

Painter

Painter

[Password](#) [Bugs](#) [ToDo](#) [Tips](#)

New in Painter 17

Quick JPEG optimizing tool; Oval Buttons; Clone brushes; Effects.

[Shapes](#) [Color](#) [Text](#) [Buttons](#) [Copy](#) [File](#) [Blocks](#) [Patterns](#) [Pens](#) [Options](#) [Gadgets](#)

Images

[Images and Filters](#) [Ranges](#)

[Backgrounds](#)

[Background Functions](#)

[Filters](#) [Cells](#) [Graph](#) [Fractals](#) [Frames](#) [Buttons](#) [Quick 3D](#) [Painter Whirl](#) [Quick JPEG](#) [Painter Effects](#)

Scripting

[Overview](#) [Language](#)

[Calculations](#) [procedures](#)

Overviews

[Copy Modes](#) [Pen Modes](#) [Pen Styles](#)

Coordinates

Click in the image with the right mouse button to set the target point. When you are painting a shape or text you can use the shift button when clicking the left mouse button. The cursor will then jump to the target point. For some type of painting this can be very convenient. When you hold down the **ctrl** key when clicking in the image, the coordinates will be scripted in the Script Editor.

Display of Coordinates

Introduced in Painter 13, coordinates are displayed when you move over the image. To display the coordinates when you are drawing a shape, you have to hold down the **Alt** key. This is done so that on slower machines the drawing process is not disturbed by the coordinate update.

Painter Shapes

Painter Shapes

You paint a shape by *first* selecting a shape and then click-drag-release with the mouse its position and size.

Some shapes (like Bezier curve, skew rectangle, triangle and the cube) require that you first click-drag-release the starting shape and then click-drag-release to set the final shape.

Clone Brush

With the clone brush you can copy parts of your drawing. Before using the clone brush you set the target by right-clicking at the desired spot. Then you select one of the two clone brush modes: **all colors** or **not target color**. Move with the mouse to an area on the drawing and start painting like you do with the freehand tool. In case of **not target color** all colors will be copied except the target color.

Pen Style

There are two pen styles, *Clear* and *Solid*.

Pen Mode

You can select one of the three pen mode.

Copy, is the default mode. The paint is copied on the surface, like with normal paper.

Xor, every pixel that you paint is bitwise Xor'd with the painting surface. You can create some interesting effects with Xor.

Not Xor, the same as Xor, only this time the result of the Xor is inverted. This will give you a transparency effect.

Line width

Select from one of the four line widths.

Painter Color

Painter Color

Select line and fill color in the color bar on the right side. Click with your left mouse-button to select line color, with the right button to select fill color. The little color flag below the color bar shows the current selection. Left-click or right-click the color flag to set a custom color.

Painter Text

Painter Text

Set *Font...* and *Text...* before applying the text shape.

Press the *T* button to set text shape. Then click in the drawing where the text should be placed.

Shadow Text

Select Shadow shape and then click in the drawing where the shadowed text must be placed. Make sure that the pen style is set to *clear* before using shadow.

The default shadow color is silver. The default offset for x and y is -2. You can set your own shadow color and offset. Return to the default values by clicking Shadow Default.

Rotated Text

Set font and text options before you apply the rotated text tool.

Then select *Text Rotate Text* from the menu. Now click in the painting where the text should start and drag a line indicating the direction of the text. When you release the mouse-button the text will be drawn rotated. As with all different painting tools: when you set shapes to a value higher than 1, Painter will draw multiple text strings. If *shapes* is 4, your text will be drawn 4 times at 90 degree angles (360 /4).

Painter Buttons

Painter Buttons

Create a default (88 pixels wide by 31 pixels high) button with *Make*.

You can set the width, height and color of the button before you create it with *Make*.

Depressed button

Using the Xor and Not Xor pen mode you can create depressed buttons.

Painter Copy

Painter Copy-Select

Before a copy operation select an area by first pressing the select button and then selecting an area with the mouse.

Paste

After having selected an area click the *Paste* button and click in the drawing where you want the copy to be placed.

Note that by holding the shift button you can copy continue when moving the mouse.

Paste Color

Paste only a specific color by first selecting that color with the color picker (pipet) and then clicking the Paste Color button.

Paste Color Inverted

The same as Paste Color only now and Xor operation is applied to the copied color pixels.

Floodfill

Fills a bounded area with a color.

Fill Selection

Fills the selected area with a color.

Painter File

Painter File

You can open the following file formats:

bmp, gif, jpg, ico, wmf, emf

And save in the following formats:

bmp, gif, jpg

Transparent Gif

To save an image as a transparent gif you have to check *Options GIF Image Transparent*. The color of the lower left pixel of your image will be used as the transparent color.

File conversion

You can convert from one format to another format by opening a file and saving it in one of the three output formats.

File New

Enter the new width and height and a new blank image is created.

File New Banner

This will create a blank new image with the "standard" banner size as used so much on the internet.

Image Clear

Click the Clear button to clear the current image.

Image Resize

Click the resize button and enter the new width and height. The current image will be reduced or enlarged accordingly.

Split Save

Introduced in version 9, you can split save an image. This can be of use when you want to use a series of images as "buttons" on a Web-page, and you want the buttons to be matching exactly. Before you split save you have to set the Image Count with *Split Save Image Count*. You can split horizontally with *Split Save Horizontal Split* or vertically with *Split Save Vertical Split*.

When you are asked for a filename give *.gif* or *.jpg* as the file extension. Any other extension will not be accepted.

The splitted images will be saved with the same name as you entered only with a sequence number starting from 0 behind the name. As a bonus, Painter will automatically create an HTML document that aligns these images vertically or horizontally, depending on the choice you made. This HTML document has the same name as the filename you entered, only with a *.htm* extension.

Split Save Puzzle

This will split save your image in an array of images. When you did set *Split Save Image Count* to 4. The image will be saved as a 4*4 array of images. The image names are derived from the name you give in the file save dialogbox with the array index added. If you give *puzzle.gif* as the name, then the first image is saved as *puzzle01.gif*, the last of the 16 images is saved as *puzzle33.gif*. In my WebLab site you can see an example of how these puzzle images can be used to create an on-line puzzle where the user can drag-and-drop the images in their proper position, solving the puzzle.

Print

Introduced in version 9, you can print and print preview an image. Print preview with *File Print....* The Previewer pops up with the image displayed. There are buttons to increase and decrease the size of the image and a printer setup button.

Painter Blocks

Painter Blocks

Painter does not have a zoom facility. So how do you create a 32x32 bit bitmap, or an 8x8 pattern? The answer is: **Blocks**. You can set the number of blocks horizontal and vertical with *Options Blocks*. The number of horizontal and vertical blocks is always the same.

Drawing with Blocks

Select blocks in the shapes dropdown menu on the toolbar. You can now draw freehand with blocks. Make sure you have selected a fill color before or you won't see anything.

Save Blocks

Save your current drawing as a block drawing in bmp format.

Load Blocks

Loads a bmp file as blocks. Basically the file is stretch drawn on the painting canvas.

Painter Patterns

Painter Patterns

You can define up to 16 fill patterns.

Use Patterns

Click a pattern in the pattern bar and then draw a closed shape like a rectangle or ellips. The shape is drawn with the selected pattern. If you want to go back to a solid fill just right-click a fill color in the color bar.

Edit Patterns

You can change an existing pattern by left-click selecting it and then right-click edit. The pattern now fills the image in block format. Select *Blocks* as the drawing shape and modify the pattern using different fill colors. Please note that this works best with the image set to 256x256 pixels (the default).

Update Patterns

After you have changed the pattern you want to update the pattern bar. Left-click the pattern that you want to update and then right-click Update in the pattern bar. The pattern.bar is automatically saved when you close Painter. The file *Patterns.bmp* holds all the 16 patterns. Just experiment.

Please note that you can use any image as a pattern. You do not need to use blocks. Just draw a big circle in blue using the ellips tool and update a pattern. You can get funny effects by painting with a pattern on an image and then create a new pattern from this image. As with all painting: just experiment and also try the XOR and NOT XOR mode in creating patterns.

Painter Pens

Painter Pens

Behind the little brush on the toolbar a whole range of pen modes is hidden. Difficult to explain what each mode will achieve. Just experiment.

Some Tips

Set the line width to large than 5 pixels with *Other Size...* Try 20 pixels. Select **red** as the pen color and *mirror4* as the shape. Now draw your fancy pattern. Change the *pen mode* to for instance *Not Xor* and draw again. See what happens! This is just an example of what you can achieve with Painter.

Painter Options

Painter Options

This menu selection allows you to customize some of the shapes.

Blocks

Sets the number of horizontal and vertical blocks when using the *block* shape.

Line Tolerance

Sets the line tolerance in pixels when using the *straight line* shape. This number determines when to change from vertical to horizontal and the other way round.

Polygon

Toggles between polygon and star mode when using the *star* shape.

Rounding

Sets the rounding in pixels for a rectangle with rounded corners.

Shapes

Sets the number of concentric shapes that will be drawn when using a shape. The number can be from 1 thru 25. The default is 1.

Spirals

Set the *Spiral* options like: *spiral direction* and *number* of spirals.

Starpoints

Sets the number of starpoints used with the *star* shape. The number can be from 3 thru 12. This number is also used with the polygon mode.

SpiroGraph Turns..

Sets the number of turns that the Spirograph shape will make. The default value is 40 and this will do for most cases.

Painter Script

Painter Script

With *Painter Script* you can give a sequence of commands to the painting area, the *Canvas*. You can store often used command scripts in the *Script Library*.

Script Editor

Make the script editor visible with *Script View Editor*.

You can enter command lines in the editor. See [Script Language](#) [Script Calculations](#) and [Script Procedures](#) for details on creating a script. You can also use [Macros](#) to enter scripting lines.

Save Script

Save a script with *Script Save As*.

Open Script

Open an existing script with *Script Open*.

Running a Script

You can run a script with *Script Run* or by pressing the *F9* key.

You can run an item from the script library by selecting the item and the *Script Run Item* or by pressing *shift+F9*.

The Script library

The script library is stored in the money green listbox. You can add scripts to the library, delete and update.

Add to Script Library

Add to the script library by first selecting all the text that you want to add. Then select *Script Add*. You are asked for a script name.

Delete from Script Library

Delete a script from the library by first selecting it in the listbox and then *Script Delete*.

Update a Script in the Library

Update a script by loading it in the editor first (as explained below), do the changes and then select *Script Update*.

Loading from the script library

You can load a script from the library into the editor by double-clicking an entry in the library. The text from the library is inserted at the cursor position.

Script Language

Script Language

This is the first topic on the PainterScript language, see also [Script Calculations](#) and [Script Procedures](#)

Some of you may remember the computer language Logo with its turtle that you could move around the screen, making it turn left and right, having the pen up or down. In many cases shapes can be create much easier with turtle graphics then with the entering of coordinates. That is why I introduced Turtle Graphics into Painter in release 10. The old PainterScript commands are now completely replaced by the new turtle graphics commands.

Script Editing

Script Editing takes place in the syntax highlighted Script Editor. Make the editor visible with *Script View Editor*.

PainterScript in release 10 is no longer line orientated. Everything in de script editor is considered as one long string of characters made up of symbols (commands, numbers, text etc). Symbols are seperated by spaces. The only time that spaces are included is in text commands when they are placed between double quotation marks.

Multiple statements

You can enter multiple script statements on one line by seperating them with a space. In the script editor you can put as many statements behind each other as you want. Painter converts carriage returns in a double space and tabs in a single space. You may want to use tabs and carriage returns just for visual clarity.

Repeat statements

With the new release of PainterScript you can repeat blocks of code with the *do loop* statement. You may nest 128 *do...loops*, more than you will ever need.

PainterScript Commands

Below you find a summary of all the commands. Please use spaces to seperate commands and values.

default

Sets default values. It is the same as:

pos 0 0 angle 0 up pencolor cblack brushcolor clwhite copymode srcrcopy area 0 0 0.

Use it at the beginning of your script to start with defined conditions.

pos

Sets the position of the turtle.

Syntax: *pos x y*

where x and y are integers specifying the absolute position.

move

Moves the position of the turtle

Syntax: *move dx dy*

where dx and dy are integers specifying the relative displacement.

lineto

Draws a line from the current position to a relative point given by the dx and dy parameters.

Syntax *lineto dx dy*

where dx and dy are integers specifying the relative displacement.

angle

Sets the angle or heading of the turtle.

Syntax: *angle Aangle*

where *Aangle* is a positive or negative integer number.

down

Sets the pen down.

Syntax *down*

up

Lifts the pen up.

Syntax *up*

pensize

Sets the thickness of the pen.

Syntax *pensize Asize*

where *Asize* is an integer number.

pencolor

Sets the color of the pen.

Syntax *pencolor Acolor*

where *Acolor* is a valid color name or integer hex value.

Hex values start with the \$ character, e.g: \$ff00cd

Valid color names are: claqua, clblack, clblue, clgray, clfuchsia, clgreen, cllime, clmaroon, clnavy, clolive, clpurple, clred, clsilver, cleteal, clwhite and clyellow.

addpencolor

Add color to the current pen color

Syntax *addpencolor Acolor*

where *Acolor* is a valid color name or a valid (hexadecimal) integer.

left

Turns the turtle left.

Syntax *left Aangle*

where *Aangle* is an integer number.

right

Turns the turtle right.

Syntax *right Aangle*

where *Aangle* is an integer number.

turn

Turns the turtle left (positive angle) or right (negative angle)

Syntax *turn Aangle*

where *Aangle* is an integer number.

go

Moves the turtle. If the pen is down, a line is drawn.

Syntax *go distance*

Where *distance* is an integer number. If *distance* is a positive number then the turtle moves forward, in case of a neagative *distance* the turtle moves backward.

goleft

Moves the turtle to the left edge of the image.

goright

Moves the turtle to the right edge of the image.

gotop

Moves the turtle to the top edge of the image.

gobottom

Moves the turtle to the bottom edge of the image.

gocenter

Moves the turtle to the center of the image.

textfont

Sets the font name of the drawing canvas.

Syntax *textfont Afontname*

where *Afontname* is a valid font name like "arial".

Please note that in release 11 of Painter the syntax for *textfont* has changed. You must now enclose the *fontname* with double quotation marks. In the previous release you could not set a fontname like "Courier New".

textsize

Sets the text size of the drawing canvas.

Syntax *textsize Asize*

where *Asize* is an integer number.

textcolor

Sets the text color of the drawing canvas.

Syntax *textcolor Acolor*

where *Acolor* is a valid color name or color integer. See above for valid color names.

text

Draws text on the canvas in the current font name, size, style and color and in the current direction of the turtle. This makes it very easy to draw rotated text under script control!

Syntax *text Atext*

where *Atext* is the text you want to draw. You must surround the whole text with double quotation marks. e.g "Jan Verhoeven".

textout

Draws straight text on the canvas, independent of the current angle.

Syntax *textout Atext*

where *Atext* is the the text to draw between double quotation marks.

bold

Sets the text style of the canvas to bold.

italic

Sets the text style of the canvas to italic.

underline

Sets the text style of the canvas to underline.

normal

Sets the text style of the canvas to normal.

bssolid

Sets the brush style to solid

bsclear

Sets the brush style to clear.

Please note that when you are working with text you usually set the brush style to clear first.

brushcolor

Sets the color of the brush.

Syntax *brushcolor Acolor*

where *Acolor* is a valid color name or color integer. See above for valid color names.

addbrushcolor

Add color to the current brush color.

Syntax *addbrushcolor Acolor*

where *Acolor* is one of the valid color names or a valid (hexadecimal) integer.

rectangle

Draws a rectangle on the canvas.

Syntax *rectangle x y*

where *x* and *y* are integer numbers.

A rectangle is drawn using the position of the turtle as one corner and the relative point given with *x y* as the opposite corner.

To draw a rectangle, 15 wide and 10 high from the current position you would enter **rect 15 10**.

Depending on the brush style the rectangle is filled with the brush color or left clear.

roundrect

Draws a rectangle with rounded corners on the canvas.

Syntax *roundrect x y rx ry*

where *x*, *y*, *rx* and *ry* are integer numbers.

A rounded rectangle is drawn using the position of the turtle as one corner and the relative point given with *x y* as the opposite corner.

The curve of the rounded corners matches the curvature of an ellipse with width *rx* and height *ry*.

ellipse

Draws an ellipse on the canvas.

Syntax *ellipse x y*

where *x* and *y* are integer numbers.

An ellipse is drawn within an imaginary bounding rectangle using the position of the turtle as one corner and the relative point given with *x y* as the opposite corner.

Depending on the brush style the ellipse is filled with the brush color or left clear.

diamond

Draws a diamond on the canvas in the current direction.

Syntax *diamond size*

where *size* is the size of the diamond.

polygon

Draws a polygon on the canvas in the current direction.

Syntax *polygon sides size*

where *sides* is the number of sides, and *size* is the length of a side. Both *sides* and *size* must be non-zero.

star

Draws a star on the canvas in the current direction.

Syntax *star sides size*

where *sides* is the number of sides, and *size* is the length of a side. Both *sides* and *size* must be non-zero.

curve

Draws a smooth curve on the canvas.

Syntax *curve pt1 pt2 pt3*

where *pt1*, *pt2* and *pt3* are control points of the curve.

Each point has an x and a y value. These values are taken relative to the current position.

The curve is drawn from the current position to *pt3*, using *pt1* and *pt2* as control points for the curvature.

Example:

curve 20 10 40 -10 60 0

where 20 10 are the x and y coordinates of *pt1* etc.

Placing the *curve* in a loop allows you to create waves. See the *Waves* example Script.

mark

Marks the current position.

gomark

Moves the pen to the previously marked position. A line is drawn when the pen is down.

markangle

Marks the current angle.

gomarkangle

Resets the angle to the previously marked value.

penmode

Sets the pen mode.

Syntax *penmode Apenmode*

where *Apenmode* is one of the valid penmodes.

See [Overview Pen Modes](#). Please note that in PainterScript you leave away the *pm*

To set the penmode to *pmXor* you would script:

penmode xor.

copymode

Sets the copy mode.

Syntax *copymode Acopymode*

where *Acopymode* is one of the valid copy modes.

See [Overview Copy Modes](#).

Please note that in PainterScript you leave away the *cm*. To set the copy mode to *cmSrcCopy* you would script:

copymode srccopy.

area

Sets an rectangular area on the canvas for subsequent use in copying.

Syntax *area x1 y1 x2 y2*

where *x1* is left, *y1* is top, *x2* is right and *y2* is bottom of the rectangle.

copy

Copies the *area* to the current position.

Syntax *copy*

do... loop

Create a loop in the script.

Syntax *do Anumber ... any number of statements... loop*

where *Anumber* is an integer number specifying the number of times the block of code must be repeated.

Example:

do 4 go 30 right 90 loop

This will draw a square, provided you have set the pen *down*.

flood

Floods an area with the current brush color up to the color border of the area.

Syntax *flood x y*

where *x* and *y* are offset values. When you enter 0 for both *x* and *y* the flooding will start from the current position. When you enter *flood 1 1*, the flooding will start at a point one right and one down from the current position. The offset is convenient if you have just drawn a shape and you want to flood it without moving the current point.

background

Draws one of the named backgrounds.

Syntax *background "kings"*

where *kings* is a named background (as is the case with the samples that come with Painter). If the name is not found in the background library then the statement is ignored. You will not get an error message. The background name must be placed between double quotation marks.

filter

Applies one of the name filters.

Syntax *filter name*

where *name* is one of the valid filter names:

trace, emboss or invert.

The filter name must be placed between double quotation marks.

Script Calculations

Script Calculations

In PainterScript you can use calculated values using the postfix notation. Values are pushed on and popped from a stack. This gives you much freedom but requires precaution.

Numbers

Numbers are pushed on the stack just by entering them in the script. When Painter interpretes the script it first checks if a word is a valid command, then it checks if it is a valid integer. If the word is a valid integer, the integer is pushed on the stack.

Variables

Variables are introduced in release 12 of Painter. The creation of a variable is simple:

40 in bike

where 40 is the mandatory default value and **in** is the command that creates a *variable* of the next token, assigning the default value. In this case the variable **bike** is created. A variable name shall not contain spaces, underscores are allowed.

To change the value of an existing variable you use the same syntax:

67 in bike

When Painter excecutes the **in** command it checks if the variable allready exists, if it does the new value is assigned, if it does not exist it is created.

Using a variable in calculations is simple:

50 in bike down go bike

This will assign 50 to **bike**, set the pen *down* and *go* the value of *bike*. You can use a variable name anywhere where you can use a number.

When you use a variable on its own its value will be pushed on the stack, the same as with numbers. This means you can also do calculations with variables:

40 in bike 50 in car bike car + down go =

Variable Operators

Painter Script has 6 variable operators that directly act on variables.

inadd

Will add the number on the stack to the variable.

Example **50 inadd avariable**

insub

Will subtract the number on the stack from the variable.

Example **50 insub avariable**

inmul

Will multiply the variable by the number on the stack.

Example **50 inmul avariable**

indiv

Will divide the variable by the number on the stack.

Example **50 indiv avariable.**

If the number on the stack is zero, an error message is generated.

ininc

Will increment a variable by 1.

Example **ininc avariable**

indec

Will decrement a variable by 1.

Example **indec avariable**

Colors

Colors are pushed on the stack the same way as numbers, just by including their hex value (e.g \$FF0A89) or their color name (e.g clblue) in the script.

Besides *numbers* and *colors* there are several attributes that you can push on the stack. They are summarized below.

=posx

Pushes the x part of the current position on the stack.

=posy

Pushes the y part of the current position on the stack.

=pencolor

Push pencolor on stack.

=pensize

Push the pensize on stack.

=textcolor

Push the textcolor on the stack.

=textsize

Push the textsize on the stack.

=brushcolor

Push brushcolor on stack.

=angle

Push current angle on stack.

=markx

Push the x part of mark on stack.

=marky

Push the y part of mark on stack.

=loop

Push the current value of the loop counter on stack.

=left

Push the image left value on stack.

=top

Push the image top value on stack.

=right

Push the image right value on stack.

=bottom

Push the image bottom value on stack.

Arithmetic

You can use the usual arithmetic operators + - * and / in postfix notation.

To calculate 3+4 you would enter 3 4 +

To calculate 3/4 you would enter 3 4 / etc.

In this way you do not need brackets.

abs

Changes top of stack number to its absolute value.

neg

Negates (changes the sign) of the top of stack number.

swap

Swaps the two top of stack numbers with each other.

max

Takes the two top of stack numbers and puts the maximum back.

min

Takes the two top of stack numbers and puts the minimum back.

sqr

Takes the top of stack number and puts the squared number back.

sqrt

Takes the top of stack number and puts the square root back. An error will result when the number is 0.

inc

Increments the top of stack number by 1.

dec

Decrements the top of stack number by 1.

Using stack values

Using a stack value is easy, just put the = where you would otherwise enter an integer or a color value.

Example:

=bottom =top - 2 / =right =left - 2 / pos = =

will put the pen in the center of the image. There is of course the *gocenter* command, but this is just an example.

dup and drop

dup will duplicate the top of stack value. This is useful when you want to keep a value for later use.

drop will drop the top of stack value from the stack.

e.g 23 drop has the net effect as if nothing happened.

Comparisons

You can do the following comparisons on the stack:

.gt

Greater than, e.g. 4 5 .gt will return false.

.ge

Greater or equals, e.g. **6 6 .ge** will return true.

.eq

Equals, e.g. **6 5** will return false.

.lt

Less than, e.g. **4 10 .lt** will return true.

.le

Less or equals, e.g. **4 4** will return true.

Logical operators

The following logical stack operators are available.

.and

Will return true if both top of stack values are true. A value is true when it is not zero.

.or

Will return true when one of the top of stack values is true.

.not

Will negate the top of stack value. If it was non-zero it will become true (1) , of it was true it will become false (0).

Script Procedures

Script Procedures

In release 10 of Painter the use of Script procedures was introduced. Procedures are name routines that you can call from anywhere in the script. They make your script more compact, because you can re-use code, and also much easier to read because you can use a descriptive name.

Defining a procedure

You define a procedure by placing code between square brackets and adding a name after the opening bracket. The brackets shall have a space in front and behind.

example:

```
[ myshape up pos 10 10 down go 40 right 90 go 25 ]
```

This is not a very usefull example but just illustrates the syntax of a procedure.

Using a procedure

You call a procedure by entering its name in the scrip.

example:

```
up pos 100 100 myshape go 50 myshape
```

Conditional procedure

You can conditionally call a procedure using the **if** statements after some logic comparisons (see [Script Calculations](#)).

If the top of stack is true the procedure name following the **if** will be executed, otherwise it will be skipped.

Example:

```
3 5 .eq if myshape go 50
```

In this case *myshape* will not be executed since 3 is not equal to 5.

Overview Copy Modes

Overview Copy Modes

When you copy a range you have a choice of various copy modes. Selecting the right copy mode sometimes means that you can achieve an effect just in one go. Find the various modes and their operation summarized below:

Value	Meaning
cmBlackness	Fills the destination rectangle on the canvas with black.
cmDstInvert	Inverts the image on the canvas and ignores the source.
cmMergeCopy	Combines the image on the canvas and the source bitmap by using the Boolean AND operator.
cmMergePaint	Combines the inverted source bitmap with the image on the canvas by using the Boolean OR operator.
cmNotSrcCopy	Copies the inverted source bitmap to the canvas.
cmNotSrcErase	Combines the image on the canvas and the source bitmap by using the Boolean OR operator, and inverts the result.
cmPatCopy	Copies the source

pattern to the canvas.

cmPatInvert Combines the source pattern with the image on the canvas using the Boolean XOR operator

cmPatPaint Combines the inverted source bitmap with the source pattern by using the Boolean OR operator. Combines the result of this operation with the image on the canvas by using the Boolean OR operator.

cmSrcAnd Combines the image on the canvas and source bitmap by using the Boolean AND operator.

cmSrcCopy Copies the source bitmap to the canvas.

cmSrcErase Inverts the image on the canvas and combines the result with the source bitmap by using the Boolean AND operator.

cmSrcInvert Combines the image on the canvas and the source bitmap by using the Boolean XOR operator.

cmSrcPaint Combines the image on the canvas and the source bitmap by using the Boolean OR operator.

cmWhiteness Fills the destination rectangle on the canvas with white.

Overview Pen Modes

Overview Pen Modes

Mode Pixel color

pmBlack Always black

pmWhite Always white

pmNop Unchanged

pmNot Inverse of canvas background color

pmCopy Pen color specified in Color property

pmNotCopy Inverse of pen color

pmMergePenNot Combination of pen color and inverse of canvas background

pmMaskPenNot Combination of colors common to both pen and inverse of canvas background.

pmMergeNotPen Combination of canvas background color and inverse of pen color

pmMaskNotPen Combination of colors common to both canvas background and inverse of pen

pmMerge Combination of pen color and canvas background color

pmNotMerge Inverse of pmMerge: combination of pen color and canvas background color

pmMask Combination of colors common to both pen and canvas background

pmNotMask Inverse of pmMask: combination of colors common to both pen and canvas background

pmXor Combination of colors in either pen or canvas background, but not both

pmNotXor Inverse of pmXor: combination of colors in either pen or canvas background, but not both

Overview Pen Styles

Overview Pen Styles

psSolid A solid line.

psDash A line made up of a series of dashes

psDot A line made up of a series of dots

psDashDot A line made up of alternating dashes and dots

psDashDotDot A line made up of a series of dash-dot-dot combinations

psClear No line is drawn (used to omit the line around shapes that draw an outline using the current pen).

Scripting Macros

Scripting Macros

Macros allow you to translate mouse actions in script text. For this to happen two things are required:

1. The Script editor must be visible
2. The Alt-key must be depressed

You can use macros with the following shapes:

- Line
- Rectangle
- RoundRect
- Ellipse
- Bezier (curve)

When you release the mouse button you will see how the correct scripting instructions are inserted in the script editor.

Using scripting macros allows you to create an optimized image, combining the precise position power of scripting with the ease of entry of the mouse.

Enter coordinates

Typing a script involves entering coordinates. To help you in entering coordinates you can ctrl-click the left mouse-button on a spot in the image. The coordinates of that spot will be inserted in the script.

X and Y in the syntax descriptions below are integer coordinate values.

Painter Images

Images

A whole range of functions is available under the *Image* menu.

Quick Background

With this you can create quickly fantastic backgrounds.

See [Painter Backgrounds](#) for details.

Image Quick Filter..

Allows you to transform the image in a variety of ways. Some effects are impossible to achieve manually. Here is where a filter comes in. See [Painter Filters](#) for details.

Image Seamless Pattern

When you create a background image for your web-site, you have the problem that when the image is tiled, you get nasty seams. With *Image Seamless Pattern* your image is modified so that you will get a Seamless Pattern when viewed with a browser.

Image Trace

This will trace color borders in your image with a black line. The trace function in Painter is superfast.

Introduced in Painter 16, the trace function has been improved. You can now set the **color** byte and the **intensity** of the trace. By using a right combination of color byte and intensity you can create a **glow** effect.

Image Trace Range

Introduced in Painter 16, you can limit the tracing to the selected range. In the [Text Outline Tip](#) you can find an example of how to work with Trace Ranges.

Image Crop

This will crop the image to the range that you have selected. Ideal for cutting away unwanted parts.

Image Invert

This will, as the name suggests, invert the colors of your image. Inverting a second time brings back the original image.

Image Magnify and Reduce

This will magnify or reduce the size of the image with the factor indicated.

Image Move

Allows you to move the whole image up, left, right or down, one pixel at the time. The line that disappears on one side will re-appear on the opposite side. As you will see in the menu you can use short-cut keys for moving the image.

Image Rotate

Introduced in Painter 13 you can rotate an image with any angle. To rotate an Image press **ctrl+alt+R**. This will rotate the image with the preset angle of 90 degrees. You can change this angle with *Image Rotation Angle...*

Image Flip right or down

This will flip the image over the horizontal or vertical axis.

Image Mirror

This will mirror the top, left, right or bottom part of the image. You can also mirror the top left quadrant to the other three quadrants. Use the *image flip* function to get the right quadrant in the top left corner.

Image Colors

This will roll and/ or exchange the *Red, Green and Blue* pixels of the image. It retains the form but changes the color.

Image Mask

Masks are ideal for clipping a portion of the image. You can create a mask from the current image with *Image Mask from Image* or you can load a mask from disk with *Image Mask from File*. See [Painter Ranges](#) for more details.

Image Projection

Introduced in Painter 16, you can project the image on to a Globe with *Image Projection Globe*.

There is a factor that you can adjust between 0.1 and 1.0 to achieve different effects. You can also check *Sinus* mode. If *Sinus* mode is not checked then the *Cosinus* mode will be used. Just experiment yourself.

Painter Ranges

Painter Ranges

Select a rectangular range with the *range* select tool. After you have selected a range you can copy it in various ways or apply range operations to it.

Transitions

Select *Range Transition* to get at the various transitions options. You can set the *start end* color of a transition, the number of *transitions bands* and the *xstep* and *ystep* that is taken with each band.

Pixel operations

You can *and* and *or* the pixels of a range with a color value.

Masks

Three masks are available: Squares, Oval and Diamond. set the copy mode before you apply a mask to a range. The result depends on the copy mode that you have set. Use *Merge Paint* for a normal mask operation.

User Masks

You can also create a mask yourself. Either by selecting *Image Mask from Image* or *Image Mask from File*.

Please note that the black areas in your mask will act as erasers when applied. A mask from file must be a *.bmp* image.

An easy way of starting a mask is by clearing the drawing surface and then select *Image Invert*. You now have a black surface. Select *white* as pen or brush color and create your mask shape. Specially with the symmetrical drawing tool you can create beautiful masks. Just be sure to create a closed area and then use *floodfill* to fill the shape with white.

Remember to set the *copymode* to **Merge Paint** before you use a mask.

Range Tile

Select a *range* and then select *Range Tile* from the menu. The complete image will now be *tiled* with the selected range.

Range Border

Select a *range* and then select either *Range Border Vertical* or *Range Border Horizontal* from the menu. Now a vertical or horizontal border will be drawn on the image using the selected range.

Range Mirror

Select a *range* and then select *Range Mirror* from the menu. The selected range will now be mirrored down.

Copy Transparent

Select a *range* and then select *Copy Transparent Copy*. Now click on the canvas where you want to place a transparent copy of the range. A transparent copy will be placed when you release the left mouse button. The *Transparent color* in the range will be replaced by the current *brush color* of the canvas. Most of the time you will set the *brush mode* to *clear* before using *Copy Transparent*.

Select *Color* if you want to change the default (white) transparent color.

Painter Bugs

Painter bugs

No program is 100% perfect. Like all the programs that I publish, Painter is part of my on-going exercise: Learn to program in Delphi. This means that for some problems I have no direct answer.

When nothing happens after a command

If you start Painter and select *Image Invert* nothing happens. The same is true with various other operations. This has to do with painting on the canvas or painting on a bitmap. When something like this happens to you most of the time the following works:

Select the line tool and just click in the drawing area. Then select the previous command again.

Filters

At the moment I am still working with my 7 years old computer that only can display 256 colors. This means that I have no idea how painter behaves with 16 bit or 24 bit colors. As my present computer is due to be replaced by a state-of-the-art-one, I can modify Painter with more options.

Password

Password

This program is freeware but in order for me, the writer, to have any idea how many people are using my programs, every program needs a password. This password is completely free and you only have to enter it once.

Please take care when entering the password. Type it exactly as given, that is: lowercase and no spaces!.

Where do I get it?

You can get the password for all my programs (current and previous versions) on the following web-page:

<http://jans.hypermart.net/passwords.htm> or on one of the sites given in the password entry screen.

Contact the author

If you have any problems or find any bugs you may contact me at by email at:

jan1.verhoeven@wxs.nl

But please try by yourself first, since I get many emails every day and there is a life besides programming as well!

Painter Backgrounds

Painter Backgrounds

Image Backgrounds are a quick way of producing a starter that would be very difficult to create otherwise. Select *Image Quick Background* from the main menu to display the *Background tool*.

Background Functions

To help you in creating Backgrounds a small understanding of the used functions is useful, although you can also try and click. See [Background Functions](#) for a summary of the available functions.

Use

Using Quick backgrounds is easy. First select the *red, green or blue* radio button. Then select a function from the list. This function will be attached to the selected color. At the same time Painter will show the result. As explained in [Background Functions](#), Painter adds a color offset to the result of a function. You adjust these offsets with the color trackerbars. Some functions use a factor that you can set with the factor trackerbar (the rightmost bar).

Background Library

Users of my programs may know by now that I like the use of libraires in programs. This makes it easy to save something for use later on. You can save the current settings of the *Quick Background* tool by right-clicking *Add Backdrop* over the library dropdown list. Enter a descriptive name and press enter or click OK. The Backgrounds are automatically saved (and loaded when you start Painter) in the *PainterQB.txt* file. Although this is a very simple file, I would suggest not to change this file manually, but always within Painter. Just to avoid errors.

Update a library item by first selecting it from the dropdown list. Then you modify the settings and finally you right-click *Update backdrop* over the dropdown list. You are asked for a name, with the present name given as default.

Special Trick

After you have created a quick background try *Image Filter Trace*, to get a stained glass effect.

Preview

Introduced in Painter 16 you can check the *Preview* box in the Background tool. Painter will display the background in a preview window. If you want to use the background, just click the **Use** button on the preview window.

Please note that the preview images is copied to your drawing using the current copy mode. You may change the copy mode before using the preview image to obtain special effects like merging etc.

Background Functions

Overview of the Background functions

When Painter creates a quick background it cycles through all the pixels starting with the top left pixel and ending with the bottom right pixel. Every pixel is made up of three colors: red, green and blue. You can assign a separate function to each color. In the outer loop the lines are scanned, top to bottom. In the inner loop the pixels are scanned left to right.

Example

If you select for *red* the *Prod* function from the list then Painter will apply the *BGProd* function (see below) to the *red* part of every pixel. *ImgDrawFactor* in the formula is the rightmost tracker on the *Quick backdrop tool*.

To the result of the function a color offset is added. To all red parts the *Red* offset is added. You adjust the offsets with the *red, green and blue* trackerbars.

```
function BGProd(inner,outer:Integer):integer;
begin
result:=outer*inner mod ImgDrawFactor;
end;
```

```
function BGSum(inner,outer:Integer):integer;
begin
result:=(outer+inner) mod ImgDrawFactor;
end;
```

```
function BGSub(inner,outer:Integer):integer;
begin
result:=(outer - inner) mod ImgDrawFactor;
end;
```

```
function BGXor(inner,outer:Integer):integer;
begin
result:=(outer xor inner) mod ImgDrawFactor;
end;
```

```
function BGAnd(inner,outer:Integer):integer;
begin
result:=(outer and inner) mod ImgDrawFactor;
end;
```

```
function BGOuXor(inner,outer:Integer):integer;
begin
result:=outer xor ImgDrawFactor;
end;
```

```
function BGInXor(inner,outer:Integer):integer;
begin
result:=inner xor ImgDrawFactor;
end;
```

```
function BGOuAnd(inner,outer:Integer):integer;
begin
result:=outer and ImgDrawFactor;
end;
```

```
function BGInAnd(inner,outer:Integer):integer;
begin
result:=inner and ImgDrawFactor;
end;
```

```
function BGOutMod(inner,outer:Integer):integer;
begin
result:=outer mod ImgDrawFactor;
end;
```

```
function BGInMod(inner,outer:Integer):integer;
begin
result:=inner mod ImgDrawFactor;
end;
```

```
function BGProdXor(inner,outer:Integer):integer;
begin
result:=(outer*inner) xor ImgDrawFactor;
end;
```

```
function BGSumXor(inner,outer:Integer):integer;
begin
result:=(outer+inner) xor ImgDrawFactor;
end;
```

```
function BGSubXor(inner,outer:Integer):integer;
begin
result:=(outer-inner) xor ImgDrawFactor;
end;
```

```
function BGProdAnd(inner,outer:Integer):integer;
begin
result:=(outer*inner) and ImgDrawFactor;
end;
```

```
function BGSumAnd(inner,outer:Integer):integer;
begin
result:=(outer+inner) and ImgDrawFactor;
end;
```

```
function BGSubAnd(inner,outer:Integer):integer;
begin
result:=(outer-inner) and ImgDrawFactor;
end;
```

```
function BGInner(inner,outer:Integer):integer;
begin
result:=inner;
end;
```

```
function BGOuter(inner,outer:Integer):integer;
begin
result:=outer;
end;
```

```
function BGOuRed(inner,outer:Integer):integer;
begin
result:=QBDRed * outer;
end;
```

```
function BGIInRed(inner,outer:Integer):integer;
begin
result:=QBDRed * inner;
end;
```

```
function BGOuGreen(inner,outer:Integer):integer;
begin
result:=QBDGreen * outer;
end;
```

```
function BGIInGreen(inner,outer:Integer):integer;
begin
result:=QBDGreen * inner;
end;
```

```
function BGOuBlue(inner,outer:Integer):integer;
begin
result:=QBDBlue * outer;
end;
```

```
function BGIInBlue(inner,outer:Integer):integer;
begin
result:=QBDBlue * inner;
end;
```

```
function BGIInModOut(inner,outer:Integer):integer;
begin
if outer<ImgDrawFactor then
  outer:=ImgDrawFactor;
result:=inner mod outer;
end;
```

```
function BGOuModIn(inner,outer:Integer):integer;
begin
if inner<ImgDrawFactor then
  inner:=ImgDrawFactor;
result:=outer mod inner;
end;
```

Painter ToDo

Painter ToDo

With a painting or drawing program you are never finished. One thing is important for me: the size of the program should stay small so that it always fits on a floppy together with the library files and some sample drawings.

Some things on my wish list are:

- Layers
- Support for the png drawing format
- Improved selection

Painter Filters

Painter Filters

Quick Filters are introduced in Painter 11.

Open the filter tool with *Image Quick Filter...* You can select from a variety of basic filter functions and also create and name your own filters.

Select a Filter

Select a filter from the list and click the long button to apply the filter to the image. If you want you can use the 24 bit filter format, where you can set individual filter functions for **Red, Green and Blue**. To set a filter function for a specific color you first select the color and then you select a function from the list. The slider works as an offset with some filter functions. Just experiment.

User Filter

One filter function is called **UserMatrix**. This filter uses the matrix values from the *UserMatrix* table. You can set your own values.

The *UserMatrix* works as follows:

The value of a pixel (the center one in the table) is calculated by multiplying the pixel and its surrounding pixels with the constants from the usermatrix, adding these products, dividing the sum by the Divisor and finally adding the Bias. Maybe this sounds complicated but just try and experiment.

If you like a particular *UserMatrix* you can save the matrix for later use in the *Filter Library*. Right-click *Add* over the green dropdown list and enter a descriptive name when prompted. You can also later on *Delete* and *Update* entries in the *Filter Library*. The filter library is automatically saved in the file *PainterFL.txt*.

Preview

Introduced in Painter 16 you can check the *Preview* box in the Filter tool. Painter will display the background in a preview window. If you want to use the background, just click the **Use** button on the preview window.

Please note that the preview images is copied to your drawing using the current copy mode. You may change the copy mode before using the preview image to obtain special effects like merging etc.

Painter Cells

Painter Cells

Cells are introduced in Painter 12.

To use *Quick Cells* select *Image Quick Cells* from the main menu to display the *Cells* tool.

What are Cells?

Cells are images created by applying a rule to a starting set of pixels. In *Quick Cells* a starting pattern is set whenever you click something in the *Cells* tool or when you change the position of a slider. This gives you instant feedback. Cells are introduced because with them you can generate visual effects difficult to achieve otherwise.

Setting the Pattern

Use the *Checklist* to set pattern pixels 0 to 7. Use the horizontal slider at the top to determine where on the top line of the image the 8 pixels pattern must begin.

Selecting a Rule

Select a cell generating rule by clicking one of the pat0 thru pat8 entries in the rules list. I have not given the rules a name because they can behave very different depending on your starting image.

Cell colors

When applying a rule, *Quick Cells* uses both a rule=true color and a rule=false color. The default *true* color is *white*, the default *false* color is black. You can modify the two colors by adjusting the vertical sliders. Just experiment.

Random

When you check *Random* then extra random pixels will be generated at the top line, thus changing the generated cells pattern.

Split

Normally the cells are generated from top to bottom. When *Split* is checked the cells are generated in a top-down, bottom-up mirror style.

Preview

Introduced in Painter 16 you can check the *Preview* box in the *Cells* tool. Painter will display the background in a preview window. If you want to use the background, just click the **Use** button on the preview window.

Please note that the preview images is copied to your drawing using the current copy mode. You may change the copy mode before using the preview image to obtain special effects like merging etc.

Painter Graph

Painter Graph

Painter Graph is introduced in Painter 12.

To use Painter Graph select *Image Quick Graph* from the main menu. Create a Graph in the *Graph* module, click the *Copy* button and in the main *Painter* form select *Edit Paste from Clipboard* to get the *Graph* as an editable image. To learn more about creating Graphs click the *Help* button in the *Graph* module.

Painter Fractals

Painter Fractals

Introduced in Painter 13 you can create Fractal images, which are a graphical representation of a complex number equation.

Creating a Fractal

To create a fractal image select *Image Quick Fractals...* from the main menu. This will popup the *Fractals Tool*. Just play with various numbers for **X0,Y0,X1,Y1** and the number of **Iterations** and **Colors** to see what happens. Click the *Show* button when you want the image displayed.

Default Settings

Click the *Default* button to restore the default settings.

Fractal Library

Similar to QuickBackgrounds and QuickFilters you can save settings in a Library. If you like a certain Fractal Image just right-click *Add* over the library dropdown list, enter a descriptive name and click OK. To *Delete* an entry from the Library you first have to select it, then you right-click *Delete*.

To *Change* an entry you first select it, change the settings and then you right-click *Update*. You use the Library by selecting an entry and then click the *Show* button.

Painter Gadgets

Painter Gadgets

Gadgets are free floating objects that you can drag around the screen, change the properties like color, size and text, and which you can stamp on to your drawing.

Gadgets are introduced in Painter 14.

Currently there are three types of Gadgets **Image** , **Text** and **Button**.

Image Gadget

Painter has 1 Image gadget that you can freely move. You can Display/Hide the image gadget with *Gadgets Image GadImage*.

Image Gadget Load Image...

With this context menu option you can load an existing *bmp, gif or jpg* image in the gadget.

Image Gadget Load Range

Load a copy of the currently selected range in the gadget.

Image Gadget Load Quick Button

Load a copy of the Quick Button in the gadget. See [Quick Buttons](#) for details on creating a quick button.

Image Gadget Save As...

Introduced in Painter 16 you can save the Image Gadget as a **bmp, gif or jpg** image.

Image Gadget Paint

Paints a copy of the gadget at the current location. Use click-drag-release on the Image Gadget to move it to the desired position at your drawing before Painting the gadget.

Image Gadget Transparency

Introduced in Painter 16 you can make your Image Gadget transparent. To make an image gadget transparent, right-click *transparent*. To make the gadget opaque, just right-click *transparent* a second time.

Initially when you set an Image Gadget to transparent, the color of the bottom left-most pixel is used as the transparent color. You can set any color of the Image Gadget as the transparent color by right-clicking *Capture Transparent Color* at the right spot on the Image Gadget.

There are 3 Image Gadgets. Before you can set a second Image Gadget as transparent, you have to make the one that is currently transparent: opaque. There can only be one Image Gadget in transparent mode at the time. This has to do with the fact that the palette of a transparent Image Gadget is linked to the palette of your drawing.

Text Gadgets

Painter has 3 *Text Gadgets* that you can manipulate individually. You can display/hide a text gadget by selecting *Gadgets Labels GadLabel1 , 2 or 3* from the main menu.

Move Text Gadget

You can move a text gadget with click-drag-releasing the gadget.

Text Gadget Caption

Change the caption of a text gadget with the right-click context menu **Text....**

To create mult-line text just place the | symbol at the place where you want a line break.

Text Gadget Font

Change the font of a text gadget with right-click **Font....**
Here you can set in the font dialogbox the Font Name, Style, Size and Color.

Text Gadget Text Style

Right-click *Text Style* on a Text Gadget to set one of the available text styles.

Text Gadget Paint Style

You can check the **Select** option. When you now paint a text gadget, the painted area is selected for subsequent range commands or for loading into an Image Gadget.

Text Gadget Get Symbol

Introduced in Painter 17 it is easy to use the Text Gadget as a symbol stamp. When you select Get Symbol, a keymap is displayed where you can select a font. Go over the grid with the mouse down and see an enlarged copy of the character. Click **Use** when you want to transfer the character to the Text Gadget. The Font Symbol form will close when you click **Use**. There are several fonts available that have symbols instead of character. You can enlarge them , give them a color etc. As a start you can use the **Wingdings** font, or maybe you have the **Webdings** font on your machine.

Paint Text Gadget

To paint a text gadget on to your drawing, move it to the desired position and right-click **Paint**. A copy of the gadget will be drawn and the gadget itself can be moved to a different location.

Paint Extrude Text Gadget

To paint Extrude a text gadget on to your drawing, move it to the desired position and right-click **Paint Extrude**. A copy of the gadget with its extrusion will be drawn and the gadget itself can be moved to a different location.

Button Gadgets

Painter has 8 free floating button gadgets that you can display/hide by selecting **Gadgets Buttons GadBut1 ... 8** from the main menu. A lot of context menu options are available with button gadgets. This allows you to create fancy buttons with ease. You can paint your buttons in three different shapes:

Rectangular, Oval and Rounded.

Move Button Gadget

You can move a button gadget by click-drag-releasing it to the desired location.

Button Gadget Caption

Change the caption of the button with right-click **Caption**

Button Gadget Font

Change the Font Name, Style, Size and Color of the button caption with right-click **Font...**

Button Gadget Text Style

Right-click *Text Style* on a Button Gadget to set one of the available text styles.

Button Gadget Font All

Right-click *Font All* on a Button Gadget to set the same font for all the Button Gadgets.

Button Gadget Size

Change the size of the button with either **Height, Width or Autosize.**

Button Gadget Align

If you are working with several buttons you can align them with either **Left or Top**. All the buttons will be aligned to the one selected.

Button Gadget Size

You can make the size of all the buttons equal to the selected one with either **Equal Width, Equal Height or Equal Width and Height**.

Button Gadget Painting

You set the way a button is painted with the right-click Paint Style menu:

- Silver, will paint the button with a silver face.
- Colored, will paint the button with a colored face. A color dialog will allow you to select this color.
- Transparent, will paint the button with a transparent face.

With the **PaintSave** option you select if a button will be saved when you paint it. You are prompted for a filename and you can save in **bmp, gif or jpg** format.

The **Border** option allows you to paint a button with or without a black outline border.

The **Select** option allows you to set the select range when you paint the button. In this way you can easily transfer a button image to a **Image Gadget**. First paint the button gadget with **Select** checked and then load the selection in an image gadget with **Load Range**.

Painter Frames

Painter Frames

Frames are introduced in Painter 14. With the release of **Movies**, my GIF-animator program I felt the need of having the possibility to treat a Painter Image as a series of equal sized frames that I could select and save as a series of bitmaps to be used with *Movies*. Although *Movies* has several drawing tools, they are sometimes limiting.

All Frame operations are available via **Image Frames** from the main menu.

You may not use frames every day, but like most features in Painter: I have added them because I needed them myself.

Frames Set

This will allow you to set the number of frames in your image. Frames are positioned from left to right. Make sure that you resize the image to the correct (long) width and (relatively) short height. When you enter the number of frames, Painter will divide the image width by the number of frames to get the frame width. As Painter works with integer values you want to make the image with an integer multiple of the desired frame width.

If you want 10 frames with a height of 60 pixels and a width of 120 pixels, you resize the image to a height of 60 and a width of 1200 (10 times 120).

Frames Go...

With Frames Go or with **ctrl+G** you can go to a desired frame. Painter will now show you the borders of the frame. When you move with the mouse over a border the hint will display the current frame. Please note that the frame borders are just markers, they are not part of the drawing.

Frames Hide

Allows you to hide the frame borders after a Frames Go. On the next Frames Go the borders will be displayed again.

Frames Clip

This will set the Clip range to the current frame. Now you have all the Painter Range commands available for working on the frame, like Range Transitions, filters etc.

Frames Paste

Will paste the previously clipped frame to the current frame. This allows you to make copies of a frame.

Frames Clear

There is no separate menu option to clear a frame, but with Frames Go, Frames Clip and Range Erase you can achieve this.

Frames Save as BMP's

This will save all the frames as a series of equal sized bitmap files in BMP format. You are asked for a name. Do not add the extension (Painter will do that). If you have 10 frames and you enter *Button* as the filename then Painter will create images *Button1.bmp ... Button10.bmp*.

Quick Buttons

Quick Buttons

Painter Quick buttons are introduced in Painter 14. With the Quick Button tool you can create buttons for your homepage in just a few seconds.

Display Button Tool

Select *Image Quick Quick Buttons* from the main menu to display the Button Tool.

Create Quick Button

Click the color grids, adjust the sliders etc, to create the button. There is no need to explain much in this help file because the Button Tool is purely visual with direct feedback.

Save Quick Button

Click the **Save** button in the Button tool and enter a filename when prompted. Use the *bmp*, *gif* or *jpg* extension to tell Painter in which file format you want to save the button.

Copy Button

Click the **Copy** button to place a copy of the button on the windows clipboard, From there you can paste it in Painter or any other program.

Show Button

Sometimes you need the **Show** button to display the button. You need to do this after you changed the button text, or when you changed the backdrop.

Button Backdrop

Click the **Backdrop** button to load a *bmp*, *gif* or *jpg* image as the backdrop that will appear on the button face.

You can also obtain a backdrop from the painter drawing by selecting a range in the drawing and then **Range Assign To Quick Button**.

Transparent Edges

Check the *Edges* box if you use a backdrop image and you want to backdrop visible thru the edges.

Text Styles

Introduced in Painter 16 you can apply the *Raised* and *Inset* style to the button text, just by selecting one of the radio button in combination with checking the **Style** checkbox.

Painter Whirl

Painter Whirl

Images whirled are introduced in Painter 15. They allow you to give a whirl distortion to your image.

Whirl Image

To whirl your image select **Image Whirl** and then one of the four options:

- Whirl In
- Whirl Out
- Whirl Center, will only whirl the center part of the image
- Whirl Double

Please note that by applying the Whirl In or Whirl Out a couple of times you can create very strong whirled images.

Whirl Range

To whirl a range first select a range, then select **Range Whirl** and finally one of the four options:

- Whirl In
- Whirl Out
- Whirl Center, will only whirl the center part of the image
- Whirl Double

Painter 3D

Painter 3D

Painter 3D is introduced in Painter 15. It allows you to rotate a cube or a globe and set viewpoint and camera angle. The most interesting part of Painter 3D is the possibility to create a 3D graph from user defined functions. These functions can be stored in a Library. A couple of examples are included with the program as a starter. Just experiment.

Start Painter 3D

You open the 3D module by selecting **Image Quick Quick 3D**. When you have created an image copy it to the clipboard by clicking the **Copy** button. You can now close the 3D-tool and select **Edit Paste from Clipboard** in the main Painter menu. You now have the image available for further processing with image manipulation or by adding Gadgets.

3D Colors

Figures are drawn in blue, but you can set the Foreground (fill) color and the Background color. Set the Foreground color by left clicking the color in the color grid. Right-click to set the background color.

Use 3D Colors

Check the **use color** checkbox to use Foreground and Background colors. If you leave the box unchecked, default white is used.

Graph 3D functions

Enter formulae in the **f(x,y)** and **f(y)** edit boxes. You may use the following operators and functions:

Predefined variables:

PI

Accepted operators:

+ , - , * , / , ^ , MOD, DIV
[MOD and DIV implicitly perform a trunc() on their operands]

Functions:

The following functions are supported; it doesn't matter if you use lower or upper case:

COS, SIN, SINH, COSH, TAN, COTAN, ARCTAN, ARG,

EXP, LN, LOG10, LOG2, LOGN,

SQRT, SQR, POWER, INTPOWER,

MIN, MAX, ABS, TRUNC, INT, CEIL, FLOOR,

HEAV (heav(x) is =1 for x>0 and =0 for x<=0),

SIGN (sign(x) is 1 for x>0, 0 for x=0, -1 for x<0),

ZERO (zero(x) is 0 for x=0, 1 for x<>0),

PH (ph(x) = x - 2*pi*round(x/2/pi))

RND (rnd(x) = int(x) * Random)

RANDOM (random(X) = Random; the argument X is not used)

IMPORTANT:

DO NOT USE BLANKS IN THE EXPRESSION.

3D Library

You can save your 3D functions in the Library. Just right-click **Add** over the green dropdown list and enter a descriptive name when prompted.

Use Library

To use a function from the Library, click the name in the list and then click the **Show** button after you have set the figure type to **Surface**.

Delete from Library

To delete a 3D function from the library, first select it and then right-click *Delete* from the context menu.

Update in Library

To update a function already in the Library, first select it, modify the function in the edit boxes and then right-click **Update**, change also the name if you want.

Adjust View

In Quick 3D you can adjust the view with the various spinners. Use the spinners up/down buttons to change the value, or directly type a new value. Just experiment to see the effect of the various spinners.

Tips

Painter

Painter Tips

In this section of the helpfile you will find tips for working with Painter.

- [Web Tiles](#)
- [Text Outline](#)

Tips - Web Tiles

Painter Tips

Tips - Web Tiles

To create a tiling image for your homepage take the following steps. As an example we will use the Tapestry 4 quickback.

1. Make the Tapestry 4 quickback by selecting *Image Quick Quick Background* and then you select *Tapestry 4 from the library*.
2. Create a Seamless Pattern from the image with *Image Seamless Pattern*
3. Select a range somewhere on the image, the exact spot is not important.
4. Reduce the whole image to the selected range with *Range Reduce*
5. Show the Image Gadget with *Gadgets Image GadImage*
6. Load the range in the Gadget with right-click *Load Range*
7. Tile the range with *Range Tile All*
8. If you like the result save the Image Gadget to a gif or jpg file with right-click *Save As...*

That is all.

Tips - Text Outline

Tips - Text Outline

In this tip we will create a text outline with a transparent fill, so that a background will shine through.

1. First we start with a default white drawing
2. Next we create a Text Gadget with font size 24, font style bold and font color red.
3. Paint the Gadget on the drawing
4. Select Image Trace Trace; you now have a red text with a black outline.
5. Now right-click red in the color bar to set the brush color to red
6. Select the floodfill tool and click outside the text to make the rest of the drawing red. Also carefully fill the remaining white parts in the text with red. Click the undo button if you make a mistake.
7. Draw a selection rectangle around the text.
8. Display the first Image Gadget and right-click Load Range to capture the selection.
9. Now right-click Transparent on the Image Gadget. Everything but the outline is now transparent. This is just because everything was red. If this would not be the case you would have to select the transparent color in the Image Gadget and right-click CaptureTransparent Color
10. Create a background with Image Quick Quick Background, and select the Day and Night background.
11. Close the QuickBack tool and drag the image gadget over the drawing to the desired location. Notice that the outline is indeed transparent.
12. Finally right-click Paint on the Image Gadget and drag the Gadget away to see the result.

Painter JPEG

Painter JPEG

Introduced in Painter 17 you can use this tool to optimize the saving of images in JPEG format.

Display the tool from the main menu with **Image Quick Quick JPEG...**

Load the image in the JPEG tool by clicking the **Show** button on the JPEG tool.

Adjust the compression slider and click the **Show** button again. The image is reloaded with the new settings.

Press the **Save** button once you are satisfied with the visible **Quality** and the displayed **File Size**.

Check the **Grayscale** checkbox if you want your image in grayscales.

Painter Effects

Painter Effects

Painter Effects are introduced in Painter 17. The switch from a 33MHz to a 400MHz PC enables me to further explore the world of image manipulation.

Effects Tool

Display the Effects Tool by selecting *View Effects Panel* from the main menu. The effects panel houses a series of scrollbars for various effects. Changing a scrollbar position will give you immediate feedback on the preview panel. When you want to use the preview just click the **Use** button on the preview panel.

Please note that Effects are allways applied to the main image, not to the preview image. This means that if you want to apply several effects to an image you have to click the **Use** button before you apply the next effect.

Please experiment with the various effects. I will not try to describe them in this help file.

Effects Menu

A couple of effects do not require a scrollbar. These you will find on the main menu under *Image Effects*.

The following options are available:

- Grayscale
- Spray
- Anti-Alias

The Spray effect and the Anti-Alias effect are not very fast at the moment. In future releases of Painter I will try to speed them up.

