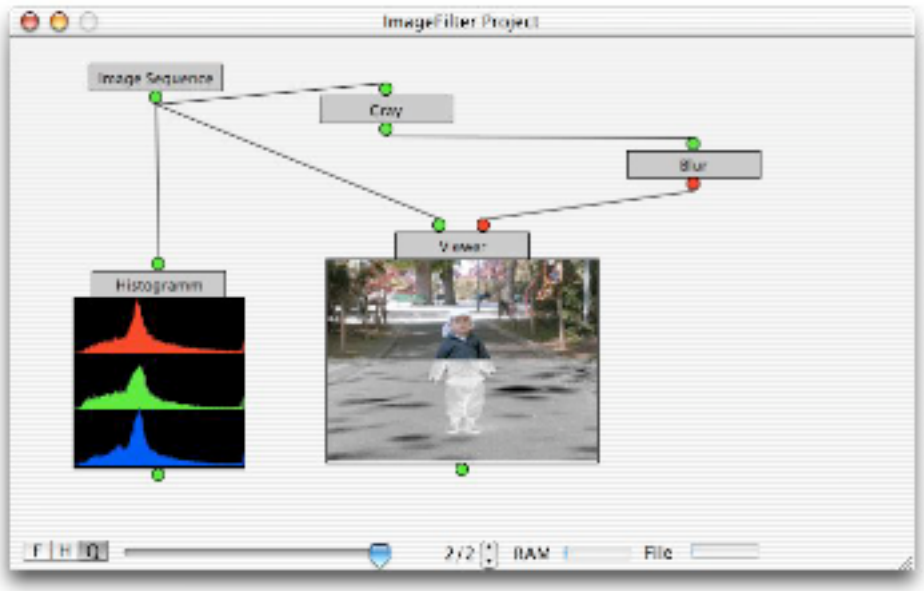
Belle Nuit Imagefilter 1.0

Programm to batch-process image sequences. Build process trees with a node interface. Apply a wide variety of nodes to color correct, filter, transform and compose images. Automate the processing of the images.



Features

- Node-Interface
- Resolution independent, non-square pixel support
- Import of many picture formats, Quicktime, PDF and textfiles.
- Scriptable Quartz Graphic Engine
- Unlimited undo and intelligent caching
- Wide variety of nodes:
    - Input: Image Sequence, QuickTime, PDF, Image Capture, Solid, Color Bars, Test Chart, LUT, Text Input, Text File, Current
    - Color: RGB Correction, RGB Matrix, Level, Color Balance, Color Temperature, Chroma, Gray, Duotone, Bitmap, Invert, Color Difference, Color Distance, Apply LUT 1d, Apply LUT 3d, Mimikri, Difference LUT, Color Space, Color Script
    - Filter: Blur, Motion Blur, Deflicker, Sharpen, Unsharp Mask, Convolution, Convolution 2d
    - Transform: Resize, Crop, Border, Mirror, Rotate, Free Rotate
    - Time: Interlace, Deinterlace
    - Composite: Matte Key, Blend, Add, Sub, Max, Min, Mult, Channel Mix, Unmultiply
    - Graphic: Rect, Text, Quartz Script
    - Output: Viewer, Viewer Window, Histogramm, PICT Export, TIFF Export, JPEG Export

Content

Specifications

- Power Macintosh G4 or G5
- OS X 10.2 or newer.
- Monitor resolution 1024 * 768 pixels or more

Download

Belle Nuit Imagefilter 1.0
http://www.belle-nuit.com/download/imagefilter100.dmg (3.3 MB)

Printable Documentation

http://www.belle-nuit.com/download/imagefilter.pfd (53 pages, 350 KB)

Installation

The downloaded file is a disk image. Drag the Program to the Applications folder.

When you want to use Belle Nuit Imagefilter for rendering titles, you need to register. Go to the the **Apple:Register...** menu and enter **Username** and **Serial Number**.

License

This program is commercial software. If you want to use it, you need a license.

Demo License

You can freely distribute this program and use on any numbers of computers. The offline version is fully functional, but export is limited to pictures 512 * 384.

Online License

You can acquire the online license at kagi (http://order.kagi.com/?ZSI).
Online licene fee is 120 USD. After paying, a username and a serial number will be sent to you within 3 days.
With the online license, you are allowed to use this program on one Macintosh. You may make backup copies, but you may not register the online license on more than

one Macintosh.
Broadcast stations and postproduction houses may opt for a site licence (600 USD). Owners of a online license will have the right to free support by email (matti@belle-nuit.com). Please include your username when you are writing for support.

With both licenses, Belle Nuit Montage keeps ownership of the program. The license only allows you to use the program for an unlimited duration.

If you use this program, you agree with the terms of this license.

Working with Nodes

A new Imagefilter document is a quite empty window. You only find some status items on the bottom:

- Quality button: (F)ull, (H)alf ad (Q)uarter resolution
  These options allow you to work with subsampled resolution while you design you node-tree.
- Offset-slider
  Allows you to define the image you are working on.
- RAM and File Cache
  Shows you the amount of used RAM or File Cache.

The rest of the interface is up to you. With this program, you will design a node-tree which define the processing of the image. Think of a node as of a black box with one ore more inputs and one output. There are nodes for importing images, for processing the image, for viewing it and for rendering (exporting) it.

You create new nodes from the menus. The new node is drawn as a rectangle with its name. It has three states:

- Normal: gray
- Selected: dark red
- Activated: light red (an activated node is always selected, too)

You can activate a node by clicking on it. You can deactivate the node by clicking outside of the node. You can activate and drag a node immediately. You can select parent and child nodes relative to the activated node to drag them together.

Use the tab key to navigate between nodes (and use shift tag to navigate back).

You can select and deselect nodes with shift-clicking. You can duplicate a group of selected nodes and copy-paste them into a new document.

Nodes are connected at their input and outputs. The inputs are on top and the output is on the bottom. All nodes have an output, but some nodes (importers and generators) do not have inputs. The connectors can have three colors:

- Not connected: transparent

- Connected, but no image available: red
- Connected and image available: green

You can draw connections from an activated node to another node. Click into an input or an output connector and drag to another node. You can drag to the connection of the other node or to the rectangle. If you drag to the rectangle, then it will connect either to the output or to the first input. To remove an input connection, drag a connection outside of the nodes.

Every node has parameters. You can open the parameters with three ways: Double-click the node, pressing the return or the enter key or using the menu Edit:Open Node Editor. The Node Editor appears below the node. You can navigate through the parameters with the mouse or with the up- and down-arrow keys. There are several types of parameters:

- String. Click on it or open it with Enter or Return: A sheet with a textbox appears.
- Checkbox. Click on it or use the left- and right-arrow keys.
- Number. Type the number or use the left- and right-arrow keys. Use option-left and option-right to go faster. Use home and end to go to the minimum and the maximum. You can also click to the optical slider to set the value approximately.
- Color. Click on it or open it with Enter or Return: A window with a color select box opens. You can define the color with one of the menus or use the eyeglass to select a color on screen.
  If the row is selected, you can also control-click anywhere on a pixel in a viewer to sample the color without using the color dialog.
- List. Use the left- and right-arrow keys. Use home and end to go to the first and the last value. Or type the first letter of a value to select it directly. Click or press enter to see the all options of the list.
- File / Folder. Click on it or open it with Enter or Return: A sheet with a file dialog appears.

To leave the node editor, press the escape key or click outside of the editor.

Importing

You start always with existing images. These may either be images you import or images you create with the generators. All these nodes are in the Input menu.

You can import single image sequences with the **Image Sequence** node. The Image Sequence node supports all file types known by QuickTime. Click on the **Folder** parameter to locate the images. The node imports all images of the folder, ordered alphabetically.

You can import a **QuickTime** movie. The node will import each frame of the movie.

You can import **PDF** files. You define the size of the rendered files. Each page gives a frame.

You also have generators like **Color Bars** and **Solid**.

The **Current** node is a special input node. It uses the output of the current active node.

You can also import text files. Textfiles combined with Graphic nodes can be used to create serial titles like subtitles or lower thirds.

For all input nodes, you will also have to define the aspect ratio. The correct aspect ratio allows for corrrect display of non-square pixels of the video formats and also allows for aspect ratio correction of some of the nodes.

Viewing

You can view either with **Viewers** or with an external **Viewer Window**. Viewers are also nodes you can choose from the Output window. Actually, nodes are only processed if they have an output node at the end of the node tree.

Viewer can be of double, full, half or quarter, eightth and sixteenth size (relative to original size).

Viewer windows can be of any size, and also Full Screen, if you have a second monitor.

The Viewers can be updated automatically and manually:

- Use the menu **Output:Refresh** to update the Viewers manually.
- Use the menu **Output:AutoRefresh** to update the Viewers on each change of a parameter.

If you have two inputs, then you can view them in split screen. Use the wipe parameter to define the split position (50 being center).

The images are cached in RAM and as files, so when you change parameters and view back and forth, they are not recalclated again. You can set the size of the RAM and the file cache in the preferences.

Note: You can always stop an update pressing the command and the option key.

Rendering

Select one of the Output nodes **PICT Export**, **TIFF Export** or **JPEG Export** and define the location of the exported files. PICT is faster, but TIFF allows for saving a mask. For JPEG, you can set the compression quality. Select **Render All** or define a range of images to render.

Now choose the menu command **Render** to render the selected output nodes or **Render All** to Render All output nodes.

Note: You always render at the current resolution.

Note: You can always stop a rendering by pressing on the stop button, by pressing command-period or escape. Note that the current frame will still be exported.

Performance

It isn't fast enough! - You are impatient.

Remind that you are not on a workstation with dedicated hardware. This porgram does not ship with a PCI-card and you are using it probably on your powerbook with a harddisk which is slower than a RAID 0 diskarray.

We know that speed is a concern for you and try to find fast solution both for rendering and user interaction.

In the mean time, speed also depends on how you configure Belle Nuit Imagefilter. There are some settings, which can increase performance by a factor of more than ten (or decrease, if used wrong). The good news is that you can change these settings yourself:

- In the interface, work with **half** and **quarter** resolution as long as you can. In quarter resolution, the node has to process only 1/16 of the pixels. The node are aware of the resolution and adapt, except the filter nodes which always work at full resolution.
- With half and quarter resolution, however, the disk access time for the image sequence and the quicktime node is still the same. If you plan to stay longer in the interface, consider rendering proxies. This is very easy from within Imagefilter. Just connect an output node to the input node, set half or quarter resolution and render it. Work with a node with the new folder and just reconnect to the old node wehen you go definitive.
- Set your **RAM cache** big enough. All rendered images and intermediate results are cached, so that on a change only the changed nodes and their outputs must be reprocessed. Set it big enough for your needs, but do not cheat. You can set it larger than the real RAM, but then it will use virtual memory and swap and become very slow.
- The **file cache** is the second net to cache images. It makes only sense when it is larger than the file cache, because images for the RAM cache are also stored in the file cache. If the file cache speeds up your processing depends on the speed of your harddisk and the complexity of your nodes. For simple color correction and compositing reporcessing may be faster than reading the disk. Therefore you can also **turn the file cache off**.
- You also may generally better turn off the file cache on long renders. But not always. If you render to multiple nodes the result of a very slow node, it is

better if it is cached.

- When you render to multiple output nodes, Imagefilter renders by offset and not by node. This way it can use intermediate results which cached. Set the RAM cache enough big to accomodate such a scenario.
- Normally, you want to work with **Auto-Refresh** so that you do not have to select Refresh on every parameter change. But sometimes, Auto-Refresh acts to fast, while you are still fiddling with the parameters. This is where the **Render Wait Time** comes in. This is the time in milliseconds the program will wait after a parameter change to render. Set it small enough to have a reactive system and big enough to not be overwhelmed. *Note that the program always renders immeditalye when you change the offset to allow you to compare pictures fast.*
- In the initial design, we had this fancy chasing arrows, which dance from node to node. While they show didactically the workflow from node to node, they slow down the computer lot. On my powerbook, every time i show them, they let the program 0.3 seconds! This is whay we have now the option **Interface update time**. This time in milliseconds defines the time a node runs silently, before the chasing arrows are shown. You will not get a feedback from every node, but everything runs faster. The "Transformations" example on my powerbook runs ten times faster with a interface update time 300 than 0. So if you want to see every node, use 0. For normal work, use 300-500. For rendering, set it to 1000 or 2000.
- Finally note that you can measure the time a Refresh needs: With the **Debug Profile Option** the Console shows timestamps of each important step of the Refresh. This way you can see if the changed settings improve your perfomance. *Note: Use a **Force Refresh** to ensure that you measure the whole refresh and do not use cached intermediate results.*

Node Reference

Image Sequence

Imports a folder of still images. All formats that Quicktime can read are recognized. The node imports all images in a folder, by their alphabetical order.

Parameters:

- Folder: Locate the folder of the images
- Loop: Loop over the images, when the sequence is shorter than other sequences in the project.
- Apect ratio: Square, PAL, PAL 16:9, NTSC, NTSC 16:9

Note: You can import one global image, if you put it alone into a folder and set the loop parameter. You can use loop also to reuse patterns.

QuickTime

Imports Quicktime movie.

Parameters:

- File
- Frame Rate: 24, 25, 29.97, 30
- Apect ratio: Square, PAL, PAL 16:9, NTSC, NTSC 16:9

PDF

Rasterizes as PDF file, one image per page. You can create graphics in any application, print them to PDF and render in ImageFile the PDF to a video file with non-square pixels. You can also use the PDF as a mask for picture effects.

Parameters:

- File
- Width

- Height
- Apect ratio: Square, PAL, PAL 16:9, NTSC, NTSC 16:9

You may want to create custom paper sizes to preview the titles more accurately in your application. Creating custom paper sizes is very easy in OS X. Once you have set a custom paper size, it is available for all programs:

1. Go to the **System Preferences:International**, select the **Numbers** tab and set the Measurement system to Standard. (You can set it back to Metric later.)
2. Open an Application that can print.
3. Open the **File:Page Setup** dialog. From the **Settings** popup-menu, choose **Custom Paper Size**
4. Click on **New** to add a size and name it. Name it immediately, because you cannot change the name afterwards. Add the following parameters for Height and Width and set all Margins to zero.

Name	Height	Width
PAL	8 inch	10.67 inch
PAL 16:9	8 inch	14.22 inch
NTSC	6.75 inch	8 inch
NTSC 16:9	6.75 inch	9 inch
HD 720p	10 inch	17.78 inch
HD 1080p	15 inch	26.67 inch

   Note: 1 pixel is 1/72 inch.

5. These paper sizes are now available in the Page Setup dialogs.
   Note: Because OS X expects all paper sizes to be tall, you still need to choose the wide orientation manually.

Image Capture

Captures images form a still camera that supports capture through the camera (like a Nikon Coolpix 3700). The picture is captured when you press command-0 (menu **Input:Capture Now**) and downloaded to a local folder on the Macintosh.

Parameters:

- Folder: Download Folder

- Apect ratio: Square, PAL, PAL 16:9, NTSC, NTSC 16:9

Note: All images are downloaded when you refresh and then deleted on the camera. We recommend you to use an empty memory chip.
Note: Only JPEG formats are supported.
Note: Because the Download to the Macintosh, the Auto-Refresh does not work properly. Use Refresh manually.

Solid

Uniform rectangle of a chosen color.

Parameters:

- Color
- Width
- Height
- Apect ratio: Square, PAL, PAL 16:9, NTSC, NTSC 16:9

Color Bars

Color bars

Parameters:

- Levels: Full, Video 100%, Video 75%
- Width
- Height
- Apect ratio: Square, PAL, PAL 16:9, NTSC, NTSC 16:9

Test Chart

Returns the Belle Nuit Testchart as documented on the Belle Nuit Website: http://www.belle-nuit.com/testchart.html.

Parameters:

- Size: PAL, PAL 16:9, NTSC, NTSC 16:9, NTSC DV, NTSC DV 16:9, 720p,

1080p

LUT

Create an identity LUT. Ti speed up multiple step color correction, correct the LUT and not the actual picture and finally apply the LUT with the apply LUT node. The LUT uses a 3D-interpolation with 16*16*16 values. Using a LUT over multiple color correction steps also minimizes stepping artefacts.

This node has no Parameters.

Text Input

Enter manually text to use with Graphic nodes. The text can be split into paragraphs for each image, either by single or double returns (as in Subtitler).

The viewers show the text, if there is no input image present ad the connection.

Parameters:

- Separator: Global (no separator), Single return, double return
- Text Encoding: UTF-8 (Unicode), MacRoman, Windows Latin1

Text File

Input Text from a file to use with Graphic nodes. The text can be split into paragraphs for each image, either by single or double returns (as in Subtitler).

The viewers show the text, if there is no input image present ad the connection.

Parameters:

- File
- Separator: Global (no separator), Single return, double return
- Text Encoding: UTF-8 (Unicode), MacRoman, Windows Latin1

Current

Get the image from the current active node. Use this to create a global viewer.

RGB Correction

Classical primary color correction.

Parameters:

- Master gain, gamma, setup
- Red gain, gamma, setup
- Green gain, gamma, setup
- Blue gain, gamma, setup

RGB Matrix

Remaps RGB values in a matrix. The default values are the identity matrix. In offsets are added before, Out offsets after the matrix multiplication.

Parameters

- Red, green and blue in offset
- Red from red, green and blue
- Green from red, green and blue
- Blue from red, green and blue
- Red, green and blue out offset

Level

Performs a histogram equalization, like for example to adapt a picture from video (601) levels to RGB levels.

Parameters:

- Black In, Black Out, White In, White Out, Gray.

Color Balance

Performs a white balance correction on low, mid and high levels.

Parameters:

- Black, Gray, White: Choose pixels in the source pictures which should be neutral density.

Note: As an alternative for bad exposuure, you may want to use the Color Temperature node.

Color Temperature

Corrects color temperature. The input is assumed Computer RGB (black = 0), converted to linear RGB and corrected with the color of the black bodies of both input and output temperature.

Parameters

- Input Temperature: 2000 - 29000 Kelvin, lower than 2000 K can not be chosen, because there is no blue any more.
- Output Temperature: 1000 - 29000 Kelvin.

Chroma

Set Saturation.

Parameters:

- Saturation (default 100)
- Saturation Gamma (default 1.0): allows to increase low chroma without affecting the saturated parts of the image.

Gray

Creates a monochrome image. The parameters define the weight of each channel to luminance.

Parameters:

- Red, green and blue

Duotone

Creates a monochrome image with a tint for black and a tint for white.

Parameters:

- Red, green and blue: weight for each channel to luminance
- Black, white: clip values
- Black color, white color: low and high duotone color

Bitmap

Creates a black and white bitmap image.

Parameters:

- Threshold

Invert

Inverts the image.

Color Difference

Calculates the difference between color channels. This can be uses to create a matte for a green screen.

Parameters:

- Channel: Red, Green, Blue
- Method: Max, Mean, Min: The function of the two other channels to compare against.

Color Distance

Calculates the distance to a key color in the RGB space. This can be uses to create a matte for a green screen.

Parameters:

- Key Color

Apply LUT 1d

Applies one, three or four 1D-LUT to a picture. You can provide up to 256 values. If both the master and the RGB LUTs are defined, the master LUT is applied first.

Parameters:

- Master, Red, Green, Blue: Text parameters. Add value pairs in ascending orders, input and output separated with space, like in the example:
  ```
  0 16
  64 80
  128 128
  255 235
  ```
- Reverse: Reverses input and output of the LUT

Apply LUT 3d

Applies a 3D-LUT to a picture. The picture is on the first and the LUT on the second input.

This node has no parameters.

Mimikri

Applies the color correction between input 2 and 3 to input 1. Mimikri is documented on the Mimikri website http://www.mimikri.ch.

Parameters:

- Method:
  - Color: Color for normal color correction and when the SD material is not a downconversion of the HD
  - Spot: Spot for spot color correction and sophisticated secondary color

correction (slower).

Difference LUT

Creates a 3D-LUT out of the difference of iunput 1 and input 2. You can then apply this LUT to another image.

This node has no parameters.

Color Space

Transform between color spaces.

Parameters

- Input Space, Output Space
- RGB: Computer RGB with black at 0 and white at 255
- Studio RGB: Video RGB with black at 16 and white at 235
- Component 601: Component Video (PAL, NTSC). Y is mapped to green, Cr to red and Cb to blue.

Color Script

Fully programmable Color Correction. Each color of the picture is processed once.

Parameters:

- Script: Script text

The Color Script uses the RBScript language, defined below. The following properties are accessible:

- Red, Green, Blue as read/write
- Luma read-only

Note: Values not set use the input value, not black.

Blur

Binominal blur.

Parameters:

- Horizontal, Vertical: Width and Height of filter.
- Red, Green, Blue: Channel to perform the filtering.

Motion Blur

Directional blur.

Parameters:

- Repetitions: Number of iterations
- Angle: 0-360 degrees.
- Distance: Distance between iterations
- Method: Linear, Zoom
- Red, Green, Blue: Channel to perform the filtering.

Deflicker

Vertical filtering for titles and graphics to be used in interlaced video.

This node has no parameter.

Sharpen

Simple 3-pixel sharpening algorithm

Parameters:

- Horizontal, Vertical: Level of filtering.
- Red, Green, Blue: Channel to perform the filtering.

Unsharp Mask

Better sharpening algorithm. Substracts the blurred picture from the source.

Parameters:

- Horizontal, Vertical: Width and Height of filter.
- Level: Level of filtering.
- Red, Green, Blue: Channel to perform the filtering.

Convolution

Performs two one-dimensional convolution filtering.

Parameters:

- Horizontal, Vertical: The list of the filter weights, provided as space-delimited text.
- Scale to unity: Scales the weights so that their sum is 1.
- Red, Green, Blue: Channel to perform the filtering.
- Border: Black, White, Pixel: Values to add on the border.

Convolution 2D

Performs a two dimensional convolution filtering on a 5x5 matrix.

Parameters:

- Matrix value: Only integer values are allowed.
- Scale to unity: Scales the weights so that their sum is 1.
- Red, Green, Blue: Channel to perform the filtering.
- Border: Black, White, Pixel: Values to add on the border.

Resize

Resamples the image

Parameters:

- Absolute: Use absolute values for with and height. If you do not use absolute values, then the sizes are measured per thousand.

- Width, Height
- Method: QuickDraw, Bilinear, QuickTime

Crop

Crops from the border of the image

Parameters:

- Absolute: Use absolute values for top, left, bottom and right. If you do not use absolute values, then the sizes are measured per thousand.
- Top, Left, Bottom, Right

Border

Adds a border outside of the image. Use this to enlargen your workspace

Parameters:

- Absolute: Use absolute values for top, left, bottom and right. If you do not use absolute values, then the sizes are measured per thousand.
- Top, Left, Bottom, Right
- Background: Color of the border.

Mirror

Mirrors the image

Parameters:

- Horizontal, Vertical

Rotate

Rotates the image counter-clockwise in 90-degrees steps.

Parameters:

- Angle

Free Rotate

Rotates the image counter-clockwise in fine steps.

Parameters:

- Angle
- Background: Color of the background

Interlace

Interlaces the odd field on input 1 with the even field on input 2.

This node has no parameters.

Deinterlace

Deinterlaces the video picture.

Parameters:

- Keep: Odd (first lines etc.), Even (second lines etc.)
- Othe Field: Throw away, Duplicate, Interpolate

Matte Key

The background image (input 1) is replaced by the foreground image (input 2) depending on the mask (input 3). The dimensions of the output are the same as the dimensions of the background.

Parameters:

- Invert Matte: By default, black is foreground and white is background.
- Absolute: Use absolute values for left and top. If you do not use absolute values, then the sizes are measured per thousand.
- Left, Top: Offset of the foreground image. By default, the foreground image is

put on the left top corner.

Note: If you do not provide a mask, then a picture-in-picture operation is done.

Blend

The background image (input 1) blended with the foreground image (input 2). The dimensions of the output are the same as the dimensions of the background.

Parameters:

- Level: Opacity of the foreground
- Absolute: Use absolute values for left and top. If you do not use absolute values, then the sizes are measured per thousand.
- Left, Top: Offset of the foreground image. By default, the foreground image is put on the left top corner.

Add

The foreground image (input 2) is added to the background image (input 1). The dimensions of the output are the same as the dimensions of the background.

Parameters:

- Absolute: Use absolute values for left and top. If you do not use absolute values, then the sizes are measured per thousand.
- Left, Top: Offset of the foreground image. By default, the foreground image is put on the left top corner.

Sub

The foreground image (input 2) is substracted to the background image (input 1). The dimensions of the output are the same as the dimensions of the background.

Parameters:

- Absolute: Use absolute values for left and top. If you do not use absolute values, then the sizes are measured per thousand.
- Left, Top: Offset of the foreground image. By default, the foreground image is

put on the left top corner.

- Absolute Difference: If the foreground pixel is brighter than the background pixel, absolute value of the substraction is taken.

Max

On each channel, the brighter value is taken. The dimensions of the output are the same as the dimensions of the background.

Parameters:

- Absolute: Use absolute values for left and top. If you do not use absolute values, then the sizes are measured per thousand.
- Left, Top: Offset of the foreground image. By default, the foreground image is put on the left top corner.

Min

On each channel, the darker value is taken. The dimensions of the output are the same as the dimensions of the background.

Parameters:

- Absolute: Use absolute values for left and top. If you do not use absolute values, then the sizes are measured per thousand.
- Left, Top: Offset of the foreground image. By default, the foreground image is put on the left top corner.

Mult

The foreground image (input 2) is multiplied with the background image (input 1). The dimensions of the output are the same as the dimensions of the background.

Parameters:

- Absolute: Use absolute values for left and top. If you do not use absolute values, then the sizes are measured per thousand.
- Left, Top: Offset of the foreground image. By default, the foreground image is put on the left top corner.

Channel Mix

Swap Channels from up to 3 inputs.

Parameters:

- Red, Green, Blue: Source Channel: Input 1/2/3 Red/Green/Blue or Black

Unmultiply

Unmultiplies a premultiplied composed image.

Parameters:

- Invert Mask: Use this is white is foreground.

Rect

Displays a Rectangle.

Parameters:

- Absolute, Left, Top, Width, Height: Relative values in thousands (1/1000), absolute values in pixel.
- Fill Color, Fill Level
- Border Color, Border Level, Border Width

Text

Displays Text, one or more lines.

Parameters:

- Textfont
- Bold, Italic
- Text Size
- Spacing, Leading

- Horizontal Align: Left, Center, Right
- Vertical Align: Top, Center, Bottom
- Absolute, X, Y: Reference point of text. Relative values in thousands (1/1000), absolute values in pixel.
- Fill Color, Fill Level
- Border Color, Border Level, Border Width

Note: Only MacRoman text is supported.
Note: Some fonts may not properly display certain styles. The text will then fall back to Helvetica.

Quartz Script

Does give direct access to the Quartz graphics engine. Text and Graphics are fully antialised and can be transformed freely in a 2D-space. The Script has two inputs: The first for the background image and the second for text (you can retrieve with GetLine).

Parameters:

- Absolute: Use absolute pixel values for dimensions. If you use relative values, width is assumed 1000 and height is assumed 1000.
- Script: Script text

The Quartz Script uses the RBScript language, defined below. The following commands are available:

- AddArc(x as double, y as double, radius as double, startangle as double, endangle as double, clockwise as boolean)
- AddArcToPoint(x1 as double, y1 as double, x2 as double, y2 as double, radius as double)
- AddCurveToPoint(cp1x as double, cp1y as double, cp2x as double, cp2y as double, x as double, y as double)
- AddLineToPoint(x as double, y as double)
- AddQuadCurveToPoint(cpx as double, cpy as double, x as double, y as double)
- AddRect(l as double, t as double, w as double, h as double)
- BeginPath
- ClearRect(l as double, t as double, w as double, h as double)

- Clip
- ClipToRect(l as double, t as double, w as double, h as double)
- ClosePath
- ConsolePrint(s as string)
- DrawPath(mode as integer)
- EOClip
- EOFillPath
- FilllPath
- FillRect(l as double, t as double, w as double, h as double)
- GetTextPosition(byref x as double, byref y as double)
- Height as integer
- LineCount as integer
- MoveToPoint(x as double, y as double)
- RestoreGState
- RotateCTM(angle as double)
- SaveGState
- ScaleCTM(sx as double, sy as double)
- SelectFont(lbl as string, sz as double)
- SetAlpha(al as integer)
- SetCharacterSpacing(spacing as double)
- SetFlatness(flatness as double)
- SetFontSize(sz as double)
- SetGrayFillColor(gray as integer, al as integer)
- SetGrayStrokeColor(gray as integer, al as integer)
- SetLineCap(cap as integer)
- SetLineJoin(join as integer)
- SetLineWidth(w as double)
- SetMiterLimit(limit as double)
- SetRGBFillColor(c as color, al as integer
- SetRGBStrokeColor(c as color, al as integer
- SetTextDrawingMode(fill as boolean, stroke as boolean, clip as boolean)
- SetTextPosition(x as double, y as double)
- ShowText(s as string)
- ShowTextAtPoint(s as string, x as double, y as double
- StrokePath

- StrokeRect(l as double, t as double, w as double, h as double)
- StrokeRectWithWidth(l as double, t as double, w as double, h as double, wid as double)
- Text as string
- TextLine(ind as integer) as string
- TranslateCTM(tx as double, ty as double)
- Width as integer

Note: Only MacRoman fonts are supported.

Note: The origin (x=0, y=0 is bottom left) like in PostScript.

Viewer

Shows the image. Use the menu **Output:Refresh** or **Output:AutoRefresh** to update the viewer. The output is the first input. If there are two inputs, the images are showed as split-screen. If there is only text at the input, then the text is displayed.

Parameters:

- Scale: Double, Pixel (shown as square pixel), Full, Half, Quarter, Eighth, Sixteenth.
- Split Type: vertical, horizontal or blend
- Split Value: Position of split screen

Viewer Window

Shows the image on an external window. The window can be on the same monitor or on another monitor. You can also set the window to fullscreen, for example use the S-Video output of your powerbook to preview the image on a videomonitor. Use the menu **Output:Refresh** or **Output:AutoRefresh** to update the viewer. The output is the first input. If there are two inputs, the images are showed as split-screen. If there is only text at the input, then the text is displayed.

Parameters:

- Scale: Double, Pixel (shown as square pixel), Full, Half, Quarter, Eighth, Sixteenth and Fit (fit to window size)

- Split Type: vertical, horizontal or blend
- Split Value: Position of split screen
- Full Screen

Available Menu commands in the Viewer Window:

- Refresh, Auto Refresh, Force Refresh, First, Previous, Next, Last.

Available Keyboard shortcuts in Viewer Window:

- ESC: Show document window into the foreground.
- Home, Left, Right, End: Navigate the Offset.

Note: If you make FullScreen on the main monitor, the other windows will be hidden. Hit Escape to put the document window into the foreground.

Histogramm

Parameters

- Scale: Full, Half, Quarter: Full is 256 * 256 pixels.

PICT Export

Exports as PICT, no mask.

Parameters:

- Folder
- Render All or First-Last (0-based)
- Name String: The 0000 are replaced by a sequential number starting with 0.
- Script: Alternatively to the name string, you can use a script to define the filename.
  The following commands are available:
  - Filename as string (read and write)
  - Shortfilename as string (read only): the filename without the extension
  - Extension as string (read only): ".pct"
  - Offset as integer

Note: The export does not empty the folder, but overwrite any existing file with the same names.
Note: The initial filename for a script is the one of the image sequence at the source (on a tree with multiple inputs, the source on the first input).

TIFF Export

Exports as uncompressed TIFF with optional mask on second input.

Parameters:

- Folder
- Render All or First-Last (0-based)
- Name String: The 0000 are replaced by a sequential number starting with 0.
- Script: Alternatively to the name string, you can use a script to define the filename.
  The following commands are available:
  - Filename as string (read and write)
  - Shortfilename as string (read only): the filename without the extension
  - Extension as string (read only): ".tif"
  - Offset as integer

Note: The export does not empty the folder, but overwrite any existing file with the same names.
Note: The initial filename for a script is the one of the image sequence at the source (on a tree with multiple inputs, the source on the first input).

JPEG Export

Exports as JPEG, no mask.

Parameters:

- Folder
- Render All or First-Last (0-based)
- Name String: The 0000 are replaced by a sequential number starting with 0.
- Quality: 100% is uncompressed.
- Script: Alternatively to the name string, you can use a script to define the

filename.
The following commands are available:

- Filename as string (read and write)
- Shortfilename as string (read only): the filename without the extension
- Extension as string (read only): ".jpg"
- Offset as integer

Note: The export does not empty the folder, but overwrite any existing file with the same names.
Note: The initial filename for a script is the one of the image sequence at the source (on a tree with multiple inputs, the source on the first input).

Menu Reference

- **Apple:About Imagefilter:** Displays Username.
- **Apple:Register**
- **Apple:Preferences**
    - RAM Cache in MB
    - File Cache in MB: The file cache is in the folder of the application and emptied at the end of each session.
    - Disable File Cache: May speed up rendering
    - Render Wait Time: Time in msec to wait before AutoRefresh
    - Interface Update Time: Time in msec to wait before chasing arrows are shown in processing nodes (Trade-off between interactivity and overall speed)
    - Render Errors, Warnings, Profile, Log: Selection of messages to be displayed in the Console window.
- **File:New, Open, Recent, Close, Save, Save As...**
- **File: Page Setup**
- **File: Print Picture:** Prints the current picture of the active node. If the picture is bigger than the page, it is resized.
- **File: Print Node Tree:** Prints the node tree and a list of the parameters of each node.
- **Edit:Undo, Redo:** Limited only by application memory.
- **Edit:Cut, Copy, Paste, Clear, Duplicate, Select All:** You can copy paste nodes between documents.
  Copy also copies the output picture of the node into the pasteboard, so that you can use it in another application.
  You can also use the keyboard-shortcut command-delete to delete selected nodes.
- **Edit:Open/Close Node Editor:** You can only edit the active node.
- **Edit:Reset Node**Resets node to default parameters. The name and the connections are preserved.
- **Edit:Select Parent Nodes** , **Edit:Select Child Nodes**: Use them to drag or copy-paste a group of nodes
- **Input, Color, Filter, Transform, Time, Composite, Graphic, Plugin, Output** see filter reference.
- **Input:Capture Now:** Capture with the Capture Image node.

- **Input:Full Resolution, Half Resolution, Quarter Resolution:** Sets the Resolution of the current document. Do not forget to set to full resolution before you render definitely.
- **Output:Refresh:** Refreshes the viewers manually. Press option to force refresh.
- **Output:AutoRefresh:** Refreshes the viewers automatically at each change. You can set the waiting time in the preferences.
- **Output:ForceRefresh:** Clears the Cache and refreshes.
- **Output:Render:** Renders the selected output nodes.
- **Output:Render All:** Renders all output nodes.
- **Output:Render Current frame:** Renders current frame of the selected output nodes.
- **Output:First, Previous, Next, Last:** Changes the offset. These menu commands are only for completeness. You will be faster using the Home, Left, Right and the End key.
- **Window:Console**

Keyboard Shortcuts

Imagefilter is optimized to use direct keyboard shortcuts whenever possible, so that you do not need the mouse. Following a list of the available keyboard shortcuts in the different contexts. These shortcuts add to the shortcuts available in the menus

Document level

- Home: Set Offset to 0
- End: Set Offset to last image
- Right Arrow: Set Offset +1
- Option-Right Arrow: Set Offset +10
- Left Arrow: Set Offset -1
- Option-Left Arrow: Set Offset -10
- Tab: Activate the next node
- Shift-Tab: Activate the previous node
- Command-Delete: Delete the active node
- Enter, Return: Open the Node editor
- Escape: Close the Node editor

Node editor

- Down Arrow: Activate next parameter
- Up Arrow: Activate previous parameter
- Enter, Return: Open the parameter editor
- Home: Set parameter to the first or minimum value
- End: Set parameter to the last or maximum value
- Right Arrow: Set parameter +1 (or + grain)
- Option-Right Arrow: Set parameter +10 (or + 10*grain)
- Left Arrow: Set parameter -1 (or - grain)
- Option-Left Arrow: Set parameter -10 (or - 10*grain)
- Letter: Select a value in the list with the starting letter
- Number, ".", "-": Start to type the number

Script Language Reference

Introduction

Imagefilter uses the Realbasic Script language for its scripts and for the plugins, expanded with Imagefilter functions. The following text will only explain the most important functions. For full documentation, refer to the Realsoftware website: http://www.realsoftware.com.

The script is one main function, for which you do not define a header and a footer. Inside the script, however, you can define functions before the code of the main function.

While you are writing the script, you may want to put the script filter on bypass, so that it is not executed with invalid code.

This simple script will just print "Hello World" to the console:

```
print "Hello World"
```

Introduction

Variables can be of the type integer, double, boolean and string. All variables have to be defined at the beginning of the code. Variables are defined with the Dim statement, constants with the Const statement.

Examples:

```
dim i as integer
dim t as text
const parselength=1024
```

Do not use any reserved word for variable names or you will have a severe parsing error. Reserved words are the names of the operators, the control structures and of all built-in functions.

Variable names are not case-sensitive.

Variables are automatically initialized (integer and double: 0, string: ""; boolean:

false).

Arrays

You can define arrays of variables in single and multiple dimensions like

```
dim lines(5) as string
dim cells(5,10) as string
```

where lines is an array of 6 elements going from 0 to 5. You can also define an empty array with

```
dim lines(-1) as string
```

Later in the code, you can either use the redim statement

```
redim cells(10,20) as string
redim lines(0) as string
```

If you try to access an array element which is not in the dimension, you will get an "Out of Bounds Exception".

Operators

The following operators are supported:

- +
- -
- *
- /
- \ (integer division)
- mod
- <
- >
- = (assignment or comparision dependend on context)
- >=
- <=
- <>

- and
- not
- or

Paranthesises can be used to define the execution order.

You can use the operators + , <, <=, >, >= and = for the strings.

Comments

Text after ' or // are ignored by the interpreter until the next line.

Control Structures

The following control structures are supported:

- if condition then
  statement
  [elseif condition then
  statement]
  [else
  statement]
  end if
- select case expression
  [case expression:
  statement]
  [else
  statement]
  end select
- for counter = start to / downto end [step stepvalue]
  statement
  [exit]
  statement
  next
- while condition
  expression
  wend
- do
  expression

loop until
- return

The "if" in "end if" and the "select" in "end select" are not optional.

You can also write functions and subroutines.

- function name ([parameterlist]) as type
  statement
  [return [value of type]]
  [statement]
  end function
- Sub([parameterlist]
  statement
  [return [value of type]]
  [statement]
  end sub

Functions and subroutines must be defined before they are used.

Parameters in functions and subroutines can be defined **byref** or **byval**, byval being the default, except for arrays.

Notes:

- You cannot use function within functions.
- Do not write endless loops, as this will block your computer.
- Avoid to create very big text strings within your loops or your programm will crash.

Functions

In the following list of functions, only the textfunctions are explained. For full documentation, refer to the Realsoftware website.

Note that all string functions return a copy of the string and leave the input string unchanged.

abs(double) as double

acos(double) as double

asc(string) as integer

Returns the Ascii-value for the first character of a string.

ascb(string) as string

Returns the Ascii-value for the first byte of a string. This makes a difference for double-byte languages like japanese.

asin(double) as double

atan(double) as double

atan2(double,double) as double

bitwiseand(integer,integer) as integer

bitwiseor(integer,integer) as integer

bitwisexor(integer,integer) as integer

cdbl(string) as double

Returns the numerical value of a string

ceil(double) as double

chr(double) as string

Returns the character of a given Ascii-value.

Newline is chr(13) and tabulator is chr(9). While Windows textfiles use chr(13)+chr(10) for newline, for the filters only chr(13) is used. The parsing of the correct line separator should be done with the input and the output filters. To write platform independent code, you should better use getnewline than chr(13).

chrb(double) as string

Returns a singlebyte-character of a given Ascii-value.

closeprogress

Makes the progress bar invisible.

cos(double) as double

countfields(string1,string2) as integer

Returns the number of fields in string1, given the separator string2. The separator is not case-sensitive.

Returns 1 if the separator is not present, and 0 if the string1 is empty.

cstr(double) as string

Converts a number to a string using local conventions.

exp(double) as double

false

floor(double) as double

format(double,string) as string

Converts a number to a string using a given formatstring. The following characters are allowed:

- # placeholder showing digit, when it is present
- 0 placeholder showing digit anyway
- . placeholder for decimal point
- , placeholder for thousands separator
- % percentage whith number multiplied by 100
- (open parenthesis
-) close paranthesis
- + diplays plus sign, if number is positive or minus, if it is negative.
- - dsplays minus sign, if a number is negative
- E or e scientific notation
- \character any character

Use the following format for integers:

format(5958020,"0") = 595820

Use the following format for timecodes:

format(5958020,"00\:00\:00:\00") = 00:59:58:20

hex(integer) as string

Formats the number using hexadecimal notation.

instr(integer,string1,string2) as integer

Looks for an occurence of string2 in string1, starting at position of integer. If it finds one, it returns the position, if it doesn't, it returns zero.

Instr is not case-sensitive.

instrb(integer,string1,string2) as integer

Looks for an occurence of string2 in string1, starting at position of integer, using raw bytes. If it finds one, it returns the position, if it doesn't, it returns zero.

left(string,integer) as string

Returns the leftmost characters of a string.

leftb(string,integer) as string

Returns the leftmost characters of a string, using bytecount.

log(double) as double)

lowercase(string) as string

Returns the string with all lowercase characters.

ltrim(string) as string

Removes the spaces at the beginning of the string

max(double,double) as double

microseconds

mid(string,integer1,integer2)

Returns a substring of string starting at position integer2 and with a length of integer2.

midb(string,integer1,integer2)

Returns a substring of string starting at position integer2 and with a length of integer2, using byte count.

min(double,double) as double

nthfield(string1,string2,integer) as string

Returns the nth field in string1, using string2 as a separator. The string2 is not case-sensitive.

oct(integer) as string

Formats the number using octal (base 8) notation.

pow(double,double) as double

print string

Prints to the console.

redim array(integer)

replace(thestring,oldstring,newstring) as string

Replaces the first occurence of oldstring in thestring with newstring. The replace is not case-sensitive.

replaceall(thestring,oldstring,newstring) as string

Replaces all occurences of oldstring in thestring with newstring. The replace is not case-sensitive.

right(string,integer) as string

Returns the rightmost characters of a string

rightb(string,integer) as string

Returns the rightmost characters of a string, suing byte count

rnd as double

round(double) as double

rtrim(string) as string

Removes spaces at the end of the string

sin(double) as double)

sqrt(double) as double

str(double) as string

Converts a number to a string, using default formats. Is fast coded, but for big numbers, format(double,"0") does a better job.

strcomp(string1,string2,integer) as integer

Compares string1 with string2, using criterium integer

- 0: binary mode
- 1: text mode (lexicographic)

It returns -1, if the string2 is later in the alphabet, +1 if it is before, and 0, if both strings are the same. This function with criterium 0 is more exact than the operator =, which is case-insensitive

tan(double) as double

ticks as integer

titlecase(string) as string

Returns the string with every first character of a word in capitals.

trim(string) as string

Removes spaces at both the beginning and the end of the character

true

ubound(array)

Returns the highest index of a single-dimensioned array.

uppercase(string) as string

Returns the string all letters in capitals

val(string) as double)

Returns the numeric value of a string, using the point as decimal separator.

>Error messages

Error messages are displayed in the console window.

- 1 Syntax does not make sense.
- 2 Type mismatch.
- 3 Select Case does not support that type of expression.
- 4 The compiler is not implemented.
- 5 The parser's internal stack has overflowed.
- 6 Too many parameters for this function.
- 7 Not enough parameters for this function call.
- 8 Wrong number of parameters for this function call.
- 9 Parameters are incompatible with this function.
- 10 Assignment of incompatible data type.

- 11 Undefined identifier.
- 12 Undefined operator.
- 13 Logic operations require Boolean operands.
- 14 Array bounds must be integers.
- 15 Can't call a non-function.
- 16 Can't get an element from something that isn't an array.
- 17 Not enough subscripts for this array's dimension.
- 18 Too many subscripts for this array's dimension.
- 19 Can't assign an entire array.
- 20 Can't use an entire array in an expression.
- 21 Can't pass an expression as ByRef parameter.
- 22 Duplicate identifier.
- 23 The backend code generator failed.
- 24 Ambiguous call to overloaded method.
- 25 Multiple inheritance is not allowed.
- 26 Cannot create an instance of an interface.
- 27 Cannot imkplement a class as though it were an interface.
- 28 Cannot inherit from something that is not a class.
- 29 This class does not fully implement the specified interface.
- 30 Event handlers cannot live outside of a a class.
- 31 It is not legal to ignore the result of a function call.
- 32 Can't use the Self keyword outside of a class.
- 33 Can't use the Me keyword outside of a class.
- 34 Can't return a value from a Sub.
- 35 An exception object required here.
- 36, 37, 38, 39 Obsolete.
- 40 Destructors can't have parameters.
- 41 Can't use the Super keyword outside of a class.
- 42 Can't use the Super keyword in a class that has no parent.

Known bugs and limitations

- The application will itself avoid duplicate node names when you create new nodes, duplicate or paste them, but it will not prevent you from renaming a node to the name of an existing nodes. Duplicate node names will however break connections.
- The application does not always prevent you from creating circular references. This can lead to an infinite loop. Stop the calculation by pressing the command and the option key at the same time.
- Sometimes, the nodes are not reprocessed, even if a parameter has changed. Use Force Refresh.
- Text Input nodes outputs are always red, because they do not return a picture.
- Some fonts may not properly display certain styles in Text node and Quartz Script node. The text will then fall back to Helvetica.
- Viewer Windows have a little grow icon even when in full screen mode.

Copyright and Disclaimer

The software is provided "as-is" and without warranty of any kind, express, implied or otherwise, including without limitation, any warranty of merchandtabality for a particular purpose. In no event shall Belle Nuit Montage be liable for any special incidental, indirect or consequential damages of any kind, or any damages whatsoever resulting from loss of use, data or profits, wether or not advised of the possibility of damage, and on any theory of liability, arising out of or in connection with the use of this software.
Product specification are subject to change without notice and do not represent a commitment on the part of Belle Nuit Montage. The software described in this document is furnished under a license agreement. The software may not be reverse assembled and may be used or copied only in accordance with the terms of the license agreement. It is against the law to copy the software on any medium except as specially allowed in the license agreement.

Trademarks

Finder, QuickTime and Final Cut Pro are trademarks of Apple Computer Inc. PDF is a trademark from Adobe. All other trademarks and registered trademarks used herein are the property of their respective owners.

Comments please to matti@belle-nuit.com

History

17.6.4 1.0 release

18.5.4 - 15.6.4: 1.0 beta 1-9

Printed with Belle Nuit HTMLBook