

AGAiff

COLLABORATORS

	<i>TITLE :</i> AGAiiff		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 31, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	AGAff	1
1.1	AGAff Docs	1
1.2	AGAff Introduction	2
1.3	AGAff Installation	3
1.4	AGAff Using AGAff	3
1.5	AGAff Main Window	4
1.6	AGAff Main Window - Save Format Gadgets	5
1.7	AGAff Main Window - Color Format Gadgets	7
1.8	AGAff Picture Screen	8
1.9	AGAff Animation Support	9
1.10	AGAff Preferences	9
1.11	AGAff - ARexx Port	10
1.12	AGAff - ARexx Commands	12
1.13	AGAff - ARexx Commands	13
1.14	AGAff - ARexx Commands	13
1.15	AGAff - ARexx Commands	13
1.16	AGAff - ARexx Commands	14
1.17	AGAff - ARexx Commands	14
1.18	AGAff - ARexx Commands	14
1.19	AGAff - ARexx Commands	15
1.20	AGAff - ARexx Commands	15
1.21	AGAff - ARexx Commands	15
1.22	AGAff - ARexx Commands	16
1.23	AGAff - ARexx Commands	16
1.24	AGAff - ARexx Commands	16
1.25	AGAff - ARexx Commands	17
1.26	AGAff - ARexx Commands	17
1.27	AGAff - ARexx Commands	18
1.28	AGAff - ARexx Commands	18
1.29	AGAff - ARexx Commands	18

1.30	AGAiiff - ARexx Commands	19
1.31	AGAiiff - ARexx Commands	19
1.32	AGAiiff - ARexx Commands	20
1.33	AGAiiff - ARexx Commands	20
1.34	AGAiiff - ARexx Commands	21
1.35	AGAiiff - ARexx Commands	21
1.36	AGAiiff - ARexx Commands	22
1.37	AGAiiff - ARexx Commands	22
1.38	AGAiiff - ARexx Commands	23
1.39	AGAiiff - ARexx Commands	23
1.40	AGAiiff - ARexx Commands	24
1.41	AGAiiff - ARexx Commands	24
1.42	AGAiiff - ARexx Commands	24
1.43	AGAiiff - ARexx Commands	25
1.44	AGAiiff - ARexx Commands	25
1.45	AGAiiff - ARexx Script 'Startup.agaiff'	26
1.46	AGAiiff - ARexx Script 'AutoScan.agaiff'	26
1.47	AGAiiff - ARexx Script 'GridSave.agaiff'	26
1.48	AGAiiff - ARexx Script 'SplitAnim.agaiff'	27
1.49	rxs_info	27
1.50	AGAiiff - Running ARexx Scripts	27
1.51	AGAiiff Development History	27
1.52	AGAiiff What's left ?	30

Chapter 1

AGAiff

1.1 AGAiff Docs

```

++*****
++
++          ++
++   A G A i f f Version 2.8           ++
++       by Michael Krause           ++
++       (Raw Style/Lego)             ++
++   © 1993-1997           ++
++
++           What's new, then?
++
++
++   Join the Mekka & Symposium 1998 Easter party in Germany!!!   ++
++   http://ms.demo.org/                                           ++
++
++           ++
++           This Production is FreeWare!           ++
++           ++
++*****

```

This is the low-level image converter used by famous
demo scene guys such as:

Azure/Artwork
Tron/Artwork
Crash/Polka Brothers
Copper/DSW ;)

After half a year of pessimism and depression concerning this program, I
have finally restarted working on it!

01. Introduction
What is this AGAiff?
02. Installation

- What do I need?
- 03. Using AGAiiff
 How to start
 - 04. Main Window
 GUI Features
 - 05. Picture Screen
 GUI Features
 - 06. Animation Support
 Animation Handling
 - 07. Preferences
 What can I change?
 - 08. ARexx Port
 How to add own functions
 - 09. History
 How AGAiiff developed
 - 10. What's left ?
 Address, Copyright etc.

Reference

Save Modes

Color Modes

Note to the very experienced ones...:

Don't try to find out all functions of this program on your own, because there are too much of them being badly named.

At least skim this doc through to get an impression of what it is able to do.

1.2 AGAiiff Introduction

AGAiiff is the type of program which (in the beginnings of the Amiga demo scene) was called "IFF-Converter". These programs could load IFF (the standard Amiga graphics file format) files and save them as unpacked raw data which is necessary for demo and game coders. There are lots of "IFF-Converters" out there but AGAiiff is probably the most advanced program of this type (because of animation and the unique ARexx support and many output formats) and it was one of the first ones to support the AGA graphics chipset which was introduced in 1992 (therefore it's name). The only competitor in business worth thinking of is Morten Eriksen's PicCon,

which has even more save formats, but no ARexx port.

These are some of AGAiiff's features:

- Running entirely under OS
- Coded as a Commodity
- Coded completely in fast Assembler, thus divinely short.
- Font-sensitive User Interface
- <MANY> different save formats:
the ones worth mentioning are e.g. all aga sprite types, some chunky save modes, converting of HAM pictures into pure color data etc., and -- 9 -- different color save formats.
- Can convert big pictures from fast memory
- Animation support, loading frames directly from disk and saving memory!
- Powerful ARexx-Port, supporting all stuff from the GUI and more
- AutoScanning and GridSaving of Pictures (see ARexx chapter)
- Can load pictures via datatypes.library
- ... and more !

1.3 AGAiiff Installation

AGAiiff requires:

- MC68020
- ECS (big blitter), if you want to save blitter masks.
- Kickstart 3.0 (Version 39)
- LIBS:asl.library
- Some display data base entries (at least one monitor activated)

Making life much easier:

- LIBS:commodities.library
- LIBS:rexxsyslib.library
- REXXMaster running
- Provided ARexx-Scripts in REXX:
- HardDisk

Note:

To enable AA-ChipSet don't forget 'SetPatch' in your startup-sequence.
Nevertheless, AGAiiff runs even on the old ChipSet !

1.4 AGAiiff Using AGAiiff

AGAiiff can be started from both Workbench or CLI. Only one copy of AGAiiff can be run.

Workbench Tooltypes

CX_POPUP Standard commodity tooltype.

If ARexx is active, the CX_POPUP specification has no direct meaning. The opening of the window must be done in the Startup script via 'SHOWGUI'. This is because otherwise you would see on the GUI how the presets are built etc., which would slow down the initialization. If this flag was set to TRUE, STATUS 7 returns a 1.

CX_POPKEY Standard commodity tooltype.

PUBSCREEN =name. Open AGAiiff windows on public screen called 'name'.
If this screen doesn't exist, AGAiiff falls back to 'Workbench'.

Shell Template

```
-----
CX_POPKEY/K
CX_POPUP/T
PUBSCREEN/K e.g. PUBSCREEN CygnusEdScreen1
```

Preferences must be put into the ARexx startup script!

1.5 AGAiiff Main Window

Close Window

Hides the GUI. Use hotkey to get it back.

Screen Mode Selection

The listview gadget shows all available monitors and displaymodes. If the program can't find any entries for this list - that's the case when Workbench is not running - a default list is used.

Selecting 'RESHOW' when a picture is in memory forces AGAiiff to redisplay it with the new display mode. 'RESTORE' restores the old one.

Save Range Selection

If you don't want to save the whole picture, you can select a range here. After loading a picture, the range is set to include the whole picture. This is ofcourse not the case if range locking is activated. (see <Preferences>)

Save Format Settings

This button opens the save format presets window. The entries are created in the 'Startup.agaiiff' ARexx Script.

Save Format

See

Save Format

By the way, better have Cycle2Menu installed if you want your left mousebutton to live for some more years...

Colors Format

The format of the color data which is appended to the picture file. See

Colors Format

Frame

This field contains information about the currently loaded picture. "ilbm0" is uncompressed IFF data, "ilbm1" means byte-run-1-compressed IFF data, "dlta5" is an ANIM-5-DLTA-crunched frame of an animation. It can also contain the base name of a datatypes picture subclass.

Next

This button only works when AGAiiff is in animation mode, that means you have loaded a valid animation. Then, by pressing this button or the right cursor key, you can load the next frame.

Exit

Quits AGAiiff. Everything will be cleared up, the commodity will be removed.

Use the right mousebutton or the 'Return' key to switch between main window and picture screen.

The 'Load' function supports IFF-ILBM, datatypes and IFF-ANIM-5 loading. If there is not enough chip memory available, the loader will automatically fall back to a fast mem buffer, but then AGAiiff can't display the picture on its screen. The internal ILBM loader ignores mask planes.

1.6 AGAiiff Main Window - Save Format Gadgets

Upper gadget:

IFF-ILBM

Hmm, saves IFF-ILBM data I think..
 Saving doesn't depend on color format gadgets.
 The BODY chunk is crunched if the picture is big enough.

RAW Standard

Saves raw, uncrunched data, one bitplane after the previous one, beginning with the lowest-order plane.

RAW Interleaved

('RAW-BLIT', Modulo)

The planes are interleaved, that means e.g. when you have a 3 planes picture:

- 1. Line / 1. Plane
- 1. Line / 2. Plane
- 1. Line / 3. Plane
- 2. Line / 1. Plane
- 2. Line / 2. Plane
- etc.

[no picture data]

Saves no bitplane data and enables you to save pure color data.

Copperlist

Saves a copperlist according to the lower color gadget (12Bit or 24Bit), initializing the color values.

Note: No color data else is attached - the "top/bottom" gadget has no meaning.

Sprite

Saves sprite data which is the same as RAW Interleaved data, but with Control data at the top and at the bottom. The control data's length corresponds to the width of one line. Sprites must have a width of 16, 32 or 64 pixels and a depth of 2 or 4 planes ! 4-planed Sprites must be displayed using the attaching of two Sprite channels, so it is divided into two sprites with each 2 planes. The sprite with the lower two planes is saved first.

Mask

Saves mask data, where one pixel represents a pixel with a color different than 0 in the original picture. Saves the same plane copied for each other plane in RAW Interleaved format. Note: No color data else is attached (Top/Bottom gadget) !

This mode requires an ECS blitter and doesn't work when the picture resides in FastMem.

Mask (1 plane)

Same as "Mask", but saves only 1 plane.

This mode requires an ECS blitter and doesn't work when the picture resides in FastMem.

Unpacked Chunky

Instead of saving bitplanes, this mode looks at the pixels and saves the number of the color the pixel has as a byte. You can append color data using the color format gadgets. The selected range will not be rounded up to 16-pixel boundaries.

Chunky Colors

It's quite similar to the previous mode, but instead of saving the number of the color register, it inserts the real color value (\$f00, \$2cb399 etc). You can choose the format of the color values using the lower color format gadget. No extra color data will be appended. The selected range will not be rounded up to 16-pixel boundaries.

SPECIAL FEATURE: This mode also converts HAM and EHB pictures into the right colors !!

Lower gadget:

File

Saves data directly into a file

dc.w

Saves an ASCII Assembler dump using the dc.w directive. In the preferences you can specify what data width to use (dc.b, dc.w or dc.l)!

dc (PC)

Saves a PC Assembler dump using the db directive. The data will always be saved as bytes.

1.7 AGAiiff Main Window - Color Format Gadgets

Upper gadget:

No Colors

Don't append any color data.

Top

Attach color data at the top of the file.

Bottom

Append color data.

Lower gadget:

12 Bit

Only 12 Bit word data, good old A500 style (\$000,\$f00,\$aaa etc.)

24 Bit ORGB

24 Bit data, one longword per color, e.g.

\$00ff0000 for red

\$00ff8800 for brown etc.

This is the mode which was called "24 Bit" prior to V2.10.

24 Bit RGB0

\$rrggb00

24 Bit RGB

3 Bytes per color: red, green and blue.

24 Bit HL

24 Bit data, saves higher and lower nibble of the RGB values separately (as words); first all higher parts of all colors, then all lower parts of them.

24 Bit HLi

Works like the one before, but 'interleaves' the High / Low data of each color, that means: higher part of first color, lower part of first color, higher part of second color, lower part of second color etc.

24 Bit LH

Works like 24 Bit HL, but first saves lower part, then the higher one.

24 Bit LHi

Works like 24 Bit HLi, but first saves lower part, then the higher one.

96 Bit

Internally, AGAiff handles all colors in 96Bit like OS3.0, perhaps this mode is a bit useful in future. Save format is 32r,32g,32b.

1.8 AGAiff Picture Screen

Once you've loaded a picture, AGAiff tries to open a screen of \leftrightarrow the desired resolution and depth. The resolution is changed to a default resolution if necessary. When this Screen is active, you can switch between there and the

Main Window
using the right mousebutton.

Using the left mouse button, you can select a range the save functions will operate on. Pressing it a single time, i.e. selecting a 1x1 range, resets the range to the whole picture.

Pressing SPACE opens a small color editor which should be self-explaining...

If you are using AGAiff with CyberGFX, you might discover that sometimes the colors are not set correctly on the picture screen. This is because CyberGFX can't share colors of the picture screen and the coordinates screen. Switch off the latter one if it disturbs you.

1.9 AGAiff Animation Support

Just click on the 'Load' button. AGAiff will recognize an IFF-ANIM file and automatically load it.

The first thing that happens while loading is that AGAiff scans the whole file, trying to figure out how many frames it has. Then the first picture is initialized and displayed on the normal picture screen. You can treat the frame as if it was a single picture -- no reduction of editing possibilities!

To jump to the next frame, press 'Next' in the main window or use the right cursor key in either the main window or the picture screen.

The animation loader completely supports the ANIM-5 format, even with multiple CMAP chunks.

1.10 AGAiff Preferences

dc.X Size

Specifies the kind of data files written as 'dc.w'--Data will have (dc.b / dc.w / dc.l). Default is long.

PC assembler dumps are always written in bytes.

Use Topaz/8

When this switch is on, the program uses topaz/8 instead of the standard screen font. Default is off.

Lock Range

Locking the save range means that the current range is stored and automatically activated for the next picture that is loaded. Default is off.

ScreenPromotion

When this switch is activated, the current monitor ID selected in ListView-Gadget of the Main Window is stored and automatically used for every new picture that is loaded. Default is off.

Picture NoShow

If you don't want AGAiff to show all pictures you load, then select this switch. Default is off.

Note to ARexx programmers:

This has not the same effect as setting the NOSHOW flag when LOADING files.

The NOSHOW flag is only valid for the one loading action. There is a special ARexx-Command for the 'global' NoShow.

Range W/H

This switch changes the range display in the main window in that the range is not chosen by setting two points (upper left and lower right one), but merely by setting one point and the width and height of the range from there. That simplifies e.g. selecting of ranges sized 128x128 from a larger picture. Just click on a specific point in the picture and then type 128 in the last two range gadgets. Default is two-points-range-selection (switch off).

Activate Window

On every load and save the window is brought to front, being activated. This can be disturbing when operating on huge amounts of pictures via ARexx. Deactivate this switch to get your environment a bit more comfortable.. Default is on.

Save=ClosePic

While saving, the picture screen can be closed in order to save memory. This is rather disturbing when operating on huge amounts of pictures via ARexx. Activate this switch if you really want the picture to be closed. Default is off = no closing.

Show Coords

This switch is on by default and gives you a line attached to the picture screen containing the current mouse coordinates, range size and frame number. When using CyberGFX, you can disable this function and the screen colors will always be correct.

Use Activates the new selections.

Close-Gadget Cancels the new selections.

If you want to make the changes resident, you have to edit the ARexx startup script!

1.11 AGAiiff - ARexx Port

This is AGAiiff's big advantage to all the other low-level image ↔ converters.

Enjoy it!

Please note that arguments containing spaces must be quoted using the " character. This is new for 1.99beta, but was not mentioned in the docs, which confused some users. See the Startup.agaiff script for an example.

Control Functions / Preferences

USETOPAZ

DCSIZE

GLOBALNOSHOW

SCREENPROMO

LOCKRANGE

VERSION

STATUS

ACTIVEWIN

SAVECLOSE

COORDS

User Communication Functions

REQUESTFILE

REQUEST

TYPETEXT

Graphical User Interface Control Functions

LOCKGUI

UNLOCKGUI

SHOWGUI

HIDEGUI

SHOWPIC

Preset List Control Functions

CLEARPRESETS

ADDPRESET

SETPRESET

I/O Functions

SFORMAT

CFORMAT

LOAD

SAVE

RESHOW

NOSHOW

RESTORE
Picture Functions

DISPMODE

RANGE

GETPIXEL

SEARCHPIXEL
Animation Functions

ANEXT
ARexx Scripts provided with this package

Startup

AutoScan

GridSave

SplitAnim

Info

How To Run Scripts

Note:

Parameters in <>'s are decimal unsigned integer numbers !

Parameters in []'s are strings !

Parameters in {}'s don't need to be specified !

Note for V1.60:

PICSIZE and DEPTH commands have been removed for reasons described in the main window description. Should not disturb compatibility 'coz it's never been used hopefully..

ARexx List Functions have been removed, too, because the ARexx Window has been replaced by an ASL Requester.

In Version 2.00, I have reworked the design a bit - the commands work like before, but some side effects with the GUI have been removed, such as automatic window reactivation when saving a picture etc., which could be disturbing when processing large amounts of pictures.

Besides that, V2.00 doesn't provide the QUIT command because it has not been working a single time :)

1.12 AGAiiff - ARexx Commands

SYNTAX

UseTopaz {OFF}

FUNCTION

Switches on / off fontsensitivity.

INPUTS

If 'off' is specified, AGAiiff uses the Workbench's Screen Font.
Else it uses topaz/8.

RESULTS

RESULT =
RC = 0

BUGS**SEE ALSO**

1.13 AGAiiff - ARexx Commands

SYNTAX

DCSize <n>

FUNCTION

Selects dc.x data size

INPUTS

n: 0 Byte (dc.b)
 1 Word (dc.w)
 2 Long (dc.l)

RESULTS

RESULT =
RC = 0, or 10 if error

BUGS**SEE ALSO**

1.14 AGAiiff - ARexx Commands

SYNTAX

GLOBALNOSHOW {OFF}

FUNCTION

Disables/Enables picture showing until the next call
of this function.

BUGS**SEE ALSO**

1.15 AGAiiff - ARexx Commands

SYNTAX

SCREENPROMO {OFF}

FUNCTION

Disables/Enables screen promotion (--> Description of Prefs Window)

BUGS

SEE ALSO

1.16 AGAiiff - ARexx Commands

SYNTAX

LOCKRANGE {OFF}

FUNCTION

Disables/Enables range locking (--> Description of Prefs Window)

BUGS

SEE ALSO

1.17 AGAiiff - ARexx Commands

SYNTAX

Version

FUNCTION

Returns a versionstring.

INPUTS

None

RESULTS

RESULT = [versionstring]

RC = 0

BUGS

SEE ALSO

1.18 AGAiiff - ARexx Commands

SYNTAX

Status <n>

FUNCTION

Returns information about AGAiiff's status.

INPUTS

n: returns:

0 pathname of current picture or empty string.

1 picture size (<X> <Y>)
2 depth
3 selected save range (<X1> <Y1> <X2> <Y2>)
4 DisplayMode (decimal number)
5 "1" if GUI is hidden, else "0"
6 address and length of the original decrunched
ILBM data (RAW-Modulo) as (<address> <length>)
7 0 if the user selected CX_POPUP=NO, else 1
8 data about currently loaded animation
as (<totalframes> <currentframe>). In single-picture mode,
<totalframes> is "0".

RESULTS

RESULT = [result as described above]
RC = 0, or 10 in case of error (wrong <n>)

BUGS

SEE ALSO

1.19 AGAiiff - ARexx Commands

SYNTAX

ACTIVEWIN {OFF}

FUNCTION

Preferences/Activate Window

BUGS

SEE ALSO

1.20 AGAiiff - ARexx Commands

SYNTAX

SAVECLOSE {OFF}

FUNCTION

Preferences/Save=ClosePic

BUGS

SEE ALSO

1.21 AGAiiff - ARexx Commands

SYNTAX

COORDS {OFF}

FUNCTION

Preferences/Show coords

BUGS

SEE ALSO

1.22 AGAiiff - ARexx Commands

SYNTAX

```
Requestfile [windowtitle]
```

FUNCTION

Displays an ASL-Filerequester with the desired windowtitle

INPUTS

windowtitle: Title of the requesterwindow

RESULTS

```
RESULT = [pathname] or empty string  
RC = 0
```

BUGS

SEE ALSO

1.23 AGAiiff - ARexx Commands

SYNTAX

```
Request [text]
```

FUNCTION

Displays a normal requester with OK / Cancel gadgets using the Intuition/EasyRequestArgs function.

INPUTS

text: Requester text.

RESULTS

```
RESULT =  
RC = 1 for OK, 0 for Cancel
```

BUGS

SEE ALSO

1.24 AGAiiff - ARexx Commands

SYNTAX

```
TypeText [text]
```

FUNCTION

Prints a text into the Main Window's message line.

INPUTS

text: Textline to be printed. No LF's or similar.

RESULTS

RESULT =
RC = 0

BUGS

SEE ALSO

1.25 AGAiiff - ARexx Commands

SYNTAX

LockGUI

FUNCTION

Locks the Graphical User Interface, so that all clicks etc. won't be noticed. Use this in ARexx scripts that must not be interfered by the user.
This call uses a nest counter.

INPUTS

None

RESULTS

RESULT =
RC = 0

BUGS

SEE ALSO

UNLOCKGUI

1.26 AGAiiff - ARexx Commands

SYNTAX

UnlockGUI

FUNCTION

Unlocks the GUI.

INPUTS

None

RESULTS

RESULT =
RC = 0

BUGS

SEE ALSO

LOCKGUI

1.27 AGAiiff - ARexx Commands

SYNTAX

ShowGUI

FUNCTION

Opens the Main Window.

INPUTS

None

RESULTS

RESULT =

RC = 0 if success, 10 if error.

BUGS

SEE ALSO

HIDEGUI

1.28 AGAiiff - ARexx Commands

SYNTAX

HideGUI

FUNCTION

Closes the Main Window and unloads a possibly loaded picture.

INPUTS

None

RESULTS

RESULT =

RC = 0

BUGS

SEE ALSO

SHOWGUI

1.29 AGAiiff - ARexx Commands

SYNTAX

ShowPic

FUNCTION

Puts the picture screen to front. If no picture screen is open, this function is no-op.

INPUTS

None

RESULTS

RESULT =
RC = 0
1 if no picture screen was open

BUGS

SEE ALSO

1.30 AGAiiff - ARexx Commands

SYNTAX

ClearPresets

FUNCTION

Clears the save format preset list.

INPUTS

None

RESULTS

RESULT =
RC = 0

BUGS

SEE ALSO

ADDPRESET

,

SETPRESET

1.31 AGAiiff - ARexx Commands

SYNTAX

AddPreset [name]

FUNCTION

Adds a preset to the save format preset list. The current save format and color format gadgets are saved.

INPUTS

name: Name the preset will appear under.

RESULTS

RESULT =
RC = 0

BUGS

SEE ALSO

CLEARPRESETS
,
SETPRESET

1.32 AGAiiff - ARexx Commands

SYNTAX

SetPreset [name]

FUNCTION

Set the save format and color format gadgets according to the preset called [name].

INPUTS

name: Preset to set up.

RESULTS

RESULT =
RC = 0, or 10 if error (preset [name] doesn't exist)

BUGS

SEE ALSO

CLEARPRESETS
,
ADDPRESET

1.33 AGAiiff - ARexx Commands

SYNTAX

SFormat [sformatstring]

FUNCTION

Sets the save format

INPUTS

sformatstring: IFF, RAW, RAWI, COPPER, SPRITE, MASK, MASK1, NOPIC,
UNPKEDCHUNKY, CHUNKYCOLS
FILE, DC, PC

RESULTS

```
RESULT =  
RC = 0
```

NOTE

"UNPKEDCHUNKY" was called "CHUNKYREGS" in pre-2.10.
"RAWI" was called "RAWBLIT" earlier;
"RAW" was called "RAWNORM" earlier.

BUGS

SEE ALSO

CFORMAT

1.34 AGAiiff - ARexx Commands

SYNTAX

```
CFormat [cformatstring]
```

FUNCTION

Sets the colors format.

INPUTS

```
cformatstring: 12BIT,  
                24BIT0RGB,24BITRGB0,24BITRGB,  
                24BITLH,24BITLHI,24BITHL,24BITHLI,96BIT,  
                NOCOLS, TOP, BOTTOM
```

NOTE

"24BIT0RGB" was called "24BIT" in pre-2.10.

RESULTS

```
RESULT =  
RC = 0
```

BUGS

SEE ALSO

SFORMAT

1.35 AGAiiff - ARexx Commands

SYNTAX

```
Load [filename] {NOSHOW}
```

FUNCTION

Loads a picture.

INPUTS

```
filename: Pathname of the picture.
```

NOSHOW: If specified, don't show the picture.

RESULTS

RESULT =
RC = 0, or 5 if Error (can't load file)

BUGS

SEE ALSO

LFORMAT

1.36 AGAiiff - ARexx Commands

SYNTAX

Save [filename]

FUNCTION

Saves a file.

INPUTS

filename: Pathname of the file.

RESULTS

RESULT =
RC = 5, if an error occurred while saving.

BUGS

Doesn't return error codes.

SEE ALSO

SFORMAT

,
CFORMAT

1.37 AGAiiff - ARexx Commands

SYNTAX

Reshow

FUNCTION

The same as the RESHOW button in the Main Window.

INPUTS

None

RESULTS

RESULT =
RC = 0

BUGS

SEE ALSO

NOSHOW
,
RESTORE

1.38 AGAiiff - ARexx Commands

SYNTAX

Noshow

FUNCTION

Disable showing the picture until the next RESHOW.

INPUTS

None

RESULTS

RESULT =
RC = 0

BUGS

SEE ALSO

RESHOW
,
RESTORE

1.39 AGAiiff - ARexx Commands

SYNTAX

Restore

FUNCTION

Restore the picture data like depth, picsize and displaymode and do a RESHOW.

INPUTS

None

RESULTS

RESULT =
RC = 0

BUGS

SEE ALSO

RESHOW

1.40 AGAiiff - ARexx Commands

SYNTAX

```
DispMode <a>
```

FUNCTION

```
Set DisplayMode.
```

INPUTS

```
a: DisplayMode ID (decimal long int)
```

RESULTS

```
RESULT =  
RC = 0, or 5 if no picture was loaded.
```

BUGS

SEE ALSO

1.41 AGAiiff - ARexx Commands

SYNTAX

```
Range <x1> <y1> <x2> <y2> {FULLPIC}
```

FUNCTION

```
Set save range.
```

INPUTS

```
x1/y1: One Corner of the range.  
x2/y2: Two Corner of the range. (?)  
FULLPIC: Specify this to select the whole picture.
```

RESULTS

```
RESULT =  
RC = 0
```

BUGS

SEE ALSO

1.42 AGAiiff - ARexx Commands

SYNTAX

```
GetPixel <x> <y>
```

FUNCTION

```
Get the colorregister of the pixel specified by 'x' and 'y'.  
The #?PIXEL-Functions are very useful for picture scanning  
functions.
```

INPUTS

```
x: X-Position
```

y: Y-Position

RESULTS

RESULT = colorregister
RC = 0 or 1 if x/y was out of picture size
5 if no picture was loaded.

BUGS

SEE ALSO

SEARCHPIXEL

1.43 AGAiiff - ARexx Commands

SYNTAX

SearchPixel <x> <y> <colorregister>

FUNCTION

Search the line specified by 'y' for a color specified by 'colorregister'. Don't start at the beginning but at X-position specified by 'x'. The #?PIXEL-Functions are very useful for picture scanning functions.

INPUTS

x: X-Position where to start search.
y: Line to search in.
colorregister: The color to search for.

RESULTS

RESULT = x-position where pixel was found.
RC = 0 if a pixel was found, 1 if no pixel was found, and
5 if no picture was loaded.

BUGS

SEE ALSO

GETPIXEL

1.44 AGAiiff - ARexx Commands

SYNTAX

ANext

FUNCTION

Activate the next frame in the currently loaded animation.

INPUTS

none

RESULTS

```
RESULT =  
RC = 0 or something different if there is no next frame.
```

BUGS

SEE ALSO

1.45 AGAiff - ARexx Script 'Startup.agaiff'

This script is called once when AGAiff is started. So you should put your Preferences, Presets and other Setup stuff here. The script by me contains some simple save formats.

1.46 AGAiff - ARexx Script 'AutoScan.agaiff'

This script is very useful for game programmers that want to create sprite animations. Imagine you have some animation frames for your ship for the newest shoot'em up game, in one picture in a random order. If you don't want to cut these 10 frames and save them on your own, just let the graphician frame these areas with a specific color. A 32x16 sprite will be framed with an 34x18 frame.

Then you call AutoScan.agaiff: First you will be asked for the color the areas are framed with, then you must give the script a filename. The picture is searched through and all framed areas are saved with the filename extended with '.*d' where %d is a numbering from 0 to <number of frames>. .. the frame won't be saved of course.. Don't forget to set the right save format !!

The only bad thing about this script is its speed. On my standard A1200 WITHOUT fast mem (:-(()), it needs a quarter of a minute or more to look through a picture. Anyway, i'm not the best ARexx coder, so why don't you try to write a faster one ??

1.47 AGAiff - ARexx Script 'GridSave.agaiff'

Imagine you put a grid on a picture. The picture is splitted into several blocks. Then imagine you save these blocks from left-right or top-down. This is exactly what this script does.

```
Grid X Block Size: Horizontal Size of one grid block  
Grid Y Block Size: Vertical Size of one grid block  
Number of Blocks X: Horizontal number of blocks  
Number of Blocks Y: Vertical number of blocks  
Add X: Additional space between to grid blocks.  
Add Y: Additional space between to grid blocks.  
Start X: The first grid block ordinate.  
Start Y: The first grid block ordinate.  
Save mode: Top-Down, then left-right, or vice versa.  
Append files: Whether to have all blocks joined in one file,
```

or saved as 'filename.%d', where %d is a numbering from 0 to <number of blocks>.

1.48 AGAiff - ARexx Script 'SplitAnim.agaiff'

Demonstrates how to use the one and only 'ANEXT' command ;-)

1.49 rxs_info

Simply shows how to use the STATUS command.

1.50 AGAiff - Running ARexx Scripts

The ARexx requester from pre-V1.60 has been replaced by a normal ASL requester showing the contents of REXX: with a default pattern of '~(Startup).agaiff', thus reading all files ending with '.agaiff' except the Startup script. Double-Click one of the files to start it.

1.51 AGAiff Development History

Started coding in the middle of 1993, when I got my A1200. AGAiff was an experiment - I wanted to learn more about that new cool OS and the AGA chipset.

Version 1.0 Release A (01-Mar-94)

First public release.

Version 1.21 Release B (01-Apr-94) (kein Aprilscherz)

- 07-Mar-94 Bug fix: Sprites' control words were too short. Now twice as big.
- 07-Mar-94 Minor fix: ASL Requesters now remember last Directory, window is put into sleep status while requesting.
- 13-Mar-94 Bug fix: Additional mask plane is now ignored.
- 13-Mar-94 Several small 'security' bug fixes.
- 20-Mar-94 Added Color Requester
- 21-Mar-94 Added IFF Saving
- 21-Mar-94 Additional Tooltypes

Version 1.60 Release C (16-Sep-94)

- 27-Jun-94 Chunky save modes added.
 - 14-Jul-94 ASL-Requesters: no '.icon's & some other trinkets..
 - 24-Jul-94 Range locking added.
 - 25-Jul-94 Screen Promotion & global NoShow added.
 - 26-Jul-94 IFF crunching added.
 - 02-Aug-94 Additional tooltypes
 - 02-Aug-94 Bug fix: Starting AGAiff twice could lead to unpredictable results.
-

- 02-Aug-94 CLI-Interface.
- 02-Aug-94 Public Screen support.
- 03-Aug-94 Key equivalents for most gadgets.
- 03-Aug-94 ARexx STATUS command enhanced.
- 06-Aug-94 !! on vacation.. !!
- 26-Aug-94 Replaced ARexx Window by ASL-Requester.
- 28-Aug-94 New ARexx commands #?PIXEL..
- 29-Aug-94 New ARexx scripts AutoScan & GridSave
- ... and lots of bugs fixed ...

The first release of V1.60 featured a wrong window title bar with 'PRIVATE VERSION ## DON'T SPREAD' in it. I left it in there by accident - it was a fully working version. However, the version I sent to Fred Fish and AmiNet was correct!

Version 1.99 Release D (10-Jun-95) (public Beta)

- 19-Nov-94 Window opening could cause oversized windows with big fonts (hmm, thought i fixed this before :-))
- 20-Nov-94 Bug fix: Forgot docs for SHOWPIC Arexx command..
- 21-Nov-94 Started animation support..
- 26-Nov-94 Bug fix: loaded ColorTables were never freed..
- 01-Dec-94 Bug fix: Mode Promotion could clear HAM/EHB flags..
- 24-Dec-94 Datatypes loading
- 01-Jan-95 ARexx QUIT command removed - didn't work anyway..
- 02-Jan-95 Added several new Preferences Items
- ... after a long, long break I continued coding:
- 01-May-95 Framenumber is mentioned in Coords-Screen
- 01-May-95 Range selection re-coded.
- 04-May-95 ARexx STATUS 8, (UN)LOCKGUI improved
- 05-Jun-95 ARexx-Server didn't take care of RXFB_RESULT.. shit
- 08-Jun-95 ARexx-Server totally recoded.

... and again most probably some bugs fixed ...

Several intermediate versions have been spread between my local scene friends meanwhile.

Version 2.10 Release F (13-May-96)

- 15-Jun-95 ARexx-Window remembers last directory...
- 15-Jun-95 ARexx-REQUESTFILE didn't show correct title since V1.99
- 15-Jul-95 Loading: e.g. Width of 17 pixels is not expanded to 32 pixels any more!!
- DLTA5 Animation loading works 100% now.
- ... long break ...
- 05-May-96 General code & GUI clean-up.
- Full-Anim support completely removed.
- 10-May-96 Pics >8 planes are now rejected.
- 11-May-96 Bug fix: ARexx Port didn't recognize when Arguments were missing ;-(
- 12-May-96 Bug fix: Damn bug in the ColorRequester code, which made the machine crash quite often (forgotten RTS ;-(
- ASL-Requesters always start with height 400 now.
- New Color Save Modes.
- Additional ARexx Tokens for them.
- 13-May-96 Removed some hardware dependencies, AGAiiff also runs on graphics cards now.
- Color Requester always works in 8 bit now.

Added PC assembler dump save mode.
IFF-Loader clears the rightmost bits of every line now,
which some savers don't clear when the width is
not at word boundaries.
New ARexx Commands: ACTIVEWIN and SAVECLOSE

Version 2.50 (05-Feb-97)

- 31-Dec-96 24 Bit color save modes RGB0 and 0RGB where accidentally used vice-versa.
4pl-Sprite saving could mask out parts of the planes.
- 11-Jan-97 Error messages while saving were always overwritten by "Saving done!".
Started support for FastMem buffers
- 12-Jan-97 Unpacked Chunky can be saved from FastMem. Doesn't round up to 16-pixel boundaries any more!
Testing for ECS Blitter now.
Rewrote screen mode adjusting code (BestModeID).
Coordinates display screen can now be switched off (it would bug with CyberGFX).
- 13-Jan-97 Cleaned up and optimized the whole source code in some areas.
- 20-Jan-97 Cleaned up the whole loading code.
STATUS PATHNAME works again.
- 23-Jan-97 Cleaned up ARexx port code.
- 28-Jan-97 Updated some parts of the documentation.
- 03-Feb-97 Loading datatypes cleanly without temporary files, displaying base name in frame gadget.
Range selector is much more responsive and doesn't bug on CyberGFX now.
Savers do much more error checking and return nice error messages!
ARexx 'SAVE' command can return an error!
- 05-Feb-97 Rewrote writerange routine to 'CutRangeX' which uses the processor now -> planar saving from fast mem!
IFF saver rewritten to use CutRangeI.
Removed most of the ToolTypes - use the ARexx Startup script now!!
Rewrote sprite saver to use CutRangeI.

Version 2.60 (13-May-97)

- 03-May-97 dc.X saver was broken.
- 12-May-97 Color requester was very broken.

Version 2.7 (02-Nov-97)

- 02-Nov-97 Fixed bug in OpenPic() which caused crashes when opening the coords screen.
Added collection of ancient IFF-Converters!
(back to school... ;D)

Version 2.8 (25-Dec-97)

- 11-Dec-97 Made picscreen window simple refresh -> less memory consumption!
The color requester is asynchronous now.
Datatypes loading uses PDTA_CRegs instead of PDTA_ColorRegisters now (was bugging with akPNG datatype, reported by Niels de Koning - thanks!)
When browsing through an animation on the display

screen, the color requester may stay open now

1.52 AGAff What's left ?

This program is FreeWare.

If you have suggestions or bug reports, then contact the author at:

Michael Krause
Mannesallee 24
21107 Hamburg
GERMANY

rawstyle@ms.demo.org or m.krause@tu-harburg.de
<http://ms.demo.org/rst>
