier, and for providing the well documented, structured source for JBSpool (FF656). (Most of the commodity stuff for ColorSaver was yanked out of JBSpool, and I hope he doesn't mind me including his link library "dcbacks.lib", though I could find no limitations to distribution in his documentation).

- \* Also many thanks to Matt Dillon and the folks at Obvious Implementations Corp. for the excellent DICE environment.
- \* And of course, thanks to Fred, his library, and all the hundreds of talented people who have submitted very high quality programs, without which the Amiga might very well still be just "An Interesting Curiosity" ....as was the Atari ST!

## 1.30 quicktest

For those of you who hate long-winded documentation, (don't we all??) here's some quick exercises to show off ColorSaver. (Note: Use LAlt-RAlt-C to "popup" ColorSaver)

1. Double-Click on ColorSaver, select "Load", you should have a choice of several palettes, Try loading a couple... You can also fiddle with the colors, then save your own palette. (Probably have to save it somewhere else, or write-enable the disk!)
2. When you get tired of that, you can try out ColorSaver's binary file patching ability.
  - a.) Double-Click on "ColorTest" and note the colors (black & white with a single red bar)
  - b.) Popup ColorSaver and adjust the colors until you like them.
  - c.) Select "Scan-Binary"
  - d.) Select "ColorTest" from the file requester.
  - e.) ColorSaver should respond that it found one Color Table match (Cycle gadget should display offset of 4196)
  - f.) Select "Write" and confirm the requester (probably have to write-enable the disk if you haven't copied all the files somewhere)
  - g.) Hide ColorSaver (select "Cancel")
  - h.) Exit ColorTest

Now run ColorTest again and note that the colors have been permanently and forever changed to the ones you selected. (Or at least until you run ColorSaver again!)

3. Try out the various supplied ARexx scripts.

## 1.31 Notes (and other Bugs...;(

The following are known ...um, er, Peculiarities!

- \* Does not properly handle HAM palettes
- \* Loading "non-palette" files should not be fatal, but will result in "bizarre" color schemes (that will likely include the pointer)
- \* Does not use LockPubScreen(), Had problems with this, instead ColorSaver pops up on IntuitionBase->FirstScreen, thus you should avoid closing a screen that you have ColorSaver popped up on.

(Close ColorSaver window FIRST!)

- \* Popping up the palette on some screens with unusual color settings may render ColorSaver nearly (or totally) invisible.
- \* If something other than ColorSaver is responsible for changing the colors of a screen, (I.E. another palette tool, system calls, etc.), the sliders will become "out of sync" with the current screen colors. Clicking on a slider will cause that color to "jump back" to what ColorSaver thinks it should be. You can avoid this effect by making a palette selection. This will cause ColorSaver to reset the sliders to the CURRENT screen colors. (This effect is probably normal for all palette tools).

## 1.32 Future plans...

ColorSaver now does pretty much what I started out intending it to do... Improvements depend largely on the requests I get... (I.E. If I'm the only one to use this thing, why bother?)

Some possible enhancements:

- \* Font Sensitivity (suggested by Klaus Alexander Seistrup)
  - \* Control of pointer/sprite colors (suggested by Tim Shaunessey)
  - \* Localization of the Gadgetry
  - \* Enhance ARexx command set, better error checking
  - \* Screen selection by name
  - \* Recognize/Load/Save SYS:Prefs/Presets type palettes
  - \* Better handling of palettes with a large number of colors, (I.E. With 128-256 colors, you currently need a magnifying glass to select the proper color...;)
-