MPaint

COLLABORATORS			
	TITLE : MPaint		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 26, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

MPaint

1.1 MicroPaint 2.0.4 © 1995-1999 by FR-Softworks

1.2 Installing MicroPaint

Amiga OS 2.0+ is required (OS 3.0+ for datatypes support). Notice that only screen modes with up to 256 colours are supported at this moment.

o You should have at least 512 KB of RAM. Large pictures with many colours and the "undo" feature will require additional memory.

o You need amigaguide.library version 36 or above to be able to display the online documentation.

» Run the Installer script supplied with MicroPaint to install the program.

The following files will be copied:

File	Directory
Program files	Program directory
CBSI	C:
cxb.library	LIBS:
Catalog files (*)	LOCALE:Catalogs

(*) The catalog files are optional. You do only need them if you want MicroPaint to use another language than English.

Notes: If MicroPaint doesn't start, try to copy »cxb.library« to your LIBS: directory. Due to the way Blitz Basic opens library files, MicroPaint cannot detect the absence of »cxb.library«.

If MicroPaint opens a dialog box stating that your keyfile is corrupt, please contact the author in order to get a new one.

Undo

This feature requires additional memory, because a copy of the entire screen must be kept.

The "FASTUNDO" feature requires even more memory, because it uses a second screen to store the "undo" buffer.

1.3 Overwiew

MicroPaint has the following features:

o up to 256 colours (requires an AGA machine)
o supports ECS and AGA
o supports Intellifonts, styles, draw modes like inverse, colour fonts
o supports all graphics modes except HAM
o flexible spray tool
o multi spray uses random colours
o calibration tool: make yout pictures brighter, more green and so on

o remap and revert pictures
o convert to black and white
o swap colours
o print pictures
o grab screens
o cut, rotate, resize, and mirror brushes
o edit, open, and save colour palettes
o save a selected area of your pictures in .BMP format
o supports datatypes
o line styles
o "character gap" definable: "abc" or "a b c"
o show details about IFF files
o view files
o can be run on a "CyberVision" graphics board
o localized
o toolbox
o can play animations
o ARexx port "MicroPaint"
o lists the last recently used files at the end of the Project menu
o provides an AppIcon
o "Mosaic" effect
o clipboard support
o standard brush size, shape, and line thickness are definable
o supports macros
o 3D text engine
o the last saved version of the current picture can be recalled
o configuration via ToolTypes
o "undo" option
o "Emboss" effect (relief and 3D)
o Remove feature
o colours can be picked
o you can select the GUI font
o HAM support
o you may crop the image
o display of the mouse coordinates

The following features are not implemented yet:

o support for graphic cards

» Open Demo.IFF to see how the supported modes look like.

» Have a look at
               History
               to see which bugs have been fixed.

1.4 Passing arguments to MicroPaint

Passing CLI arguments

Call MicroPaint like follows:

```
  mpaint <file name>
```

MicroPaint will then try to open <file name>.

Passing Workbench arguments

Two techniques are possible:

1. Create an icon for your picture file and change the "default tool" to "mpaint". MicroPaint saves icons automatically.

2. Hold down the shift key and click on the picture's icon. Then double click on the icon of MicroPaint. (If you have activated the Workbench menu item "Window/Show Contents/All Files", reverse the process, please.)

How to use the AppIcon

You can open a picture file by dragging its icon on the AppIcon of MicroPaint, too. If the file is an animation, then MicroPaint will play it.

Doubleclicking on the AppIcon brings the MicroPaint screen to the front.

Remarks

MicroPaint will try to open the file in all cases. You will be asked which format the screen to be opened should have, but your answer is not really important because MicroPaint will open a new screen corresponding to the settings the graphics file contains.

1.5 ToolTypes

The following ToolTypes are recognized:

ToolType	Description
ASPECT	Definition of the aspect ratio (defaults to 1.5)
BRUSHPTR	If this option is enabled, MicroPaint uses the current brush instead of the normal mouse pointer. (You have to MOVE the mouse when in "Points" mode to see the brush.)
COORDINATES	Enables the display of the mouse coordinates.
DEPTH	Definition of the screen depth in planes. The screen has max. 2^depth colours. (*)
FAIL	Might be needed by some old or slow Amigas. o mode names ("Lo Res" are shown as numbers o monitor flags are ignored o "undo" is disabled

FASTUNDO Enables the faster version of the undo option which requires more memory. (Ensure that UNDO is enabled. This entry only determines which "undo" version has to be used.)

FONT Selects the GUI font (default: "XEN.font"). If the font cannot be found, "topaz.font" (8 points) is used.

FONTHEIGHT Sets the height of the GUI font (default: 8) in points.

HEIGHT Definition of the screen height in pixels. (*)

SCREEN Definition of the screen's mode id. (The mode id of the current screen can be obtained by the "Project > About" menu item.

If this option is used, MicroPaint will not open its screen mode requester on startup.

Hexadecimal values must be preceeded by a dollar sign ($). Leading zeros are ignored.

$29004 equals $00029004.

TEMP Definition of the temporary file (path and name). This defaults to "SwapFile" in the program directory.

UNDO Enables the "undo" feature. See FASTUNDO for how to get a faster version of "undo".

UNDOFILE Definition of the temporary "undo" file. This defaults to "T:UndoFile".

WIDTH Definition of the screen width in pixels. (*)

(*) This option must be used in conjunction with the SCREEN option.

1.6 Frequently asked questions

Q: I want to see the current brush instead of the standard mouse pointer. It would be easier to place a brush if I could see it this way.

A: Enable the BRUSHPTR ToolType.

Q: If I set the background colour to black, the pointer disappears.

A: This is caused by the AmigaOS. It uses fixed colour palette entries for its mouse pointer.

(i) Change the colours used for the pointer via Commodore's "Pointer" preferences utility or

(ii) edit the colour palette

or

(iii) open another colour palette, such as "DPaint.COL".

Q: Where will the brush be placed if I hit the mouse button?

A: Imagine a rectangle surrounding the brush. The mouse pointer indicates the upper left corner of this rectangle.

Q: If "undo" is enabled, MicroPaint is too fast - I cannot alter the size of boxes, circles etc.

A: MicroPaint stores the current screen when you hit the left mouse button. Just keep the mouse button down until the busy pointer disappears. After that you can move the mouse to alter the size of the object and then release the mouse button.

Q: The toolbox disappears.

A: Have a look at the description of the "Mode » Points" command.

Q: When I select a menu item, parts of the menu do not disappear from the screen.

A: Have a look at the description of the "Options » Show Toolbox" command.

Q: When I switch to the Workbench screen, the window I selected is deactivated at once.

A: Double click within the window to select it.

1.7 menus

MicroPaint features the following menus:

Project
File operations like open, save, print; LRU files

Mode
Tools selection

Tools
Calibration, effects, new CLI, help

```
Brush
      Mirror, rotate, resize, clipboard

Macro
      Record, execute, open, save

Options
    Font, palette, line style, toolbox, HAM
```

1.8 Project

```
                  New...           This command clears the screen using the current ←
                     background
                   colour.

                   Note: The background colour can be changed by altering the
                         RGB values of colour #0.

Open...            This command reads a picture and lets you open a compatible
                   screen. The default file name extension for picture files
                   is ".IFF".

                   HAM files are opened correctly if the HAM support has been
                   enabled via the Options menu. Otherwise, an AA picture is
                   assumed.

                   Optionally, you may keep the current screen. The picture
                   may be loaded partially only in this case, and the screen
                   will not be cleared. However, the colour palette of the
                   picture will be used in *all* cases.

                   If the selected file is an animation, then MicroPaint will
                   play it.

                   If necessary, MicroPaint will use OS 3.X datatypes to open
                   the picture. The utility
                   DT2IFF
                   is used to convert
                   datatypes to standard IFF-ILBM files. The picture will be
                   converted to a temporary file in the program directory.

                   MicroPaint lists the last recently used files at the end of
                   the Project menu.

                   Note: A new "undo" buffer will be created if the "Fast Undo"
                         option is enabled. The update of the screen contents
                         can be delayed by this process.

Save               This command saves the current picture.

Save As...         This command saves the current picture and allows you to
                   select a new file name via an ASL file requester.

Delete...          This command allows you to delete a file.
```

Details... This command shows width, height, depth and display mode of an IFF file. The display mode is shown as a view mode ID an as a string (for example "00000000 - LoRes").

Datatype pictures will not be converted by this function.

MicroPaint displays a warning if it is not able to open the picture without conversion.

View... This command uses SYS:Utilities/MultiView to display the file. This is usefull to see if all necessary datatypes have been installed correctly.

Note: Ensure that MultiView is running on its own screen.

Last Saved... This command allows you to revert to the last saved version of the current picture and can be used to undo unwanted alterations of the graphic.

Format... This command allows you to select a new screen mode view an ASL requester. The depth must not exceed 8 bit planes in MicroPaint.

The current picture can be kept. It will be saved to a temporary file during this operation (use the command Options/Swap File to select a file name.) MicroPaint will crop the picture if necessary.

Using swap files allows you to convert pictures easily from one screen mode to another.

Note: Enable the "Force HAM screens" option to be able to open a HAM screen.

Note: A new "undo" buffer will be created if the "Fast Undo" option is enabled. The update of the screen contents can be delayed by this process.

About This command displays copyrights, version number, and revision date.

Additionally, the following details are shown:

o current file name or "unknown"
o current screen format
o current paint mode
o currently used colour (register #)
o current view mode id (hexadecimal number and string)
o free memory (chip and fast)

Print... This command prints the pictures using the printer currently selected in Preferences.

To get an exaptable result, do not select the black/white option in Preferences. Select grey scale or colours instead.

Quit... This command quits the program.

1..5 The file listed in the menu will be opened - see above for details.

1.9 Mode

All paint modes except multi spray use the current colour. "Line[s]" and "Box" use the selected line style and thickness.

Most modes use the "rubber band" technique to show a preview.

Note: You can select the paint mode via the toolbox, too. The associated icon will be highlighted and the corresponding menu item will be checked, too.

The toolbox will be deactivated temporarily while you are painting.

Points This mode draws single dots as long as the left mouse button is held down (using the current brush).

» Hit the left Amiga and Alt keys simultaneously to plot a single dot.

If no brush is defined, the standard brush size and shape definition is used. These definitions can be changed via the "Options" menu.

Note: If the standard brush size is too small, the brush shape "Circle" can lead to unexpected results.

If the "BRUSHPTR" ToolType has been activated, the current brush is shown when the mouse is moved.

Note: In this case, you should not move the mouse when you are hittung the right mouse button and the mouse pointer is near the top of the screen. Otherwise strange things might occur.

Free This paint mode allows you to draw free hand figures.

Line Click at the start position and move the mouse pointer to the end of the line, and release the mouse button.

Lines Click at the start point of the line, release the mouse button and click at the end of the line. This will be the start position of the next line.

Box Click at the start point, move the pointer to the opposite corner of the box and release the mouse button to draw a rectangle.

Filled Box This mode allows you to draw filled boxes (see above for details.) The line pattern is used to fill the box.

Ellipse Move the mouse pointer to the middle of the ellipse to be drawn, hold the left mouse button down and move the mouse to any direction to define the radius.

Filled Ellipse   This mode draws filled ellipses (see above.)

Circle   This paint mode allows you to draw circles using the technique described above.

If the circles are not round, select the command Aspect Ratio of the Options menu to fix this.

Filled Circle   This command draws filled circles.

Text...   This opens a small window enabling you to edit up to 255 characters. The text will be added to the picture when you select the close gadget.

Use the command Options/Font... to select font, text colour, and draw mode. The "character gap" can be defined by using the corresponding command in the Options menu.

Note: If enabled, 3D text will be generated.

Fill   This paint mode fills areas surrounded by lines, boxes or anything else.

Spray   This paint mode works like an air brush. This mode can be configured by using the corresponding menu item of the Options menu.

Multi Spray   This mode is like the one explained above but uses all colours.

1.10 Options

Font...   This command calls the ASL font requester and lets you select font, text colour and draw mode.

The font size is limited by the window height only. (Font sizes of 500 points have been tested successfully.)

Palette

Edit: You may select the colour to be used.

The "Colour Editor" allows you to alter the red, green, and blue RGB values of the currently selected colour.

"Reset" restores the last recently used colours.

"Swap" allows you to swap colours.

"Range" allows you to create a colour range.

Open: This command opens a palette file.

Save: This command saves a palette file.

Default: This command opens the palette file "Standard.COL" which is located in the main program directory.

Notes:

- o The background colour can be changed by altering the RGB values of colour #0.
- o Open the palette window via the toolbox.

Spray Settings... You can specify the "spray delay", the "spray distance" and the "spray density".

The distance is measured from the mouse coordinates. The density is the number of points to be drawn.

The "spray delay" is used to make the spray function slower and is computed at the program start.

Aspect Ratio... If circles are not displayed correctly, you may have to alter this value.

The correct value depends on your monitor.

Swap File... This command lets you specify the file name for the temporary swap file which is used to store the current picture if you change the screen format.

The file name defaults to "SwapFile".

Line Style This command allows you to choose a line style.

If you press the "Define..." button, you can define your own line style. Simply type a 1 for a point which should be set, and a 0 otherwise. You can state up to 16 digits.

Input	Line style
1111111111111111	Default line style (---).
1010101010101010	A line made up of separated points (. . .).
1111111100000000	A dashed line (- -).

Character Gap... This command lets you specify the "gap" between individual characters (from 0 to 32767 points). The default value is 0 (zero).

This feature can be very usefull for birthday invitations cards and so on...

Animation...

This command allows you to define the animation speed, which defaults to 5. This value defines the delay between two frames and is measured in 1/50 seconds.

Brush Size..

This command allows you to define the size of the standard brush (in pixels).

Note: If the standard brush size is too small, the brush shape "Circle" can lead to unexpected results.

Line Thickness..

This command allows you to define the line thickness in pixels. This value is used by the paint modes "Lines[s]" and "Box".

The aspect ratio is taken into coniderationa.

Brush Shape

This command allows you to select the brush shape: "Circle" or "Square".

The aspect ratio is taken into coniderationa.

Note: If the standard brush size is too small, the brush shape "Circle" can lead to unexpected results.

3D Text Settings...

This command allows you to alter some parameters of the 3D text engine:

o text colour
o outline colour
o magnification factor
o character gap
o shadow depth

The option "3D text active" is used to switch the 3D text engine on (or off). The option "Perspective view" will generate a 2D text output with shadows only if it is disabled.

Suggestion: Apply the "Emboss" effect to your 3D texts!

Undo Settings...

This command allows you to enable (disable) the "undo" button and to define the location of the swap file.

o everything you draw and the commands of the "Tools" menu can be undone

o use a fast device like "RAM:" to store the "undo" file

Refer to
Using 'undo'
for details.

The standard version of "undo" uses the swap file to store the current picture. "Fast Undo" saves the colour palette to this file.

If you disable the "undo" feature, the buffer or swap file repectively will be deleted.

A new "undo" buffer will be created if the "Fast Undo" option is enabled. The update of the screen contents can be delayed by this process.

HAM Settings... This command allows you to enable or disable the HAM support of MicroPaint.

option	result
Enable HAM6	If the picture has six bitplanes (64 colours), a HAM screen is used.
Enable HAM8	If the picture has eight bitplanes (256 colours), a HAM screen is opened.
Force HAM screens	If you select a screen depth of 6 or 8 bitplanes, a HAM screen is opened.

If your Amiga does not have the AA chipset, MicroPaint enables HAM6 and forces HAM screens.

Select Printer... This command runs "SYS:Prefs/Printer". Refer to your Workbench manual for details.

Printer Setup... This command runs "SYS:Prefs/PrinterGfx". Refer to your Workbench manual for details.

Show Toolbox This command opens the toolbox or hides the associated window.

If the toolbox cannot be created, the current screen might be too small.

Note: MicroPaint creates a temporary file called »T:ToolBox.ICL« which contains the toolbox definition. Use a utility like "SnoopDOS" to check if the file could be created successfully.

(The toolbox will be hidden by MicroPaint if this is required by an operation.)

The toolbox should be deactivated before executing a macro.

Note: The toolbox cannot be displayed if the "Points" mode and the "BRUSHPTR" ToolType are active.

(Otherwise, parts of the screen might be covered by parts of the toolbox when the mouse is moved.)

Show Coordinates — This commands opens a window which contains the current mouse coordinates.

1.11 Tools

If the "FASTUNDO" version of the "undo" feature has been enabled, ↩ none
of the following effects can be undone. MicroPaint will create a new "undo" buffer in this case, however.

Calibrate... — This command brings up the "Calibration requester".

"XXX times brighter": The current contents of the colour registers will be multiplicated with this value (or divided by it if you specify a negative number).
The default is "1" (no change).

"Alter components": You can specify the colour components which should be affected by the calibration process. The default is "RGB" (all components are affected).

"0 components := ": Registers which current content equals to zero are set to this value. The maximum value supported by MicroPaint is "15". The default value is 0.

"Registers XX to XX": The colour registers affected by the operation. This defaults to all registers.

The calibration settings can be stored and retrieved.

Click on "OKAY" to perform the calibration.

Grey Scale... — The colour palette of the current picture is converted to grey scale. The brightness of the colours will be preserved.

This process cannot be undone.

Revert — All colours will be reversed. This can be undone be calling this menu item once more.

Negative — This command creates the negative of the current picture and can be undone by calling this menu item once more.

Swap Colours... — This command swaps the contents of two colour registers.

Colour Range... This command allows you to create a colour range between two colour registers.

If you set one register to white and the other to black, the picture is converted to grey scale.

Screen Grabber You can grab a screen of another program which is actually running.

The following features are available:

o open a new screen (according to the grabbed screen)
o scale the picture to fit on the current screen

If you do not accept the new screen format, the current picture will be preserved.

MicroPaint can grab HAM screens.

Note: A new "undo" buffer will be created if the "Fast Undo" option is enabled. The update of the screen contents can be delayed by this process.

Remap... Points of a specified colour will be drawn in the other colour specified.

Monochrome... This command converts the current picture to monochrome. The default threshold value is 10.

Mosaic... This command converts the current picture to something similar to a mosaic. The size of the tiles to be created can be changed (default: 5 pixels).

Note: Values below 2 are ignored. The tile size must not exceed the size of the current screen.

Emboss... This command converts the current picture to something similar to a relief.

The brightness difference (delta) between the pixels can be altered.

The "emboss" process changes the colours of all pixels according to the brightness difference between the current pixel and the pixel left and above.

Brightness difference	New pixel colour
+ delta	darkest colour
- delta	brightest colour
else	background colour

The colour of the graphics objects will be preserved if the "Keep objects" option is enabled.

The colours stated above are swapped if "reverse direction" has been selected.

Note: If some objects have not been processed, their brightness might be the same as the brightness of the background colour.

Try to alter the delta value in that case.

Rem. feature... This command allows you to remove isolated pixels from your picture.

Pixels which are not surrounded by the background colour will not be removed.

Isolated pixels might have been created by scanning a document or by converting a picture file.

Pick Colour... This command allows you to get information about the colour of a pixel.

When you click on a pixel, a dialog box with the following information is displayed:

o X position
o Y position
o colour (register)

If you select the "Pick" button, this colour will become the active colour.

New CLI... This command opens a new CLI window.

Play Anim... This command plays the selected animation file using the

playanim
utility.

Execute ARexx s. This command allows you to execute an ARexx script.

The default filename extension is ".mprx". The ARexx port is called "MicroPaint".

Refer to the "ARexx Commands" documentation for details.

Help... This command displays the online documentation using the amigaguide.library directly.

1.12 Brush

Select the »Points« tool to use a brush!

Cut... Move the mouse pointer to the upper left corner of the area you wish to become a brush. Drag the mouse to the lower right corner of the desired area while holding down the left mouse button.

This is similar to drawing a rectangle.

Clear The current brush is deleted without a warning.

Open... This command lets you load a brush from disk. The colour palette of the brush will not be used.

Animated brushes are not supported by MicroPaint. If you try to open an anim brush, an error message is generated ("No IFF file").

Save... The current brush is stored on the disk.

Brush Size The following options are available:

Double X, Double Y, Double X+Y, Half X, Half Y, and Half X+Y.

Mirror This command mirrors the brush horizontally or vertically.

Rotate You can rotate the brush by 45 or 90 degrees orby a user-defined value.

Save S. As BMP...You can save a section of the picture in the common BMP format by selecting the desired area with the mouse and chosing a file name via an ASL requester.
The resulting file will be uncompressed and with 256 colours.

Notes: Commodore’s BMP datatype is somewhat faulty, because it cannot open pictures which had less than 256 colours before converting them to BMP.

You can use Personal Paint or XView to deal with BMP files, for instance.

Crop Selection You can crop the image by drawing a rectangle around the parts to be preserved (like cutting a brush). Anything outside the selection will be erased.

MicroPaints opens a new screen for the altered picture. The "undo" buffer is cleared by this command.

Note: If the new screen is too small, the user interface might look strange.

Copy to Clipb. You can save the current brush to the clipboard (unit 0) for an exchange with other programs.

Paste f. Clipb. You can load a brush from the clipboard (unit 0).

Clear Clipboard The content of the clipboard (unit 0) is cleared.

1.13 Macro

Record — The following key strokes, mouse moves, etc. are recorded until the "Stop" command is selected.

Notes:

- use key strokes rather than mouse buttons

- close windows by moving the mouse pointer into the upper left corner of the screen before hitting the mouse button;

- use the maximum screen size and depth to avoid problems with other screen resolutions;

- do not use the toolbox;

- hit the left Amiga key and v simultaneously (or left Amiga and b respectively) rather than clicking on the "Ok" or "Cancel" buttons;

Stop — Stops the recoding of the current macro.

Execute — The current macro is executed. MicroPaint opens a requester after terminating the macro.

The toolbox should be deactivated before executing a macro to ensure that the whole screen is accessible.

Open... — You can load a macro file (.mac).

Save... — MicroPaint adds the extension ».mac« to the selected filename.

1.14 Tips & Tricks

Convert a picture into another screen format

Rotate / mirror text lines

How to use the colour palette of brushes

Macro problems

1.15 Macro problems

Windows cannot be closed

Move the mouse pointer into the upper left corner of the screen before clicking the left mouse button.

This is necessary to avoid problems with other screen resolutions, because macros use the current position of the mouse pointer.

The macro does not terminate

Macros should be recorded using screens with maximum size and depth.

Some operations need a long time if big screens are used. Thus the Amiga might be busy when the macro wants to terminate and ignores this command.

The recommended technique might result in a waste of time if low screen resolutions are used.

1.16 Convert a picture into another screen format

This is very easy. Simply perform the following steps:

1 Start MicroPaint and open the picture.

2 Press the left Amiga key and 'm' to see the Workbench screen.

3 Load MicroPaint and grab the screen of the MicroPaint picture.

4 Now you can exit the MicroPaint program loaded in step #1.

1.17 Rotate / mirror text lines

Convert the text line to a brush and mirror / rotate the brush!

1 Cut the text line using "Cut" in the "Brush" menu.

2 Perform "Mirror" or "Rotate" on the brush.

3 Use "Mode/Points" to put the text on the screen.

1.18 How to use the colour palette of brushes

Use the Command "Options/Palette»Open..." to load the colour palette of the brush.

Note: This will affect the whole picture!

1.19 History

Version	remarks
1.7.3	First localized version.

1.7.4 Introduction of GadTools menus.

1.7.5 Introduction of the toolbox. The menu items and the documentation have been revised.

1.7.6 MicroPaint can play animations now. Old versions could not be used as default tool in icons - this has been fixed.

1.7.7 The online help is displayed by calling the amigaguide.library directly now - Multiview is no longer needed and thus the online documentation can be displayed on systems using Workbench 2.x, too.

1.7.8 The ARexx port "MicroPaint" has been implemented.

1.7.9 All external files (mouse pointer, toolbox) have been inegrated into the executable file. The creation of filled circles and ellipses is much faster now.

1.8.0 All event handlers (menu, gadgets, ARexx) have been rewritten. The brush handling has been optimized. The toolbox will be deactivated while you are painting.

1.8.1 The last recently used files are listed at the end of the Project menu.

1.8.2 The screen grabber has been revised. MicroPaint has been optimized. Screens are scaled with more precision now.

1.8.3 The palette requester uses the "gadtools.library" now.

1.8.4 The error checks of the Blitz Basic 2 run-time library have been disabled. All errors are handled by Micropaint now.
The handling of the toolbox has been optimized.

1.8.5 ARexx scripts can run from within MicroPaint. An AppIcon is provided. The "locale.library" support has been rewritten from scratch. (Numbers are localized, too.) The name of the current view mode is displayed now.

1.8.6 ARexx commands can provide a result using the ClipList of the ARexx interpreter. (The commmand FILENAME returns the name of the current file, for example.) The handling of the .LRU file has been improved.

1.8.7 Circles and ellipses are filled correctly now, even if they cover other parts of the picture (the preview is filled, too). A new effect called "Mosaic" has been introduced. Clipboard support has been added. Standard brush size and shape are definable. The line thickness can be defined. AGA is fully supported now.

1.8.8 Macros are supported now.

1.8.9 Results are passed to ARexx via a RexxMsg structure instead of using the ClipList. Thus the execution of ARexx commands is much faster now.

1.9.0 A 3D text engine has been added. The area behind the toolbox is filled correctly.

2.0.0 alpha The last saved version of the current picture can be recalled by the new menu item "Project/Last Saved". The Palette Editor provides access to the "Swap" and "Range" dialog boxes now. New ToolTypes can be used to alter some preferences of MicroPaint. The palette window can be opened via the toolbox. An "undo" option has been introduced.
(The new features have been suggested by Jen Allen - Thanks!)

beta A progress indicator has been introduced. A new effect allows you to convert the current picture to a "relief". The "remove feature" removes isolated pixels.

final The "emboss" effect can preserve the colours of the graphics objects. A "Fast Undo" feature has been added. The palette window will be opened immediately even if the "undo" feature has been activated. ARexx scripts and documentation have been revised. MicroPaint's user interface has been improved - the dialog boxes contain "Ok" and "Cancel" buttons. The "emboss" effect did not use the "Delta" value - fixed. All effects use the "graphics.library" directly. All palette files have been fixed. The Amiga 500 Index value of »CBSI« is more accurate. The calibration settings can be stored and retrieved. BMP files are created faster now. MicroPaint requires the new »cxb.library«. Due to a new startup code, MicroPaint no longer crashes if you don't select a screen (previous versions ran in the background, doing nothing but consuming CPU time). You can pick a colour now. You can select the GUI font MicroPaint uses. HAM is supported. The checks for IFF files have been improved. The current brush can be displayed instead of the mouse pointer.

2.0.1 The image can be cropped.

Palette files are saved in a format that can be read by DPaint now.

The ARexx command CRSRSIZE can be used to set brush size and line thickness.

The "asl.library" is called directly to be able to remove the MaximumHeight limit of 24 points.

»CBSI« uses another method for calculating MIPS. MFLOPS are also calculated now.

2.0.2 The current mouse coordinates are optionally displayed in a window.

The event handler of the toolbox had to be rewritten and is faster now.

The "BRUSHPTR" ToolType doesn't make MicroPaint unstable any longer. (Unfortunately, the toolbox cannot be used in conjunction with the "BRUSHPTR" ToolType.)

Brushes are saved with colour palettes now.

MicroPaint uses the "clipboard.device" directly rather than a file called "CLIPS:0" to be compatible with Personal Paint. (Some programs use an incompatible method for acessing the clipboard. In this case, it cannot be used by MicroPaint.)

MicroPaint is about 1 KB shorter due to optimzations.

2.0.3 The colour sliders of the colour editor cold not be used - this has been fixed by simplifying the event handler code.

The new ToolType "FAIL" might be needed if MicroPaint is run on old (slow) Amigas.

2.0.4 The executable file is smaller by using constants instead of functions for boolean values.

Use the "reverse direction" option of the "relief" tool to swap the colours MicroPaint uses to draw the shadow with.

1.20 Copyright & liability

Copyright (C) 1995 - 1999 by FR-SoftWorks.

*** BMP-Support: (c) 1995 by Christopher Jennings, Enchanted Blade Associates.

*** Datatypes-Support: (c) 1994 by Markus Hillenbrand.

The author shall not be liable for any damages caused by the usage of this program.

This program must not be distributed, selled, disassembled, nor translated without the written permission of the author.

You may distribute the demo version of MicroPaint. Ensure that you always distribute the whole archive, however.

Write bug reports and / or suggestions to:

Frank Reibold
Ottberger Weg 13
D-31737 Rinteln

GERMANY

eMail: Peter.Reibold@T-Online.de

SaveBMP docs

DT2IFF docs
Please
register
!

Have fun!

1.21 How to register

Print the "regist_UK.frm" file, fill it out and send it to the author.

The current price is $ 15.00 or something similar to this in another currency.

I will then send you a keyfile which enables the save function.

1.22 DT2IFF Documentation

MicroPaint uses this utility to load datatypes pictures.

Original documentation:

```
------------------------------------------------------------------------------

NAME

        DT2IFF
```

FORMAT

```
DT2IFF INFILE/A,OUTPATH/A,ICON/K,DEFTOOL/K
```

PURPOSE

Converts any picture-datatype to standard IFF-ILBM format.

SPECIFICATION

Since Workbench 3.0 a lot of datatypes are supported. But not all programs can use these data. Thus DT2IFF converts a picture of any type to the standard IFF-ILBM format, so nearly all applications can use the pictures. If the old file has an extension (e.g. .GIF, .JPG), it will be replaced by .IFF, if there is no extension, a new one will be added.
The original files WILL NOT be overridden!

OPTIONS

```
INFILE          This is neccessary.
                Type in the file you want to convert.
OUTPATH         This is neccessary.
                Type in the path where the new picture shall be
                written to. (If you want to have it in the current
                directory you will have to use "")
ICON            This is not neccessary.
                Here you can enter an icon name (without .info at
                the end of the filename) in order to create an
                icon for the new file.
DEFTOOL         This is not neccessary.
                Here you can enter a new default tool for your icon.
                The old one will be overridden.
```

No matter what kind of icon you select for your new picture (drawer, disk, or something else), the new icon will be a project icon.

EXAMPLES

```
1> DT2IFF Pics:Mouse.GIF RAM:
This will convert 'Mouse.GIF'~to 'RAM:Mouse.IFF'

1> DT2IFF Pics:Mouse.JPG RAM: ICON=ENV:def_Picture
This will convert 'Mouse.JPG' to 'RAM:Mouse.IFF'. In addition, a new
icon 'RAM:Mouse.IFF.info' will be created.

1> DT2IFF Pics:Mouse.PCX RAM: ICON=ENV:def_Picture DEFTOOL=PPaint:PPaint
Like example number 2, but the new default tool will be 'PPaint:PPaint'
```

BUGS

NONE.

AUTHOR

Markus Hillenbrand
Weserstrasse 9
36124 Eichenzell-Loeschenrod
Germany

E-mail: m_hillen@informatik.uni-kl.de

COPYRIGHT NOTICE

This man page was written by:

Markus Hillenbrand
Weserstrasse 9
36124 Eichenzell-Loeschenrod
Germany

E-mail: m_hillen@informatik.uni-kl.de

SEE ALSO

ilbm2ppm (UNIX), FileFormats (AmigaDOS)

1.23 Documentation for SaveBMP

MicroPaint uses these sub routines to save .BMP files. The generated files
are not compressed and have 256 colours.

Original documentation:

```
                   +++++++++++++++++;;;;;;;;;;;;;;;;;
                   SaveBMP Routine for Blitz Basic II
                   +++++++++++++++++;;;;;;;;;;;;;;;;;
```

7 Contents 7

7 Abstract 7

The routine "SaveBMP" provided in SaveBMP.src can save a bitmap object in Blitz Basic as a ".BMP" file, the file format used by the OS/2 Presentation Manager and Microsoft Windows 3.x+ (including "Windows ’95").

7 Introduction 7

We all know that the best bitmap graphics editors are on the Amiga (it’s too bad for them that the IBM version of DPaint died at "version 2 enhanced"). Games programmers for other platforms have used Amiga editors for years to great effect (Check the graphics files for the PC version of UFP: Enemy Unknown, for example. You’ll notice they all end in ".LBM", as in IFF ILBM. They can be loaded directly into Amiga graphics editors!)

It is very common for someone who uses both Amiga and PC systems to want to share data or work on a project for one platform with tools from another platform. This is often the case especially with graphical data. Since the Amiga relies upon the IFF ILBM standard and Windoze uses BMPs, it would be useful to have a routine that could create BMP files.

This can cause problems when you want to send over a graphic, though, because virtually no applications in the IBM world support the IFF file format. The statement contained in the file SaveBMP.src does exactly this.

7 SaveBMP 7

The function will take any Blitz Basic "bitmap" object and write out an equivalent "BMP" file. As written, it uses the colour map of the currently used screen or slice for colour data. The calling sequence is as follows:

```
SaveBMP{ofile.s, bmpobj.w, width.w, height.w, numcol.w}
```

ofile	The filename to use for the BMP file which will be written.
bmpobj	Number of the bitmap object to use.
width	Width of the bitmap, or highest x-coordinate to save.
height	Height of the bitmap, or highest y-coordinate to save.
numcol	Number of colours in the bitmap (and palette).

7 Usage Notes 7

[1] You can save only a rectangular area of the region from (0,0) to (x,y) instead of the full image if you use width=x and height=y.

[2] The SaveBMP routine could easily be changed to save any aribtrary rectangle of the image by modifying the horizontal and vertical run loops.

[3] The constant #BMPFH should be set to the number of the Blitz file object that you wish the routine to use when it writes the file. The SaveBMP.src file sets this to ‘0’, but you can change it to any legal value.

[4] The routine does no special conversion for HAM images, so they will not come out properly with this routine alone.

[5] All images are saved as 256 colour uncompressed images; this is highly compatible with Windows, and many newer DOS applications, but tends to greatly increase the file size compared to the equivalent IFF file.

[6] The source code is easy to follow and well-commented, and contains further notes on program operation and BMP file format.

7 IFF Conversion 7

It is especially easy to write an IFF -> BMP conversion program using this routine and Blitz's built-in IFF handling commands. This would be a good project for new Blitz programmers looking for something to increase there programming skills and/or familiarity with the language.

7 Implementation Notes 7

[1] A few words on 12 -> 24-bit colour conversion: Most programs that convert images from 12 -> 24-bit colour data do so incorrectly. In fact, even the IFF standard bungles this task. To see why, you first need to understand a few things about how the colours are stored. Recall that colours are coded as triplets of red, green and blue values. One pre-AGA Amigas, these colours are values from 0-15 (one hexadecimal digit), while on AGA and other 24-bit systems, they are values from 0-255 (two hex digits). So "pure white" on an old system is R=15=$F, G=15=$F, B=15=$F, and on an AGA system R=255=$FF, G=255=$FF, B=255=$FF.

Most conversion software multiples a 12-bit value by 16, which effectively shifts the hex digit to the left, like this: R=240=$F0, G=240=$F0, B=240=$F0. While this simplifies conversion in general, it does not properly duplicate the original colours (colours become darker overall). What we want to do instead of shifting digits is to duplicate the digit, so that $F=$FF, $8=$88, and so on. The way to do this is to multiply by 16 and add the original value again: NewCol = Col * 16 + Col, or, in other words, to multiply by 17 instead of 16.

[2] The code assumes that you were reading and writing with StdIn and StdOut (DefaultInput / DefaultOutput). You will want to adjust this for most applictaions. Sections needing to be changed are commented in the listing.

7 Copyright/Disclaimers 7

Software and documentation written by Christopher Jennings.

Use of or reliance on this software is entirely at the risk of the user.

I hope you enjoy and use this software.
Comments and questions may be directed to:

Christopher Jennings, Soft. Eng. Dept.
Enchanted Blade Associates
R. R. 4 Langton, ON
Canada   N0E 1G0

or to u9303286@muss.cis.mcmaster.ca
or lynnjenn@village.ca

I hope that this is at least somewhat lucid. As usual, it's a last-minute write documentation and do two other things at the same time job.

END OF TEXT

1.24 Image processing with MicroPaint

Yes, it’s right: you can process images with MicroPaint very easily.

This is done using the "Calibration requester" which resides in the "Tools" menu. The following examples may be usefull to you:

The default values are:

```
  [1]   times brighter
  Alter components   [RGB]
  Zero components :=  [0]
  Registers [0] to [max. colours]
```

How to correct a picture which is too dark

Enter the following values:

```
  [1.1]   times brighter
  Alter components   [RGB]
  Zero components :=  [0]
  Registers [0] to [max. colours]
```

This will make the picture 10 1ghter by increasing the RGB values of all colour registers.

To make this undone, enter [-1.1] instead of [1.1]

How to increase the blue component

Enter the following values:

```
  [1.1]   times brighter
  Alter components   [B]
  Zero components :=  [0]
  Registers [0] to [max. colours]
```

As you see, only the [B] (blue) components are increased.

How to increase the yellow component of a picture

Enter the following values:

```
  [1.1]   times brighter
  Alter components   [RG]
  Zero components :=  [0]
  Registers [0] to [max. colours]
```

This is somewhat tricky: yellow is the addition of red and green, so we changed the [R] and [G] components!

How to make the background lighter

Enter the following values:

```
  [1.1]   times brighter
  Alter components   [RGB]
  Zero components :=  [0]
  Registers [0] to [0]
```

As you see, only register 0 (zero) is affected, because this holds the background colour.

How to correct a very dark picture

Enter the following:

```
  [1]   times brighter
  Alter components   [RGB]
  Zero components :=  [3]
  Registers [0] to [max. colours]
```

The colour registers containing the colour "black" are set to a dark grey. Hence, the picture is lighter then before. Other colours are not affected.

How to correct a picture which is too green

```
  [-1.1]   times brighter
  Alter components   [G]
  Zero components :=  [0]
  Registers [0] to [max. colours]
```

The green components are decreased by 10 %.

1.25 Tutorial

We suppose that you can select menu items and that you know how to use a mouse. The required files are in the "Bilder" subdirectory.

Note: If your Amiga does not support the AA chipset, open a new screen prior to opening the picture files. Select a "LoRes Lace" screen with 32 colours via the Project/Format... menu item.

Then open the picture files, answering the question with "No".

*

Starting MicroPaint

Start MicroPaint by double-clicking on its icon. The program asks you to select a screen mode ("hires" is recommended for this tutorial"). Then an "Info" dialogue is displayed which you should cancel by clicking on "Ok".

Opening a picture file

Select the menu item "Project/Open..." an click on the file Tutor1.IFF . This picture shows Deluxe Paint with the image »Tut256.lores« opened and was created using MicroPaint's screen grabber.
Answer the following requester with "Ok", and you will see king Tut on your screen.

Remark: The picture can also be opended by dragging its icon on the AppIcon
        of MicroPaint.

The aim

The picture is somewhat too dark, and DPaints tool bar is annoying. We will corrrect this and convert the picture to Lores-Lace, too.

Changing the screen format

Press the left Amiga key and 'm' to see the Workbench screen. Load a new copy of MicroPaint and select a lores-lace screen (320x400) with 256 colours.
Select "Tools/screen grabber..." and grab the "Unnamed Screen: 320x200x256".
Answer the following requesters with "Yes", "No", and "Yes". MicroPaint will then convert the image.

Note: If your Amiga does not support the AA chipset, select only 32 colours.

Image processing

Select "Tools/Calibrate..." and enter the following values:

```
  [1.3]   times brighter
  Alter components   [RGB]
  Zero components :=  [0]
  Registers [0] to [max. colours]
```

(The image will become 30 % brighter.)

Then enter the following values:

```
  [1.2]   times brighter
  Alter components   [RG]
  Zero components :=  [0]
  Registers [0] to [max. colours]
```

(This will increase the yellow component of the picture by 20 %.)

The picture is nicer now, isn't it?

Getting rid of DPaint's tool bar

Select the menu item "Brush/Cut..." and drag a rectangle around king Tut.
Remember the position of Tutankhamun.

Then select "Project/New..." and answer the requester with "Yes". MicroPaint will clear the screen using the background colour. To put the king on the screen, move the mouse pointer to the remembered position and press the left Amiga key and the left ALT key simultaneously.
The result is a simulation of a mouse click: you used the brush! You should call "Brush/Clear" to delete the brush, because we do not need it any longer.

Saving your masterpiece

Use the "Project/Save as..." command to save the image as »TutAnchAmun.IFF«. You should compare it with Tutor2.IFF using "Project/View".

The end

The tutorial ends here.

You may now improve the picture using MicroPaints paint tools - what about a head line?

1.26 Support and Utilities

Documentation for SaveBMP

Documentation for DT 2 IFF

Documentation for CBSI

Documentation for iff_view

Documentation for playanim

Documentation for grabscreen

1.27 Documentation for grabscreen

This utility must be started via CLI and saves screens to IFF files.

Grabscreen requires a file name as first argument and accepts a screen name as second argument (if no screen name is stated, the current screen will be grabbed).

You can use "?" as file name to get some hints.

1.28 Documentation for iff_view

This utility may be started via Workbench or CLI and displays IFF pictures.

If no arguments are supplied, an ASL requester will allow you to select a file.

To quit the program, hit a mouse button.

1.29 Documentation for playanim

This utility may be started via Workbench or CLI and plays IFF animations.

If no arguments are supplied, an ASL requester will allow you to select a file. A second argument defining the animation speed will be accepted, too. The animation speed defaults to 5.

To quit the program, hit a mouse button.

Only the DPaint III animation format (without colour palette changes) is supported.

1.30 Documentation for CBSI

CBSI is the "Crossbones System Information" utility.

CBSI must be run in the CLI and displays the following if you call it with »?« as argument:

```
Crossbones System Information 1.0.7


Usage: CBSI [drive [buffer length]]
```

CBSI examines your Amiga - for example CPU, FPU, display, operating system, and CPU speed.

Additionally, the transfer speed of your hard disk drive can be measured. The block length defaults to 1 MB (the file created will consist of 8 blocks).

Sample Output

```
Crossbones System Information 1.1.4
Chip RAM:          1.815.184 bytes
Fast RAM:          15.573.136 bytes
Attn Flags:        1000000001111111
  CPU:             Motorola 68040
  FPU:             Motorola 68040 FPU
  Cache:           INST: Cache Burst DATA: Cache Burst CopyBack
VBL Frequency:     50 Hz
Power Supply:      50 Hz
```

```
EClock Frequency:  709.379 Hz
VBR Address:       $0704DBBC
Operating System:  Amiga OS 3.1
  Kickstart:       Version 40.68
  Workbench:       Version 40.42
  SetPatch:        Version 43.6
RevBits:           00011111
  DMA:             AA Alice
  Display:         AA (M)Lisa
  Chip set:        AA
MIPS:              19,13187
  Amiga 500 Index: 27,10358
MFLOPS:            4,799689
Device "hd1:":
Buffer: 1.000.000 bytes
  Write file:      1.046 KB/s
  Read file:       1.208 KB/s
Workbench Screen:  $07043B58
  Format:          640 x 256 x 256
  Mode:            $00029000 - PAL:HighRes
```

1.31 Using the Toolbox

The Toolbox provides an easy access to all paint modes of MicroPaint.

The following commands can also be called via the toolbox:

o the palette window
o the "undo" option

Refer to
Using 'undo'
for details.

Note: The "undo" button is grayed if the "undo" feature has been disabled.

1.32 Using 'undo'

How to enable 'undo'

The "undo" option is activated via the "options" menu.

MicroPaint offers two versions of 'undo':

'undo' version	Remarks
Standard	MicroPaint saves the "undo" buffer to an IFF file. This option is *very* slow, but it requires only a small amount of memory.
Fast "Undo"	The "undo" buffer is kept in a second screen. Thus you

need a lot of graphics memory if you want to use this version of "undo". FASTUNDO is significantly faster than the standard version, but it is not able to undo the effects of the "Tools" menu.

Note: This option has to be chosen via the ToolType FASTUNDO prior to enabling it via the "options" menu.

Example: A "PAL: Superhighres Lace" screen with 1280x512 pixels and 256 colours requires twice 640 KB of graphics memory.

How to use ’undo’

MicroPaint acts like version 1.x if the "undo" option is disabled.

The activation of the "undo" option slows MicroPaint down, because the current screen has to be saved to the "undo" buffer before drawing something or executing a menu command respectively.

To draw an object, perform the following steps:

1. Select a paint mode (optional).
2. Move the mouse pointer to the position desired.
3. CLICK the left mouse button and HOLD it down as long as the busy pointer is displayed on the screen. (The busy pointer indicates that the "undo" buffer is saved.)
4. CONTINUE holding the mouse button down and move the mouse to alter the size or direction of the object.
5. RELEASE the mouse button. MicroPaint creates the graphics object.
6. To undo the result, simply click the "undo" button.

1.33 Effect samples

Original picture

Original

Picture after applying the effects

Revert
Negative
Greyscale
Monochrome
Mosaic
Emboss
Emboss (option ’Keep objects’ enabled)
Emboss (option ’reverse direction’ enabled)

1.34 Index