

Natron Documentation

Release 2.3.15

The Natron documentation authors

Apr 23, 2019

Contents

1	User Guide	3
1.1	What is compositing?	3
1.1.1	Theory	3
1.1.2	Practice	4
1.2	Getting started	4
1.2.1	About	4
1.2.2	Installation	6
1.2.3	Environment	15
1.2.4	Troubleshooting	16
1.3	Compositing	17
1.3.1	Managing Projects	17
1.3.2	Reformatting elements (empty)	18
1.3.3	Channels (empty)	18
1.3.4	Merging images (empty)	18
1.3.5	Noise removal (empty)	18
1.3.6	Keying (empty)	18
1.3.7	Using Rotopaint (empty)	18
1.3.8	Tracking and stabilizing	18
1.3.9	Transforming elements (empty)	20
1.3.10	Working with color (empty)	20
1.3.11	Stereoscopic compositing	20
1.3.12	Preview and rendering (empty)	24
1.3.13	Expressions (empty)	24
1.4	Tutorials	24
1.4.1	Writing documentation	24
1.4.2	HSVTool	25
1.4.3	FFmpeg (empty)	31
1.4.4	Using PanoTools projects within Natron	33
1.4.5	Vector graphics workflow	36
1.4.6	Alternative Matte Extraction Tutorial	42
2	Reference Guide	47
2.1	Preferences	47
2.1.1	General	47
2.1.2	Threading	47
2.1.3	Rendering	48
2.1.4	GPU Rendering	49
2.1.5	Project Setup	49
2.1.6	Documentation	49
2.1.7	User Interface	49
2.1.8	Color Management	50

2.1.9	Caching	50
2.1.10	Viewer	51
2.1.11	Nodegraph	52
2.1.12	Plug-ins	52
2.1.13	Python	53
2.1.14	Appearance	53
2.2	Image nodes	55
2.2.1	CheckerBoard node	55
2.2.2	ColorBars node	57
2.2.3	ColorWheel node	59
2.2.4	Constant node	61
2.2.5	OpenRaster node	62
2.2.6	Read node	67
2.2.7	ReadCDR node	71
2.2.8	ReadFFmpeg node	75
2.2.9	ReadKrita node	80
2.2.10	ReadMisc node	85
2.2.11	ReadOIIO node	90
2.2.12	ReadPDF node	98
2.2.13	ReadPFM node	102
2.2.14	ReadPNG node	107
2.2.15	ReadPSD node	111
2.2.16	ReadSVG node	122
2.2.17	RunScript node	127
2.2.18	Solid node	133
2.2.19	Write node	135
2.2.20	WriteFFmpeg node	137
2.2.21	WriteOIIO node	148
2.2.22	WritePFM node	155
2.2.23	WritePNG node	159
2.3	Draw nodes	164
2.3.1	LightWrap node	164
2.3.2	Noise node	169
2.3.3	Plasma node	170
2.3.4	Radial node	171
2.3.5	Ramp node	173
2.3.6	Rand node	174
2.3.7	Rectangle node	176
2.3.8	Roto node	177
2.3.9	RotoPaint node	180
2.3.10	SeGrain node	183
2.3.11	SeNoise node	185
2.3.12	Text node	188
2.4	Time nodes	191
2.4.1	AppendClip node	191
2.4.2	Deinterlace node	192
2.4.3	FrameBlend node	193
2.4.4	FrameHold node	194
2.4.5	FrameRange node	195
2.4.6	NoTimeBlur node	196
2.4.7	Retime node	196
2.4.8	SlitScan node	197
2.4.9	TimeBlur node	199
2.4.10	TimeOffset node	200
2.5	Channel nodes	200
2.5.1	Shuffle node	200
2.6	Color nodes	204
2.6.1	Add node	204

2.6.2	Clamp node	204
2.6.3	ClipTest node	206
2.6.4	ColorCorrect node	206
2.6.5	ColorLookup node	209
2.6.6	ColorMatrix node	212
2.6.7	ColorSuppress node	213
2.6.8	Equalize node	214
2.6.9	Gamma node	215
2.6.10	Grade node	215
2.6.11	HSIToRGB node	217
2.6.12	HSLToRGB node	218
2.6.13	HSVToRGB node	218
2.6.14	HSVTool node	219
2.6.15	HistEQ node	221
2.6.16	HueCorrect node	222
2.6.17	Invert node	223
2.6.18	LabToRGB709 node	224
2.6.19	LabToXYZ node	224
2.6.20	Log2Lin node	225
2.6.21	Multiply node	226
2.6.22	OCIOCDLTransform node	226
2.6.23	OCIOColorSpace node	228
2.6.24	OCIODisplay node	230
2.6.25	OCIOFileTransform node	233
2.6.26	OCIOLogConvert node	235
2.6.27	OCIOLookTransform node	236
2.6.28	PLogLin node	241
2.6.29	Quantize node	242
2.6.30	RGB709ToLab node	243
2.6.31	RGB709ToXYZ node	244
2.6.32	RGBToHSI node	244
2.6.33	RGBToHSL node	245
2.6.34	RGBToHSV node	245
2.6.35	RGBToYCbCr601 node	246
2.6.36	RGBToYCbCr709 node	246
2.6.37	RGBToYPbPr601 node	247
2.6.38	RGBToYPbPr709 node	247
2.6.39	RGBToYUV601 node	248
2.6.40	RGBToYUV709 node	248
2.6.41	Saturation node	249
2.6.42	VectorToColor node	250
2.6.43	XYZToLab node	250
2.6.44	XYZToRGB709 node	251
2.6.45	XYZToxyY node	251
2.6.46	YCbCrToRGB601 node	252
2.6.47	YCbCrToRGB709 node	252
2.6.48	YPbPrToRGB601 node	253
2.6.49	YPbPrToRGB709 node	253
2.6.50	YUVToRGB601 node	254
2.6.51	YUVToRGB709 node	254
2.6.52	xyYToXYZ node	255
2.7	Filter nodes	255
2.7.1	AngleBlur node	255
2.7.2	Bloom node	256
2.7.3	Blur node	258
2.7.4	ChromaBlur node	260
2.7.5	DenoiseSharpen node	261
2.7.6	Dilate node	267

2.7.7	DirBlur node	268
2.7.8	Distance node	269
2.7.9	DropShadow node	270
2.7.10	EdgeBlur node	271
2.7.11	EdgeDetect node	272
2.7.12	EdgeExtend node	274
2.7.13	Erode node	276
2.7.14	ErodeBlur node	276
2.7.15	ErodeSmooth node	277
2.7.16	Fill node	279
2.7.17	GMICExpr node	281
2.7.18	Glow node	287
2.7.19	GodRays node	290
2.7.20	Inpaint node	292
2.7.21	Laplacian node	293
2.7.22	Matrix3x3 node	294
2.7.23	Matrix5x5 node	295
2.7.24	Median node	296
2.7.25	Shadertoy node	297
2.7.26	Sharpen node	336
2.7.27	SharpenInvDiff node	337
2.7.28	SharpenShock node	338
2.7.29	SmoothAnisotropic node	339
2.7.30	SmoothBilateral node	340
2.7.31	SmoothBilateralGuided node	341
2.7.32	SmoothGuided node	342
2.7.33	SmoothPatchBased node	343
2.7.34	SmoothRollingGuidance node	344
2.7.35	Soften node	345
2.7.36	ZMask node	346
2.7.37	ZRemap node	347
2.8	Keyer nodes	348
2.8.1	ChromaKeyer node	348
2.8.2	Despill node	350
2.8.3	Difference node	351
2.8.4	HueKeyer node	352
2.8.5	Keyer node	353
2.8.6	MatteMonitor node	355
2.8.7	PIK node	356
2.8.8	PIKColor node	360
2.9	Merge nodes	362
2.9.1	Absminus node	362
2.9.2	ContactSheet node	362
2.9.3	CopyRectangle node	363
2.9.4	Dissolve node	364
2.9.5	In node	364
2.9.6	KeyMix node	364
2.9.7	LayerContactSheet node	365
2.9.8	Matte node	366
2.9.9	Max node	366
2.9.10	Merge node	366
2.9.11	Min node	370
2.9.12	Multiply node	370
2.9.13	Out node	370
2.9.14	Plus node	370
2.9.15	Premult node	371
2.9.16	RotoMerge node	372
2.9.17	Screen node	376

2.9.18	SeExpr node	376
2.9.19	SeExprSimple node	383
2.9.20	Switch node	390
2.9.21	TimeDissolve node	391
2.9.22	Unpremult node	391
2.10	Transform nodes	392
2.10.1	AdjustRoD node	392
2.10.2	Card3D node	393
2.10.3	CornerPin node	400
2.10.4	CornerPinMasked node	403
2.10.5	Crop node	406
2.10.6	IDistort node	407
2.10.7	LensDistortion node	410
2.10.8	Mirror node	415
2.10.9	Position node	416
2.10.10	Reformat node	417
2.10.11	Resize node	419
2.10.12	STMap node	421
2.10.13	SpriteSheet node	424
2.10.14	Tracker node	424
2.10.15	Transform node	427
2.10.16	TransformMasked node	429
2.11	Views nodes	431
2.11.1	Anaglyph node	431
2.11.2	JoinViews node	432
2.11.3	MixViews node	432
2.11.4	OneView node	433
2.11.5	SideBySide node	433
2.12	Other nodes	434
2.12.1	Backdrop node	434
2.12.2	DiskCache node	434
2.12.3	Dot node	435
2.12.4	Group node	435
2.12.5	ImageStatistics node	436
2.12.6	Input node	439
2.12.7	NoOp node	440
2.12.8	Output node	442
2.12.9	Precomp node	442
2.13	GMIC nodes	443
2.13.1	About G'MIC node	443
2.13.2	G'MIC 3d blocks node	444
2.13.3	G'MIC 3d colored object node	446
2.13.4	G'MIC 3d elevation node	448
2.13.5	G'MIC 3d extrusion node	449
2.13.6	G'MIC 3d image object node	451
2.13.7	G'MIC 3d lathing node	453
2.13.8	G'MIC 3d random objects node	455
2.13.9	G'MIC 3d text pointcloud node	457
2.13.10	G'MIC 3d tiles node	458
2.13.11	G'MIC Abstraction node	460
2.13.12	G'MIC Add grain node	462
2.13.13	G'MIC Align layers node	464
2.13.14	G'MIC Array faded node	466
2.13.15	G'MIC Array mirrored node	467
2.13.16	G'MIC Array random colors node	469
2.13.17	G'MIC Array regular node	470
2.13.18	G'MIC Ascii art node	472
2.13.19	G'MIC B&W films node	474

2.13.20	G'MIC B&W films collage node	477
2.13.21	G'MIC B&W stencil node	478
2.13.22	G'MIC Ball node	480
2.13.23	G'MIC Bandpass node	481
2.13.24	G'MIC Barnsley fern node	484
2.13.25	G'MIC Basic adjustments node	485
2.13.26	G'MIC Black & white node	487
2.13.27	G'MIC Blend average all node	490
2.13.28	G'MIC Blend edges node	491
2.13.29	G'MIC Blend fade node	492
2.13.30	G'MIC Blend median node	494
2.13.31	G'MIC Blend seamless node	495
2.13.32	G'MIC Blend standard node	497
2.13.33	G'MIC Blur angular node	501
2.13.34	G'MIC Blur bloom node	503
2.13.35	G'MIC Blur depth-of-field node	507
2.13.36	G'MIC Blur gaussian node	508
2.13.37	G'MIC Blur glow node	512
2.13.38	G'MIC Blur linear node	515
2.13.39	G'MIC Blur radial node	518
2.13.40	G'MIC Bokeh node	520
2.13.41	G'MIC Boost chromaticity node	523
2.13.42	G'MIC Boost-fade node	525
2.13.43	G'MIC Box fitting node	526
2.13.44	G'MIC Brushify node	528
2.13.45	G'MIC Burn node	530
2.13.46	G'MIC Camouflage node	533
2.13.47	G'MIC Canvas node	534
2.13.48	G'MIC Cartesian transform node	536
2.13.49	G'MIC Cartoon node	538
2.13.50	G'MIC Channel processing node	540
2.13.51	G'MIC Channels to layers node	544
2.13.52	G'MIC Charcoal node	545
2.13.53	G'MIC Chessboard node	547
2.13.54	G'MIC Chromatic aberrations node	549
2.13.55	G'MIC Circle abstraction node	551
2.13.56	G'MIC Circle art node	553
2.13.57	G'MIC Circle transform node	554
2.13.58	G'MIC Color balance node	556
2.13.59	G'MIC Color blindness node	558
2.13.60	G'MIC Colorize lineart auto-fill node	560
2.13.61	G'MIC Colorize lineart propagation node	561
2.13.62	G'MIC Colorize lineart smart coloring node	563
2.13.63	G'MIC Colorize photographs node	566
2.13.64	G'MIC Colorize with colormap node	567
2.13.65	G'MIC Colormap node	570
2.13.66	G'MIC Colors to layers node	573
2.13.67	G'MIC Conformal maps node	574
2.13.68	G'MIC Contrast swiss mask node	576
2.13.69	G'MIC Convolve node	577
2.13.70	G'MIC Cracks node	581
2.13.71	G'MIC Crease node	584
2.13.72	G'MIC Crystal node	585
2.13.73	G'MIC Crystal background node	587
2.13.74	G'MIC Cubism node	588
2.13.75	G'MIC Cupid node	590
2.13.76	G'MIC Curvature node	591
2.13.77	G'MIC Customize CLUT node	593

2.13.78	G'MIC Cutout node	601
2.13.79	G'MIC Decompose channels node	603
2.13.80	G'MIC Deinterlace node	604
2.13.81	G'MIC Details equalizer node	606
2.13.82	G'MIC Detect skin node	610
2.13.83	G'MIC Dices node	612
2.13.84	G'MIC Difference of gaussians node	613
2.13.85	G'MIC Diffusion tensors node	615
2.13.86	G'MIC Dirty node	617
2.13.87	G'MIC Distance transform node	620
2.13.88	G'MIC Distort lens node	622
2.13.89	G'MIC Dithering node	623
2.13.90	G'MIC Drawn montage node	625
2.13.91	G'MIC Drop shadow node	627
2.13.92	G'MIC Drop shadow 3d node	628
2.13.93	G'MIC Drop water node	630
2.13.94	G'MIC Droste node	631
2.13.95	G'MIC Edges node	633
2.13.96	G'MIC Edges offsets node	635
2.13.97	G'MIC Edges on fire node	637
2.13.98	G'MIC Ellipsionism node	639
2.13.99	G'MIC Engrave node	641
2.13.100	G'MIC Equalize HSI-HSL-HSV node	643
2.13.101	G'MIC Equalize HSV node	646
2.13.102	G'MIC Equalize local histograms node	647
2.13.103	G'MIC Equalize shadow node	651
2.13.104	G'MIC Equation plot parametric node	652
2.13.105	G'MIC Equation plot $y=f(x)$ node	654
2.13.106	G'MIC Equirectangular to nadir-zenith node	656
2.13.107	G'MIC Euclidean polar node	657
2.13.108	G'MIC Extract objects node	658
2.13.109	G'MIC Fade layers node	660
2.13.110	G'MIC Felt pen node	661
2.13.111	G'MIC Flip & rotate blocs node	663
2.13.112	G'MIC Flower node	666
2.13.113	G'MIC Fourier analysis node	667
2.13.114	G'MIC Fourier transform node	668
2.13.115	G'MIC Fourier watermark node	670
2.13.116	G'MIC Frame blur node	671
2.13.117	G'MIC Frame fuzzy node	673
2.13.118	G'MIC Frame painting node	675
2.13.119	G'MIC Frame pattern node	677
2.13.120	G'MIC Frame regular node	678
2.13.121	G'MIC Frame round node	680
2.13.122	G'MIC Frame smooth node	681
2.13.123	G'MIC Freaky B&W node	682
2.13.124	G'MIC Freaky details node	684
2.13.125	G'MIC Fuji xtrans node	687
2.13.126	G'MIC Games & demos node	689
2.13.127	G'MIC Gear node	691
2.13.128	G'MIC Gradient RGB node	692
2.13.129	G'MIC Gradient corners node	694
2.13.130	G'MIC Gradient custom shape node	695
2.13.131	G'MIC Gradient from line node	698
2.13.132	G'MIC Gradient linear node	700
2.13.133	G'MIC Gradient norm node	701
2.13.134	G'MIC Gradient radial node	703
2.13.135	G'MIC Gradient random node	704

2.13.136G'MIC Grid cartesian node	706
2.13.137G'MIC Grid hexagonal node	707
2.13.138G'MIC Grid triangular node	708
2.13.139G'MIC Halftone node	710
2.13.140G'MIC Hard sketch node	712
2.13.141G'MIC Heart node	714
2.13.142G'MIC Hearts node	715
2.13.143G'MIC Highlight bloom node	718
2.13.144G'MIC Histogram analysis node	720
2.13.145G'MIC Hope poster node	722
2.13.146G'MIC Hough sketch node	724
2.13.147G'MIC Illuminate 2d shape node	726
2.13.148G'MIC Import data node	729
2.13.149G'MIC Ink wash node	730
2.13.150G'MIC Inpaint holes node	732
2.13.151G'MIC Inpaint morphological node	733
2.13.152G'MIC Inpaint multi-scale node	734
2.13.153G'MIC Inpaint patch-based node	736
2.13.154G'MIC Inpaint transport-diffusion node	737
2.13.155G'MIC Instant collage consumer node	739
2.13.156G'MIC Instant collage pro node	740
2.13.157G'MIC Instant consumer node	742
2.13.158G'MIC Instant pro node	746
2.13.159G'MIC Intarsia node	750
2.13.160G'MIC Isophotes node	752
2.13.161G'MIC JPEG artefacts node	754
2.13.162G'MIC Kaleidoscope blended node	755
2.13.163G'MIC Kaleidoscope polar node	756
2.13.164G'MIC Kaleidoscope symmetry node	758
2.13.165G'MIC Kuwahara node	759
2.13.166G'MIC Laplacian node	762
2.13.167G'MIC Lava node	764
2.13.168G'MIC Lava lamp node	766
2.13.169G'MIC Layers to tiles node	767
2.13.170G'MIC Light glow node	768
2.13.171G'MIC Light leaks node	772
2.13.172G'MIC Light patch node	774
2.13.173G'MIC Light rays node	776
2.13.174G'MIC Lightning node	777
2.13.175G'MIC Linify node	779
2.13.176G'MIC Lissajous node	781
2.13.177G'MIC Local normalization node	783
2.13.178G'MIC Local orientation node	786
2.13.179G'MIC Local processing node	789
2.13.180G'MIC Lomo node	792
2.13.181G'MIC Lylejk's painting node	793
2.13.182G'MIC Magic details node	795
2.13.183G'MIC Make seamless diffusion node	798
2.13.184G'MIC Make seamless patch-based node	800
2.13.185G'MIC Mandelbrot Julia sets node	802
2.13.186G'MIC Marble node	804
2.13.187G'MIC Maze node	805
2.13.188G'MIC Mighty details node	807
2.13.189G'MIC Mineral mosaic node	810
2.13.190G'MIC Ministeck node	811
2.13.191G'MIC Mixer CMYK node	812
2.13.192G'MIC Mixer HSV node	815
2.13.193G'MIC Mixer Lab node	817

2.13.194G'MIC Mixer RGB node	819
2.13.195G'MIC Mixer YCbCr node	821
2.13.196G'MIC Montage node	823
2.13.197G'MIC Morph layers node	825
2.13.198G'MIC Morphological filter node	827
2.13.199G'MIC Mosaic node	830
2.13.200G'MIC Multiscale operator node	833
2.13.201G'MIC Negative collage color node	835
2.13.202G'MIC Negative collage new node	836
2.13.203G'MIC Negative collage old node	838
2.13.204G'MIC Negative color node	839
2.13.205G'MIC Negative new node	842
2.13.206G'MIC Negative old node	844
2.13.207G'MIC Neon lightning node	847
2.13.208G'MIC Noise additive node	848
2.13.209G'MIC Noise spread node	851
2.13.210G'MIC Old photograph node	854
2.13.211G'MIC Old-movie stripes node	855
2.13.212G'MIC Oldschool 8bits node	858
2.13.213G'MIC Op art node	859
2.13.214G'MIC Pack node	862
2.13.215G'MIC Pack sprites node	863
2.13.216G'MIC Painting node	865
2.13.217G'MIC Paper texture node	867
2.13.218G'MIC Pen drawing node	870
2.13.219G'MIC Pencil node	871
2.13.220G'MIC Pencil portrait node	873
2.13.221G'MIC Perspective node	875
2.13.222G'MIC PictureFX node	876
2.13.223G'MIC PictureFX collage node	879
2.13.224G'MIC Pixel sort node	880
2.13.225G'MIC Plaid node	882
2.13.226G'MIC Plasma node	883
2.13.227G'MIC Polar transform node	885
2.13.228G'MIC Polaroid node	886
2.13.229G'MIC Polka dots node	888
2.13.230G'MIC Polygonize delaunay node	889
2.13.231G'MIC Polygonize energy node	891
2.13.232G'MIC Pop shadows node	893
2.13.233G'MIC Poster edges node	895
2.13.234G'MIC Posterize node	897
2.13.235G'MIC Print films node	899
2.13.236G'MIC Print films collage node	901
2.13.237G'MIC Puzzle node	902
2.13.238G'MIC Quadrangle node	904
2.13.239G'MIC Quadtree variations node	906
2.13.240G'MIC Quick copyright node	908
2.13.241G'MIC Rain & snow node	910
2.13.242G'MIC Rainbow node	911
2.13.243G'MIC Random color ellipses node	912
2.13.244G'MIC Random shade stripes node	914
2.13.245G'MIC Reflection node	917
2.13.246G'MIC Relief light node	919
2.13.247G'MIC Remove hot pixels node	920
2.13.248G'MIC Resynthesize texture FFT node	922
2.13.249G'MIC Resynthesize texture patch-based node	924
2.13.250G'MIC Retinex node	926
2.13.251G'MIC Retro fade node	928

2.13.252G'MIC Rodilius node	929
2.13.253G'MIC Rorschach node	933
2.13.254G'MIC Sample image node	934
2.13.255G'MIC Satin node	936
2.13.256G'MIC Scanlines node	938
2.13.257G'MIC Seamcarve node	941
2.13.258G'MIC Seamless turbulence node	942
2.13.259G'MIC Segmentation node	943
2.13.260G'MIC Select-replace color node	947
2.13.261G'MIC Selective desaturation node	949
2.13.262G'MIC Sepia node	951
2.13.263G'MIC Shade bobs node	953
2.13.264G'MIC Shadow patch node	954
2.13.265G'MIC Shapeism node	956
2.13.266G'MIC Sharp abstract node	958
2.13.267G'MIC Sharpen Gold-Meinell node	961
2.13.268G'MIC Sharpen Richardson-Lucy node	964
2.13.269G'MIC Sharpen deblur node	966
2.13.270G'MIC Sharpen inv. diffusion node	969
2.13.271G'MIC Sharpen octave sharpening node	972
2.13.272G'MIC Sharpen shock filters node	975
2.13.273G'MIC Sharpen texture node	978
2.13.274G'MIC Sharpen unsharp mask node	981
2.13.275G'MIC Shock waves node	984
2.13.276G'MIC Sierpinski triangle node	987
2.13.277G'MIC Skeleton node	988
2.13.278G'MIC Sketch node	990
2.13.279G'MIC Slice luminosity node	992
2.13.280G'MIC Slide collage color node	994
2.13.281G'MIC Slide color node	996
2.13.282G'MIC Smooth abstract node	999
2.13.283G'MIC Smooth anisotropic node	1001
2.13.284G'MIC Smooth antialias node	1004
2.13.285G'MIC Smooth bilateral node	1006
2.13.286G'MIC Smooth diffusion node	1009
2.13.287G'MIC Smooth guided node	1012
2.13.288G'MIC Smooth mean-curvature node	1015
2.13.289G'MIC Smooth median node	1018
2.13.290G'MIC Smooth nlmeans node	1021
2.13.291G'MIC Smooth patch-based node	1024
2.13.292G'MIC Smooth patch-pca node	1027
2.13.293G'MIC Smooth perona-malik node	1030
2.13.294G'MIC Smooth selective gaussian node	1033
2.13.295G'MIC Smooth skin node	1036
2.13.296G'MIC Smooth total variation node	1038
2.13.297G'MIC Smooth wavelets node	1043
2.13.298G'MIC Snowflake node	1046
2.13.299G'MIC Solidify node	1047
2.13.300G'MIC Solve maze node	1049
2.13.301G'MIC Spatial transition node	1050
2.13.302G'MIC Sphere node	1052
2.13.303G'MIC Spherize node	1054
2.13.304G'MIC Split details alpha node	1055
2.13.305G'MIC Split details gaussian node	1056
2.13.306G'MIC Split details wavelets node	1058
2.13.307G'MIC Sponge node	1059
2.13.308G'MIC Square to circle node	1062
2.13.309G'MIC Stained glass node	1063

2.13.310G'MIC Stamp node	1065
2.13.311G'MIC Stars node	1067
2.13.312G'MIC Stencil node	1068
2.13.313G'MIC Stereographic projection node	1070
2.13.314G'MIC Streak node	1072
2.13.315G'MIC Stroke node	1073
2.13.316G'MIC Super-pixels node	1075
2.13.317G'MIC Superformula node	1077
2.13.318G'MIC Symmetrize node	1078
2.13.319G'MIC Taquin node	1080
2.13.320G'MIC Tetris node	1081
2.13.321G'MIC Textured glass node	1082
2.13.322G'MIC Thin edges node	1084
2.13.323G'MIC Tileable rotation node	1086
2.13.324G'MIC Tiled isolation node	1087
2.13.325G'MIC Tiled normalization node	1088
2.13.326G'MIC Tiled parameterization node	1091
2.13.327G'MIC Tiled random shifts node	1092
2.13.328G'MIC Tiled rotation node	1093
2.13.329G'MIC Tiles to layers node	1095
2.13.330G'MIC Tone mapping node	1096
2.13.331G'MIC Tone mapping fast node	1099
2.13.332G'MIC Tones to layers node	1102
2.13.333G'MIC Transfer colors advanced node	1103
2.13.334G'MIC Transfer colors basic node	1106
2.13.335G'MIC Truchet node	1107
2.13.336G'MIC Tunnel node	1108
2.13.337G'MIC Turbulence node	1110
2.13.338G'MIC Twirl node	1111
2.13.339G'MIC Upscale dcci2x node	1112
2.13.340G'MIC Upscale diffusion node	1114
2.13.341G'MIC Upscale scale2x node	1115
2.13.342G'MIC User-defined node	1117
2.13.343G'MIC Various node	1118
2.13.344G'MIC Various collage node	1123
2.13.345G'MIC Vector painting node	1124
2.13.346G'MIC Vignette node	1126
2.13.347G'MIC Visible watermark node	1127
2.13.348G'MIC Warp by intensity node	1128
2.13.349G'MIC Whirl drawing node	1132
2.13.350G'MIC Whirls node	1134
2.13.351G'MIC Wind node	1137
2.13.352G'MIC Zoom node	1140
2.14 Extra nodes	1141
2.14.1 Arc node	1141
2.14.2 Charcoal node	1142
2.14.3 Edges node	1143
2.14.4 Implode node	1144
2.14.5 Modulate node	1145
2.14.6 Oilpaint node	1146
2.14.7 Polar node	1146
2.14.8 Polaroid node	1147
2.14.9 Reflection node	1148
2.14.10 Roll node	1149
2.14.11 Sketch node	1150
2.14.12 Swirl node	1151
2.14.13 Texture node	1152
2.14.14 Tile node	1153

2.14.15	Wave node	1154
3	Developers Guide	1157
3.1	Python API	1157
3.1.1	NatronEngine	1157
3.1.2	NatronGui	1258
3.2	Introduction	1279
3.2.1	Natron plug-in paths	1279
3.2.2	Python Auto-declared variables	1280
3.2.3	Start-up scripts	1283
3.2.4	Examples	1283
3.2.5	Natron in command-line	1284
3.2.6	Example	1287
3.2.7	Objects hierarchy Overview	1289
3.3	Getting started	1290
3.3.1	Creating and controlling nodes	1290
3.3.2	Controlling parameters	1292
3.3.3	Parameters expressions	1294
3.3.4	Working with groups	1301
3.3.5	Using Callbacks	1311
3.3.6	Rendering	1321
3.3.7	Using the rotoscoping functionalities	1321
3.3.8	Using the tracker functionalities	1323
3.3.9	Modal dialogs	1324
3.3.10	User menu commands	1326
3.3.11	PySide panels	1327
3.3.12	Controlling the viewer	1332
	Bibliography	1333
	Python Module Index	1335

The Natron documentation is organized as follows:

- The *User Guide* contains everything to get started with Natron, including tutorials.
- The *Reference Guide* contains the documentation about the user settings in Natron, as well as the documentation
- The *Developers Guide* contains documentation about using the Python scripting language within Natron, and using the Natron command-line interface (a.k.a. NatronRenderer).

This documentation is also [available online](#) and can be downloaded as a [PDF](#), [HTML zip](#) or [ePub](#) file.

This manual is maintained largely by volunteers.

The [Creative Commons Attribution-ShareAlike 4.0 International License \(CC-BY-SA 4.0\)](#) is used for this manual, which is a free and open license. Though there are certain restrictions that come with this license you may in general freely reproduce it and even make changes to it. However, rather than distribute your own version of this manual, we would much prefer if you would send any corrections or changes to the Natron project.

The *Natron documentation authors* are:

- Alexandre Gauthier-Foichat
- Ole-André Rodlie
- Omar Brown
- Frédéric Devernay

Natron is an open source video compositing and special effects software for Windows, macOS and Linux.

This guide will help you getting started using Natron for compositing and visual effects.

1.1 What is compositing?

Compositing is the combining of visual elements from separate sources into single images, often to create the illusion that all those elements are parts of the same scene [[Wikipedia](#)].

Typical examples of compositing are, for example:

- The superimposition of a character filmed on a green background over a scene shot in another place, at another time, or a computer-generated scene;
- The manual detouring (also called rotoscoping) of an element in a video to embed it in another video, possibly with a different motion;
- Artistic modifications of a video, after shooting a live-action scene or rendering a CGI scene, in order to modify its lighting, colors, depth of field, camera motion, or to remove noise or add film grain.

A video compositing software is not a 3D computer graphics software, like Blender or Maya, but it is perfectly suited for combining computer-generated elements produced by other software with live-action video or 2D animation. Rather than rendering a full 3D scene with the 3D software, which may cost many hours of computation, the video compositing software can assemble the elements produced separately with a much more reactive interface and an almost instantaneous visual feedback.

1.1.1 Theory

The math behind compositing was formalized by Porter & Duff [[PorterDuff1984](#)] after the preliminary work by Wallace [[Wallace1981](#)]. More informing about the theory behind compositing can be found in the works of Jim Blinn [[Blinn1994a](#)] [[Blinn1994b](#)] and Alvy Ray Smith [[Smith1995](#)].

The compositing theory also introduces the notion of “premultiplied” RGB values, or “associated alpha”, and there is still a lot of [debate](#) about [premultiplying or not](#).

Natron made the choice of using premultiplied alpha by default in the compositing workflow, like all modern compositing software, because images are stored internally with floating-point values.

1.1.2 Practice

There are excellent books that introduce how to do compositing in practice, and using compositing software: [Wright2010], [Brinkmann2008], [Lanier2009], [VES2014].

Most of what is described in these books also apply to Natron. It is thus strongly recommended to become familiar with the techniques and workflows described in these books before starting to use Natron.

There are also video tutorials available on video streaming platforms (youtube, vimeo) for Natron or other reference compositing software, such as Nuke or Fusion. These tutorials can be used to get acquainted with compositing.

1.2 Getting started

1.2.1 About

Features

- 32 bits floating point linear color processing pipeline.
- Colorspace management handled by the famous open-source OpenColorIO library.
- Dozens of file formats supported: EXR, DPX, TIFF, PSD, SVG, Raw, JPG, PNG ...
- **Support for many free and open-source OpenFX plugins:** * [OpenFX-IO](#) * [OpenFX-Misc](#) * [OpenFX-Arena](#) * [OpenFX-G'MIC](#) * [OpenFX-OpenCV](#) * [OpenFX-Yadif](#) deinterlacer * [OpenFX-Vegas](#) SDK samples * [OpenFX](#) samples * [TuttleOFX](#)
- **Support for commercial OpenFX plugins:** * [RevisionFX](#) products * [NeatVideo](#) denoiser * [Furnace](#) by The Foundry * [KeyLight](#) by The Foundry * [GenArts Sapphire](#) * Other [GenArts](#) products * And many more... * [OpenFX v1.4](#) supported
- **Intuitive user interface:** Natron aims not to break habits by providing an intuitive and familiar user interface. It is possible to separate on any number of screens the graphical user interface. It supports Retina screens on MacOSX.
- **Performances:** Never wait for anything to be rendered, in Natron anything you do produces real-time feedback thanks to its optimised multi-threaded rendering pipeline and its support for proxy rendering (i.e: the render pipeline can be computed at lower res to speed-up rendering).
- **Multi-task:** Natron can render multiple graphs at the same time and make use of 100% of the compute power of your CPU.
- **Network rendering:** Natron can be used as a command-line tool and can be integrated on a render farm manager such as [Afanasy](#).
- **NatronRenderer:** A command line tool for execution of project files and python scripts. The command line version is executable from ssh on a computer without any display.
- **Fast & interactive Viewer** – Smooth & accurate zooming/panning even for very large image sizes (tested on 27k x 30k images).
- **Real-time playback:** Natron offers a real-time playback with thanks to its RAM/Disk cache technology. Once a frame is rendered, it can be reproduced instantly afterwards, even for large image sizes.
- **Low hardware requirements:** All you need is an x86 64 bits or 32 bits processor, at least 3 GB of RAM and a graphic card that supports OpenGL 2.0 or OpenGL 1.5 with some extensions.
- **Motion editing:** Natron offers a simple and efficient way to deal with keyframes with a very accurate and intuitive curve editor. You can set expressions on animation curves to create easy and believable motion for objects. Natron also incorporates a fully featured dope-sheet to quickly edit clips and keyframes in time-space.

- **Multi-view workflow:** Natron saves time by keeping all the views in the same stream. You can separate the views at any time with the OneView node.
- **Rotoscoping/Rotopainting:** Edit your masks and animate them to work with complex shots
- **Tracker node:** A point tracker is embedded in Natron to track multiple points. Version 2.1 of Natron will incorporate the Tracker from Blender.

FAQ

Can I use Natron for commercial work?

Yes. Anything you create with Natron is yours and you're free to do anything you want with it.

What operating systems are supported by Natron?

Natron officially supports:

- Windows 7, 8 and 10 with latest service packs.
- MacOSX 10.6 or greater
- Linux 2.6.18 or greater (Glibc 2.12+/libgcc 4.4+)

Why did you make Natron free of charge?

Our original motives were to create a tool for people who needed it and that may felt left-aside by the software editors pricing plans, that is:

- Students who want to learn compositing at home
- Schools that may not be able to buy expensive software licenses

Another reason why we develop Natron here at [INRIA](#) is because a compositing software is a playground that enables scientists in computer vision/graphics to develop, test exchange and publish results easily on such platform.

One great mission of a free open-source software is to aim to create common practises so everyone can benefit of it.

On the other hand, being free of charge, Natron can be installed on large-scale render farms without wondering about licensing issues.

What is OpenFX?

[OpenFX](#) is a standard for creating visual effects plug-ins for compositing and editor applications.

As of today several applications are compatible with this plug-in format: (meaning you can use the same plug-ins in all of them)

- Nuke 5.1+, by The Foundry
- Vegas 10+, by Sony
- SCRATCH 6.1+, by Assimilate
- Fusion 5.1+, by Blackmagic Design (formerly by eyeon)
- DaVinci Resolve 10+, by Blackmagic Design
- DustBuster+ 4.5+, by HS-ART
- Baselight 2.2+ by FilmLight
- Nucoda Film Master 2011.2.058+

- SGO Mistika 6.5.35+
- Autodesk Toxik 2009+
- Avid DS 10.3+
- Natron
- ButtleOFX
- TuttleOFX

Can I use commercial and proprietary plug-ins within Natron?

Yes. Natron doesn't limit you to open-source plug-ins.

Is my graphics card supported?

An OpenGL 2.0 compatible graphics card is needed to run Natron (2.1+) with hardware-accelerated rendering. Other graphics cards work with software-only rendering (see below).

The following graphics cards are supported for hardware-accelerated rendering:

- Intel GMA 3150 (Linux-only)
- Intel GMA X3xxx (Linux-only)
- Intel GMA X4xxx (Windows 7 & Linux)
- Intel HD (Ironlake) (Windows 7 & Linux)
- Intel HD 2000/3000 (Sandy Bridge) (Windows 7/Linux/Mac)
- Intel HD 4000 and greater (All platforms)
- Nvidia GeForce 6 series and greater
- Nvidia Quadro FX and greater
- Nvidia Quadro NVS 285 and greater
- ATI/AMD Radeon R300 and greater
- ATI/AMD FireGL T2-64 and greater (FirePro)

Cards not listed here will probably not support hardware-accelerated rendering.

On Windows and Linux you can enable software rendering. On Linux, enable the environment variable `LIBGL_ALWAYS_SOFTWARE=1` before running Natron. On Windows, enable the legacy hardware package in the installer.

1.2.2 Installation

This chapter will guide you through the installation of Natron on Windows, Mac and Linux.

Windows

This chapter will guide your through the installation and maintenance of Natron on Windows.

Requirements

Natron will work on Windows 7, 8.x, 10 with latest updates.

The basic requirements are:

- x86 compatible CPU (Core2 x86_64 or higher recommended)
- **OpenGL 2.0 or higher with the following extensions:**
 - **GL_ARB_texture_non_power_of_two** (*Viewer and OpenGL rendering*)
 - **GL_ARB_shader_objects** (*Viewer and OpenGL rendering*)
 - **GL_ARB_vertex_buffer_object** (*Viewer and OpenGL rendering*)
 - **GL_ARB_pixel_buffer_object** (*Viewer and OpenGL rendering*)
 - **GL_ARB_vertex_array_object** or **GL_APPLE_vertex_array_object** (*OpenGL rendering only*)
 - **GL_ARB_framebuffer_object** or **GL_EXT_framebuffer_object** (*OpenGL rendering only*)
 - **GL_ARB_texture_float** (*OpenGL rendering only*)

If you don't have the minimum required OpenGL extensions we provide a Software OpenGL solution, install the package *Software OpenGL* from the installer. If you have the portable ZIP file copy *bin\mesa\opengl32.dll* to *bin*.

Download



Navigate to <https://natrongithub.github.io/#download> and download the latest version. This documentation will assume that you downloaded the installer (our default and recommended choice).





Download



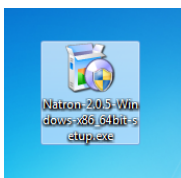
[Changelog](#) for this version.

Alternative Downloads:  

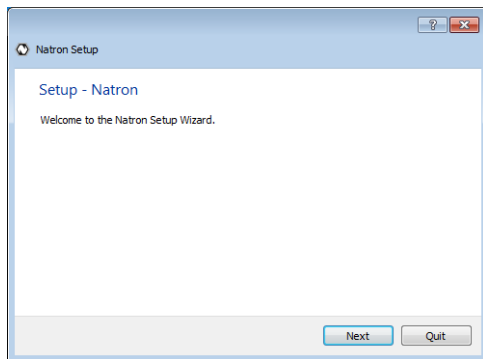
Wrong platform?  

Install

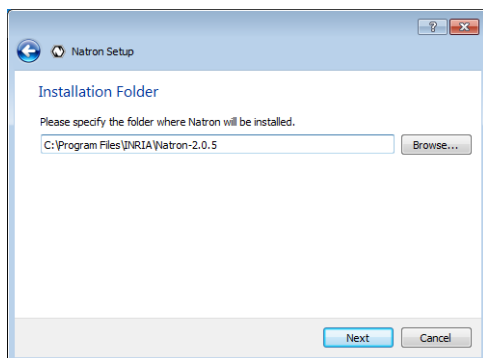
You are now ready to start the installation, double-click on the setup file to start the installation.



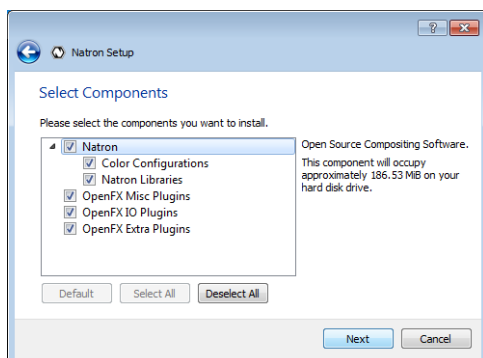
You should now be greeted with the installation wizard.



Click 'Next' to start the installation, your first option is where to install Natron. The default location is recommended.



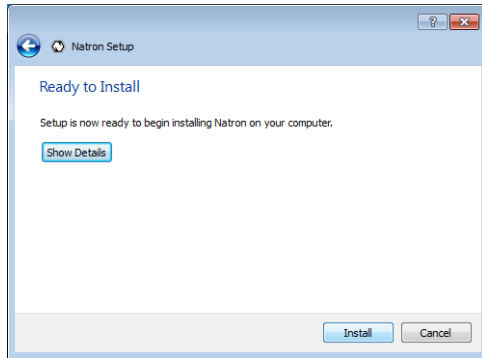
Your next option is the package selection, most users should accept the default. Each package has an more in-depth description if you want to know what they provide.



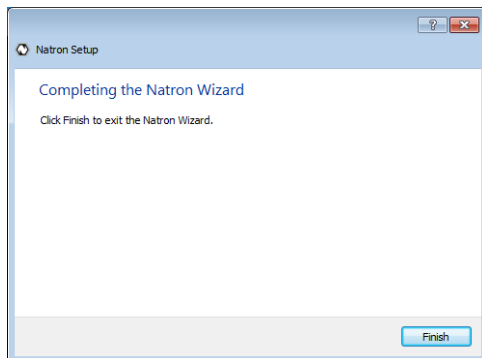
Then comes the standard license agreement, Natron and its plug-ins are licensed under the GPL version 2. You can read more about the licenses for each component included in Natron after installation (in menu Help→About).



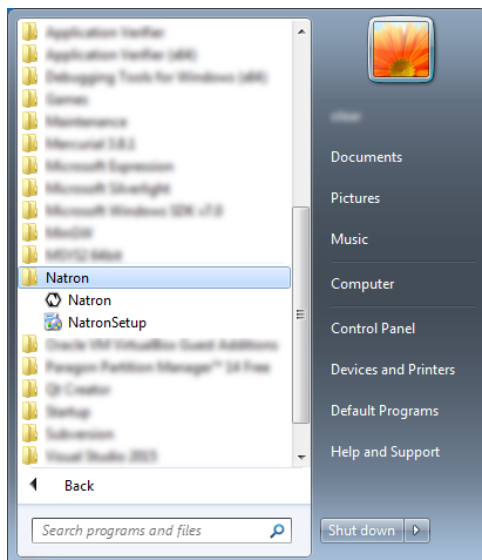
The installation wizard is now ready to install Natron on your computer. The process should not take more than a minute or two (depending on your computer).



The installation is now over! Start Natron and enjoy.

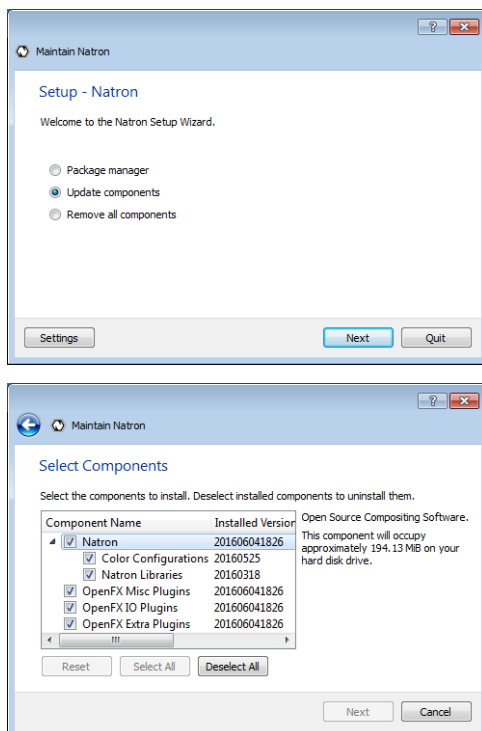


Natron can now be started from the desktop icon or from the start menu.



Maintenance

Natron includes a maintenance tool called 'NatronSetup', with this application you can easily upgrade Natron and it's components when a new version is available. You can also add or remove individual packages, or remove Natron completely. The application should be in the same start menu folder as Natron, or you can start it from the folder where you installed Natron.



The application also include a basic settings category, where you can configure proxy and other advanced options.

macOS

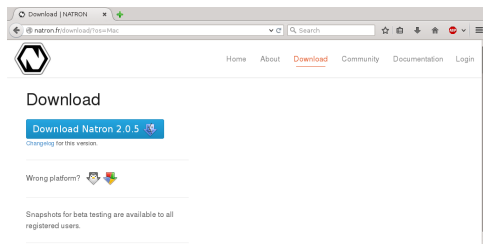
This chapter will guide your through the installation of Natron on macOS (formerly known as Mac OS X or OS X).

Requirements

- Mac OS X 10.6 (Snow Leopard) or higher
- x86 compatible CPU (Core2 x86_64 or higher recommended)
- **OpenGL 2.0 or higher with the following extensions:**
 - **GL_ARB_texture_non_power_of_two** (*Viewer and OpenGL rendering*)
 - **GL_ARB_shader_objects** (*Viewer and OpenGL rendering*)
 - **GL_ARB_vertex_buffer_object** (*Viewer and OpenGL rendering*)
 - **GL_ARB_pixel_buffer_object** (*Viewer and OpenGL rendering*)
 - **GL_ARB_vertex_array_object** or **GL_APPLE_vertex_array_object** (*OpenGL rendering only*)
 - **GL_ARB_framebuffer_object** or **GL_EXT_framebuffer_object** (*OpenGL rendering only*)
 - **GL_ARB_texture_float** (*OpenGL rendering only*)

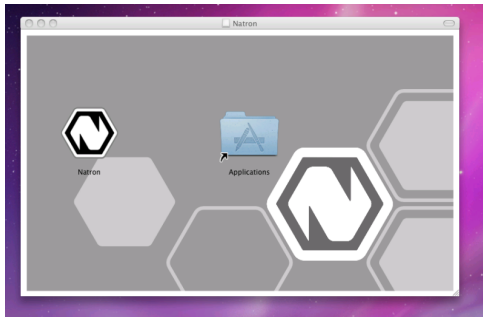
Download

Navigate to <https://natrongithub.github.io/#download> and download the latest version.



Install

Double-click the DMG file and copy Natron where you want it.



Run

On OS X 10.7 and later, you may get the message “*Natron has not been signed by a recognized distributor and may damage your computer. You should move it to the trash.*”

The macOS binaries are not signed with an Apple Developer ID, because of incompatibilities between the Apple code signing tools and the compiler (GCC 4.9) and target OS (Mac OS X 10.6) we use.

There are at least four options to launch Natron on macOS:

- rather than double-clicking on the Natron application, right-click or control-click on it and select Open
- after an unsuccessful launch of Natron, go to the Security & Privacy preferences panel, and enable it.
- from the terminal command-line, execute `spctl --add /Applications/Natron.app`, as explained in [this OSXDaily article](#).
- (not recommended) click “Allow apps downloaded from: Anywhere” in the Security & Privacy preferences panel. Since macOS 10.12 Sierra, this option is not available anymore, but it is possible to re-enable it, as explained in [that OSXDaily article](#).

Linux

This chapter will guide your through the installation and maintenance of Natron on Linux.

Requirements

Natron will work on any distribution released after 2010, this includes (but not limited to):

- CentOS/RHEL 6.4 and higher
- Fedora 14 and higher
- Ubuntu 10.04 and higher
- Debian 7 and higher

The basic requirements are:

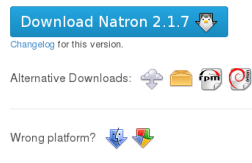
- x86 compatible CPU (Core2 x86_64 or higher recommended)
- Linux 2.6.18 and higher
- Glibc 2.12 and higher
- libgcc 4.4 and higher
- **OpenGL 2.0 or higher with the following extensions:**
 - **GL_ARB_texture_non_power_of_two** (*Viewer and OpenGL rendering*)
 - **GL_ARB_shader_objects** (*Viewer and OpenGL rendering*)
 - **GL_ARB_vertex_buffer_object** (*Viewer and OpenGL rendering*)
 - **GL_ARB_pixel_buffer_object** (*Viewer and OpenGL rendering*)
 - **GL_ARB_vertex_array_object** or **GL_APPLE_vertex_array_object** (*OpenGL rendering only*)
 - **GL_ARB_framebuffer_object** or **GL_EXT_framebuffer_object** (*OpenGL rendering only*)
 - **GL_ARB_texture_float** (*OpenGL rendering only*)

Download

Navigate to <https://natrongithub.github.io/1#download> and download the latest version. This documentation will assume that you downloaded the installer (our default and recommended choice).

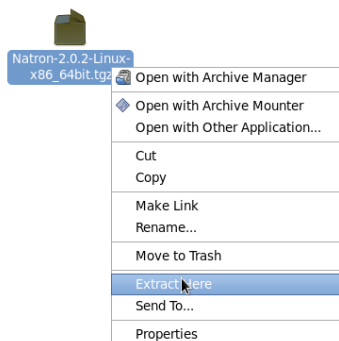


Download



Extract

When the file has been downloaded, extract the file. This can be done in your file browser, usually just right-click and select 'Extract Here'.



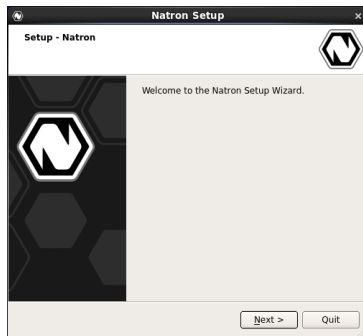
Install

You are now ready to start the installation, double-click on the extracted file to start the installation.

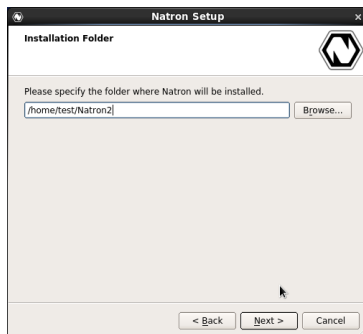


On some installations you are not allowed to execute downloaded files, right-click and select properties, then tick the 'Execute file as program' option. This option may have a different name depending on your distribution and desktop environment. You can also make the file executable through the terminal, type `chmod +x filename`.

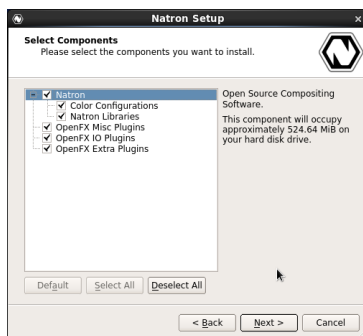
You should now be greeted with the installation wizard.



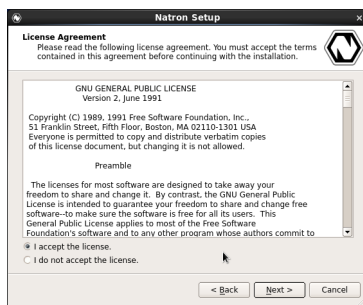
Click 'Next' to start the installation, you first option is where to install Natron. Usually the default location is good enough. If you select a installation path outside your home directory you will need to supply the root (administrator) password before you can continue.



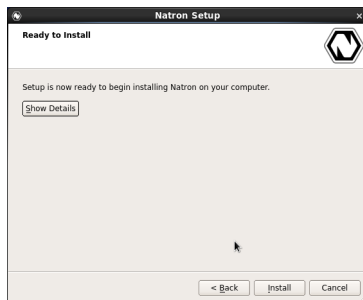
Your next option is the package selection, most users should accept the default. Each package has an more in-depth description if you want to know what they provide.



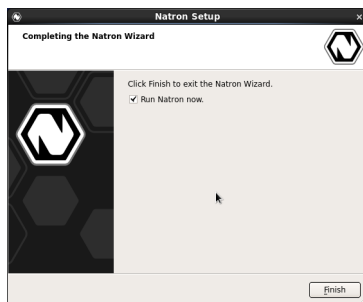
Then comes the standard license agreement, Natron and it's plug-ins are licensed under the GPL version 2. You can read more about the licenses for each component included in Natron after installation (in Help=>About).



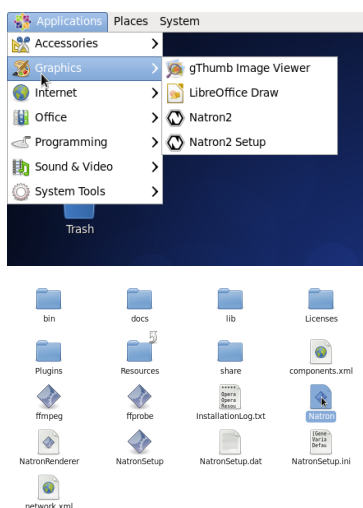
The installation wizard is now ready to install Natron on your computer. The process should not take more than a minute or two (depending on your computer).



The installation is now over! Start Natron and enjoy.

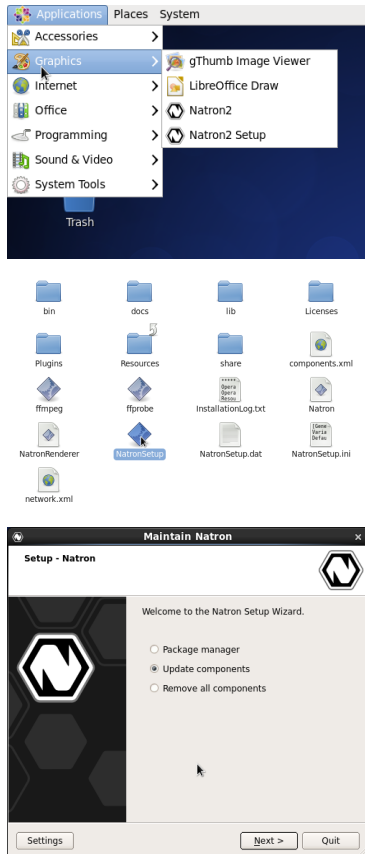


Natron can be started from the desktop menu (under Graphics) or by executing the 'Natron' file in the folder you installed Natron.



Maintenance

Natron includes a maintenance tool called ‘NatronSetup’, with this application you can easily upgrade Natron and it’s components when a new version is available. You can also add or remove individual packages, or remove Natron completely. The application is in the ‘Graphics’ section in the desktop menu, or you can start it from the folder where you installed Natron.



The application also include a basic settings category, where you can configure proxy and other advanced options.

Advanced installation

Natron also has RPM and DEB packages, these are recommended for multi-user installations or for deployment on more than one machine. You can find more information on our website at <https://natrongithub.github.io/>.

1.2.3 Environment

Toolbar, menu bar, and context menus

Using the toolbar

Working with nodes

Navigating inside the node graph

Properties panels

Using the color controls

Using node presets

Animating parameters

Compositing viewers

Using the file browser

Undoing and redoing

Progress bars

1.2.4 Troubleshooting

Natron has bugs, as any software does.

Natron is also a free and open-source software, and bugs are fixed by volunteers when they have some spare time, so please be tolerant and do not expect your bug to be fixed within the hour. It may take days, weeks, or it may even never be fixed.

Properly reporting a bug takes time, but the time spent reporting a bug will certainly help you and the community a lot. It is also the best way to find a temporary solution or a workaround.

Identifying the Kind of Bug

Natron may fail in several ways:

1. It crashes while doing some specific user interaction with the GUI.
2. It crashes while rendering the project.
3. Rendered images are wrong, or contain black areas.
4. Natron hangs and the GUI is not responsive (i.e. menus and buttons do not respond). This is probably a *deadlock* in the GUI code of Natron.
5. Rendering stops before the end of the sequence. This is probably caused by an OpenFX error: check the error log from the “Display/Show Project Errors Log...” menu: there may be an indication of the problem (but it can still be a Natron bug).
6. Rendering hangs or Natron hangs, but the GUI is responsive. This is probably a *deadlock* in the rendering code, and this is the hardest kind of bug to reproduce or fix. If it cannot be reproduced easily, then your best bet is to use one of the workarounds below.

Searching and Reporting Bugs

Bugs may come from OpenFX plugins that were not bundled with Natron, so before reporting anything, if you have any extra OpenFX plugins installed, uncheck “Enable default OpenFX plugins location” in “Preferences/Plugins”, save preferences, relaunch Natron, and check that the bug is still here.

The best way to have your bug considered for fixing is to first search on the [Natron forum](#) and in the [Natron issues](#) if this is a known bug. If yes, then read about it, and try some workarounds given in these bug reports (see below for more workarounds).

If this bug does not seem to be a known issue, then post a [new issue](#) on the Natron github, and follow strictly the guidelines to report the bug. The issue title should be as precise as possible (“Natron crash” is *not* a correct title, see existing issues for title examples). If possible, also post a project that exhibits the issue. Make the project as small as possible: remove extra assets or replace them by small JPEG sequences, checkerboards or colorwheels, etc. You can then either attach your project as a zip file to the github issue, or post a link to a file sharing service.

Common Workarounds

Luckily, there are workarounds for most Natron crashes or hangs. Here are a few one worth trying, but of course your mileage may vary or you may find another workaround (which you should describe in the proper [Natron issue](#)).

1. Avoid using videos with inter-frame compression as inputs and outputs. This includes H.264 (eg AVCHD) and H.265 (HEVC) video. ProRes is OK but slow, especially for writing. DNxHR is OK. Individual frames are *best* (DPX, EXR, TIFF, PNG, JPEG, whatever suits your input video quality and bit depth). The video reader is here for convenience, but it may have difficulties decoding some videos. The video writer may also be a source of bugs, and should be avoided for long sequences: if Natron crashes in the middle, then the whole sequence has to be rendered. Extract individual frames, do your compositing, then compress the frames (and optionally mux the audio) with an external tool. To extract frames, you may use a simple Natron project or any other tool (e.g. [FFmpeg](#)). To compress frames to a video, there are also many tools available, e.g. [FFmpeg](#), [MEncoder](#), or [VirtualDub](#) (windows-only). This is the standard compositing workflow and the preferred method of running Natron.
2. If Natron hangs or crashes when rendering an image sequence (this does not work when rendering to a video), check that the rendered frames are OK, relaunch Natron and in the parameters of the Write node uncheck “Overwrite”. That way, only the missing frames will be rendered.
3. If you have a large project, or a project with heavy processing, use the [DiskCache node](#) at places that make sense: downstream heavy processing in the graph, or before you use the result of processing as inputs to [Roto node](#) or [RotoPaint node](#).

You will quickly notice that using individual frames instead of videos for inputs and output give a *big* performance boost and will most probably solve your issues, so once you’ve learned how to decompress/compress any video, this will become your standard workflow. Just add extra disk space, and you’re good to do serious and fluid compositing with Natron.

OpenGL/GPU Rendering Issues

If the viewer displays some error message about OpenGL, then GPU rendering is probably going bad. Note that this kind of problem seems to only happen on Windows, so you might want to consider switching to Linux or macOS to use Natron if your GPU is not well supported by Natron under Windows.

1. Create a [Shadertoy node](#), click “Renderer Info...” and check that the OpenGL version is at least 2.1 and that the extension `GL_ARB_texture_non_power_of_two` is available. If the displayed info does not correspond to your graphics card, check that the OpenGL drivers for your card are installed. If not, install the software called “OpenGL Extension Viewer” and check that your card appears in the list of renderers. If not, then it is a drivers issue.
2. In Natron Preferences / GPU Rendering, check that the displayed is consistent with what “Renderer Info...” above gave.
3. Now uncheck “Enable GPU Render” in the Shadertoy node and click the refresh/recycle button on the top of the viewer. Click again “Renderer Info...” and it should say it now uses Mesa in the `GL_VERSION`. Does it fix the issue? If yes, you may try the next step to globally disable OpenGL rendering in Natron.
4. To temporarily fix this issue, in Natron Preferences / GPU Rendering, set “OpenGL Rendering” to “Disabled”, click the “Save” button in the Preferences window, quit Natron, launch Natron, check that GPU rendering is still disabled in the Preferences, and test your project.

1.3 Compositing

1.3.1 Managing Projects

Project setup (empty)

Proxy Mode (empty)

Loading images and video (empty)

Image caching (empty)

Frame ranges

The project frame range (in the Project Settings, key ‘s’ in the Node Graph’) is the range that will be used by default when rendering Writers.

Each clip (input or output of a node in the Node graph) also has its own frame range. This “clip frame range” may be used or even modified by plugins, such as Retime (which may change the frame range), Merge or Switch (which set the frame range to the union of their input frame ranges). The plugin may be able to render images outside of this frame range, and it is just an indication of a valid frame range. This information is available from the “Info” tab of the properties panel of each node.

Most generator plugins (e.g. CheckerBoard, ColorBars, ColorWheel, Constant, Solid) have a “Frame Range” parameter, which is (1,1) by default. The FrameRange plugin may be used to modify this frame range inside the graph.

The default framerate of an image sequence or video is the range of the sequence

1.3.2 Reformatting elements (empty)

1.3.3 Channels (empty)

1.3.4 Merging images (empty)

1.3.5 Noise removal (empty)

1.3.6 Keying (empty)

1.3.7 Using Rotopaint (empty)

1.3.8 Tracking and stabilizing

Workflow Summary

In order to track a planar shape and move a Roto mask or a texture corresponding to that shape:

- Track some points inside your mask (shape)
- In the Transform tab, set the transform to CornerPin and to match-move
- Disable the CornerPin and set the from points of the corner pin at the reference frame where you want your object to move in (basically the bounding box of the shape to track)
- Export to CornerPin
- Append your CornerPin to the Roto node

In a future version we will have a planar tracker that will do that automatically for you in a single click.

Detailed Usage

To link parameters in Natron, it is the same as in Nuke except that you drag and drop the widget of a parameter onto another one by holding the control key (or cmd on macOS).

The tracker works differently than the Nuke tracker regarding the “Transform” part. For the tracking itself, almost everything is the same. Basically, in Nuke they can only output a CornerPin with exactly 4 points, and they map 1 track to each corner of the CornerPin. For the Transform node they may use 1 (translation only), 2, or N points to find the final transformation, however that will never be something other than a [similarity](#), which means that it cannot handle perspective deformation.

In Natron, we offer the possibility to compute a CornerPin with N points, that is an [homography](#), which encompasses all distortion-free perspective transforms.

This is much better, because the more tracks you use to compute that CornerPin, the more robust it will be.

An homography is typically used to contain information about a perspective deformation, whereas a similarity is more constrained: a similarity is translation, rotation and uniform scale.

In The Transform tab, this is what we call “the model”. Basically, the problem we are trying to solve is to fit a model (i.e. similarity or homography) so it is the closest to the N point correspondences. Each correspondence is the position of a track at the reference frame and its position at the tracked time.

Hence the more correspondences you have (i.e. the more tracks), the more robust the homography is in the region where you tracked features.

The *Fitting error* parameter (in the Transform tab) is an indication of how much difference there is in pixels between the reference point on which we applied the computed transformation and the original tracked point. This is the RMS (root mean square) error across all tracks, and gives an estimate of the quality of the model found in pixel units.

For each tracked frame, the *correspondences* we use to compute the CornerPin are the tracks that are *enabled* at this frame (i.e. the Enabled parameter is checked at this time) and that have a keyframe on the center (i.e. they successfully tracked).

When you press *Compute*, it computes the model (CornerPin/Transform) with all the tracks that meet the aforementioned requirements *over all keyframes*.

When *Compute Transform Automatically* is checked, whenever a parameter that has an effect on the output model is changed, this will recompute the Corner/Pin transform *over all keyframes* again.

The parameters that have an effect on the output model are:

- The motion type
- The Transform Type (i.e. Similarity or Homography)
- The Reference Frame
- Jitter Period
- Smooth: this can be used to smooth the resulting curve to remove some of the noise in the high frequencies of the CornerPin/Transform. Note that in “Add Jitter” mode, you can increase High frequencies to simulate a camera shake that follows the original camera movements.
- Robust model: this is quite complicated, but in short: When trying to find a model that *best fits* all correspondences, you may have correspondences that are just wrong (bad tracking for example). These bad correspondences are called *outliers*, and this parameter when checked tells we should not take into account those outliers to compute the final model. In most cases this should be checked. However sometimes, the user may have for example required to compute an homography (i.e. CornerPin), but the given tracked points (correspondences) just cannot make-up an homography. In this case, if the parameter were to be checked, it would fail to compute a model. If you uncheck this, it will take into account all the points and compute a model that averages the motion of all correspondences.

Also when *Compute Transform Automatically* is checked, the model will be computed automatically when the tracking ends.

We cannot compute the model after each track step (i.e. during tracking) because the model at each frame depends on the model at other frames since we may smooth the curve or add jitter.

So all in all it works differently than Nuke, the whole transformation computation can be more robust and happens as a second pass after the tracking is actually done.

One last thing: to compute the CornerPin in the “Transform” tab of the tracker, the **to** points are computed using the **from** points as reference.

Basically what happens is that the tracking outputs a transformation matrix at each frame. Then when computing the model, this matrix is applied to the **from** points at each frame in order to obtain the **to** points.

So if you were to change the reference points (i.e. the **from** points) with the *Set to input RoD* for example, then you would need to recompute the model at all frames, because the **to** points would just not be the same.

The work is usually done in two steps:

- First, disable the CornerPin so that even if the viewer is connected to the Tracker there is no deformation going on, and set the **from** points to be the RoD (bounding box) of the Roto shape at the reference frame.
- Then, export the CornerPin. It just links the parameters of the CornerPin to the ones in the tracker, so if you change something in the tracker transform tab the changes will reflect onto the CornerPin.

Basically what the Planar tracker will do in the future is automatically do all the steps for you: it will place markers inside the mask for you, track them and output a CornerPin from the bounding box of the roto shape.

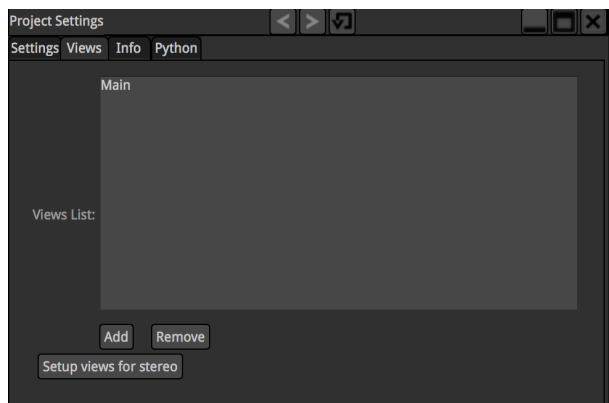
1.3.9 Transforming elements (empty)

1.3.10 Working with color (empty)

1.3.11 Stereoscopic compositing

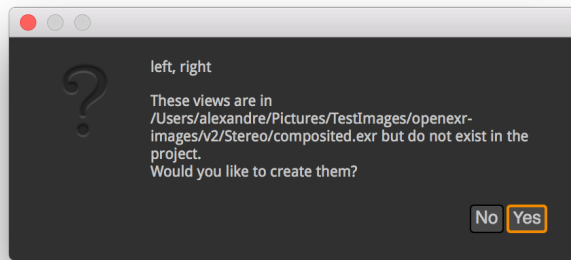
Importing stereoscopic sequences

When creating a new project in Natron, by default there is a single view present. It is called the “Main” view.

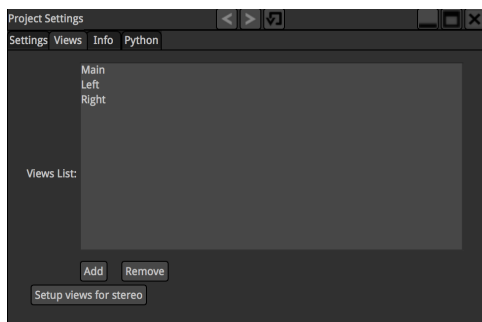


You can add, remove and rename views as desired. Clicking the “Setup views for stereo” button will make exactly 2 views named “Left” and “Right”.

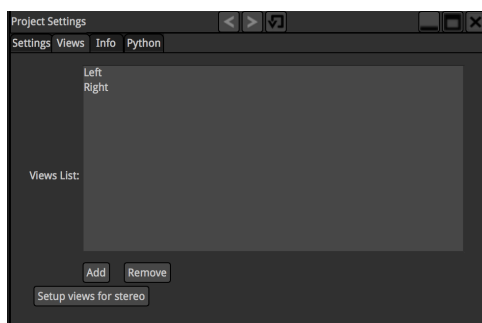
Upon importing an EXR file containing multiple views within the file, Natron will prompt you to create the appropriate views in the project:



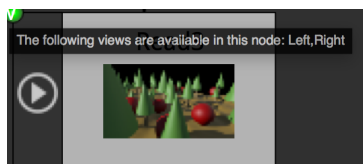
If clicking yes, Natron will create the missing views in the project:



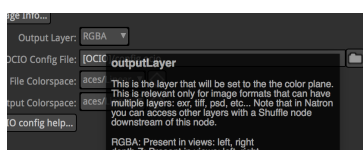
You can remove the “Main” view if needed (in our case it does not make sense to leave it in the project). You can do so by clicking either Remove and selecting the “Main” view or clicking “Setup views for stereo”:



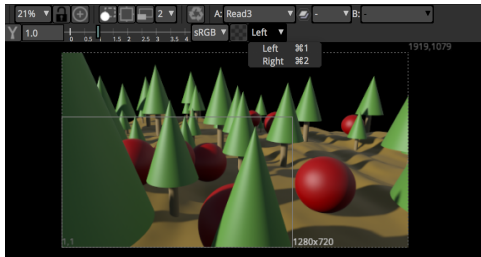
The Read node will have a special “V” mark on its top-left corner, indicating that it has multiple views available. When hovering the “V” indicator with the mouse, more informations are available regarding which views are present in this Read node



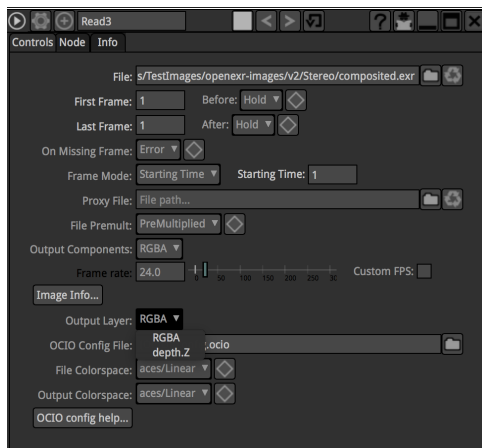
Hovering your mouse over the Output Layer parameter will detail which layer is available in which view:



When the project has multiple views available, each viewer will have a drop-down with available views. You can select with view to display:

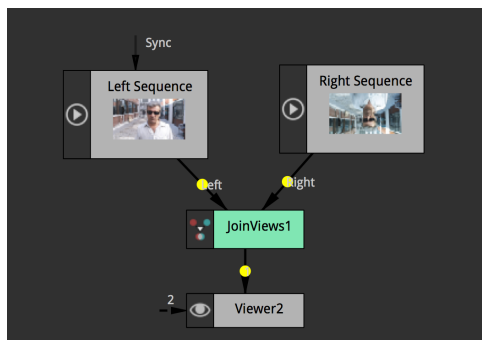


If we take a look at our Read node's properties, you can see that we have 2 layers in this file: RGBA and depth. Layers may sometimes not be present in all views in the file, but the Read node will show the union of all layers available across all views:



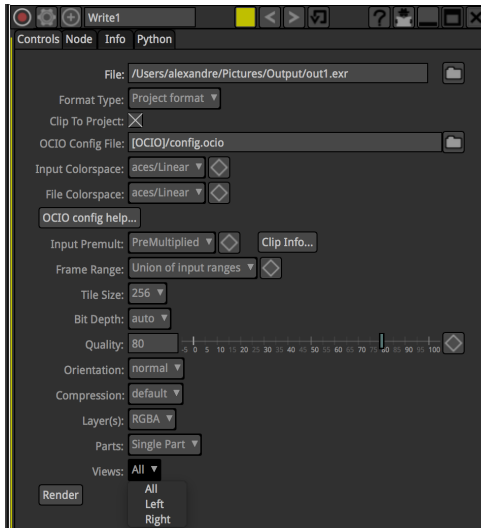
Joining separate views

In Natron you can join different simple-view files to a multiple-view stream by using the JoinViews node, like this:



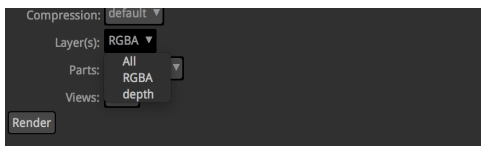
Exporting stereoscopic sequences

Exporting multi-view files in Natron can be done multiple ways: Either you want to have a separate file for each view, or you want all views to be in the same file. The later is only supported by the EXR file format. When exporting to EXR, the Write node properties panel will have an extra “Views” parameter:



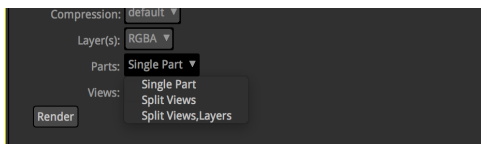
This is quite straight-forward: you can select which views you decide to export.

You can also select which layer you would like to export:



The “All” choice is only available for the EXR and TIFF file formats which allow embedding more than 1 layer per-file.

Another interesting parameter controls how the EXR/TIFF file is written, this is the “Parts” parameter:



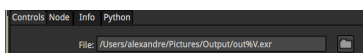
When set to “Single Part”, the Write node will aggregate all views requested to render in a single “part” of the file. Single part files are what is written by OpenEXR 1.x applications, thus selecting “Single Part” ensures compatibility with other applications that do not support OpenEXR 2.

When set to “Split Views”, all views will be scattered in different parts of the file. The resulting file will only be readable in applications that support OpenEXR 2 but with the advantage of being faster to read than the single-part files. Note that if you select “All” layers to render all layers will be aggregated for each view.

When set to “Split Views,Layers” each layer of each view will be written to a separate part, ensuring EXR optimized for decoding speed, though the file will be larger.

Writing to separate files

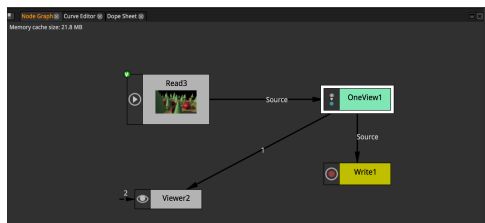
You can choose to export views to a separate files by adding a `%v` (which will expand to the letter ‘l’ or ‘r’) or `%V` (which will expand to the word “left” or “right”) to the output filename:



In that case the “Views” parameter will no longer be available. This is how you should proceed if you want to output multiple views but do not want to write EXR’s.

By default for files that do not support multi-view (I.e: anything besides EXR), if requesting to write multiple views to the same file, Natron will only write the first view in the project’s views. If you do not want to render

all views but a specific one, you can use a “OneView” node prior to your Write node to specify which view you would like to write-out:



1.3.12 Preview and rendering (empty)

1.3.13 Expressions (empty)

1.4 Tutorials

1.4.1 Writing documentation

This quick tutorial will guide you through the creation/modification of documentation for Natron and the plugins.

Plugins

Editing/adding documentation for the Natron plugins requires you to edit the CPP file for each plugin. Usually the plugin(s) has a **kPluginDescription** define where you can edit the description found when hovering or clicking the ? button in Natron.

Let's start with an example, you want to edit the description in the Checkerboard plugin.

1. Fork the <https://github.com/NatronGitHub/openfx-misc> repository on Github.
2. Open the file *Checkerboard/Checkerboard.cpp* in your favorite (plain) text-editor

Navigate to the line **#define kPluginDescription**, where you can edit the description. Line breaks are added with `\n`.

You will also notice that each parameter has a hint define, for example the Checkerboard has **#define kParamBoxSizeHint**, **#define kParamColor0Hint** etc. These describe each parameter in the plugin and shows up when you hover the parameter in Natron, or access the HTML documentation online or through Natron.

To test your modification you must build the plugin(s) and load them in Natron, refer to each plugin bundle on Github on how to build the plugin(s).

When you are done do a pull request on the master repository on Github.

Markdown

The plugin description and parameters supports [Markdown](#) format. This enables you to have more control over how the information is displayed.

Enabling Markdown on a plugin requires some modifications, as the plugin must tell the host (Natron) that it supports Markdown on the description and/or parameters. See the [Shadertoy](#) plugin for an example of how this works.

Basically you need to add **desc.setPluginDescription(kPluginDescriptionMarkdown, true);** in the **describe** function for each plugin. If you are not comfortable with this, contact the repository maintainer(s) and ask them to enable Markdown for you.

Natron

Contributing to the Natron documentation is a bit easier than contributing to the plugins. First fork the Natron repository on Github, <https://github.com/NatronGitHub/Natron>. The source for the documentation is located in the *Documentation/source* folder.

The documentation is generated using [Sphinx](#), and the source files are in [reStructuredText](#) format.

Most likely you will want to contribute to the User Guide. The guide is located in *Documentation/source/guide*. If you want to contribute to an already existing document just open the file in your favorite (plain) text-editor and do your modifications.

If you prefer editing with [LibreOffice](#) (or even [MSWord](#)), just keep the document simple (use styles for section headers, don't try to format too much, etc.), and use [pandoc](#) to get a first working version in [reStructuredText](#) format. This file will probably require a few touch-ups afterwards, but it is usually a good starting point.

To send your contributions, you will need to:

- Fork <https://github.com/NatronGitHub/Natron> using your github account.
- On your fork, create a branch from the RB-2.3 branch (do not use the master branch), and give it a name like "documentation-keying" if you are going to write the keying doc (which we really need).
- To add your doc, you can either:
 - clone the repository to your computer, edit and add files, commit your changes locally (the github desktop application is easy to use), and then push your changes,
 - or edit the files directly on github. see [tutorials-hsvtool.rst](#) for example (you will probably need to fork the repository first, see below, and browse to that file on your fork): click on the pencil icon on the top right. You get an editable text view and can get a preview by clicking on the preview tab on top.
- Then, submit a [pull request](#) to the RB-2.3 branch on the main repository from your branch (there is a button to submit a pull request when you view your fork on github), and the Natron maintainers can either accept it as it is, or ask for a few modifications.

You can view the formatted documentation on your github repository, as explained above, but you can also preview your modifications by using [pandoc](#) to convert it to another format, or install [Sphinx](#) and recompile the whole documentation. On Linux and Mac you can install Sphinx through your package manager (using MacPorts type `sudo port install py27-sphinx`, on HomeBrew type `pip install sphinx`), on Windows refer to the [Sphinx documentation](#).

When you have Sphinx installed go to the Documentation folder and launch the following command:

```
sphinx-build -b html source html
```

The Natron documentation has now been generated in the *Documentation/html* folder. Open *Documentation/html/index.html* in your web browser to review your changes.

When you are satisfied with your modifications do a pull request against the master repository on Github.

Note: do not modify the files in *Documentation/source/plugins*, these are automatically generated by Natron and updated when needed.

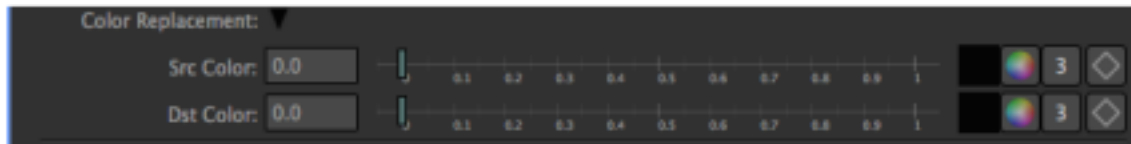
1.4.2 HSVTool

The *HSVTool node* converts the input color space to a HSV color space and converts the adjusted values back to the image input color space for further processing for the output. The *HSVTool node* is used to adjust the HSV channel components of an image the the Read node stream. It's basic operation are to adjust the Hue: which is the color of the input image, Saturation: which is the color range of the input image, and Brightness: which is the color value level and the grayscale of the input image.

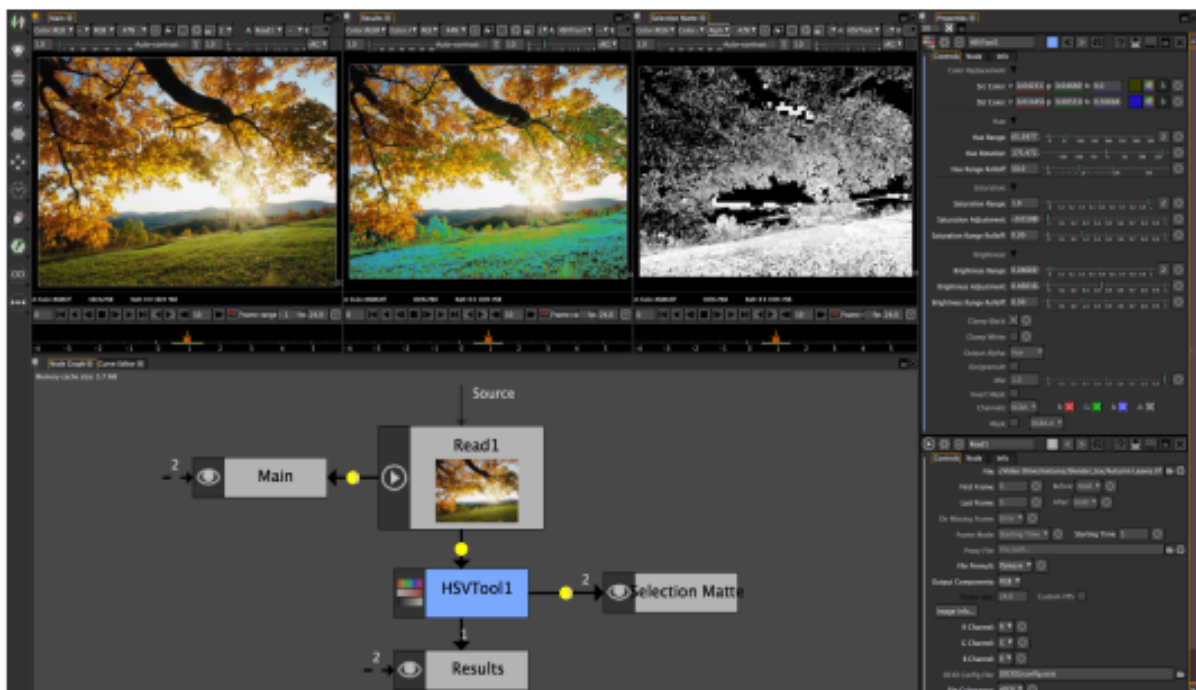
Each section has a range controls that can be adjusted to limit the effects of a node by narrowing the input color ranges.

What are some standard work-flow using *HSVTool node*?

1. You can do secondary color corrections or color replacements with the source/destination eyedropper tool.
2. You can use the *HSVTool node* for linear/color keying.
3. You can do an overall color shift by using the rotation range adjuster in the Hue parameters etc.



The following screen captures are based on a single pixel selection without manual component modifications. You will see different alpha mattes display based on the alpha output components.



The image above is the result of a single color pixel selection using the blue destination replacement color.

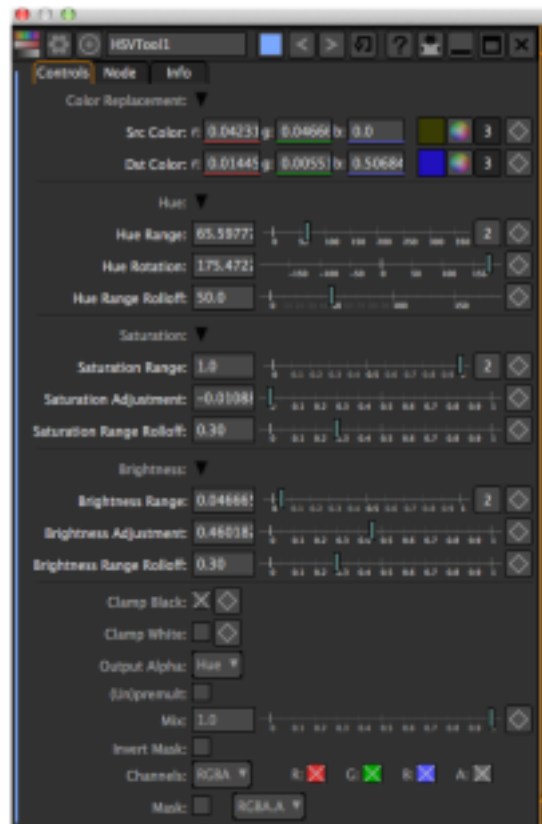
Brightness adjustment ranges

This is when you make adjustments to Hue, Saturation, and Brightness to limit the input color ranges for the desired effect. You can look at the third viewer on the right that has matte/alpha generated by the color selection key using the source color eyedropper. This is the alpha interpretation of the alpha output component which is the to Hue.

There are several alpha output component that the *HSVTool node* uses to limit effects. The next images are the results of the output alpha mode from the *HSVTool node*.

Note: Each of the matte generated displays are based from the colors that are spread across the image on a single pixel selection and it HSV values.

By using a low resolution and/or compressed image you will see the artifacts in the alpha channel. This makes it harder to do a replacement or secondary color correction. Is is evident that the blue channel has the most artifacts/noise, you can use the *SmoothPatchBased node* and blur one or two pixels in the blue channel to soften the pixel edges. This technique will not always work, remember to apply dynamic range applications when



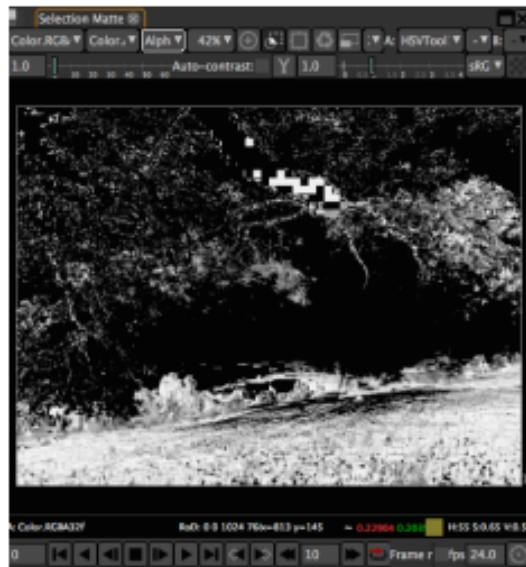




min(Hue, Saturation)



min(Saturation)



min(Hue, Brightness)



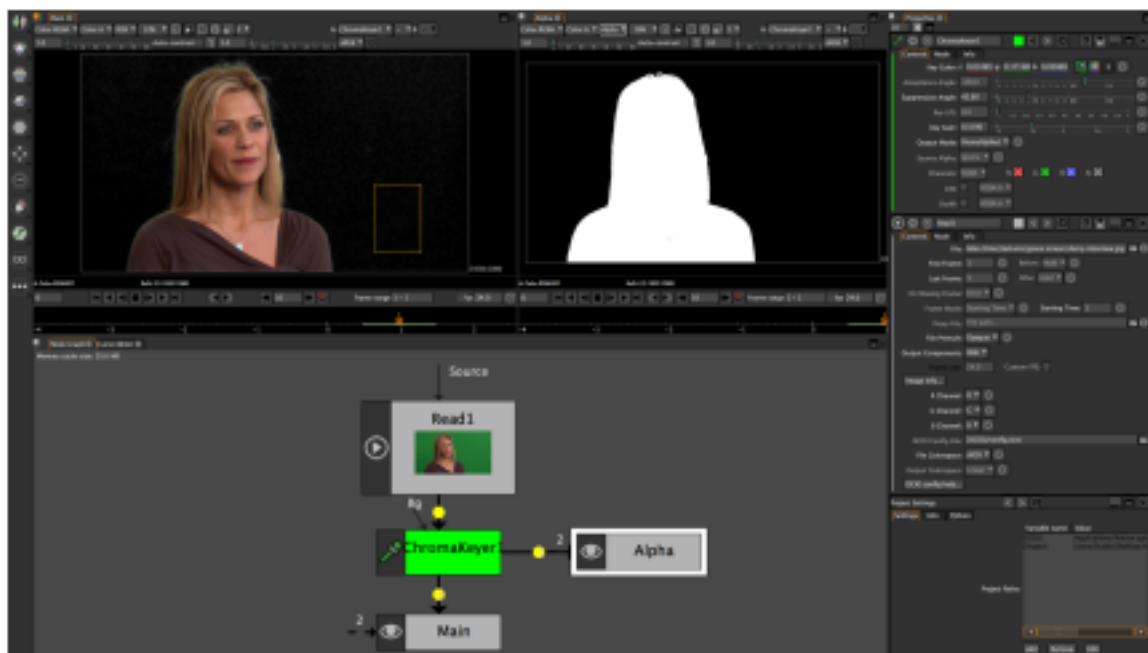
min(All)

using the *HSVTool node*. You can only push the Saturation and Brightness so far. Also remember that *HSVTool node* need color input in order to apply any color filtering even though it can output greyscale and matte data.

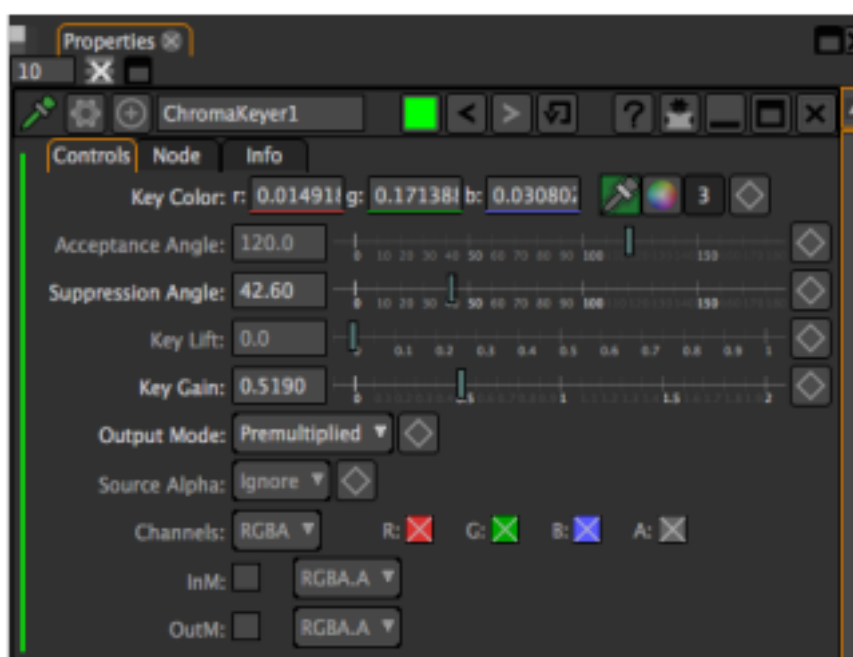
HSVTool Keyer

The *HSVTool node* can be used as a color keyer. It uses the same principles as the *ChromaKeyer node* and the *Keyer node*. What separates the *HSVTool node* as a keyer is that the matte is inverted with it's selection.

When you need to pull a key with the *ChromaKeyer node* the key color is outputted as black or represents a transparency. See node graph below for an example.



The image below is the *ChromaKeyer node* parameters panel. You can see that the eyedropper has a chroma green selection from the input image. You may also notice the acceptance angle is very high, this is because the source material is compressed.

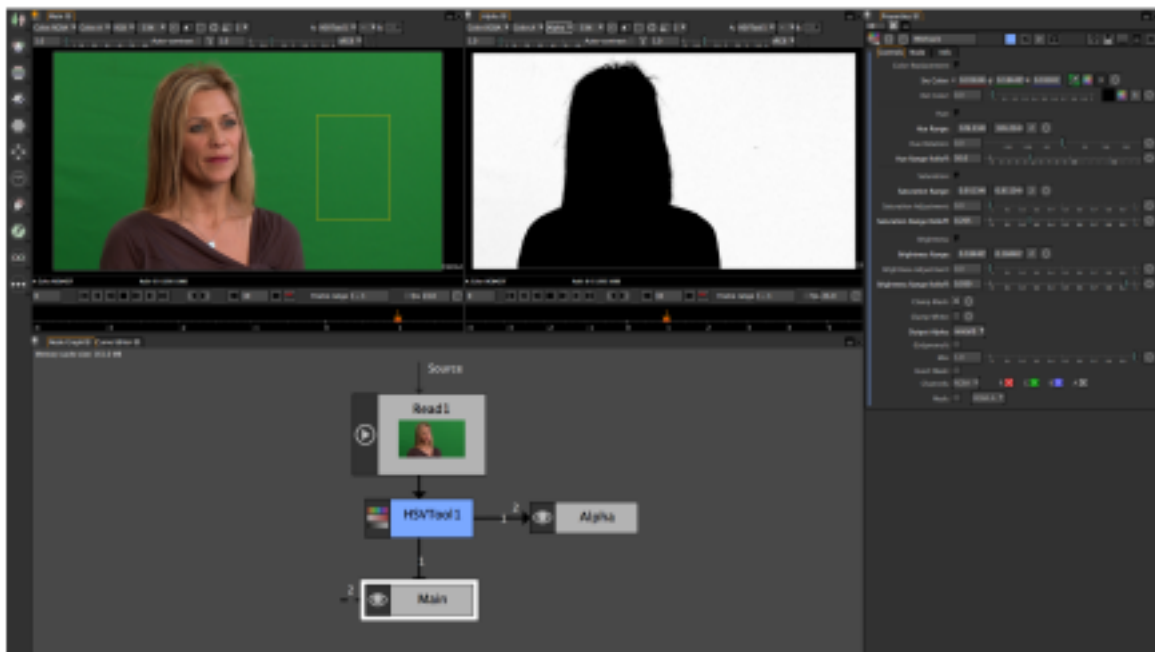


As you can see from the panel the chromakey subtracted the chroma green, replacing it with black/transparent. The “Key Lift” and “Key Gain” clamps the black/white to make a clean matte from the keyed selection.

Even if the chromakey generates transparency from the node, it still outputs a black and white matte that can be used as an inverted mask or holdout matte to apply needed filter effects.

Note: A holdout matte is a section of your image that tells the keyer not to key the selected area. A holdout matte is commonly used to define a area within your image that might have similar colors to the color being keyed.

The node graph below is using the same green screen image when pulling a key with the *HSVTool node*. The alpha output shows up as the white color. This *HSVTool node* does not subtract the color to generate a matte, as oppose to the *ChromaKeyer node* and *Keyer node* nodes that subtracts the color to generate a matte. If you are concerned about color spaces, *ChromaKeyer node* and *Keyer node* process data in RGB color space and the *HSVTool node* process in HSV color space.



The image below is the *HSVTool node* parameters panel. You can see that the source color eyedropper operator has a chroma green selected. As you look further down the panel you will notice the Hue, Saturation, and Brightness parameters which has all the values the make up the chroma green selection.

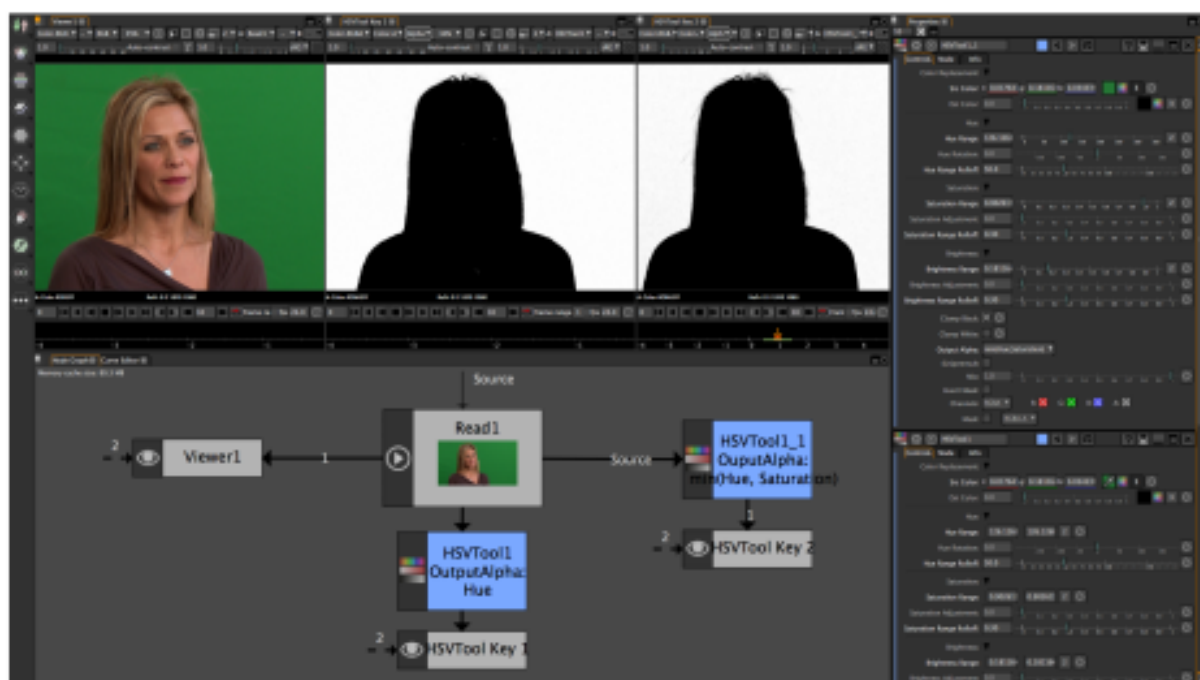
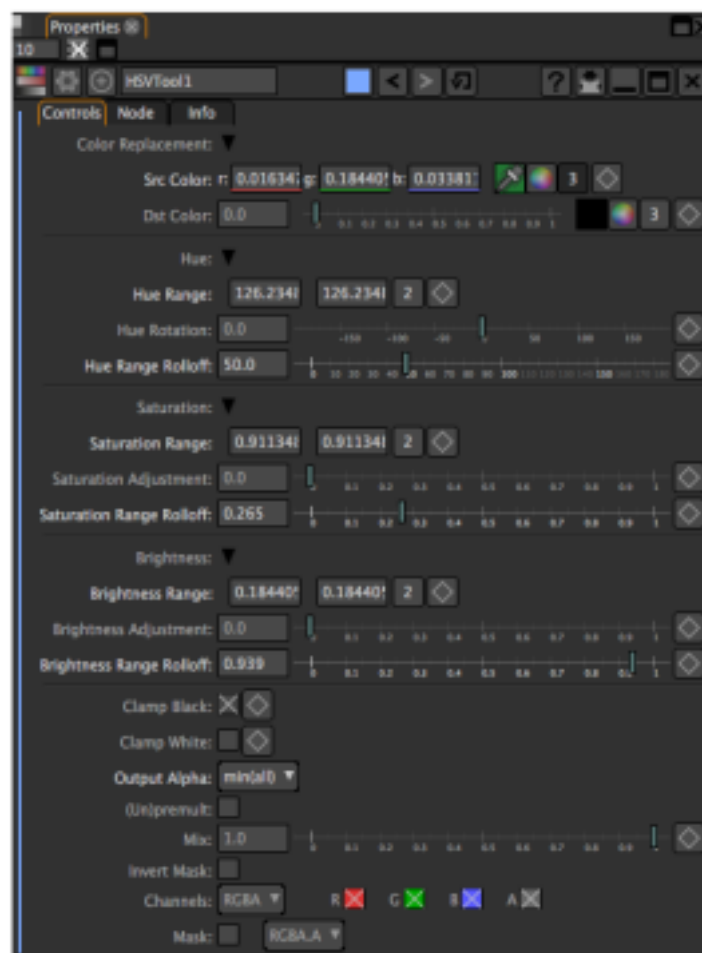
The important part to remember is that Hue, Saturation, and Brightness are used to generate and adjust the matte(s) based on the output alpha mode. The image output alpha is set to min(All), when selected, Hue, Saturation, and Brightness can be used to adjust the matte/mask output.

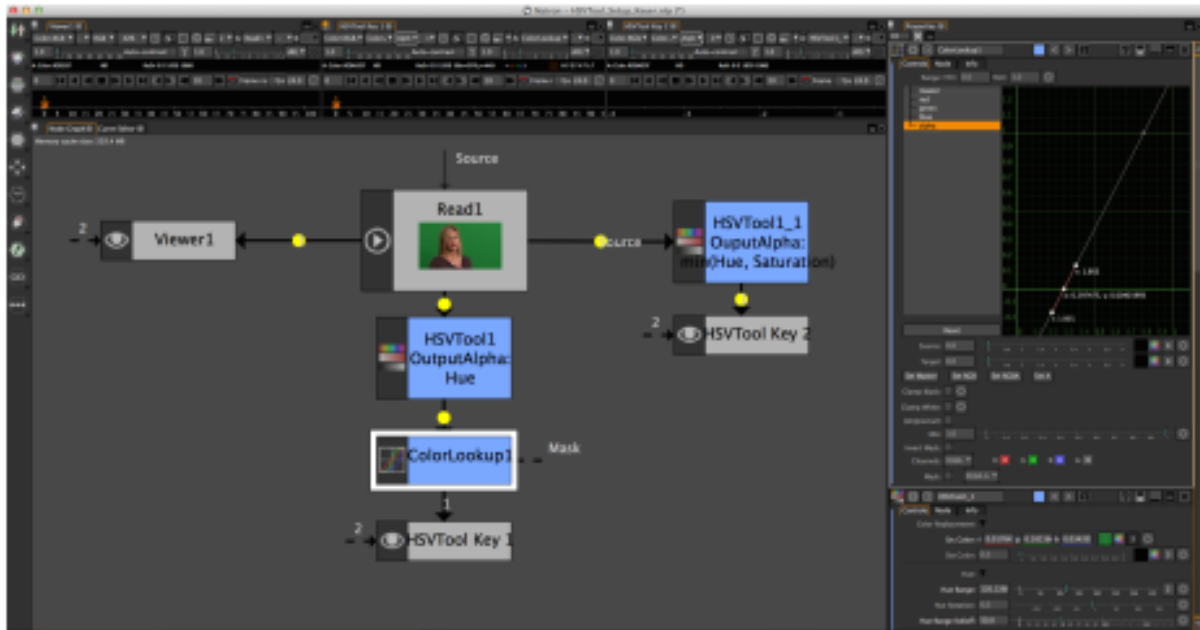
There are eight output alpha modes. The only mode that can’t generate a matte is the “Source” mode. Even if there’s no matte generated you can still do color replacements and color corrections.

The image above displays two *HSVTool node* nodes that is keying the same input, but generating two different mattes based on the output alpha. The middle viewer is the Hue mode output. The viewer on the right is the min(Hue, Saturation) mode output. When you are working with compressed source images the output alpha modes will yield different results.

Note: If you need more control on the alpha/matte output, the *ColorLookup node* can help adjusting edges.

1.4.3 FFmpeg (empty)





Producing digital intermediates (empty)

Muxing audio (empty)

Encoding HEVC (empty)

1.4.4 Using PanoTools projects within Natron

Those are preliminary notes on using PanoTools or Hugin projects from within Natron.

Creating a PanoTools (pto) project

Using the Hugin GUI

- download [Hugin](#) - on macOS this can be done using [homebrew](#), by typing the command `brew cask install hugin` in a terminal.
- read the [Hugin documentation](#) or start with a [Hugin tutorial](#)
- make your panorama

Using command-line tools

Full details on using command-line tools are given in the [Panorama scripting in a nutshell](#) documentation.

On macOS, if Hugin was installed using homebrew, the tools are available in various directories, and they can be added to the PATH using:

```
PATH="$PATH:/Applications/Hugin/tools_mac:/Applications/Hugin/Hugin.app/Contents/
↪MacOS:/Applications/Hugin/HuginStitchProject.app/Contents/MacOS"
```

Here is an example of running the panorama tools from a set of JPEG images (with suffix `.jpg`) placed in the current directory to generate a perspective panorama (other options are described in the documentation):

```
pto_gen -o project.pto *.JPG
cpfind --multirow -o project.pto project.pto

celeste_standalone -i project.pto -o project.pto
cpclean -v --output project.pto project.pto

autooptimiser -a -l -s -m -o project.pto project.pto

nona -m TIFF_m -o project project.pto

enblend --save-masks -o panorama.tif project*.tif
```

The result is:

- A `project.pto` file, which is the Hugin project itself
- The `project*.tif` images, which are images warped to the reference projection.
- The `mask-*.tif` images, which are masks generated by `enblend`.
- The `panorama.tif` image, which is the final panorama.

Using Hugin/PanoTools parameters in Natron

The `pto` file syntax is described in the [PTOptimizer](#) and [PTStitcher](#) docs.

This file can be viewed and edited in any text editor.

Project size

The project size, or format, should be set to the panorama size, which is given in pixels on the `p` line (at the start of the file).

Distortion correction

The distortion correction parameters for the input images are given in the `i` lines (one for each image).

The values for the `a`, `b`, `c`, `d`, `e`, `g`, `t` parameters should be entered in a **LensDistortion** node, with *Model/model*=PanoTools and *Direction/direction*=Undistort, placed after the **Read** node for each input image or video. The script name for these values are `pt_a`, `pt_b`, `pt_c`, `pt_d`, `pt_e`, `pt_g`, `pt_t`.

Note that if all images share the same distortion parameters (this is written as `a=0 b=0 c=0 d=0 e=0 g=0 t=0` on the `i` line), the **LensDistortion** node can be cloned in Natron (right-click on node, *Edit*, *Clone Nodes*).

Projection

The projection (which can be performed by the [Nona](#) tool) can be done using **Card3D** nodes placed after each **LensDistortion** node.

Each **Card3D** node must have its *Transform Order/cardXFormOrder* set to `STR`, and the *Rotation Order/cardRotOrder* should be left to the default value (`ZXY`). Be careful, these are neither the Axis nor the Cam Transform Order, which are in the two first groups of the **Card3D** node.

Set the *Output Format* to `Project`.

The `r p` and `y` values from the `i` line are roll, yaw, pitch angles. Their values should be put in the *Rotate* parameter of the **Card3D** (script name is `cardRotate`), using the following convention: `cardRotate.x = p`, `cardRotate.y = -y`, `cardRotate.z = -r`.

The v value from the i line corresponds to the horizontal field of view, and has to be converted to an aperture value. The **Lens-In H.Aperture**/`lensInHAperture` param should be set to $2 * \tan(v * \pi / 360)$. This expression can be directly typed in the value field, with v replaced by the actual value from the corresponding i line of the `pto` file.

Setup all the **Card3D** nodes for each input image that way.

Camera

In Natron 2, the camera used by the **Card3D** node is set in the *Cam* group at the top of the parameters list.

In the **Card3D** node for the first image, unfold this group, and unfold the *Cam Projection* group.

The v value on the p line (usually at the top of the `pto` file) gives the horizontal field of view of the output panorama.

Set the *Focal Length*/`camfocal` to 1., and set the *Horiz. Aperture*/`camhaperture` to $2 * \tan(v * \pi / 360)$, replacing v with its actual value.

You can then copy these two parameters to all the **Card3D** nodes, or - even better - link these parameters, so that the output camera for the panorama can then be modified: Right-click on the parameter from the first **Card3D**, Copy Link, then right-click on the same parameter of every other **Card3D** node, Paste Link.

If you intend to modify the camera orientation later, you can also link the same way the *Rotate*/`camRotate` parameter (those for the camera, not the card).

Note that when the Natron **Card3D** node is used in Nuke (where it appears as **Card3DOFX**), the node has an external Cam input, to which a Camera node may be connected. There may be a similar concept in future versions of Natron.

Building the panorama in Natron

First sketch: overlap the images

Make sure that the *Output Components* parameter in all readers is set to RGBA, so that images have a transparent value outside of their domain.

Now, connect the output of the first *Card3D* to the B input of a **Merge** node, connect the second to the A input, the third to the A2 input, etc. . .

The output of the **Merge** node should show a first panorama, obtained by overlapping all images.

Drawing the masks

Add a **Roto** node after each **LensDistortion**, before each **Card3D**.

Only leave connected the A and B inputs to the **Merge** node to the two first **Card3D** nodes.

Check *Premultiply* in each Roto node, check that only the A channel is affected by **Roto** in its parameters, and start editing the roto mask on the second image, for which the **LensDistortion** output is connected to the A input of the **Merge**, while viewing the output of the **Merge**.

Set the compositing operator of each roto shape to “min” instead of “over”, so that the original image alpha gets masked by the roto shape.

Do not forget to add feather, especially in the overlap area.

Then, reconnect the A2 input to the third **Card3D**, and start editing its roto mask, always in “min” compositing mode.

Do the same with A3, A4, etc. . . and you should end up with a full panorama.

Future work

Camera response

See [Camera response curve](#) and [Vig_optimize](#).

The values `Ra Rb Rc Rd Re` on the `i` line encode a color response curve (EMoR). See `EMoRParams` in the hugin source code.

Exposure and color

See [Vignetting](#) and [Vig_optimize](#).

`Eev` encodes the exposure, see `ExposureValue` in the hugin source code.

`Er` and `Eb` encode the red and blue multipliers, see `WhiteBalanceRed` and `WhiteBalanceBlue` in the hugin source code.

Vignetting

See [Vignetting](#) and [Vig_optimize](#).

`Va` is always 1, see `VigCorrMode` in the hugin source code.

`Vb Vc Vd` encode the degree 2, 4 and 6 coefficients for vignette correction, see `RadialVigCorrCoeff` in the hugin source code.

`Vx` and `Vy` encode the vignetting center shift, see `RadialVigCorrCenterShift` in the hugin source code.

Blending

- importing enblend masks, using them as roto masks
- executing enblend externally (using `RunScript` maybe?)

1.4.5 Vector graphics workflow



Software development is constantly growing. The film and multimedia industry are coming together as partners to provide a one stop shop by allowing applications to communicate via different file formats. One of the formats that have proven to be useful in both industries is the SVG (Scalable Vector Graphics) format. Its architecture algorithm is based on mathematical expressions. In simpler terms, it doesn't suffer image noise & artifacts like bitmap formats such as PNG, TIFF, JPG and etc.

Inkscape's SVG format uses multiple layers and paths, but most graphics applications can only parse this as a flat bitmap. There will times when an artist may want to animate and/or modify those layers in a compositing program but can't due to limitation in the SVG importer. Natron however supports all layers and paths in the SVG file, this enables enhanced control over the vector graphics.

This tutorial will show you how to use vector graphics from Inkscape in Natron.

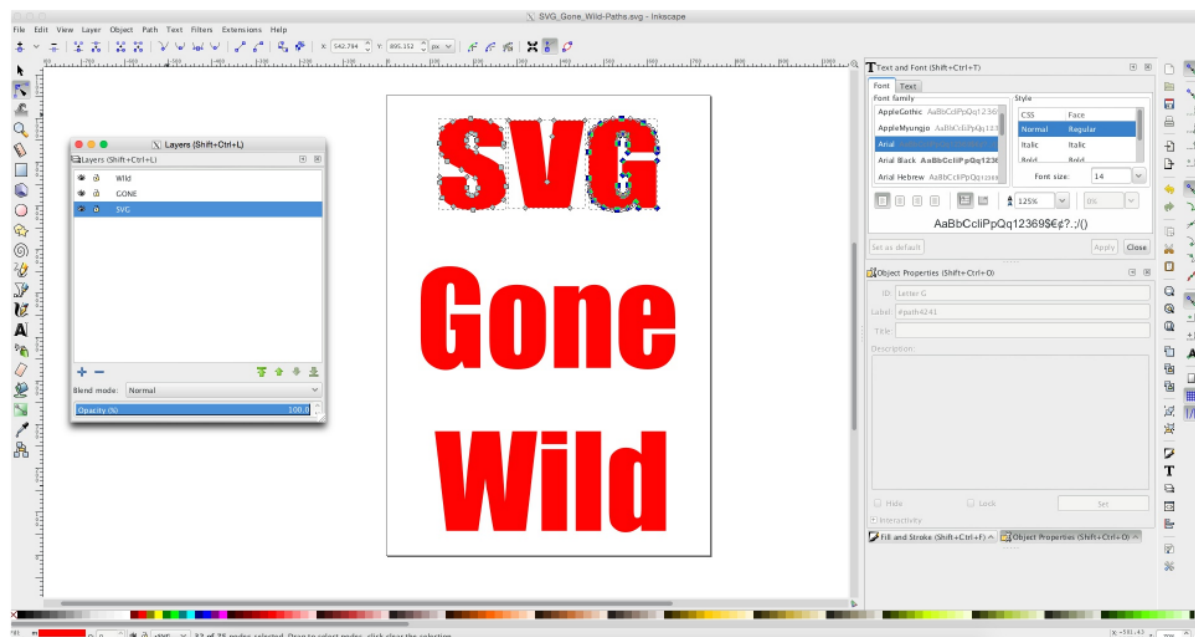
Inkscape



Inkscape has multiple ways to create vector graphics to be saved and imported into Natron. The default is to first create your document layout. This entails formatting the resolution for the project upon which your vector graphics will be displayed. Inkscape starts with one layer and you can draw and type your graphics on that layer or additional layers. You can then save the project to SVG file.

This image is SVG letters being converted to a group and layers for each word so that they may have their own layer/alpha channel in Natron.

The image below is the same vector letters being converted to paths and eventually ungrouping each of the vector letters so that they may have their own layer and alpha channel in Natron.



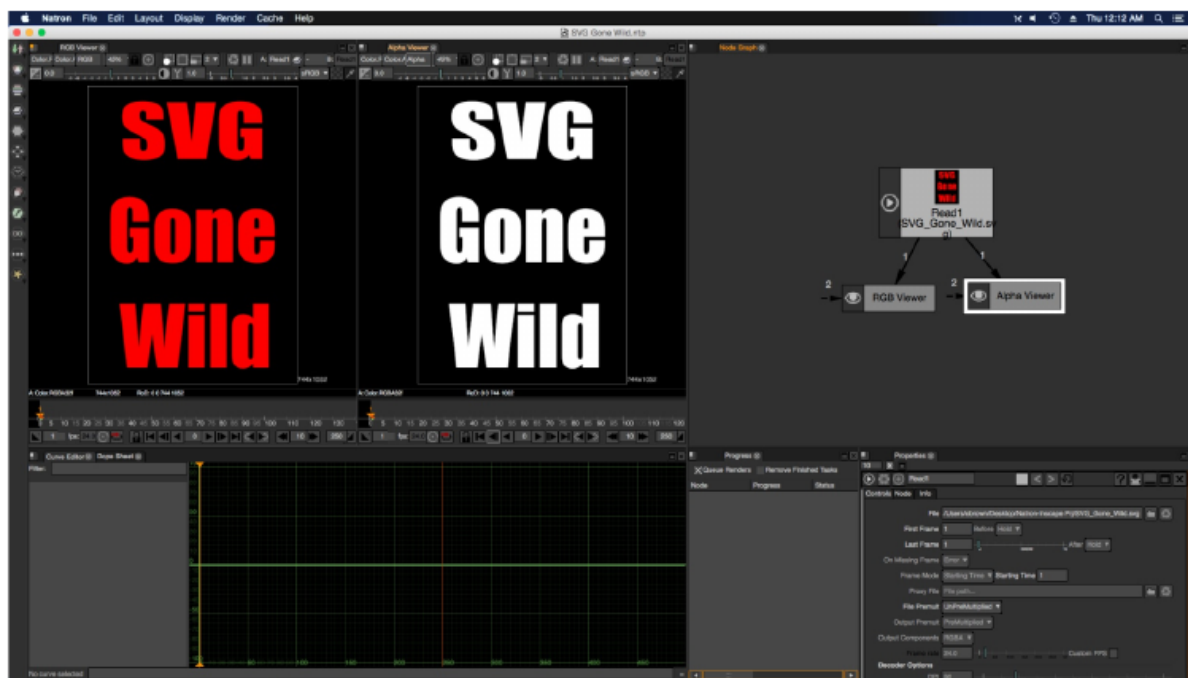
Note: Remember to make sure that you convert your objects to paths, it is highly advisable to name each path with its separate id name. It will make it much easier to find Color.RGB and alpha channel names in Natron after

loading the SVG file in the node graph.

It is also possible and good practice to select objects/paths and create groups of them. This will allow easy animation work-flows without having to duplicate animated key-frames if needed. Those groups will also show up as Color.RGB and Alpha channels in Natron.

Natron

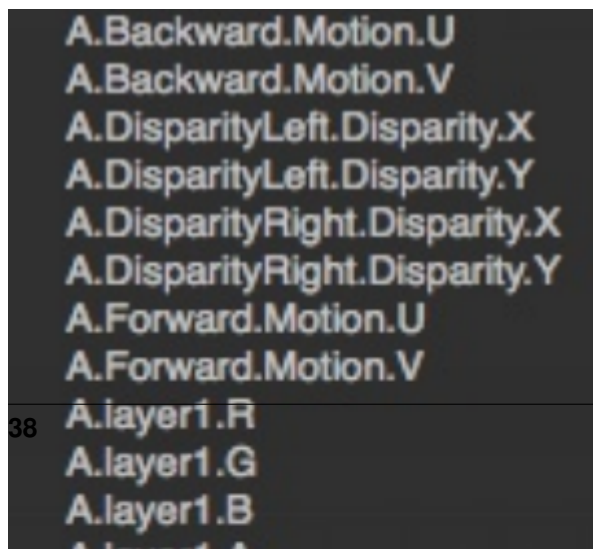
The image below is Natron with the SVG file loaded. We have two viewers displaying the Color.RGB and alpha channels generated from the SVG file.



After you have loaded the file, you can check the Color.RGB/Alpha Channels headers to see how the layers, paths and groups are read. The ReadSVG node is multi-plane aware. The next few images are Natron screen captures of the headers, merge and shuffle nodes.

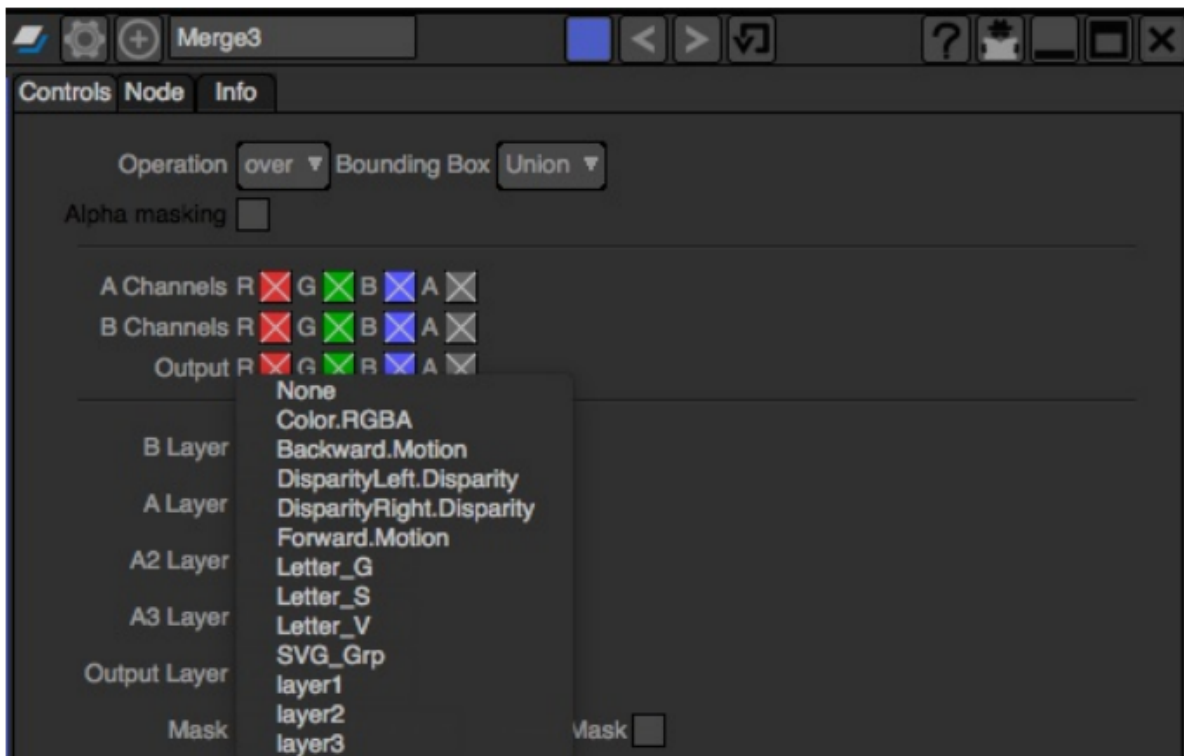
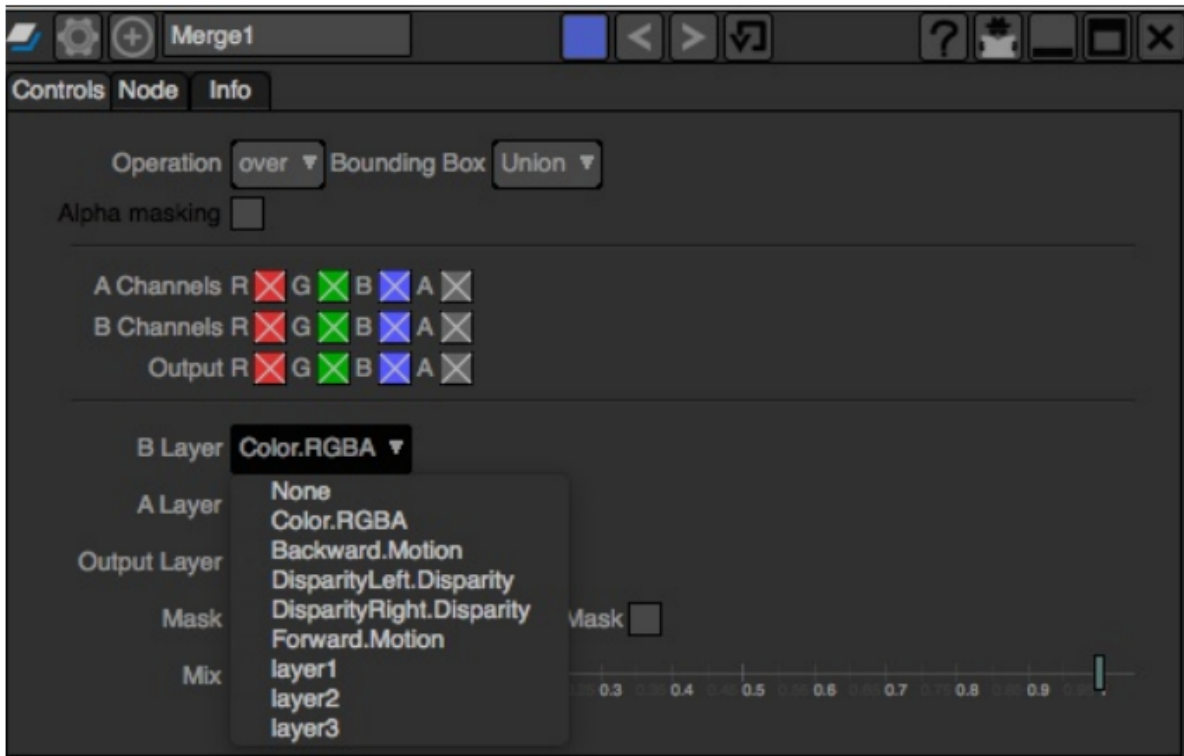
Note: To clarify the Merge: Paths Channels image. In the SVG image, I broke apart the vector characters and converted them to paths. Then each letter of the SVG was giving the name assigned to that letter. That information got saved as individual Color.RGB/Alpha channels.

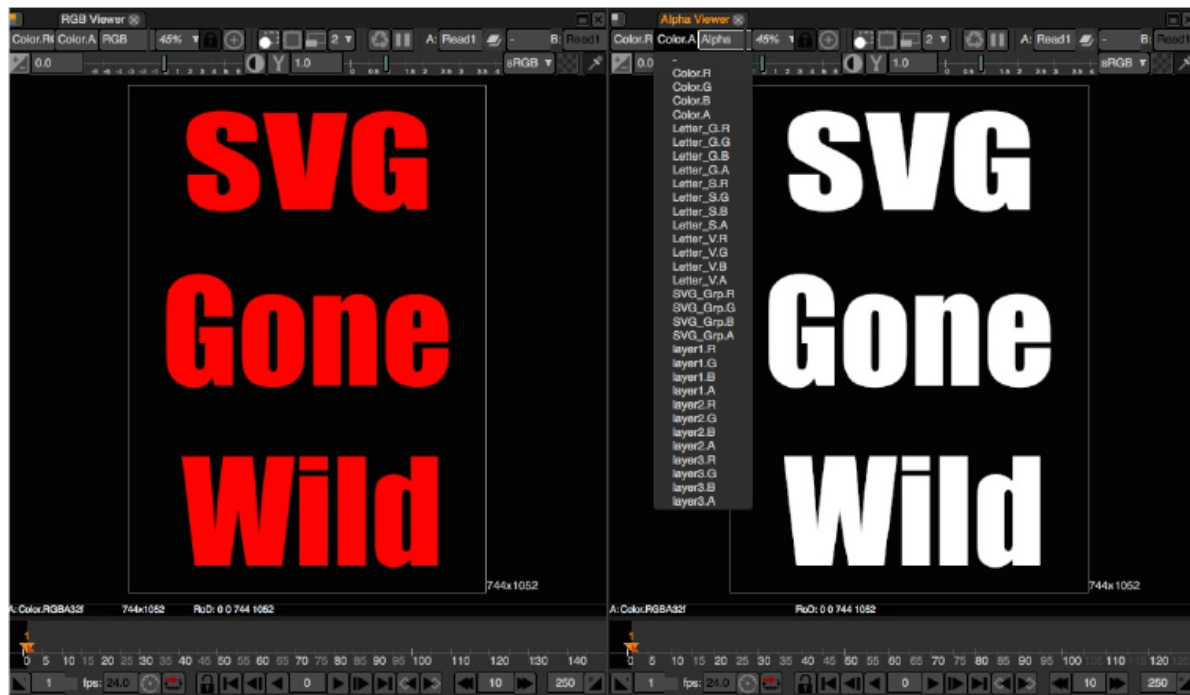
Be sure to look carefully at the Color.RGB and alpha headers in the images below. When your finished designing your vector document with layers/paths/groups, this is where the channels are displayed.



The Shuffle and Merge nodes are used to access multi-plane layers. Though they both can process the different SVG files objects, paths, layers, they work differently. As you can see in the image on the left the shuffle node display every channel(s) from a file and also copy channels from other files into the “A” input from the “B” input.

The merge node only perform mathematical blending operations on the inputs “B” and “A:A1000”. In order to access the alpha channels from the SVG file you must use the maskChannel mask inputs.

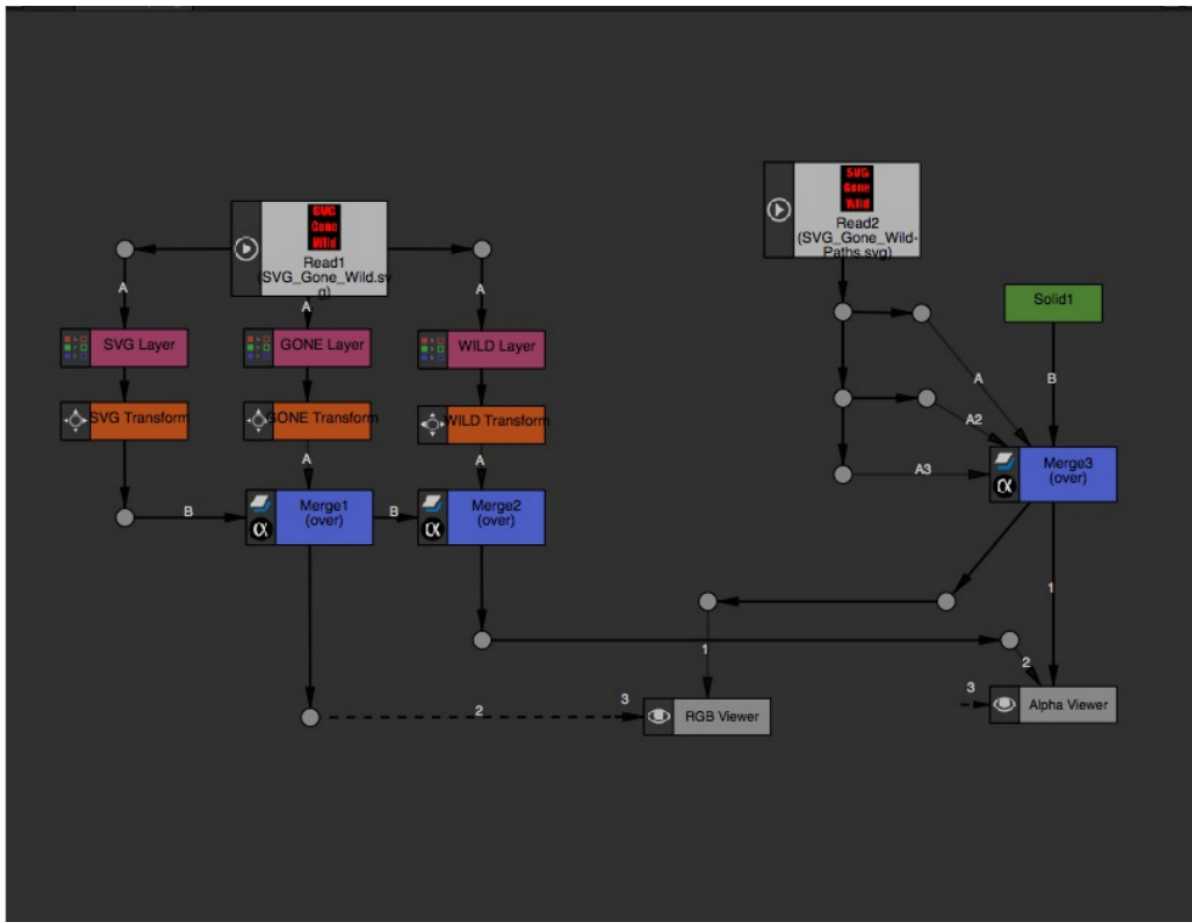




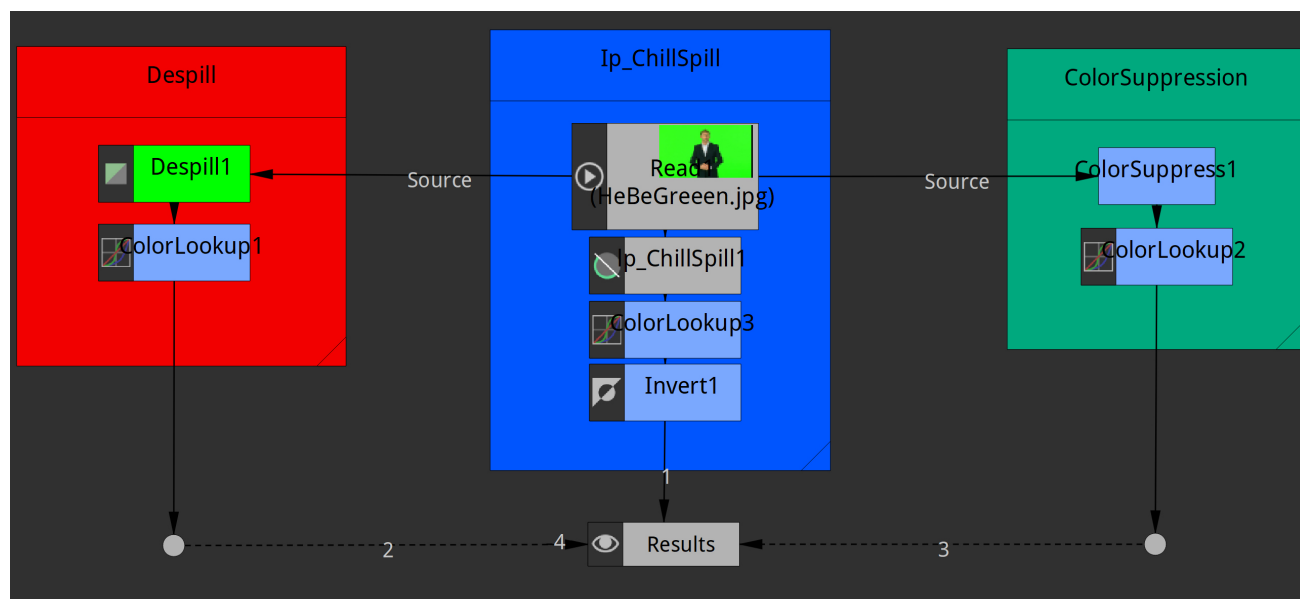
So plan your work before you start a project. Organization is key. Make sure that Layer(s), Object(s), Path(s) and Group(s) have logical naming conventions.



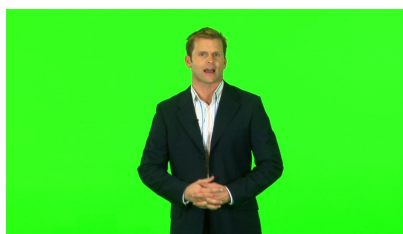
The node graph below shows how to extract individual layers using the shuffle node and the merge node. Both can extract the RGB layers and individual paths from an SVG file, but the Shuffle node allow you to pipe any of the layers or single channels to another node mask input.



1.4.6 Alternative Matte Extraction Tutorial

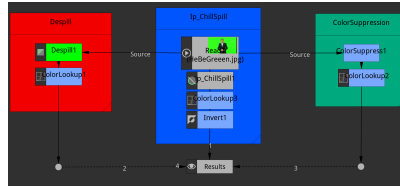


In the world of vfx in current films these days, it is hard to even phathom that **pulling keys** (aka **chromakeying**) or generally just creating mattes from images is not common place. Today I want to share some features in a few nodes that are in **Natron VFX Digital Compositor**. The nodes that I want to discuss are **Despill**, **Ip_ChillSpill** and **ColorSuppression**. You can guess by the name the functions that they perform. Basically, they **subtract any blue or green screen spillage that happens to contaminate your foreground objects during the production process**. These types of functions are common place in every post-production facilities in the industry. It doesn't matter if you are a beginner wanting to produce your own short films or a professional working on block buster films. The needs are the same. The attached image is a greenscreen image that I pulled off google to demonstrate what the Natron developers had implemented to take these despillers to another level or just added functionality.

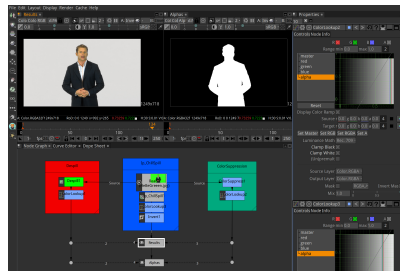


The added functionality is having the ability to use the suppressed or despill color information and convert it to a matte or alpha. I am unaware if any other compositing applications has these abilities. Natron is my main compositing app and from time to time I use these **despilling node algorithms** to help generate masks, general mattes and scaled alphas. This is Natron's node graph pipeline for each node that I will be discussing. The pipeline for each node is really simple. You just connect the green/blue screen footage to the input of the nodes, adjust whatever you have to adjust, click on the very simple knob that

says “Spillmap to Alpha” if you are using the Despill node, “Shuffle Spillmatte to Alpha” if you are using the community openfx plugin called **Ip_ChillSpill Node**, and “**Output: Image, Alpha & Image and Alpha**” if you are using the ColorSuppression node.



The first screenshot demonstration is for the node **Ip_ChillSpill**. This despilling node has the most of features and functions that I can tell that exist amongst all the despilling nodes. In the image below you will see the spill suppression on the left and the alpha channel from selecting “**Shuffle Spillmatte to Alpha**” on the right.



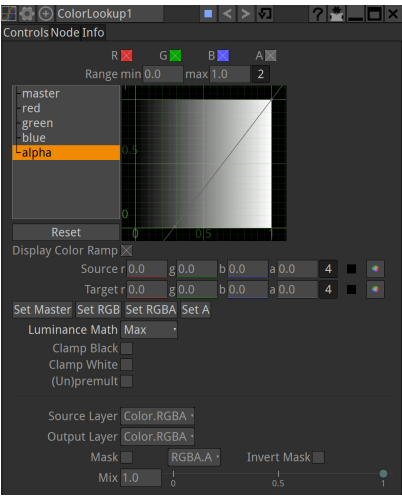
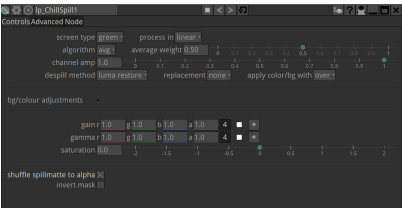
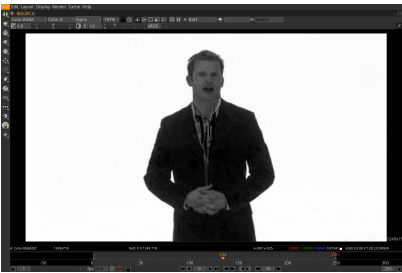
Now just selecting the **Shuffle Spillmatte to Alpha** feature is not some magic trick and you get a perfect matte, not by a long shot. For a matter of fact it's not for any of them. Attached are examples of the nodes in their default state before the scaling process begins. The first image is the **Ip_ChillSpill** default matte output and the second is the **ColorSuppression** default matte output. It looks like if I was trying to use the **HSVTool Node** to pull a **Saturation or Brightness Key**. You can read more about that in my **HSVTool node tutorial**. I used a very underrated and underused node amongst beginners called the **ColorLookup Node**. You can be very familiar with the node if you have used Photoshop or Gimp's curve tool.



Here are the nodes and their settings to show what I had to do to get it to scale my suppression mattes. The key tool is to use the **ColorLookup Node** connected after the despilling nodes. The **ColorLookup Node** has four color channels and the channel that you use to scale your mattes is the “**alpha curve channel**”. The bottom left of the **alpha curve** is used to **crush your blacks/shadows** and the top right is used to **extend your white/highlights**. In the **ColorLookup Node** you will also see a feature called “**Luminance Math**”. This feature will yield its full benefits based off the resolution and color spaces of your footage. The **ColorLookup node** is very powerful in a sense because the channel curves can have multiple points to limit its effects.

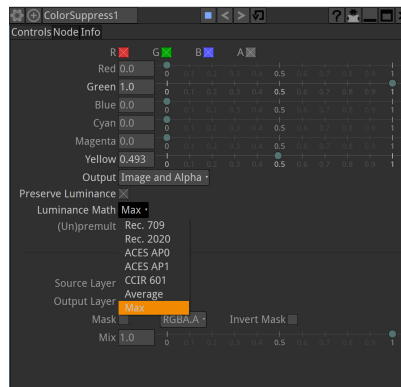
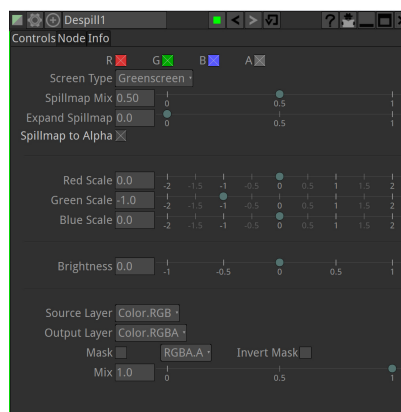
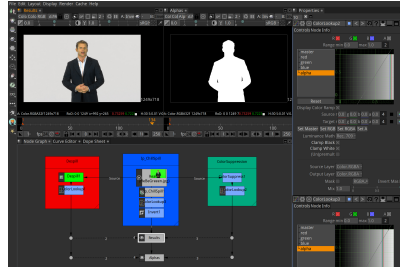
As you can see these nodes all perform the same functions but some have different parameters to accomplish the same thing and well as providing other color processing effects. Here are some screen captures of the effects using the **ColorLookup Node**.

Now this by node means a **primary replacement for powerful keying node in Natron**. The **extended functionality should only be considered as a complement to Chromakeyer, PIK/PIK Color and Keyer nodes**. Just remember that the **Here are some screen captures of the effects using the ColorLookup node is needed to scaled that matte**. Also this process doesn't treat your edges with a choking or eroding effect. You would have to experiment with some of the filters to process them. Now you can try and cheat by using the **Shuffle Node** to convert to this matte output to an real alpha channel and maybe you can process your edges directly as if you were coming



out of a keyer. You will need to do some serious testing. After you have done that, please feel free to talk about it and join [NatronNation](#) and read my [blog](#).

Despill and Color Suppression Pipeline





The first section in this manual describes the various options available from the Natron preference settings. It is followed by one section for each node group in Natron. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.1 Preferences

2.1.1 General

Always check for updates on start-up

When checked, Natron will check for new updates on start-up of the application.

Auto-save trigger delay

The number of seconds after an event that Natron should wait before auto-saving. Note that if a render is in progress, Natron will wait until it is done to actually auto-save.

Enable Auto-save for unsaved projects

When activated Natron will auto-save projects that have never been saved and will prompt you on startup if an auto-save of that unsaved project was found. Disabling this will no longer save un-saved project.

Appear to plug-ins as

Natron will appear with the name of the selected application to the OpenFX plug-ins. Changing it to the name of another application can help loading plugins which restrict their usage to specific OpenFX host(s). If a Host is not listed here, use the “Custom” entry to enter a custom host name. Changing this requires a restart of the application and requires clearing the OpenFX plugins cache from the Cache menu.

2.1.2 Threading

Number of render threads (0=“guess”)

Controls how many threads Natron should use to render.

-1: Disable multithreading totally (useful for debugging)

0: Guess the thread count from the number of cores.

Number of parallel renders (0=“guess”)

Controls the number of parallel frame that will be rendered at the same time by the renderer. A value of 0 indicate that Natron should automatically determine the best number of parallel renders to launch given your CPU activity. Setting a value different than 0 should be done only if you know what you’re doing and can lead in some situations to worse performances. Overall to get the best performances you should have your CPU at 100% activity without idle times.

Effects use the thread-pool

When checked, all effects will use a global thread-pool to do their processing instead of launching their own threads. This suppresses the overhead created by the operating system creating new threads on demand for each rendering of a special effect. As a result of this, the rendering might be faster on systems with a lot of cores (≥ 8).

WARNING: This is known not to work when using The Foundry’s Furnace plug-ins (and potentially some other plug-ins that the dev team hasn’t not tested against it). When using these plug-ins, make sure to uncheck this option first otherwise it will crash Natron.

Max threads usable per effect (0=“guess”)

Controls how many threads a specific effect can use at most to do its processing. A high value will allow 1 effect to spawn lots of thread and might not be efficient because the time spent to launch all the threads might exceed the time spent actually processing. By default (0) the renderer applies an heuristic to determine what’s the best number of threads for an effect.

Render in a separate process

If true, Natron will render frames to disk in a separate process so that if the main application crashes, the render goes on.

Append new renders to queue

When checked, renders will be queued in the Progress Panel and will start only when all other prior tasks are done.

2.1.3 Rendering

Convert NaN values

When activated, any pixel that is a Not-a-Number will be converted to 1 to avoid potential crashes from downstream nodes. These values can be produced by faulty plug-ins when they use wrong arithmetic such as division by zero. Disabling this option will keep the NaN(s) in the buffers: this may lead to an undefined behavior.

Copy input image before rendering any plug-in

If checked, when before rendering any node, Natron will copy the input image to a local temporary image. This is to work-around some plug-ins that write to the source image, thus modifying the output of the node upstream in the cache. This is a known bug of an old version of RevisionFX REmap for instance. By default, this parameter should be leaved unchecked, as this will require an extra image allocation and copy before rendering any plug-in.

RGB components support

When checked Natron is able to process images with only RGB components (support for images with RGBA and Alpha components is always enabled). Un-checking this option may prevent plugins that do not well support RGB components from crashing Natron. Changing this option requires a restart of the application.

Transforms concatenation support

When checked Natron is able to concatenate transform effects when they are chained in the compositing tree. This yields better results and faster render times because the image is only filtered once instead of as many times as there are transformations.

2.1.4 GPU Rendering

Active OpenGL renderer

The currently active OpenGL renderer.

OpenGL renderer

The renderer used to perform OpenGL rendering. Changing the OpenGL renderer requires a restart of the application.

No. of OpenGL Contexts

The number of OpenGL contexts created to perform OpenGL rendering. Each OpenGL context can be attached to a CPU thread, allowing for more frames to be rendered simultaneously. Increasing this value may increase performances for graphs with mixed CPU/GPU nodes but can drastically reduce performances if too many OpenGL contexts are active at once.

OpenGL Rendering

Select whether to activate OpenGL rendering or not. If disabled, even though a Project enable GPU rendering, it will not be activated.

2.1.5 Project Setup

First image read set project format

If checked, the project size is set to this of the first image or video read within the project.

Auto-preview enabled by default for new projects

If checked, then when creating a new project, the Auto-preview option is enabled.

Auto fix relative file-paths

If checked, when a project-path changes (either the name or the value pointed to), Natron checks all file-path parameters in the project and tries to fix them.

Use drive letters instead of server names (Windows only)

This is only relevant for Windows: If checked, Natron will not convert a path starting with a drive letter from the file dialog to a network share name. You may use this if for example you want to share a same project with several users across facilities with different servers but where users have all the same drive attached to a server.

2.1.6 Documentation

Documentation Source

Documentation source.

Documentation local port (0=auto)

The port onto which the documentation server will listen to. A value of 0 indicate that the documentation should automatically find a port by itself.

2.1.7 User Interface

Warn when a file changes externally

When checked, if a file read from a file parameter changes externally, a warning will be displayed on the viewer. Turning this off will suspend the notification system.

Prompt with file dialog when creating Write node

When checked, opens-up a file dialog when creating a Write node

Refresh viewer only when editing is finished

When checked, the viewer triggers a new render only when mouse is released when editing parameters, curves or the timeline. This setting doesn't apply to roto splines editing.

Linear color pickers

When activated, all colors picked from the color parameters are linearized before being fetched. Otherwise they are in the same colorspace as the viewer they were picked from.

Maximum number of open settings panels (0="unlimited")

This property holds the maximum number of settings panels that can be held by the properties dock at the same time. The special value of 0 indicates there can be an unlimited number of panels opened.

Value increments based on cursor position

When enabled, incrementing the value fields of parameters with the mouse wheel or with arrow keys will increment the digits on the right of the cursor.

When disabled, the value fields are incremented given what the plug-in decided it should be. You can alter this increment by holding Shift (x10) or Control (/10) while incrementing.

Default layout file

When set, Natron uses the given layout file as default layout for new projects. You can export/import a layout to/from a file from the Layout menu. If empty, the default application layout is used.

Load workspace embedded within projects

When checked, when loading a project, the workspace (windows layout) will also be loaded, otherwise it will use your current layout.

2.1.8 Color Management

OpenColorIO configuration

Select the OpenColorIO configuration you would like to use globally for all operators and plugins that use OpenColorIO, by setting the "OCIO" environment variable. Only nodes created after changing this parameter will take it into account, and it is better to restart the application after changing it. When "Custom config" is selected, the "Custom OpenColorIO config file" parameter is used.

Custom OpenColorIO configuration file

OpenColorIO configuration file (config.ocio) to use when "Custom config" is selected as the OpenColorIO config.

Warn on OpenColorIO config change

Show a warning dialog when changing the OpenColorIO config to remember that a restart is required.

2.1.9 Caching

Aggressive caching

When checked, Natron will cache the output of all images rendered by all nodes, regardless of their "Force caching" parameter. When enabling this option you need to have at least 8GiB of RAM, and 16GiB is recommended.

If not checked, Natron will only cache the nodes which have multiple outputs, or their parameter "Force caching" checked or if one of its output has its settings panel opened.

Maximum amount of RAM memory used for caching (% of total RAM)

This setting indicates the percentage of the total RAM which can be used by the memory caches.

System RAM to keep free (% of total RAM)

This determines how much RAM should be kept free for other applications running on the same system. When this limit is reached, the caches start recycling memory instead of growing. This value should reflect the amount of memory you want to keep available on your computer for other usage. A low value may result in a massive slowdown and high disk usage.

Maximum playback disk cache size (GiB)

The maximum size that may be used by the playback cache on disk (in GiB)

Maximum DiskCache node disk usage (GiB)

The maximum size that may be used by the DiskCache node on disk (in GiB)

Disk cache path (empty = default)

WARNING: Changing this parameter requires a restart of the application.

This points to the location where Natron on-disk caches will be. This variable should point to your fastest disk. If the parameter is left empty or the location set is invalid, the default location will be used.

Wipe Disk Cache

Cleans-up all caches, deleting all folders that may contain cached data. This is provided in case Natron lost track of cached images for some reason.

2.1.10 Viewer

Viewer textures bit depth

Bit depth of the viewer textures used for rendering. Hover each option with the mouse for a detailed description.

Viewer tile size is 2 to the power of...

The dimension of the viewer tiles is 2^n by 2^n (i.e. 256 by 256 pixels for $n=8$). A high value means that the viewer renders large tiles, so that rendering is done less often, but on larger areas.

Checkerboard tile size (pixels)

The size (in screen pixels) of one tile of the checkerboard.

Checkerboard color 1

The first color used by the checkerboard.

Checkerboard color 2

The second color used by the checkerboard.

Automatically enable wipe

When checked, the wipe tool of the viewer will be automatically enabled when the mouse is hovering the viewer and changing an input of a viewer.

Automatically enable proxy when scrubbing the timeline

When checked, the proxy mode will be at least at the level indicated by the auto-proxy parameter.

Max. opened node viewer interface

Controls the maximum amount of nodes that can have their interface showing up at the same time in the viewer

Use number keys for the viewer

When enabled, the row of number keys on the keyboard is used for switching input (<key> connects input to A side, <shift-key> connects input to B side), even if the corresponding character in the current keyboard layout is not a number.

This may have to be disabled when using a remote display connection to Linux from a different OS.

2.1.11 Nodegraph

Auto Scroll

When checked the node graph will auto scroll if you move a node outside the current graph view.

Auto-turbo

When checked the Turbo-mode will be enabled automatically when playback is started and disabled when finished.

Snap to node

When moving nodes on the node graph, snap to positions where they are lined up with the inputs and output nodes.

Maximum undo/redo for the node graph

Set the maximum of events related to the node graph Natron remembers. Past this limit, older events will be deleted forever, allowing to re-use the RAM for other purposes.

Changing this value will clear the undo/redo stack.

Disconnected arrow length

The size of a disconnected node input arrow in pixels.

Auto hide masks inputs

When checked, any disconnected mask input of a node in the nodegraph will be visible only when the mouse is hovering the node or when it is selected.

Merge node connect to A input

If checked, upon creation of a new Merge node, or any other node with inputs named A and B, input A is preferred for auto-connection. When the node is disabled, B is always output, whether this is checked or not.

2.1.12 Plug-ins

Use bundled plug-ins

When checked, Natron also uses the plug-ins bundled with the binary distribution.

When unchecked, only system-wide plug-ins found in are loaded (more information can be found in the help for the “Extra plug-ins search paths” setting).

Prefer bundled plug-ins over system-wide plug-ins

When checked, and if “Use bundled plug-ins” is also checked, plug-ins bundled with the Natron binary distribution will take precedence over system-wide plug-ins if they have the same internal ID.

Enable default OpenFX plugins location

When checked, Natron also uses the OpenFX plug-ins found in the default location (/Library/OFX/Plugins).

OpenFX plug-ins search path

Extra search paths where Natron should scan for OpenFX plug-ins. Extra plug-ins search paths can also be specified using the OFX_PLUGIN_PATH environment variable.

The priority order for system-wide plug-ins, from high to low, is:

- plugins bundled with the binary distribution of Natron (if “Prefer bundled plug-ins over system-wide plug-ins” is checked)
- plug-ins found in OFX_PLUGIN_PATH
- plug-ins found in /Library/OFX/Plugins (if “Enable default OpenFX plug-ins location” is checked)
- plugins bundled with the binary distribution of Natron (if “Prefer bundled plug-ins over system-wide plug-ins” is not checked)

Any change will take effect on the next launch of Natron.

PyPlugs search path

Search path where Natron should scan for Python group scripts (PyPlugs). The search paths for groups can also be specified using the NATRON_PLUGIN_PATH environment variable.

2.1.13 Python

After project created

Callback called once a new project is created (this is never called when “After project loaded” is called.)

The signature of the callback is : callback(app) where:

- app: points to the current application instance

Default after project loaded

The default afterProjectLoad callback that will be set for new projects.

Default before project save

The default beforeProjectSave callback that will be set for new projects.

Default before project close

The default beforeProjectClose callback that will be set for new projects.

Default after node created

The default afterNodeCreated callback that will be set for new projects.

Default before node removal

The default beforeNodeRemoval callback that will be set for new projects.

Load PyPlugs in projects from .py if possible

When checked, if a project contains a PyPlug, it will try to first load the PyPlug from the .py file. If the version of the PyPlug has changed Natron will ask you whether you want to upgrade to the new version of the PyPlug in your project. If the .py file is not found, it will fallback to the same behavior as when this option is unchecked. When unchecked the PyPlug will load as a regular group with the informations embedded in the project file.

Print auto-declared variables in the Script Editor

When checked, Natron will print in the Script Editor all variables that are automatically declared, such as the app variable or node attributes.

2.1.14 Appearance

Font

List of all fonts available on your system

Stylesheet file (.qss)

When pointing to a valid .qss file, the stylesheet of the application will be set according to this file instead of the default stylesheet. You can adapt the default stylesheet that can be found in your distribution of Natron.

Main Window

Use black & white toolbutton icons

When checked, the tools icons in the left toolbar are greyscale. Changing this takes effect upon the next launch of the application.

Curve Editor

Dope Sheet

Node Graph

Display plug-in icon on node-graph

When checked, each node that has a plug-in icon will display it in the node-graph. Changing this option will not affect already existing nodes, unless a restart of Natron is made.

Anti-Aliasing

When checked, the node graph will be painted using anti-aliasing. Unchecking it may increase performances. Changing this requires a restart of Natron

Default node color

The default color used for newly created nodes.

Default backdrop color

The default color used for newly created backdrop nodes.

Readers

The color used for newly created Reader nodes.

Writers

The color used for newly created Writer nodes.

Generators

The color used for newly created Generator nodes.

Color group

The color used for newly created Color nodes.

Filter group

The color used for newly created Filter nodes.

Transform group

The color used for newly created Transform nodes.

Time group

The color used for newly created Time nodes.

Draw group

The color used for newly created Draw nodes.

Keyer group

The color used for newly created Keyer nodes.

Channel group

The color used for newly created Channel nodes.

Merge group

The color used for newly created Merge nodes.

Views group

The color used for newly created Views nodes.

Deep group

The color used for newly created Deep nodes.

Script Editor

Font

List of all fonts available on your system

Font Size

The font size

2.2 Image nodes

The following sections contain documentation about every node in the Image group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.2.1 CheckerBoard node



This documentation is for version 1.0 of CheckerBoard.

Description

Generate an image with a checkerboard.

A frame range may be specified for operators that need it.

See also: http://opticalenquiry.com/nuke/index.php?title=Constant,_CheckerBoard,_ColorBars,_ColorWheel

Inputs

Input	Description	Optional
Source		Yes

Controls

Parameter / script name	Type	Default	Function
Extent / extent	Choice	Default	<p>Extent (size and offset) of the output.</p> <p>Format (format): Use a pre-defined image format.</p> <p>Size (size): Use a specific extent (size and offset).</p> <p>Project (project): Use the project extent (size and offset).</p> <p>Default (default): Use the default extent (e.g. the source clip extent, if connected).</p>

Continued on next page

Table 1 – continued from previous page

Parameter / script name	Type	Default	Function
Center / recenter	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.
Reformat / reformat	Boolean	Off	Set the output format to the given extent, except if the Bottom Left or Size parameters is animated.
Format / NatronParamFormatChoice	Choice	HD 1920x1080	<p>The output format</p> <p>PC_Video 640x480 (PC_Video)</p> <p>NTSC 720x486 0.91 (NTSC)</p> <p>PAL 720x576 1.09 (PAL)</p> <p>NTSC_16:9 720x486 1.21 (NTSC_16:9)</p> <p>PAL_16:9 720x576 1.46 (PAL_16:9)</p> <p>HD_720 1280x720 (HD_720)</p> <p>HD 1920x1080 (HD)</p> <p>UHD_4K 3840x2160 (UHD_4K)</p> <p>1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap))</p> <p>1K_Cinemascope 914x778 2.00 (1K_Cinemascope)</p> <p>2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap))</p> <p>2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope)</p> <p>2K_DCP 2048x1080 (2K_DCP)</p> <p>4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap))</p> <p>4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope)</p> <p>4K_DCP 4096x2160 (4K_DCP)</p> <p>square_256 256x256 (square_256)</p> <p>square_512 512x512 (square_512)</p> <p>square_1K 1024x1024 (square_1K)</p> <p>square_2K 2048x2048 (square_2K)</p>
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Frame Range / frameRange	Integer	min: 1 max: 1	Time domain.
Output Components / outputComponents	Choice	RGBA	<p>Components in the output</p> <p>RGBA</p> <p>RGB</p> <p>Alpha</p>
Box Size / boxSize	Double	x: 64 y: 64	Size of the checkerboard boxes in pixels.
Color 0 / color0	Color	r: 0.1 g: 0.1 b: 0.1 a: 1	Color to fill the box on top-left of image center and every other row and column.
Color 1 / color1	Color	r: 0.5 g: 0.5 b: 0.5 a: 1	Color to fill the box on top-right of image center and every other row and column.

Continued on next page

Table 1 – continued from previous page

Parameter / script name	Type	Default	Function
Color 2 / <code>color2</code>	Color	r: 0.1 g: 0.1 b: 0.1 a: 1	Color to fill the box on bottom-right of image center and every other row and column.
Color 3 / <code>color3</code>	Color	r: 0.5 g: 0.5 b: 0.5 a: 1	Color to fill the box on bottom-left of image center and every other row and column.
Line Color / <code>lineColor</code>	Color	r: 1 g: 1 b: 1 a: 1	Color of the line drawn between boxes.
Line Width / <code>lineWidth</code>	Double	0	Width, in pixels, of the lines drawn between boxes.
Centerline Color / <code>centerlineColor</code>	Color	r: 1 g: 1 b: 0 a: 1	Color of the center lines.
Centerline Width / <code>centerlineWidth</code>	Double	1	Width, in pixels, of the center lines.

2.2.2 ColorBars node



This documentation is for version 1.0 of ColorBars.

Description

Generate an image with SMPTE RP 219:2002 color bars.

The output of this plugin is broadcast-safe if “Output IRE” is unchecked. Be careful that colorbars are defined in a nonlinear colorspace. In order to get linear RGB, this plug-in should be combined with a transformation from the video space to linear.

See also: http://opticalenquiry.com/nuke/index.php?title=Constant,_CheckerBoard,_ColorBars,_ColorWheel

Inputs

Input	Description	Optional
Source		Yes

Controls

Parameter / script name	Type	Default	Function
Extent / extent	Choice	Default	Extent (size and offset) of the output. Format (format): Use a pre-defined image format. Size (size): Use a specific extent (size and offset). Project (project): Use the project extent (size and offset). Default (default): Use the default extent (e.g. the source clip extent, if connected).
Center / recenter	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.
Reformat / reformat	Boolean	Off	Set the output format to the given extent, except if the Bottom Left or Size parameters is animated.
Format / NatronParamFormat	Choice	HD 1920x1080	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Frame Range / frameRange	Integer	min: 1 max: 1	Time domain.
Output Components / outputComponents	Choice	RGBA	Components in the output RGBA RGB
Bar Intensity / barIntensity	Double	75	Bar Intensity, in IRE unit.

Continued on next page

Table 2 – continued from previous page

Parameter / script name	Type	Default	Function
Output IRE / outputIRE	Boolean	Off	When checked, the output is scaled so that 0 is black, the max value is white, and the superblack (under the middle of the magenta bar) has a negative value.

2.2.3 ColorWheel node



This documentation is for version 1.0 of ColorWheel.

Description

Generate an image with a color wheel.

The color wheel occupies the full area, minus a one-pixel black and transparent border

See also: http://opticalenquiry.com/nuke/index.php?title=Constant,_CheckerBoard,_ColorBars,_ColorWheel

Inputs

Input	Description	Optional
Source		Yes

Controls

Parameter / script name	Type	Default	Function
Extent / extent	Choice	Default	Extent (size and offset) of the output. Format (format): Use a pre-defined image format. Size (size): Use a specific extent (size and offset). Project (project): Use the project extent (size and offset). Default (default): Use the default extent (e.g. the source clip extent, if connected).
Center / recenter	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.
Reformat / reformat	Boolean	Off	Set the output format to the given extent, except if the Bottom Left or Size parameters is animated.

Continued on next page

Table 3 – continued from previous page

Parameter / script name	Type	Default	Function
Format / NatronParamFormatChoice	Choice	HD 1920x1080	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Frame Range / frameRange	Integer	min: 1 max: 1	Time domain.
Output Components / outputComponents	Choice	RGBA	Components in the output RGBA RGB XY Alpha
Center Saturation / centerSaturation	Double	0	Sets the HSV saturation level in the center of the color wheel.
Edge Saturation / edgeSaturation	Double	1	Sets the HSV saturation level at the edges of the color wheel.
Center Value / centerValue	Double	1	Sets the HSV value level in the center of the color wheel.
Edge Value / edgeValue	Double	1	Sets the HSV value level at the edges of the color wheel.
Gamma / gamma	Double	0.45	Sets the overall gamma level of the color wheel.
Rotate / rotate	Double	0	Sets the amount of rotation to apply to color position in the color wheel. Negative values produce clockwise rotation and vice-versa.

2.2.4 Constant node



This documentation is for version 1.0 of Constant.

Description

Generate an image with a constant color.

See also: http://opticalenquiry.com/nuke/index.php?title=Constant,_CheckerBoard,_ColorBars,_ColorWheel

Inputs

Input	Description	Optional
Source		Yes

Controls

Parameter / script name	Type	Default	Function
Extent / extent	Choice	Default	Extent (size and offset) of the output. Format (format): Use a pre-defined image format. Size (size): Use a specific extent (size and offset). Project (project): Use the project extent (size and offset). Default (default): Use the default extent (e.g. the source clip extent, if connected).
Center / recenter	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.
Reformat / reformat	Boolean	Off	Set the output format to the given extent, except if the Bottom Left or Size parameters is animated.

Continued on next page

Table 4 – continued from previous page

Parameter / script name	Type	Default	Function
Format / NatronParamFormatChoice	Choice	HD 1920x1080	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Frame Range / frameRange	Integer	min: 1 max: 1	Time domain.
Output Components / outputComponents	Choice	RGBA	Components in the output RGBA RGB XY Alpha
Color / color	Color	r: 0 g: 0 b: 0 a: 0	Color to fill the image with.

2.2.5 OpenRaster node



This documentation is for version 2.1 of OpenRaster.

Description

Read OpenRaster image format.

Inputs

Input	Description	Optional
Sync	Sync	Yes

Controls

Parameter / script name	Type	Default	Function
File / filename	N/A		The input image sequence/video stream file(s).
First Frame / firstFrame	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the first output frame.
Before / before	Choice	Hold	What to do before the first frame of the sequence. Hold (hold) : While before the sequence, load the first frame. Loop (loop) : Repeat the sequence before the first frame Bounce (bounce) : Repeat the sequence in reverse before the first frame Black (black) : Render a black image Error (error) : Report an error
Last Frame / lastFrame	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the last output frame.
After / after	Choice	Hold	What to do after the last frame of the sequence. Hold (hold) : While before the sequence, load the first frame. Loop (loop) : Repeat the sequence before the first frame Bounce (bounce) : Repeat the sequence in reverse before the first frame Black (black) : Render a black image Error (error) : Report an error
On Missing Frame / onMissingFrame	Choice	Error	What to do when a frame is missing from the sequence/stream. Hold previous (previous) : Try to load the previous frame in the sequence/stream, if any. Load next (next) : Try to load the next frame in the sequence/stream, if any. Load nearest (nearest) : Try to load the nearest frame in the sequence/stream, if any. Error (error) : Report an error Black (black) : Render a black image

Continued on next page

Table 5 – continued from previous page

Parameter / script name	Type	Default	Function
Frame Mode / <code>frameMode</code>	Choice	Starting Time	<p>Starting Time (<code>startingTime</code>): Set at what output frame the first sequence frame is output. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>timeOffset</code>.</p> <p>Time Offset (<code>timeOffset</code>): Set an offset to be applied as a number of frames. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>firstFrame+timeOffset</code>.</p>
Starting Time / <code>startingTime</code>	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / <code>timeOffset</code>	Integer	0	Offset applied to the sequence in time units (i.e. frames).
Proxy File / <code>proxy</code>	N/A		Filename of the proxy images. They will be used instead of the images read from the File parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / <code>proxyThreshold</code>	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / <code>customProxyScale</code>	Boolean	Off	Check to enable the Proxy scale edition.

Continued on next page

Table 5 – continued from previous page

Parameter / script name	Type	Default	Function
File Premult / filePremult	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / outputPremult	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Components / outputComponents	Choice	RGBA	<p>What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these.</p> <p>RGBA</p>
Frame rate / frameRate	Double	24	<p>By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.</p>
Custom FPS / customFps	Boolean	Off	<p>If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.</p>
OCIO Config File / ocioConfigFile	N/A		<p>OpenColorIO configuration file</p>

Continued on next page

Table 5 – continued from previous page

Parameter / script name	Type	Default	Function
File Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Output Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 5 – continued from previous page

Parameter / script name	Type	Default	Function
value3 / value3	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
key4 / key4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
value4 / value4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.6 Read node

This documentation is for version 1.0 of Read.

Description

Node used to read images or videos from disk. The image/video is identified by its filename and its extension. Given the extension, the Reader selected from the Preferences to decode that specific format will be used.

Inputs

Input	Description	Optional
Sync		Yes

Controls

Parameter / script name	Type	Default	Function
File Info... / <code>fileInfo</code>	Button		Press to display informations about the file
Decoder / <code>decodingPluginChoice</code>	Choice	Default	Select the internal decoder plug-in used for this file format. By default this uses the plug-in selected for this file extension in the Preferences of Natron Default: Use the default plug-in chosen from the Preferences to read this file format
File / <code>filename</code>	N/A		The input image sequence/video stream file(s).
First Frame / <code>firstFrame</code>	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If <code>startTime</code> is 1 or <code>timeOffset</code> is 0, this is also the first output frame.
Before / <code>before</code>	Choice	Hold	What to do before the first frame of the sequence. Hold (hold): While before the sequence, load the first frame. Loop (loop): Repeat the sequence before the first frame Bounce (bounce): Repeat the sequence in reverse before the first frame Black (black): Render a black image Error (error): Report an error
Last Frame / <code>lastFrame</code>	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If <code>startTime</code> is 1 or <code>timeOffset</code> is 0, this is also the last output frame.
After / <code>after</code>	Choice	Hold	What to do after the last frame of the sequence. Hold (hold): While before the sequence, load the first frame. Loop (loop): Repeat the sequence before the first frame Bounce (bounce): Repeat the sequence in reverse before the first frame Black (black): Render a black image Error (error): Report an error
On Missing Frame / <code>onMissingFrame</code>	Choice	Error	What to do when a frame is missing from the sequence/stream. Hold previous (previous): Try to load the previous frame in the sequence/stream, if any. Load next (next): Try to load the next frame in the sequence/stream, if any. Load nearest (nearest): Try to load the nearest frame in the sequence/stream, if any. Error (error): Report an error Black (black): Render a black image

Continued on next page

Table 6 – continued from previous page

Parameter / script name	Type	Default	Function
Frame Mode / <code>frameMode</code>	Choice	Starting Time	<p>Starting Time (<code>startingTime</code>): Set at what output frame the first sequence frame is output. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>timeOffset</code>.</p> <p>Time Offset (<code>timeOffset</code>): Set an offset to be applied as a number of frames. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>firstFrame+timeOffset</code>.</p>
Starting Time / <code>startingTime</code>	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / <code>timeOffset</code>	Integer	0	Offset applied to the sequence in time units (i.e. frames).
Proxy File / <code>proxy</code>	N/A		Filename of the proxy images. They will be used instead of the images read from the File parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / <code>proxyThreshold</code>	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / <code>customProxyScale</code>	Boolean	Off	Check to enable the Proxy scale edition.

Continued on next page

Table 6 – continued from previous page

Parameter / script name	Type	Default	Function
File Premult / filePremult	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / outputPremult	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Components / outputComponents	Choice	RGBA	<p>What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these.</p> <p>RGBA RGB RG Alpha</p>
Frame rate / frameRate	Double	24	<p>By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.</p>

Continued on next page

Table 6 – continued from previous page

Parameter / script name	Type	Default	Function
Custom FPS / customFps	Boolean	Off	If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
File Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Output Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.7 ReadCDR node



This documentation is for version 1.0 of ReadCDR.

Description

Read CorelDRAW(R) document format.

This plugin is not manufactured, approved, or supported by Corel Corporation or Corel Corporation Limited.

Inputs

Input	Description	Optional
Sync	Sync	Yes

Controls

Parameter / script name	Type	Default	Function
File / filename	N/A		The input image sequence/video stream file(s).
First Frame / firstFrame	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the first output frame.
Before / before	Choice	Hold	<p>What to do before the first frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>

Continued on next page

Table 7 – continued from previous page

Parameter / script name	Type	Default	Function
Last Frame / <code>lastFrame</code>	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If <code>startingTime</code> is 1 or <code>timeOffset</code> is 0, this is also the last output frame.
After / <code>after</code>	Choice	Hold	What to do after the last frame of the sequence. Hold (hold) : While before the sequence, load the first frame. Loop (loop) : Repeat the sequence before the first frame Bounce (bounce) : Repeat the sequence in reverse before the first frame Black (black) : Render a black image Error (error) : Report an error
On Missing Frame / <code>onMissingFrame</code>	Choice	Error	What to do when a frame is missing from the sequence/stream. Hold previous (previous) : Try to load the previous frame in the sequence/stream, if any. Load next (next) : Try to load the next frame in the sequence/stream, if any. Load nearest (nearest) : Try to load the nearest frame in the sequence/stream, if any. Error (error) : Report an error Black (black) : Render a black image
Frame Mode / <code>frameMode</code>	Choice	Starting Time	Starting Time (startingTime) : Set at what output frame the first sequence frame is output. The sequence frame designated by the <code>firstFrame</code> parameter is output at <code>frameTimeOffset</code> . Time Offset (timeOffset) : Set an offset to be applied as a number of frames. The sequence frame designated by the <code>firstFrame</code> parameter is output at <code>frame firstFrame+timeOffset</code> .
Starting Time / <code>startingTime</code>	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / <code>timeOffset</code>	Integer	0	Offset applied to the sequence in time units (i.e. frames).
Proxy File / <code>proxy</code>	N/A		Filename of the proxy images. They will be used instead of the images read from the <code>File</code> parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / <code>proxyThreshold</code>	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / <code>customProxyScale</code>	Boolean	Off	Check to enable the Proxy scale edition.

Continued on next page

Table 7 – continued from previous page

Parameter / script name	Type	Default	Function
File Premult / filePremult	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / outputPremult	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Components / outputComponents	Choice	RGBA	<p>What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these.</p> <p>RGBA</p>
Frame rate / frameRate	Double	24	<p>By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.</p>
Custom FPS / customFps	Boolean	Off	<p>If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.</p>
DPI / dpi	Integer	90	<p>Dots-per-inch (90 is default)</p>

Continued on next page

Table 7 – continued from previous page

Parameter / script name	Type	Default	Function
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
File Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Output Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 7 – continued from previous page

Parameter / script name	Type	Default	Function
value3 / value3	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
key4 / key4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
value4 / value4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.8 ReadFFmpeg node



This documentation is for version 1.0 of ReadFFmpeg.

Description

Read video using FFmpeg.

All formats supported by FFmpeg should be supported, but there may be issues with some non-conform files. In this case, it is recommended to transcode the video to a digital intermediate format, which is more suitable for grading, compositing and video editing.

This can be done using the ffmpeg command-line tool, by following the instructions at <https://trac.ffmpeg.org/wiki/Encode/VFX>

Inputs

Input	Description	Optional
Sync	Sync	Yes

Controls

Parameter / script name	Type	Default	Function
File / filename	N/A		The input image sequence/video stream file(s).
First Frame / firstFrame	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the first output frame.
Before / before	Choice	Hold	<p>What to do before the first frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
Last Frame / lastFrame	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the last output frame.
After / after	Choice	Hold	<p>What to do after the last frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
On Missing Frame / onMissingFrame	Choice	Error	<p>What to do when a frame is missing from the sequence/stream.</p> <p>Hold previous (previous): Try to load the previous frame in the sequence/stream, if any.</p> <p>Load next (next): Try to load the next frame in the sequence/stream, if any.</p> <p>Load nearest (nearest): Try to load the nearest frame in the sequence/stream, if any.</p> <p>Error (error): Report an error</p> <p>Black (black): Render a black image</p>

Continued on next page

Table 8 – continued from previous page

Parameter / script name	Type	Default	Function
Frame Mode / <code>frameMode</code>	Choice	Starting Time	<p>Starting Time (<code>startingTime</code>): Set at what output frame the first sequence frame is output. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>timeOffset</code>.</p> <p>Time Offset (<code>timeOffset</code>): Set an offset to be applied as a number of frames. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>firstFrame+timeOffset</code>.</p>
Starting Time / <code>startingTime</code>	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / <code>timeOffset</code>	Integer	0	Offset applied to the sequence in time units (i.e. frames).
Proxy File / <code>proxy</code>	N/A		Filename of the proxy images. They will be used instead of the images read from the File parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / <code>proxyThreshold</code>	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / <code>customProxyScale</code>	Boolean	Off	Check to enable the Proxy scale edition.

Continued on next page

Table 8 – continued from previous page

Parameter / script name	Type	Default	Function
File Premult / filePremult	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / outputPremult	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Components / outputComponents	Choice	RGBA	<p>What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these.</p> <p>RGBA</p> <p>RGB</p>
Frame rate / frameRate	Double	24	<p>By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.</p>
Custom FPS / customFps	Boolean	Off	<p>If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.</p>

Continued on next page

Table 8 – continued from previous page

Parameter / script name	Type	Default	Function
Max retries per frame / maxRetries	Integer	10	Some video files are sometimes tricky to read and needs several re-tries before successfully decoding a frame. This parameter controls how many times we should attempt to decode the same frame before failing.
First Track Only / firstTrackOnly	Boolean	Off	Causes the reader to ignore all but the first video track it finds in the file. This should be selected in a multiview project if the file happens to contain multiple video tracks that don't correspond to different views.
FFmpeg Info... / libraryInfo	Button		Display information about the underlying library.
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
File Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Output Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 8 – continued from previous page

Parameter / script name	Type	Default	Function
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value3 / value3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key4 / key4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value4 / value4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.9 ReadKrita node



This documentation is for version 2.0 of ReadKrita.

Description

Read Krita image format.

Inputs

Input	Description	Optional
Sync	Sync	Yes

Controls

Parameter / script name	Type	Default	Function
File / filename	N/A		The input image sequence/video stream file(s).
First Frame / firstFrame	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the first output frame.
Before / before	Choice	Hold	<p>What to do before the first frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
Last Frame / lastFrame	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the last output frame.
After / after	Choice	Hold	<p>What to do after the last frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
On Missing Frame / onMissingFrame	Choice	Error	<p>What to do when a frame is missing from the sequence/stream.</p> <p>Hold previous (previous): Try to load the previous frame in the sequence/stream, if any.</p> <p>Load next (next): Try to load the next frame in the sequence/stream, if any.</p> <p>Load nearest (nearest): Try to load the nearest frame in the sequence/stream, if any.</p> <p>Error (error): Report an error</p> <p>Black (black): Render a black image</p>

Continued on next page

Table 9 – continued from previous page

Parameter / script name	Type	Default	Function
Frame Mode / <code>frameMode</code>	Choice	Starting Time	<p>Starting Time (<code>startingTime</code>): Set at what output frame the first sequence frame is output. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>timeOffset</code>.</p> <p>Time Offset (<code>timeOffset</code>): Set an offset to be applied as a number of frames. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>firstFrame+timeOffset</code>.</p>
Starting Time / <code>startingTime</code>	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / <code>timeOffset</code>	Integer	0	Offset applied to the sequence in time units (i.e. frames).
Proxy File / <code>proxy</code>	N/A		Filename of the proxy images. They will be used instead of the images read from the File parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / <code>proxyThreshold</code>	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / <code>customProxyScale</code>	Boolean	Off	Check to enable the Proxy scale edition.

Continued on next page

Table 9 – continued from previous page

Parameter / script name	Type	Default	Function
File Premult / filePremult	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / outputPremult	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Components / outputComponents	Choice	RGBA	<p>What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these.</p> <p>RGBA</p>
Frame rate / frameRate	Double	24	<p>By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.</p>
Custom FPS / customFps	Boolean	Off	<p>If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.</p>
OCIO Config File / ocioConfigFile	N/A		<p>OpenColorIO configuration file</p>

Continued on next page

Table 9 – continued from previous page

Parameter / script name	Type	Default	Function
File Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Output Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 9 – continued from previous page

Parameter / script name	Type	Default	Function
value3 / value3	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
key4 / key4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
value4 / value4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.10 ReadMisc node



This documentation is for version 1.1 of ReadMisc.

Description

Read Misc image format.

Inputs

Input	Description	Optional
Sync	Sync	Yes

Controls

Parameter / script name	Type	Default	Function
File / filename	N/A		The input image sequence/video stream file(s).
First Frame / firstFrame	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the first output frame.
Before / before	Choice	Hold	<p>What to do before the first frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
Last Frame / lastFrame	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the last output frame.
After / after	Choice	Hold	<p>What to do after the last frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
On Missing Frame / onMissingFrame	Choice	Error	<p>What to do when a frame is missing from the sequence/stream.</p> <p>Hold previous (previous): Try to load the previous frame in the sequence/stream, if any.</p> <p>Load next (next): Try to load the next frame in the sequence/stream, if any.</p> <p>Load nearest (nearest): Try to load the nearest frame in the sequence/stream, if any.</p> <p>Error (error): Report an error</p> <p>Black (black): Render a black image</p>
Frame Mode / frameMode	Choice	Starting Time	<p>Starting Time (startingTime): Set at what output frame the first sequence frame is output. The sequence frame designated by the firstFrame parameter is output at frame timeOffset.</p> <p>Time Offset (timeOffset): Set an offset to be applied as a number of frames. The sequence frame designated by the firstFrame parameter is output at frame firstFrame+timeOffset.</p>
Starting Time / startingTime	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / timeOffset	Integer	0	Offset applied to the sequence in time units (i.e. frames).

Continued on next page

Table 10 – continued from previous page

Parameter / script name	Type	Default	Function
Proxy File / <code>proxy</code>	N/A		Filename of the proxy images. They will be used instead of the images read from the File parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / <code>proxyThreshold</code>	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / <code>customProxyScale</code>	Boolean	Off	Check to enable the Proxy scale edition.
File Premult / <code>filePremult</code>	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / <code>outputPremult</code>	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>

Continued on next page

Table 10 – continued from previous page

Parameter / script name	Type	Default	Function
Output Components / <code>outputComponents</code>	Choice	RGBA	What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these. RGBA
Frame rate / <code>frameRate</code>	Double	24	By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.
Custom FPS / <code>customFps</code>	Boolean	Off	If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.
OCIO Config File / <code>ocioConfigFile</code>	N/A		OpenColorIO configuration file
File Colorspace / <code>ocioInputSpaceIndex</code>	Choice		Input data is taken to be in this colorspace.
Output Colorspace / <code>ocioOutputSpaceIndex</code>	Choice		Output data is taken to be in this colorspace.
key1 / <code>key1</code>	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
value1 / <code>value1</code>	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
key2 / <code>key2</code>	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html

Continued on next page

Table 10 – continued from previous page

Parameter / script name	Type	Default	Function
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value3 / value3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key4 / key4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value4 / value4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.11 ReadOIIO node



This documentation is for version 2.0 of ReadOIIO.

Description

Read images using OpenImageIO.

Output is always Premultiplied (alpha is associated).

The “Image Premult” parameter controls the file premultiplication state, and can be used to fix wrong file metadata (see the help for that parameter).

OpenImageIO supports reading/writing the following file formats:

BMP (*.bmp)

Cineon (*.cin)

Direct Draw Surface (*.dds)

DPX (*.dpx)

Field3D (*.f3d)

FITS (*.fits)

GIF (*.gif)

HDR/RGBE (*.hdr)

ICO (*.ico)

IFF (*.iff)

JPEG (*.jpg *.jpe *.jpeg *.jif *.jif * .jfi)

JPEG-2000 (*.jp2 *.j2k)

OpenEXR (*.exr)

PNG / Portable Network Graphics (*.png)

PNM / Netpbm (*.pbm *.pgm *.ppm *.pfm)

PSD (*.psd *.pdd *.psb)

Ptex (*.ptex)

RAW digital camera files (*.crw *.cr2 *.nef *.raf *.dng and others)

RLA (*.rla)

SGI (*.sgi *.rgb *.rgba *.bw *.int *.inta)

Softimage PIC (*.pic)

Targa (*.tga *.tpic)

TIFF (*.tif *.tiff *.tx *.env *.sm *.vsm)

Webp (*.webp)

Zfile (*.zfile)

Inputs

Input	Description	Optional
Sync	Sync	Yes

Controls

Parameter / script name	Type	Default	Function
File / filename	N/A		The input image sequence/video stream file(s).
First Frame / firstFrame	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the first output frame.
Before / before	Choice	Hold	<p>What to do before the first frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
Last Frame / lastFrame	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the last output frame.
After / after	Choice	Hold	<p>What to do after the last frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
On Missing Frame / onMissingFrame	Choice	Error	<p>What to do when a frame is missing from the sequence/stream.</p> <p>Hold previous (previous): Try to load the previous frame in the sequence/stream, if any.</p> <p>Load next (next): Try to load the next frame in the sequence/stream, if any.</p> <p>Load nearest (nearest): Try to load the nearest frame in the sequence/stream, if any.</p> <p>Error (error): Report an error</p> <p>Black (black): Render a black image</p>

Continued on next page

Table 11 – continued from previous page

Parameter / script name	Type	Default	Function
Frame Mode / <code>frameMode</code>	Choice	Starting Time	<p>Starting Time (<code>startingTime</code>): Set at what output frame the first sequence frame is output. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>timeOffset</code>.</p> <p>Time Offset (<code>timeOffset</code>): Set an offset to be applied as a number of frames. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>firstFrame+timeOffset</code>.</p>
Starting Time / <code>startingTime</code>	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / <code>timeOffset</code>	Integer	0	Offset applied to the sequence in time units (i.e. frames).
Proxy File / <code>proxy</code>	N/A		Filename of the proxy images. They will be used instead of the images read from the File parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / <code>proxyThreshold</code>	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / <code>customProxyScale</code>	Boolean	Off	Check to enable the Proxy scale edition.

Continued on next page

Table 11 – continued from previous page

Parameter / script name	Type	Default	Function
File Premult / filePremult	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / outputPremult	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Components / outputComponents	Choice	RGBA	<p>What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these.</p> <p>RGBA RGB RG Alpha</p>
Frame rate / frameRate	Double	24	<p>By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.</p>

Continued on next page

Table 11 – continued from previous page

Parameter / script name	Type	Default	Function
Custom FPS / <code>customFps</code>	Boolean	Off	If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.
Image Info... / <code>showMetadata</code>	Button		Shows information and metadata from the image at current time.
Auto Bright / <code>rawAutoBright</code>	Boolean	Off	If checked, use libraw's automatic increase of brightness by histogram (exposure correction).
Use Camera WB / <code>rawUseCameraWB</code>	Boolean	On	If checked, and if possible, use the white balance from the camera.
Adjust Maximum Thr. / <code>rawAdjustMaximumThr</code>	Double	0	<p>This parameters controls auto-adjusting of maximum value based on <code>channel_maximum[]</code> data, calculated from real frame data. If calculated maximum is greater than <code>adjust_maximum_thr*maximum</code>, than maximum is set to <code>calculated_maximum</code>.</p> <p>Default: 0. If you set this value above 0.99999, then default value will be used. If you set this value below 0.00001, then no maximum adjustment will be performed. A value of 0.75 is reasonable for still shots, but sequences should always use 0.</p> <p>Adjusting maximum should not damage any picture (esp. if you use default value) and is very useful for correcting channel overflow problems (magenta clouds on landscape shots, green-blue highlights for indoor shots).</p>
Max. value / <code>rawUserSat</code>	Integer	0	The camera sensor saturation (maximum) value. Raw values greater or equal to this are considered saturated and are processed using the algorithm specified by the <code>rawHighlightMode</code> parameter. 0 means to use the default value.
Output Colorspace / <code>rawOutputColor</code>	Choice	sRGB	<p>Output colorspace.</p> <p>Raw (raw): Raw data</p> <p>sRGB (srgb): sRGB</p> <p>Adobe (adobergb): Adobe RGB (1998)</p> <p>Wide (wide): Wide-gamut RGB color space (or Adobe Wide Gamut RGB)</p> <p>ProPhoto (prophoto): Kodak ProPhoto RGB (or ROMM RGB)</p> <p>XYZ (xyz): CIE XYZ</p> <p>ACES (aces): AMPAS ACES</p>
Camera Matrix / <code>rawUseCameraMatrix</code>	Choice	Default	<p>Use/don't use an embedded color matrix.</p> <p>None (none): Do not use the embedded color matrix.</p> <p>Default (default): Use embedded color profile (if present) for DNG files (always); for other files only if <code>rawUseCameraWb</code> is set.</p> <p>Force (force): Use embedded color data (if present) regardless of white balance setting.</p>

Continued on next page

Table 11 – continued from previous page

Parameter / script name	Type	Default	Function
Highlight Mode / <code>rawHighlightMode</code>	Choice	Clip	<p>Algorithm for restoring highlight clippings. Highlights are part of your images that are burned due to the inability of your camera to capture the highlights. Highlight recovery is applied after white balance and demosaic.</p> <p>Clip (clip): Clip all highlights to white.</p> <p>Unclip (unclip): Leave highlights unclipped in various shades of pink.</p> <p>Blend (blend): Blend clipped and unclipped values for a gradual fade to white.</p> <p>Rebuild (rebuild): Reconstruct highlights with various levels of aggressiveness.</p>
Rebuild Level / <code>rawHighlightRebuildLevel</code>	Integer	2	<p>Level of aggressiveness used to rebuild highlights. <code>rawHighlightRebuildLevel=2</code> (which corresponds to -H 5 in LibRaw/dcrw) is a good compromise. If that's not good enough, use <code>rawHighlightRebuildLevel=6</code>, cut out the non-white highlights, and paste them into an image generated with <code>rawHighlightRebuildLevel=0</code>.</p>
Exposure / <code>rawExposure</code>	Double	1	<p>Amount of exposure correction before de-mosaicing, from 0.25 (2-stop darken) to 8 (3-stop brighten). (Default: 1., meaning no correction.)</p>
Demosaic / <code>rawDemosaic</code>	Choice	AHD	<p>Force a demosaicing algorithm. Will fall back on AHD if the demosaicing algorithm is not available due to licence restrictions (AHD-Mod, AFD, VCD, Mixed, LMMSE are GPL2, AMaZE is GPL3).</p> <p>None (none): No demosaicing.</p> <p>Linear (linear): Linear interpolation.</p> <p>VNG (vng): VNG interpolation.</p> <p>PPG (ppg): PPG interpolation.</p> <p>AHD (ahd): AHD interpolation.</p> <p>DCB (dcb): DCB interpolation.</p> <p>AHD-Mod (ahdmod): Modified AHD interpolation by Paul Lee.</p> <p>AFD (afd): AFD interpolation (5-pass).</p> <p>VCD (vcd): VCD interpolation.</p> <p>Mixed (mixed): Mixed VCD/Modified AHD interpolation.</p> <p>LMMSE (lmmse): LMMSE interpolation.</p> <p>DHT (dht): DHT interpolation.</p> <p>AAHD (aahd): Modified AHD interpolation by Anton Petrushevich.</p>
Output Layer / <code>outputLayer</code>	Choice		<p>This is the layer that will be set to the the color plane. This is relevant only for image formats that can have multiple layers: exr, tiff, psd, etc... Note that in Natron you can access other layers with a Shuffle node downstream of this node.</p>

Continued on next page

Table 11 – continued from previous page

Parameter / script name	Type	Default	Function
Edge Pixels / <code>edgePixels</code>	Choice	Auto	<p>Specifies how pixels in the border of the region of definition are handled</p> <p>Auto (auto): If the region of definition and format match exactly then repeat the border pixel otherwise use black</p> <p>Edge Detect (edge): For each edge, if the region of definition and format match exactly then repeat border pixel, otherwise use black</p> <p>Repeat (repeat): Repeat pixels outside the region of definition</p> <p>Black (black): Add black pixels outside the region of definition</p>
Offset Negative Display Window / <code>offsetNegativeDispWindow</code>	Boolean	On	The EXR file format can have its “display window” origin at another location than (0,0). However in OpenFX, formats should have their origin at (0,0). If the left edge of the display window is not 0, either you can offset the display window so it goes to 0, or you can treat the negative portion as overscan and resize the format.
OpenImageIO Info... / <code>libraryInfo</code>	Button		Display information about the underlying library.
OCIO Config File / <code>ocioConfigFile</code>	N/A		OpenColorIO configuration file
File Colorspace / <code>ocioInputSpaceIndex</code>	Choice		Input data is taken to be in this colorspace.
Output Colorspace / <code>ocioOutputSpaceIndex</code>	Choice		Output data is taken to be in this colorspace.
<code>key1 / key1</code>	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
<code>value1 / value1</code>	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
<code>key2 / key2</code>	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 11 – continued from previous page

Parameter / script name	Type	Default	Function
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value3 / value3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key4 / key4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value4 / value4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.12 ReadPDF node



This documentation is for version 1.4 of ReadPDF.

Description

Read PDF documents using poppler.

Inputs

Input	Description	Optional
Sync	Sync	Yes

Controls

Parameter / script name	Type	Default	Function
File / filename	N/A		The input image sequence/video stream file(s).
First Frame / firstFrame	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the first output frame.
Before / before	Choice	Hold	What to do before the first frame of the sequence. Hold (hold) : While before the sequence, load the first frame. Loop (loop) : Repeat the sequence before the first frame Bounce (bounce) : Repeat the sequence in reverse before the first frame Black (black) : Render a black image Error (error) : Report an error
Last Frame / lastFrame	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the last output frame.
After / after	Choice	Hold	What to do after the last frame of the sequence. Hold (hold) : While before the sequence, load the first frame. Loop (loop) : Repeat the sequence before the first frame Bounce (bounce) : Repeat the sequence in reverse before the first frame Black (black) : Render a black image Error (error) : Report an error

Continued on next page

Table 12 – continued from previous page

Parameter / script name	Type	Default	Function
On Missing Frame / onMissingFrame	Choice	Error	<p>What to do when a frame is missing from the sequence/stream.</p> <p>Hold previous (previous): Try to load the previous frame in the sequence/stream, if any.</p> <p>Load next (next): Try to load the next frame in the sequence/stream, if any.</p> <p>Load nearest (nearest): Try to load the nearest frame in the sequence/stream, if any.</p> <p>Error (error): Report an error</p> <p>Black (black): Render a black image</p>
Frame Mode / frameMode	Choice	Starting Time	<p>Starting Time (startingTime): Set at what output frame the first sequence frame is output. The sequence frame designated by the firstFrame parameter is output at frame timeOffset.</p> <p>Time Offset (timeOffset): Set an offset to be applied as a number of frames. The sequence frame designated by the firstFrame parameter is output at frame firstFrame+timeOffset.</p>
Starting Time / startingTime	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / timeOffset	Integer	0	Offset applied to the sequence in time units (i.e. frames).
Proxy File / proxy	N/A		Filename of the proxy images. They will be used instead of the images read from the File parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / proxyThreshold	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / customProxyScale	Boolean	Off	Check to enable the Proxy scale edition.

Continued on next page

Table 12 – continued from previous page

Parameter / script name	Type	Default	Function
File Premult / filePremult	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / outputPremult	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Components / outputComponents	Choice	RGBA	<p>What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these.</p> <p>RGBA</p>
Frame rate / frameRate	Double	24	By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.
Custom FPS / customFps	Boolean	Off	If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.
DPI / dpi	Double	150	Dots-per-inch (150 is default)

Continued on next page

Table 12 – continued from previous page

Parameter / script name	Type	Default	Function
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
File Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Output Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 12 – continued from previous page

Parameter / script name	Type	Default	Function
value3 / value3	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
key4 / key4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
value4 / value4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.13 ReadPFM node



This documentation is for version 1.0 of ReadPFM.

Description

Read PFM (Portable Float Map) files.

Inputs

Input	Description	Optional
Sync	Sync	Yes

Controls

Parameter / script name	Type	Default	Function
File / filename	N/A		The input image sequence/video stream file(s).
First Frame / firstFrame	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the first output frame.
Before / before	Choice	Hold	<p>What to do before the first frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
Last Frame / lastFrame	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the last output frame.
After / after	Choice	Hold	<p>What to do after the last frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
On Missing Frame / onMissingFrame	Choice	Error	<p>What to do when a frame is missing from the sequence/stream.</p> <p>Hold previous (previous): Try to load the previous frame in the sequence/stream, if any.</p> <p>Load next (next): Try to load the next frame in the sequence/stream, if any.</p> <p>Load nearest (nearest): Try to load the nearest frame in the sequence/stream, if any.</p> <p>Error (error): Report an error</p> <p>Black (black): Render a black image</p>
Frame Mode / frameMode	Choice	Starting Time	<p>Starting Time (startingTime): Set at what output frame the first sequence frame is output. The sequence frame designated by the firstFrame parameter is output at frame timeOffset.</p> <p>Time Offset (timeOffset): Set an offset to be applied as a number of frames. The sequence frame designated by the firstFrame parameter is output at frame firstFrame+timeOffset.</p>
Starting Time / startingTime	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / timeOffset	Integer	0	Offset applied to the sequence in time units (i.e. frames).

Continued on next page

Table 13 – continued from previous page

Parameter / script name	Type	Default	Function
Proxy File / <code>proxy</code>	N/A		Filename of the proxy images. They will be used instead of the images read from the File parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / <code>proxyThreshold</code>	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / <code>customProxyScale</code>	Boolean	Off	Check to enable the Proxy scale edition.
File Premult / <code>filePremult</code>	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / <code>outputPremult</code>	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>

Continued on next page

Table 13 – continued from previous page

Parameter / script name	Type	Default	Function
Output Components / outputComponents	Choice	RGBA	<p>What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these.</p> <p>RGBA RGB Alpha</p>
Frame rate / frameRate	Double	24	By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.
Custom FPS / customFps	Boolean	Off	If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
File Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Output Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 13 – continued from previous page

Parameter / script name	Type	Default	Function
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value3 / value3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key4 / key4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value4 / value4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.14 ReadPNG node



This documentation is for version 1.0 of ReadPNG.

Description

Read PNG files.

Inputs

Input	Description	Optional
Sync	Sync	Yes

Controls

Parameter / script name	Type	Default	Function
File / filename	N/A		The input image sequence/video stream file(s).
First Frame / firstFrame	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the first output frame.
Before / before	Choice	Hold	What to do before the first frame of the sequence. Hold (hold) : While before the sequence, load the first frame. Loop (loop) : Repeat the sequence before the first frame Bounce (bounce) : Repeat the sequence in reverse before the first frame Black (black) : Render a black image Error (error) : Report an error
Last Frame / lastFrame	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the last output frame.
After / after	Choice	Hold	What to do after the last frame of the sequence. Hold (hold) : While before the sequence, load the first frame. Loop (loop) : Repeat the sequence before the first frame Bounce (bounce) : Repeat the sequence in reverse before the first frame Black (black) : Render a black image Error (error) : Report an error

Continued on next page

Table 14 – continued from previous page

Parameter / script name	Type	Default	Function
On Missing Frame / onMissingFrame	Choice	Error	<p>What to do when a frame is missing from the sequence/stream.</p> <p>Hold previous (previous): Try to load the previous frame in the sequence/stream, if any.</p> <p>Load next (next): Try to load the next frame in the sequence/stream, if any.</p> <p>Load nearest (nearest): Try to load the nearest frame in the sequence/stream, if any.</p> <p>Error (error): Report an error</p> <p>Black (black): Render a black image</p>
Frame Mode / frameMode	Choice	Starting Time	<p>Starting Time (startingTime): Set at what output frame the first sequence frame is output. The sequence frame designated by the firstFrame parameter is output at frame timeOffset.</p> <p>Time Offset (timeOffset): Set an offset to be applied as a number of frames. The sequence frame designated by the firstFrame parameter is output at frame firstFrame+timeOffset.</p>
Starting Time / startingTime	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / timeOffset	Integer	0	Offset applied to the sequence in time units (i.e. frames).
Proxy File / proxy	N/A		Filename of the proxy images. They will be used instead of the images read from the File parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / proxyThreshold	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / customProxyScale	Boolean	Off	Check to enable the Proxy scale edition.

Continued on next page

Table 14 – continued from previous page

Parameter / script name	Type	Default	Function
File Premult / filePremult	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / outputPremult	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Components / outputComponents	Choice	RGBA	<p>What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these.</p> <p>RGBA</p> <p>RGB</p>
Frame rate / frameRate	Double	24	<p>By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.</p>
Custom FPS / customFps	Boolean	Off	<p>If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.</p>

Continued on next page

Table 14 – continued from previous page

Parameter / script name	Type	Default	Function
Image Info... / showMetadata	Button		Shows information and metadata from the image at current time.
libpng Info... / libraryInfo	Button		Display information about the underlying library.
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
File Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Output Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 14 – continued from previous page

Parameter / script name	Type	Default	Function
key3 / key3	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
value3 / value3	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
key4 / key4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
value4 / value4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.15 ReadPSD node



This documentation is for version 2.7 of ReadPSD.

Description

Read Photoshop/GIMP/Cinepaint (RGB/CMYK/GRAY) image formats with ICC color management.

Inputs

Input	Description	Optional
Sync	Sync	Yes

Controls

Parameter / script name	Type	Default	Function
File / filename	N/A		The input image sequence/video stream file(s).
First Frame / firstFrame	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the first output frame.
Before / before	Choice	Hold	What to do before the first frame of the sequence. Hold (hold) : While before the sequence, load the first frame. Loop (loop) : Repeat the sequence before the first frame Bounce (bounce) : Repeat the sequence in reverse before the first frame Black (black) : Render a black image Error (error) : Report an error
Last Frame / lastFrame	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the last output frame.
After / after	Choice	Hold	What to do after the last frame of the sequence. Hold (hold) : While before the sequence, load the first frame. Loop (loop) : Repeat the sequence before the first frame Bounce (bounce) : Repeat the sequence in reverse before the first frame Black (black) : Render a black image Error (error) : Report an error
On Missing Frame / onMissingFrame	Choice	Error	What to do when a frame is missing from the sequence/stream. Hold previous (previous) : Try to load the previous frame in the sequence/stream, if any. Load next (next) : Try to load the next frame in the sequence/stream, if any. Load nearest (nearest) : Try to load the nearest frame in the sequence/stream, if any. Error (error) : Report an error Black (black) : Render a black image

Continued on next page

Table 15 – continued from previous page

Parameter / script name	Type	Default	Function
Frame Mode / <code>frameMode</code>	Choice	Starting Time	<p>Starting Time (<code>startingTime</code>): Set at what output frame the first sequence frame is output. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>timeOffset</code>.</p> <p>Time Offset (<code>timeOffset</code>): Set an offset to be applied as a number of frames. The sequence frame designated by the <code>firstFrame</code> parameter is output at frame <code>firstFrame+timeOffset</code>.</p>
Starting Time / <code>startingTime</code>	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / <code>timeOffset</code>	Integer	0	Offset applied to the sequence in time units (i.e. frames).
Proxy File / <code>proxy</code>	N/A		Filename of the proxy images. They will be used instead of the images read from the File parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / <code>proxyThreshold</code>	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / <code>customProxyScale</code>	Boolean	Off	Check to enable the Proxy scale edition.

Continued on next page

Table 15 – continued from previous page

Parameter / script name	Type	Default	Function
File Premult / filePremult	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / outputPremult	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Components / outputComponents	Choice	RGBA	<p>What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these.</p> <p>RGBA</p>
Frame rate / frameRate	Double	24	<p>By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.</p>
Custom FPS / customFps	Boolean	Off	<p>If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.</p>

Continued on next page

Table 15 – continued from previous page

Parameter / script name	Type	Default	Function
Image layer / <code>layer</code>	Choice	Default	<p>Select image layer</p> <p>The recommended way to access layers is through a merge/shuffle node (multi-plane).</p> <p>Default</p> <p>Layer 1</p> <p>Layer 2</p> <p>Layer 3</p> <p>Layer 4</p> <p>Layer 5</p> <p>Layer 6</p> <p>Layer 7</p> <p>Layer 8</p> <p>Layer 9</p>
Offset layers / <code>offset</code>	Boolean	On	Enable/Disable layer offset
Color management / <code>icc</code>	Boolean	Off	<p>Enable/Disable ICC color management</p> <p>Requires installed ICC v2/v4 color profiles.</p>

Continued on next page

Table 15 – continued from previous page

Parameter / script name	Type	Default	Function
Default RGB profile / iccRGB	Choice	s/sRGB IEC61966-2.1	-Default RGB profile Used when a RGB image is missing an embedded color profile. None W/Web Safe Colors P/ProPhoto RGB s/sRGB IEC61966-2.1 C/ColorMatch RGB A/Apple RGB A/Adobe RGB (1998) D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display S/Smokey H/HDTV (Rec. 709) C/ColorNegative C/CIE RGB N/NTSC (1953) R/RedBlueYellow S/SMPTE-C A/AnimePalette P/PAL/SECAM
116			<div>Chapter 2. Reference Guide</div> W/Wide Gamut RGB T/TealMagentaGold S/SDTV PAL S/SDTV NTSC

Table 15 – continued from previous page

Parameter / script name	Type	Default	Function
Default CMYK profile / <code>iccCMYK</code>	Choice	U/U.S. Web Coated (SWOP) v2	<p>Default CMYK profile</p> <p>Used when a CMYK image is missing an embedded color profile.</p> <p>None</p> <p>J/Japan Color 2002 Newspaper</p> <p>C/Coated GRACoL 2006 (ISO 12647-2:2004)</p> <p>W/Web Coated SWOP 2006 Grade 3 Paper</p> <p>U/U.S. Sheetfed Uncoated v2</p> <p>J/Japan Color 2001 Uncoated</p> <p>W/Web Coated SWOP 2006 Grade 5 Paper</p> <p>C/Coated FOGRA27 (ISO 12647-2:2004)</p> <p>U/US Newsprint (SNAP 2007)</p> <p>U/Uncoated FOGRA29 (ISO 12647-2:2004)</p> <p>U/U.S. Web Coated (SWOP) v2</p> <p>J/Japan Color 2001 Coated</p> <p>J/Japan Web Coated (Ad)</p> <p>W/Web Coated FOGRA28 (ISO 12647-2:2004)</p> <p>U/U.S. Web Uncoated v2</p> <p>C/Coated FOGRA39 (ISO 12647-2:2004)</p> <p>U/U.S. Sheetfed Coated v2</p> <p>J/Japan Color 2003 Web Coated</p> <p>E/Euroscale Coated v2</p> <p>E/Euroscale Uncoated v2</p> <p>P/Photoshop 5 Default CMYK</p> <p>P/Photoshop 4 Default CMYK</p> <p>T/Total Ink Preview</p> <p>J/Japan Color 2002 Newspaper</p> <p>C/Coated GRACoL 2006 (ISO 12647-2:2004)</p> <p>W/Web Coated SWOP 2006 Grade 3 Paper</p> <p>U/U.S. Sheetfed Uncoated v2</p> <p>J/Japan Color 2001 Uncoated</p> <p>W/Web Coated SWOP 2006 Grade 5 Paper</p> <p>C/Coated FOGRA27 (ISO 12647-2:2004)</p> <p>U/US Newsprint (SNAP 2007)</p> <p>U/Uncoated FOGRA29 (ISO 12647-2:2004)</p> <p>U/U.S. Web Coated (SWOP) v2</p> <p>J/Japan Color 2001 Coated</p> <p>J/Japan Web Coated (Ad)</p> <p>W/Web Coated FOGRA28 (ISO 12647-2:2004)</p> <p>U/U.S. Web Uncoated v2</p> <p>C/Coated FOGRA39 (ISO 12647-2:2004)</p> <p>U/U.S. Sheetfed Coated v2</p> <p>J/Japan Color 2003 Web Coated</p>

Continued on next page

Table 15 – continued from previous page

Parameter / script name	Type	Default	Function
Default GRAY profile / <code>iccGRAY</code>	Choice	None	Default GRAY profile Used when a GRAY image is missing an embedded color profile. None B/Black & White
Rendering intent / <code>renderingIntent</code>	Choice	Perceptual	Rendering intent specifies the style of reproduction to be used. Undefined Saturation Perceptual Absolute Relative
Black point / <code>blackPoint</code>	Boolean	Off	Enable/Disable black point compensation

Continued on next page

[illegible]

Table 15 – continued from previous page

Parameter / script name	Type	Default	Function
Output color profile / iccOut	Choice	s/sRGB IEC61966-2.1	-ICC RGB output profile If image is CMYK/GRAY a colorspace convert will happen. None W/Web Safe Colors P/ProPhoto RGB s/sRGB IEC61966-2.1 C/ColorMatch RGB A/Apple RGB A/Adobe RGB (1998) D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display D/Display S/Smokey H/HDTV (Rec. 709) C/ColorNegative C/CIE RGB N/NTSC (1953) R/RedBlueYellow S/SMPTE-C A/AnimePalette P/PAL/SECAM
120			W/Wide Gamut RGB T/TealMagentaGold S/SDTV PAL S/SDTV NTSC

Chapter 2. Reference Guide

Table 15 – continued from previous page

Parameter / script name	Type	Default	Function
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
File Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Output Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 15 – continued from previous page

Parameter / script name	Type	Default	Function
value3 / value3	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
key4 / key4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
value4 / value4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.16 ReadSVG node



This documentation is for version 3.3 of ReadSVG.

Description

Fast SVG (Scalable Vector Graphics) reader using libsvg and Cairo.

Inputs

Input	Description	Optional
Sync	Sync	Yes

Controls

Parameter / script name	Type	Default	Function
File / filename	N/A		The input image sequence/video stream file(s).
First Frame / firstFrame	Integer	0	The first frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the first output frame.
Before / before	Choice	Hold	<p>What to do before the first frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
Last Frame / lastFrame	Integer	0	The last frame number to read from this image sequence or video file. This cannot be less than the first frame of the image sequence or video file, and cannot be greater than the last frame of the image sequence or video file. The first frame of a video file is numbered 1. If startingTime is 1 or timeOffset is 0, this is also the last output frame.
After / after	Choice	Hold	<p>What to do after the last frame of the sequence.</p> <p>Hold (hold): While before the sequence, load the first frame.</p> <p>Loop (loop): Repeat the sequence before the first frame</p> <p>Bounce (bounce): Repeat the sequence in reverse before the first frame</p> <p>Black (black): Render a black image</p> <p>Error (error): Report an error</p>
On Missing Frame / onMissingFrame	Choice	Error	<p>What to do when a frame is missing from the sequence/stream.</p> <p>Hold previous (previous): Try to load the previous frame in the sequence/stream, if any.</p> <p>Load next (next): Try to load the next frame in the sequence/stream, if any.</p> <p>Load nearest (nearest): Try to load the nearest frame in the sequence/stream, if any.</p> <p>Error (error): Report an error</p> <p>Black (black): Render a black image</p>
Frame Mode / frameMode	Choice	Starting Time	<p>Starting Time (startingTime): Set at what output frame the first sequence frame is output. The sequence frame designated by the firstFrame parameter is output at frame timeOffset.</p> <p>Time Offset (timeOffset): Set an offset to be applied as a number of frames. The sequence frame designated by the firstFrame parameter is output at frame firstFrame+timeOffset.</p>
Starting Time / startingTime	Integer	0	At what time (on the timeline) should this sequence/video start.
Time Offset / timeOffset	Integer	0	Offset applied to the sequence in time units (i.e. frames).

Continued on next page

Table 16 – continued from previous page

Parameter / script name	Type	Default	Function
Proxy File / <code>proxy</code>	N/A		Filename of the proxy images. They will be used instead of the images read from the File parameter when the proxy mode (downscaling of the images) is activated.
Proxy threshold / <code>proxyThreshold</code>	Double	x: 1 y: 1	The scale of the proxy images. By default it will be automatically computed out of the images headers when you set the proxy file(s) path. When the render scale (proxy) is set to a scale lower or equal to this value then the proxy image files will be used instead of the original images. You can change this parameter by checking the “Custom scale” checkbox so that you can change the scale at which the proxy images should be used instead of the original images.
Custom Proxy Scale / <code>customProxyScale</code>	Boolean	Off	Check to enable the Proxy scale edition.
File Premult / <code>filePremult</code>	Choice	PreMultiplied	<p>The image file being read is considered to have this premultiplication state.</p> <p>To get UnPremultiplied (or “unassociated alpha”) images, set the “Output Premult” parameter to Unpremultiplied.</p> <p>By default the value should be correctly be guessed by the image file, but this parameter can be edited if the metadatas inside the file are wrong.</p> <ul style="list-style-type: none"> - Opaque means that the alpha channel is considered to be 1 (one), and it is not taken into account in colorspace conversion. - Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion, and re-multiplied by alpha after colorspace conversion. - UnPremultiplied, means that red, green and blue channels are not modified before applying the colorspace conversion, and are multiplied by alpha after colorspace conversion. <p>This is set automatically from the image file and the plugin, but can be adjusted if this information is wrong in the file metadata.</p> <p>RGB images can only be Opaque, and Alpha images can only be Premultiplied (the value of this parameter doesn’t matter).</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Output Premult / <code>outputPremult</code>	Choice	PreMultiplied	<p>The alpha premultiplication in output of this node will have this state.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>

Continued on next page

Table 16 – continued from previous page

Parameter / script name	Type	Default	Function
Output Components / <code>outputComponents</code>	Choice	RGBA	What type of components this effect should output when the main color plane is requested. For the Read node it will map (in number of components) the Output Layer choice to these. RGBA
Frame rate / <code>frameRate</code>	Double	24	By default this value is guessed from the file. You can override it by checking the Custom fps parameter. The value of this parameter is what will be visible by the effects down-stream.
Custom FPS / <code>customFps</code>	Boolean	Off	If checked, you can freely force the value of the frame rate parameter. The frame-rate is just the meta-data that will be passed downstream to the graph, no retime will actually take place.
DPI / <code>dpi</code>	Integer	90	Dots-per-inch (90 is default)
OCIO Config File / <code>ocioConfigFile</code>	N/A		OpenColorIO configuration file
File Colorspace / <code>ocioInputSpaceIndex</code>	Choice		Input data is taken to be in this colorspace.
Output Colorspace / <code>ocioOutputSpaceIndex</code>	Choice		Output data is taken to be in this colorspace.
key1 / <code>key1</code>	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
value1 / <code>value1</code>	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
key2 / <code>key2</code>	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html

Continued on next page

Table 16 – continued from previous page

Parameter / script name	Type	Default	Function
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value3 / value3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key4 / key4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value4 / value4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

2.2.17 RunScript node



This documentation is for version 1.0 of RunScript.

Description

Run a script with the given arguments. This is mostly useful to execute an external program on a set of input images files, which outputs image files. Writers should be connected to each input, so that the image files are written before running the script, and the output of this node should be fed into one or more Readers, which read the images written by the script.

Sample section of a node graph which uses RunScript:

```

    ...
    ^
    |
Write([Project]/scriptinput####.png)
    ^
    |
RunScript1(processes [Project]/scriptinput####.png, output is [Project]/
↳scriptoutput####.png)
    ^
    |
Read([Project]/scriptoutput####.png, set the frame range manually)
    ^
    |
RunScript2(deletes temporary files [Project]/scriptinput####.png and [Project]/
↳scriptoutput####.png, optional)
    ^
    |
    ...

```

Keep in mind that the input and output files are never removed in the above graph. The output of RunScript is a copy of its first input.

Each argument may be:

- A filename (RunScript1 and RunScript2 in the example above should have [Project]/scriptinput####.png and [Project]/scriptoutput####.png as filename parameters 1 and 2)
- A floating-point value (which can be linked to any plugin)
- An integer
- A string

Under Unix, the script should begin with a traditional shebang line, e.g. `#!/bin/sh` or `#!/usr/bin/env python`. The arguments can be accessed as usual from the script (in a Unix shell-script, argument 1 would be accessed as `$1` - use double quotes to avoid problems with spaces). For example, the script in RunScript2 in the above example would be:

```
#!/bin/sh
rm "$1" "$2"
```

This plugin uses pstream (<http://pstreams.sourceforge.net>), which is distributed under the Boost Software License, Version 1.0.

Inputs

Input	Description	Optional
1		Yes
2		Yes
3		Yes
4		Yes

Controls

Parameter / script name	Type	Default	Function
Number of Parameters / <code>paramCount</code>	Integer	0	
Type of Parameter 1 / <code>type1</code>	Choice	File Name	<p>File Name: . A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p> <p>String: A string (or sequence of characters).</p> <p>Floating Point: A floating point numerical value.</p> <p>Integer: An integer numerical value.</p>
File Name1 / <code>filename1</code>	N/A		<p>A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p>
String1 / <code>string1</code>	String		A string (or sequence of characters).
Floating Point1 / <code>double1</code>	Double	0	A floating point numerical value.
Integer1 / <code>integer1</code>	Integer	0	An integer numerical value.

Continued on next page

Table 17 – continued from previous page

Parameter / script name	Type	Default	Function
Type of Parameter 2 / type2	Choice	File Name	<p>File Name: . A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V). This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p> <p>String: A string (or sequence of characters).</p> <p>Floating Point: A floating point numerical value.</p> <p>Integer: An integer numerical value.</p>
File Name2 / filename2	N/A		<p>A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V). This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p>
String2 / string2	String		A string (or sequence of characters).
Floating Point2 / double2	Double	0	A floating point numerical value.
Integer2 / integer2	Integer	0	An integer numerical value.
Type of Parameter 3 / type3	Choice	File Name	<p>File Name: . A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V). This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p> <p>String: A string (or sequence of characters).</p> <p>Floating Point: A floating point numerical value.</p> <p>Integer: An integer numerical value.</p>
File Name3 / filename3	N/A		<p>A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V). This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p>
String3 / string3	String		A string (or sequence of characters).
Floating Point3 / double3	Double	0	A floating point numerical value.

Continued on next page

Table 17 – continued from previous page

Parameter / script name	Type	Default	Function
Integer3 / integer3	Integer	0	An integer numerical value.
Type of Parameter 4 / type4	Choice	File Name	<p>File Name: . A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p> <p>String: A string (or sequence of characters).</p> <p>Floating Point: A floating point numerical value.</p> <p>Integer: An integer numerical value.</p>
File Name4 / filename4	N/A		<p>A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p>
String4 / string4	String		A string (or sequence of characters).
Floating Point4 / double4	Double	0	A floating point numerical value.
Integer4 / integer4	Integer	0	An integer numerical value.
Type of Parameter 5 / type5	Choice	File Name	<p>File Name: . A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p> <p>String: A string (or sequence of characters).</p> <p>Floating Point: A floating point numerical value.</p> <p>Integer: An integer numerical value.</p>
File Name5 / filename5	N/A		<p>A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p>
String5 / string5	String		A string (or sequence of characters).

Continued on next page

Table 17 – continued from previous page

Parameter / script name	Type	Default	Function
Floating Point5 / double5	Double	0	A floating point numerical value.
Integer5 / integer5	Integer	0	An integer numerical value.
Type of Parameter 6 / type6	Choice	File Name	<p>File Name: . A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p> <p>String: A string (or sequence of characters).</p> <p>Floating Point: A floating point numerical value.</p> <p>Integer: An integer numerical value.</p>
File Name6 / filename6	N/A		<p>A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p>
String6 / string6	String		A string (or sequence of characters).
Floating Point6 / double6	Double	0	A floating point numerical value.
Integer6 / integer6	Integer	0	An integer numerical value.
Type of Parameter 7 / type7	Choice	File Name	<p>File Name: . A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p> <p>String: A string (or sequence of characters).</p> <p>Floating Point: A floating point numerical value.</p> <p>Integer: An integer numerical value.</p>
File Name7 / filename7	N/A		<p>A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p>

Continued on next page

Table 17 – continued from previous page

Parameter / script name	Type	Default	Function
String7 / string7	String		A string (or sequence of characters).
Floating Point7 / double7	Double	0	A floating point numerical value.
Integer7 / integer7	Integer	0	An integer numerical value.
Type of Parameter 8 / type8	Choice	File Name	<p>File Name: . A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p> <p>String: A string (or sequence of characters).</p> <p>Floating Point: A floating point numerical value.</p> <p>Integer: An integer numerical value.</p>
File Name8 / filename8	N/A		<p>A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p>
String8 / string8	String		A string (or sequence of characters).
Floating Point8 / double8	Double	0	A floating point numerical value.
Integer8 / integer8	Integer	0	An integer numerical value.
Type of Parameter 9 / type9	Choice	File Name	<p>File Name: . A constant or animated string containing a filename. If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p> <p>String: A string (or sequence of characters).</p> <p>Floating Point: A floating point numerical value.</p> <p>Integer: An integer numerical value.</p>

Continued on next page

Table 17 – continued from previous page

Parameter / script name	Type	Default	Function
File Name9 / filename9	N/A		<p>A constant or animated string containing a filename.</p> <p>If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p>
String9 / string9	String		A string (or sequence of characters).
Floating Point9 / double9	Double	0	A floating point numerical value.
Integer9 / integer9	Integer	0	An integer numerical value.
Type of Parameter 10 / type10	Choice	File Name	<p>File Name: . A constant or animated string containing a filename.</p> <p>If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p> <p>String: A string (or sequence of characters).</p> <p>Floating Point: A floating point numerical value.</p> <p>Integer: An integer numerical value.</p>
File Name10 / filename10	N/A		<p>A constant or animated string containing a filename.</p> <p>If the string contains hashes (like #####) or a printf token (like %04d), they will be replaced by the frame number, and if it contains %v or %V, it will be replaced by the view ID (“l” or “r” for %v, “left” or “right” for %V).</p> <p>This is usually linked to the output filename of an upstream Writer node, or to the input filename of a downstream Reader node.</p>
String10 / string10	String		A string (or sequence of characters).
Floating Point10 / double10	Double	0	A floating point numerical value.
Integer10 / integer10	Integer	0	An integer numerical value.
Script / script	String	#!/bin/sh	<p>Contents of the script. Under Unix, the script should begin with a traditional shebang line, e.g. ‘#!/bin/sh’ or ‘#!/usr/bin/env python’</p> <p>The arguments can be accessed as usual from the script (in a Unix shell-script, argument 1 would be accessed as “\$1” - use double quotes to avoid problems with spaces).</p>
Validate / validate	Boolean	Off	Validate the script contents and execute it on next render. This locks the script and all its parameters.

2.2.18 Solid node

This documentation is for version 1.0 of Solid.

Description

Generate an image with a constant opaque color.

Inputs

Input	Description	Optional
Source		Yes

Controls

Parameter / script name	Type	Default	Function
Extent / extent	Choice	Default	Extent (size and offset) of the output. Format (format): Use a pre-defined image format. Size (size): Use a specific extent (size and offset). Project (project): Use the project extent (size and offset). Default (default): Use the default extent (e.g. the source clip extent, if connected).
Center / recenter	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.
Reformat / reformat	Boolean	Off	Set the output format to the given extent, except if the Bottom Left or Size parameters is animated.
Format / NatronParamFormatChoice	Choice	HD 1920x1080	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)

Continued on next page

Table 18 – continued from previous page

Parameter / script name	Type	Default	Function
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Frame Range / frameRange	Integer	min: 1 max: 1	Time domain.
Output Components / outputComponents	Choice	RGB	Components in the output RGBA RGB XY Alpha
Color / color	Color	r: 0 g: 0 b: 0	Color to fill the image with.

2.2.19 Write node

This documentation is for version 1.0 of Write.

Description

Node used to write images or videos on disk. The image/video is identified by its filename and its extension. Given the extension, the Writer selected from the Preferences to encode that specific format will be used.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Frame Increment / frameIncr	Integer	1	The number of frames the timeline should step before rendering the new frame. If 1, all frames will be rendered, if 2 only 1 frame out of 2, etc. This number cannot be less than 1.
Read back file / readBack	Boolean	Off	When checked, the output of this node comes from reading the written file instead of the input node
Encoder / encodingPluginChoice	Choice	Default	Select the internal encoder plug-in used for this file format. By default this uses the plug-in selected for this file extension in the Preferences. Default: Use the default plug-in chosen from the Preferences to write this file format

Continued on next page

Table 19 – continued from previous page

Parameter / script name	Type	Default	Function
File / filename	N/A		The output image sequence/video stream file(s). The string must match the following format: path/sequenceName###.ext where the number of # (hashes) will define the number of digits to append to each file. For example path/mySequence###.jpg will be translated to path/mySequence000.jpg, path/mySequence001.jpg, etc. %d printf-like notation can also be used instead of the hashes, for example path/sequenceName%03d.ext will achieve the same than the example aforementioned. there will be at least 2 digits). The file name may not contain any # (hash) in which case it will be overridden everytime. Views can be specified using the “long” view notation %V or the “short” notation using %v.
Format Type / formatType	Choice	Project Format	Determines which rectangle of pixels will be written in output. Input Format (input): Renders the pixels included in the input format Project Format (project): Renders the pixels included in the project format Fixed Format (fixed): Renders the pixels included in the format indicated by the Format parameter.
Format / NatronParamFormatChoice	Choice	HD 1920x1080	The output format to render PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
Input Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
File Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.

Continued on next page

Table 19 – continued from previous page

Parameter / script name	Type	Default	Function
Input Premult / <code>inputPremult</code>	Choice	PreMultiplied	<p>Input is considered to have this premultiplication state.</p> <p>If it is Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion.</p> <p>This is set automatically from the input stream information, but can be adjusted if this information is wrong.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Clip Info... / <code>clipInfo</code>	Button		Display information about the inputs
Frame Range / <code>frameRange</code>	Choice	Project frame range	<p>What frame range should be rendered.</p> <p>Union of input ranges (union): The union of all inputs frame ranges will be rendered.</p> <p>Project frame range (project): The frame range delimited by the frame range of the project will be rendered.</p> <p>Manual (manual): The frame range will be the one defined by the first frame and last frame parameters.</p>
First Frame / <code>firstFrame</code>	Integer	0	
Last Frame / <code>lastFrame</code>	Integer	0	

2.2.20 WriteFFmpeg node



This documentation is for version 1.0 of WriteFFmpeg.

Description

Write a video sequence using FFmpeg.

This plugin can be used to produce either digital intermediates, i.e. videos with very high resolution and quality which can be read frame by frame for further processing, or highly compressed videos to distribute on the web. Note that this plug-in does not support audio, but audi can easily be added to the video using the ffmpeg command-line tool (see note below). In a VFX context, it is often preferable to save processed images as a sequence of individual frames (using WriteOIIO), if disk space and real-time playing are not an issue.

The preferred pixel coding (Pref. Pixel Coding) and bit depth (Pref. Bit Depth) can be selected. This is especially useful for codecs that propose multiple pixel formats (e.g. ffv1, ffvhuff, huffyuv, jpeg2000, mjpeg, mpeg2video, vc2, libopenjpeg, png, qtrle, targa, tiff, libschroedinger, libtheora, libvpx, libvpx-vp9, libx264, libx265).

The pixel format is selected from the available choices for the chosen codec using the following rules:

- First, try to find the format with the smallest BPP (bits per pixel) that fits into the preferences.
- Second, If no format fits, get the format that has a BPP equal or a bit higher than the one computed from the preferences.
- Last, if no such format is found, get the format that has the highest BPP.

The selected pixel coding, bit depth, and BPP are displayed in the Selected Pixel Coding, Bit Depth, and BPP parameters.

The recommended Codec/Container configurations for encoding digital intermediates are (see also <https://trac.ffmpeg.org/wiki/Encode/VFX>):

- ProRes inside QuickTime: all ProRes profiles are 10-bit and are intra-frame (each frame is encoded separately). Prores 4444 can also encode the alpha channel.
- Avid DNxHR inside QuickTime: the codec is intra-frame. DNxHR profiles are resolution-independent and are available with 8-bit or 10-bit depth. The alpha channel cannot be encoded.
- HEVC (hev1/libx265) inside Matroska, MP4, QuickTime or MPEG-TS and Output Quality set to Lossless or Perceptually Lossless. libx265 supports 8-bit, 10-bit and 12-bit depth (if libx265 was compiled with high bit depth support). Lossless may not be playable in real-time for high resolutions. Set the Encoding Speed to Ultra Fast for faster encoding but worse compression, or Very Slow for best compression.

To write videos intended for distribution (as media files or for streaming), the most popular codecs are mp4v (mpeg4 or libxvid), avc1 (libx264), H264 (libopenh264), hev1 (libx265), VP80 (libvpx) and VP90 (libvpx-vp9). The quality of mp4v may be set using the Global Quality parameter (between 1 and 31, 1 being the highest quality), and the quality of avc1, hev1, VP80 and VP90 may be set using the Output Quality parameter. More information can be found at <https://trac.ffmpeg.org/wiki#Encoding>

If the output video should be encoded with specific FFmpeg options, such as a given pixel format or encoding option, it is better to write the output as individual frames in an image format that has a sufficient bit depth, and to encode the set of individual frames to a video using the command-line ffmpeg tool.

The settings for the “Global Quality” and “Quality” parameters may have different meanings for different codecs. See <http://slhck.info/video/2017/02/24/vbr-settings.html> for a summary of recommended values. Using these settings should be preferred over constant bitrate-based encoding, as it usually gives a much better result.

Adding audio

If synchronized audio is available as a separate file, encoded with the right codec, it can be easily added to the video using a command like: `ffmpeg -i input.mp4 -i input.mp3 -c copy -map 0:0 -map 1:0 output.mp4` (in this example, input.mp4 contains the video, input.mp3 contains the audio, and output.mp4 contains both tracks).

This command does not re-encode the video or audio, but simply copies the data from each source file and places it in separate streams in the output.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Output Components / outputComponents	Choice	RGBA	Map the input layer to this type of components before writing it to the output file. RGB RGBA
File / filename	N/A		The output image sequence/video stream file(s). The string must match the following format: path/sequenceName###.ext where the number of # (hashes) will define the number of digits to append to each file. For example path/mySequence###.jpg will be translated to path/mySequence000.jpg, path/mySequence001.jpg, etc. %d printf-like notation can also be used instead of the hashes, for example path/sequenceName%03d.ext will achieve the same than the example aforementioned. there will be at least 2 digits). The file name may not contain any # (hash) in which case it will be overridden everytime. Views can be specified using the “long” view notation %V or the “short” notation using %v.
Overwrite / overwrite	Boolean	On	Overwrite existing files when rendering.
Format Type / formatType	Choice	Project Format	Determines which rectangle of pixels will be written in output. Input Format (input): Renders the pixels included in the input format Project Format (project): Renders the pixels included in the project format Fixed Format (fixed): Renders the pixels included in the format indicated by the Format parameter.
Format / NatronParamFormatChoice	Choice	HD 1920x1080	The output format to render PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)

Continued on next page

Table 20 – continued from previous page

Parameter / script name	Type	Default	Function
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
Input Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
File Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 20 – continued from previous page

Parameter / script name	Type	Default	Function
value3 / value3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key4 / key4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value4 / value4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.
Input Premult / inputPremult	Choice	PreMultiplied	<p>Input is considered to have this premultiplication state.</p> <p>If it is Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion.</p> <p>This is set automatically from the input stream information, but can be adjusted if this information is wrong.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Clip Info... / clipInfo	Button		Display information about the inputs

Continued on next page

Table 20 – continued from previous page

Parameter / script name	Type	Default	Function
Frame Range / frameRange	Choice	Project frame range	<p>What frame range should be rendered.</p> <p>Union of input ranges (union): The union of all inputs frame ranges will be rendered.</p> <p>Project frame range (project): The frame range delimited by the frame range of the project will be rendered.</p> <p>Manual (manual): The frame range will be the one defined by the first frame and last frame parameters.</p>
First Frame / firstFrame	Integer	0	
Last Frame / lastFrame	Integer	0	
Container / format	Choice	guess from file-name	<p>Output format/container.</p> <p>guess from filename (default)</p> <p>AVI (Audio Video Interleaved) [avi] (avi): Compatible with ayuv, cinepak, ffv1, ffvhuff, flv, h263p, huffyuv, jpeg2000, jpegls, jpeg, mjpeg, mpeg2video, mpeg4, msmpeg4v2, msmpeg4, png, svq1, targa, v308, v408, v410, vc2, libopenjpeg, libtheora, libvpx, libvpx-vp9, libx264, libx264rgb, libxvid, libopenh264.</p> <p>FLV (Flash Video) [flv] (flv): Compatible with flv, mpeg4, libx264, libx264rgb, libxvid, libopenh264.</p> <p>GIF Animation [gif] (gif): Compatible with gif.</p> <p>Matroska [matroska] (matroska): Compatible with prores_ksap4h, prores_ksapch, prores_ksapcn, prores_ksapcs, prores_ksapco, ffv1, mjpeg, mpeg2video, mpeg4, msmpeg4, vc2, libtheora, libvpx, libvpx-vp9, libx264, libx264rgb, libx265, libxvid, libopenh264.</p> <p>QuickTime / MOV [mov] (mov): Compatible with prores_ksap4h, prores_ksapch, prores_ksapcn, prores_ksapcs, prores_ksapco, avr, cinepak, dnxhd, gif, hap, jpeg2000, mjpeg, mpeg2video, mpeg4, msmpeg4, png, qtrle, svq1, targa, tiff, v308, v408, v410, vc2, libopenjpeg, libvpx, libvpx-vp9, libx264, libx264rgb, libx265, libxvid, libopenh264.</p> <p>MP4 (MPEG-4 Part 14) [mp4] (mp4): Compatible with jpeg2000, mjpeg, mpeg2video, mpeg4, png, vc2, libopenjpeg, libvpx-vp9, libx264, libx264rgb, libx265, libxvid, libopenh264.</p> <p>MPEG-1 Systems / MPEG program stream [mpeg] (mpeg): Compatible with libx264, libx264rgb, libopenh264.</p> <p>MPEG-TS (MPEG-2 Transport Stream) [mpegts] (mpegts): Compatible with mpeg2video, mpeg4, vc2, libx264, libx264rgb, libx265, libxvid, libopenh264.</p> <p>Ogg Video [ogv] (ogv): Compatible with libtheora.</p> <p>3GP2 (3GPP2 file format) [3g2] (3g2): Compatible with mpeg4, libx264, libx264rgb, libxvid, libopenh264.</p> <p>3GP (3GPP file format) [3gp] (3gp): Compatible with mpeg4, libx264, libx264rgb, libxvid, libopenh264.</p>

Continued on next page

Table 20 – continued from previous page

Parameter / script name	Type	Default	Function
Codec / codec	Choice	ap4h Apple ProRes 4444	<p>Output codec used for encoding. The general recommendation is to write either separate frames (using WriteOIIO), or an uncompressed video format, or a “digital intermediate” format (ProRes, DNxHD), and to transcode the output and mux with audio with a separate tool (such as the ffmpeg or mencoder command-line tools).</p> <p>The FFmpeg encoder codec name is given between brackets at the end of each codec description.</p> <p>Please refer to the FFmpeg documentation http://ffmpeg.org/ffmpeg-codecs.html for codec options.</p> <p>ap4h Apple ProRes 4444 (prores_ksap4h): Compatible with matroska, mov.</p> <p>apch Apple ProRes 422 HQ (prores_ksapch): Compatible with matroska, mov.</p> <p>apcn Apple ProRes 422 (prores_ksapcn): Compatible with matroska, mov.</p> <p>apcs Apple ProRes 422 LT (prores_ksapcs): Compatible with matroska, mov.</p> <p>apco Apple ProRes 422 Proxy (prores_ksapco): Compatible with matroska, mov.</p> <p>AVrp Avid 1:1 10-bit RGB Packer [avrp] (avrp): Compatible with mov.</p> <p>AYUV Uncompressed packed MS 4:4:4:4 [ayuv] (ayuv): Compatible with avi.</p> <p>cvid Cinepak [cinepak] (cinepak): Compatible with avi, mov.</p> <p>AVdn Avid DNxHD / DNxHR / SMPTE VC-3 [dnxhd] (dnxhd): Compatible with mov.</p> <p>FFV1 FFmpeg video codec #1 [ffv1] (ffv1): Compatible with avi, matroska.</p> <p>FFVH Huffvuv FFmpeg variant [ffvhuff] (ffvhuff): Compatible with avi.</p> <p>FLV1 FLV / Sorenson Spark / Sorenson H.263 (Flash Video) [flv] (flv): Compatible with avi, flv.</p> <p>gif GIF (Graphics Interchange Format) [gif] (gif): Compatible with gif, mov.</p> <p>H263 H.263+ / H.263-1998 / H.263 version 2 [h263p] (h263p): Compatible with avi.</p> <p>Hap1 Vidvox Hap [hap] (hap): Compatible with mov.</p> <p>HFYU HuffYUV [huffyuv] (huffyuv): Compatible with avi.</p> <p>mjp2 JPEG 2000 [jpeg2000] (jpeg2000): Compatible with avi, mov, mp4.</p> <p>MJLS JPEG-LS [jpegls] (jpegls): Compatible with avi.</p> <p>LJPG Lossless JPEG [ljpeg] (ljpeg): Compatible with avi.</p> <p>jpeg Photo JPEG [mjpeg] (mjpeg): Compatible with avi, matroska, mov, mp4.</p> <p>m2v1 MPEG-2 Video [mpeg2video] (mpeg2video): Compatible with avi, matroska, mov, mp4, mpegts.</p> <p>mp4v MPEG-4 part 2 [mpeg4] (mpeg4): Compatible with avi, flv, matroska, mov, mp4, mpegts, 3g2, 3gp.</p> <p>MP42 MPEG-4 part 2 Microsoft variant version 2 [msmpeg4v2] (msmpeg4v2): Compatible with avi.</p> <p>3IVD MPEG-4 part 2 Microsoft variant version 3 [msmpeg4] (msmpeg4): Compatible with avi, matroska, mov.</p>
2.2. Image nodes			<p>png PNG (Portable Network Graphics) image [png] (png) 143 Compatible with avi, mov, mp4.</p> <p>rle QuickTime Animation (RLE) video [qtrle] (qtrle): Compatible with mov.</p>

Table 20 – continued from previous page

Parameter / script name	Type	Default	Function
Codec Name / <code>codecShortName</code>	String		The codec used when the writer was configured. If this parameter is visible, this means that this codec may not be supported by this version of the plugin.
FPS / <code>fps</code>	Double	24	File frame rate
Reset FPS / <code>resetFps</code>	Button		Reset FPS from the input FPS.
Pref. Pixel Coding / <code>prefPixelCoding</code>	Choice	YUV422	Preferred pixel coding. YUV420 (yuv420) : 1 Cr & Cb sample per 2x2 Y samples. YUV422 (yuv422) : 1 Cr & Cb sample per 2x1 Y samples. YUV444 (yuv444) : 1 Cr & Cb sample per Y sample. RGB (rgb) : Separate r, g, b. XYZ (xyz) : CIE XYZ compressed with gamma=2.6, used for Digital Cinema.
Bit Depth / <code>prefBitDepth</code>	Choice	8	Preferred bit depth (number of bits per component). 8 10 12 16
Alpha / <code>enableAlpha</code>	Boolean	Off	If checked, and the input contains alpha, formats with an alpha channel are preferred.
Show Avail. / <code>prefShow</code>	Button		Show available pixel codings for this codec.

Continued on next page

Table 20 – continued from previous page

Parameter / script name	Type	Default	Function
DNxHD Codec Profile / <code>DNxHDCodecProfile</code>	Choice	DNxHR 444	<p>Only for the Avid DNxHD codec, select the target bit rate for the encoded movie. The stream may be resized to 1920x1080 if resolution is not supported. Writing in thin-raster HDV format (1440x1080) is not supported by this plug-in, although FFmpeg supports it.</p> <p>DNxHR 444 (dnxhr444): DNxHR 4:4:4 (12 bit, RGB / 4:4:4, 4.5:1 compression)</p> <p>DNxHR HQX (dnxhrhqx): DNxHR High Quality (12 bit, 4:2:2 chroma sub-sampling, 5.5:1 compression)</p> <p>DNxHR HQ (dnxhrhq): DNxHR High Quality (8 bit, 4:2:2 chroma sub-sampling, 4.5:1 compression)</p> <p>DNxHR SQ (dnxhrsq): DNxHR Standard Quality (8 bit, 4:2:2 chroma sub-sampling, 7:1 compression)</p> <p>DNxHR LB (dnxhrlb): DNxHR Low Bandwidth (8 bit, 4:2:2 chroma sub-sampling, 22:1 compression)</p> <p>DNxHD 422 10-bit 440Mbit (dnxhd422_440x): 880x in 1080p/60 or 1080p/59.94, 730x in 1080p/50, 440x in 1080p/30, 390x in 1080p/25, 350x in 1080p/24</p> <p>DNxHD 422 10-bit 220Mbit (dnxhd422_220x): 440x in 1080p/60 or 1080p/59.94, 365x in 1080p/50, 220x in 1080i/60 or 1080i/59.94, 185x in 1080i/50 or 1080p/25, 175x in 1080p/24 or 1080p/23.976, 220x in 1080p/29.97, 220x in 720p/59.94, 175x in 720p/50</p> <p>DNxHD 422 8-bit 220Mbit (dnxhd422_220): 440 in 1080p/60 or 1080p/59.94, 365 in 1080p/50, 220 in 1080i/60 or 1080i/59.94, 185 in 1080i/50 or 1080p/25, 175 in 1080p/24 or 1080p/23.976, 220 in 1080p/29.97, 220 in 720p/59.94, 175 in 720p/50</p> <p>DNxHD 422 8-bit 145Mbit (dnxhd422_145): 290 in 1080p/60 or 1080p/59.94, 240 in 1080p/50, 145 in 1080i/60 or 1080i/59.94, 120 in 1080i/50 or 1080p/25, 115 in 1080p/24 or 1080p/23.976, 145 in 1080p/29.97, 145 in 720p/59.94, 115 in 720p/50</p> <p>DNxHD 422 8-bit 36Mbit (dnxhd422_36): 90 in 1080p/60 or 1080p/59.94, 75 in 1080p/50, 45 in 1080i/60 or 1080i/59.94, 36 in 1080i/50 or 1080p/25, 36 in 1080p/24 or 1080p/23.976, 45 in 1080p/29.97, 100 in 720p/59.94, 85 in 720p/50</p>
Hap Format / <code>HapFormat</code>	Choice	Hap 1	<p>Only for the Hap codec, select the target format.</p> <p>Hap 1 (hap): DXT1 textures (FourCC Hap1)</p> <p>Hap Alpha (hap_alpha): DXT5 textures (FourCC Hap5)</p> <p>Hap Q (hap_q): DXT5-YCoCg textures (FourCC HapY)</p>
Selected Pixel Coding / <code>infoPixelFormat</code>	String		Pixel coding of images passed to the encoder. If several pixel codings are available, the coding which causes less data loss is selected. Other pixel formats may be available by transcoding with ffmpeg on the command-line, as can be seen by executing ‘ffmpeg –help encoder=codec_name’ on the command-line.
Bit Depth / <code>infoBitDepth</code>	Integer	0	Bit depth (number of bits per component) of the pixel format.
BPP / <code>infoBpp</code>	Integer	0	Bits per pixel of the pixel format.

Continued on next page

Table 20 – continued from previous page

Parameter / script name	Type	Default	Function
DNxHD Output Range / DNxHDEncodeVideoRange	Choice	Video Range	<p>When encoding using DNxHD this is used to select between full scale data range and ‘video/legal’ data range.</p> <p>Full scale data range is 0-255 for 8-bit and 0-1023 for 10-bit.</p> <p>‘Video/legal’ data range is a reduced range, 16-240 for 8-bit and 64-960 for 10-bit.</p> <p>Full Range (full)</p> <p>Video Range (video)</p>
Output Quality / <i>crf</i>	Choice	Medium Quality	<p>Constant Rate Factor (CRF); tradeoff between video quality and file size. Used by avc1, hev1, VP80, VP9, and CAVS codecs.</p> <p>Option -crf in ffmpeg.</p> <p>None (none): Use constant bit-rate rather than constant output quality</p> <p>Lossless (crf0): Corresponds to CRF = 0.</p> <p>Perceptually Lossless (crf17): Corresponds to CRF = 17.</p> <p>High Quality (crf20): Corresponds to CRF = 20.</p> <p>Medium Quality (crf23): Corresponds to CRF = 23.</p> <p>Low Quality (crf26): Corresponds to CRF = 26.</p> <p>Very Low Quality (crf29): Corresponds to CRF = 29.</p>
Encoding Speed / x26xSpeed	Choice	Medium	<p>Trade off performance for compression efficiency. Available for avc1 and hev1.</p> <p>Option -preset in ffmpeg.</p> <p>Ultra Fast (ultrafast): Fast encoding, but larger file size.</p> <p>Very Fast (veryfast)</p> <p>Faster (faster)</p> <p>Fast (fast)</p> <p>Medium (medium)</p> <p>Slow (slow)</p> <p>Slower (slower)</p> <p>Very Slow (veryslow): Slow encoding, but smaller file size.</p>
Global Quality / qscale	Double	-1	<p>For lossy encoding, this controls image quality, from 0 to 100 (the lower, the better, 0 being near-lossless). For lossless encoding, this controls the effort and time spent at compressing more. -1 or negative value means to use the codec default or CBR (constant bit rate). Used for example by FLV1, mjp2, theo, jpeg, m2v1, mp4v MP42, 3IVD, codecs.</p> <p>Option -qscale in ffmpeg.</p>

Continued on next page

Table 20 – continued from previous page

Parameter / script name	Type	Default	Function
Quality / <code>quality</code>	Integer	min: -1 max: -1	The quality range the codec is allowed to vary the image data quantiser between to attempt to hit the desired bitrate. The lower, the better: higher values mean increased image degradation is possible, but with the upside of lower bit rates. Only supported by certain codecs (e.g. VP80, VP90, avc1, but not hev1 or mp4v). -1 means to use the codec default. Options -qmin and -qmax in ffmpeg.
Bitrate / <code>bitrateMbps</code>	Double	185	The target bitrate the codec will attempt to reach (in Megabits/s), within the confines of the bitrate tolerance and quality min/max settings. Only supported by certain codecs (e.g. hev1, m2v1, MP42, 3IVD, but not mp4v, avc1 or H264). Option -b in ffmpeg (multiplied by 1000000).
Bitrate Tolerance / <code>bitrateToleranceMbps</code>	Double	0	Set video bitrate tolerance (in Megabits/s). In 1-pass mode, bitrate tolerance specifies how far ratecontrol is willing to deviate from the target average bitrate value. This is not related to min/max bitrate. Lowering tolerance too much has an adverse effect on quality. As a guideline, the minimum slider range of target bitrate/target fps is the lowest advisable setting. Anything below this value may result in failed renders. Only supported by certain codecs (e.g. MP42, 3IVD, but not avc1, hev1, m2v1, mp4v or H264). A reasonable value is $5 * \text{bitrateMbps} / \text{fps}$. Option -bt in ffmpeg (multiplied by 1000000).
Keyframe Interval / <code>gopSize</code>	Integer	-1	The keyframe interval, also called GOP size, specifies how many frames may be grouped together by the codec to form a compression GOP. Exercise caution with this control as it may impact whether the resultant file can be opened in other packages. Only supported by certain codecs. -1 means to use the codec default if bFrames is not 0, or 1 if bFrames is 0 to ensure only intra (I) frames are produced, producing a video which is easier to scrub frame-by-frame. Option -g in ffmpeg.
Max B-Frames / <code>bFrames</code>	Integer	-1	Set max number of B frames between non-B-frames. Must be an integer between -1 and 16. 0 means that B-frames are disabled. If a value of -1 is used, it will choose an automatic value depending on the encoder. Influences file size and seekability. Only supported by certain codecs. -1 means to use the codec default if Keyframe Interval is not 1, or 0 if Keyframe Interval is 1 to ensure only intra (I) frames are produced, producing a video which is easier to scrub frame-by-frame. Option -bf in ffmpeg.

Continued on next page

Table 20 – continued from previous page

Parameter / script name	Type	Default	Function
Write NCLC / writeNCLC	Boolean	On	Write nclc data in the colr atom of the video header. QuickTime only.
FFmpeg Info... / libraryInfo	Button		Display information about the underlying library.

2.2.21 WriteOIIO node



This documentation is for version 1.0 of WriteOIIO.

Description

Write images using OpenImageIO.

OpenImageIO supports writing the following file formats:

BMP (*.bmp)

Cineon (*.cin)

DPX (*.dpx)

FITS (*.fits)

HDR/RGBE (*.hdr)

Icon (*.ico)

IFF (*.iff)

JPEG (*.jpg *.jpe *.jpeg *.jif *.jfif *.jfi)

JPEG-2000 (*.jp2 *.j2k)

OpenEXR (*.exr)

Portable Network Graphics (*.png)

PNM / Netpbm (*.pbm *.pgm *.ppm)

PSD (*.psd *.pdd *.psb)

RLA (*.rla)

SGI (*.sgi *.rgb *.rgba *.bw *.int *.inta)

Softimage PIC (*.pic)

Targa (*.tga *.tpic)

TIFF (*.tif *.tiff *.tx *.env *.sm *.vsm)

Zfile (*.zfile)

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Output Components / <code>outputComponents</code>	Choice	RGBA	Map the input layer to this type of components before writing it to the output file. Alpha RGB RGBA
File / <code>filename</code>	N/A		The output image sequence/video stream file(s). The string must match the following format: <code>path/sequenceName###.ext</code> where the number of # (hashes) will define the number of digits to append to each file. For example <code>path/mySequence###.jpg</code> will be translated to <code>path/mySequence000.jpg</code> , <code>path/mySequence001.jpg</code> , etc. <code>%d</code> printf-like notation can also be used instead of the hashes, for example <code>path/sequenceName%03d.ext</code> will achieve the same than the example aforementioned. there will be at least 2 digits). The file name may not contain any # (hash) in which case it will be overridden everytime. Views can be specified using the “long” view notation <code>%V</code> or the “short” notation using <code>%v</code> .
Overwrite / <code>overwrite</code>	Boolean	On	Overwrite existing files when rendering.
Format Type / <code>formatType</code>	Choice	Project Format	Determines which rectangle of pixels will be written in output. Input Format (input) : Renders the pixels included in the input format Project Format (project) : Renders the pixels included in the project format Fixed Format (fixed) : Renders the pixels included in the format indicated by the Format parameter.

Continued on next page

Table 21 – continued from previous page

Parameter / script name	Type	Default	Function
Format / NatronParamFormatChoice	Choice	HD 1920x1080	<p>The output format to render</p> <p>PC_Video 640x480 (PC_Video)</p> <p>NTSC 720x486 0.91 (NTSC)</p> <p>PAL 720x576 1.09 (PAL)</p> <p>NTSC_16:9 720x486 1.21 (NTSC_16:9)</p> <p>PAL_16:9 720x576 1.46 (PAL_16:9)</p> <p>HD_720 1280x720 (HD_720)</p> <p>HD 1920x1080 (HD)</p> <p>UHD_4K 3840x2160 (UHD_4K)</p> <p>1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap))</p> <p>1K_Cinemascope 914x778 2.00 (1K_Cinemascope)</p> <p>2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap))</p> <p>2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope)</p> <p>2K_DCP 2048x1080 (2K_DCP)</p> <p>4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap))</p> <p>4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope)</p> <p>4K_DCP 4096x2160 (4K_DCP)</p> <p>square_256 256x256 (square_256)</p> <p>square_512 512x512 (square_512)</p> <p>square_1K 1024x1024 (square_1K)</p> <p>square_2K 2048x2048 (square_2K)</p>
Clip To RoD / clipToRoD	Boolean	On	<p>When checked, the portion of the image written will be the region of definition of the image in input and not the format selected by the Output Format parameter.</p> <p>For the EXR file format, this will distinguish the data window (size of the image in input) from the display window (the format specified by Output Format).</p>
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
Input Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
File Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html </p>

Continued on next page

Table 21 – continued from previous page

Parameter / script name	Type	Default	Function
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value3 / value3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 21 – continued from previous page

Parameter / script name	Type	Default	Function
key4 / key4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value4 / value4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.
Input Premult / inputPremult	Choice	PreMultiplied	<p>Input is considered to have this premultiplication state.</p> <p>If it is Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion.</p> <p>This is set automatically from the input stream information, but can be adjusted if this information is wrong.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Clip Info... / clipInfo	Button		Display information about the inputs
Frame Range / frameRange	Choice	Project frame range	<p>What frame range should be rendered.</p> <p>Union of input ranges (union): The union of all inputs frame ranges will be rendered.</p> <p>Project frame range (project): The frame range delimited by the frame range of the project will be rendered.</p> <p>Manual (manual): The frame range will be the one defined by the first frame and last frame parameters.</p>
First Frame / firstFrame	Integer	0	
Last Frame / lastFrame	Integer	0	

Continued on next page

Table 21 – continued from previous page

Parameter / script name	Type	Default	Function
Tile Size / <code>tileSize</code>	Choice	Scan-Line Based	Size of a tile in the output file for formats that support tiles. If scan-line based, the whole image will have a single tile. Scan-Line Based (0) 64 128 256 512
Bit Depth / <code>bitDepth</code>	Choice	auto	Number of bits per sample in the file [TIFF,DPX,TGA,DDS,ICO,IFF,PNM,PIC]. auto : Guess from the output format 8i : 8 bits integer 10i : 10 bits integer 12i : 12 bits integer 16i : 16 bits integer 16f : 16 bits floating point 32i : 32 bits integer 32f : 32 bits floating point 64i : 64 bits integer 64f : 64 bits floating point
Quality / <code>quality</code>	Integer	100	Indicates the quality of compression to use (0-100), for those plugins and compression methods that allow a variable amount of compression, with higher numbers indicating higher image fidelity. [JPEG, TIFF w/ JPEG comp., WEBP]
DWA Compression Level / <code>dwaCompressionLevel</code>	Double	45	Amount of compression when using Dreamworks DWAA or DWAB compression options. These lossy formats are variable in quality and can minimize the compression artifacts. Higher values will result in greater compression and likewise smaller file size, but increases the chance for artifacts. Values from 45 to 150 are usually correct for production shots, whereas HDR vacation photos could use up to 500. Values below 45 should give no visible improvement on photographs. [EXR w/ DWAA or DWAB comp.]

Continued on next page

Table 21 – continued from previous page

Parameter / script name	Type	Default	Function
Orientation / orientation	Choice	normal	<p>The orientation of the image data [DPX,TIFF,JPEG,HDR,FITS]. By default, image pixels are ordered from the top of the display to the bottom, and within each scanline, from left to right (i.e., the same ordering as English text and scan progression on a CRT). But the “Orientation” parameter can suggest that it should be displayed with a different orientation, according to the TIFF/EXIF conventions.</p> <p>normal: normal (top to bottom, left to right)</p> <p>flip: flipped horizontally (top to bottom, right to left)</p> <p>180: rotate 180deg (bottom to top, right to left)</p> <p>flip: flipped vertically (bottom to top, left to right)</p> <p>transposed: transposed (left to right, top to bottom)</p> <p>90clockwise: rotated 90deg clockwise (right to left, top to bottom)</p> <p>transverse: transverse (right to left, bottom to top)</p> <p>90counter-clockwise: rotated 90deg counter-clockwise (left to right, bottom to top)</p>
Compression / compression	Choice	default	<p>Compression type [TIFF,EXR,DDS,IFF,SGI,TGA]</p> <p>Indicates the type of compression the file uses. Supported compression modes will vary from format to format. As an example, the TIFF format supports “none”, “lzw”, “ccittrle”, “zip” (the default), “jpeg”, “packbits”, and the EXR format supports “none”, “rle”, “zip” (the default), “piz”, “pxr24”, “b44”, “b44a”, “dwaa” or “dwab”.</p> <p>default: Guess from the output format</p> <p>none: No compression [EXR, TIFF, IFF]</p> <p>zip: Zlib/Deflate compression (lossless) [EXR, TIFF, Zfile]</p> <p>zips: Zlib compression (lossless), one scan line at a time [EXR]</p> <p>rle: Run Length Encoding (lossless) [DPX, IFF, EXR, TGA, RLA]</p> <p>piz: Piz-based wavelet compression [EXR]</p> <p>pxr24: Lossy 24bit float compression [EXR]</p> <p>b44: Lossy 4-by-4 pixel block compression, fixed compression rate [EXR]</p> <p>b44a: Lossy 4-by-4 pixel block compression, flat fields are compressed more [EXR]</p> <p>dwaa: lossy DCT based compression, in blocks of 32 scanlines. More efficient for partial buffer access. [EXR]</p> <p>dwab: lossy DCT based compression, in blocks of 256 scanlines. More efficient space wise and faster to decode full frames than DWAA. [EXR]</p> <p>lzw: Lempel-Ziv Welsch compression (lossless) [TIFF]</p> <p>ccittrle: CCITT modified Huffman RLE (lossless) [TIFF]</p> <p>jpeg: JPEG [TIFF]</p> <p>packbits: Macintosh RLE (lossless) [TIFF]</p>

Continued on next page

Table 21 – continued from previous page

Parameter / script name	Type	Default	Function
Layer(s) / outputChannels	Choice	Color.RGBA	Select which layer to write to the file. This is either All or a single layer. This is not yet possible to append a layer to an existing file. Color.RGBA (<code>uk.co.thefoundry.OfxImagePlaneColour</code>) DisparityLeft.Disparity (<code>uk.co.thefoundry.OfxImagePlaneStereoDisparityLeft</code>) DisparityRight.Disparity (<code>uk.co.thefoundry.OfxImagePlaneStereoDisparityRight</code>) Backward.Motion (<code>uk.co.thefoundry.OfxImagePlaneBackMotionVector</code>) Forward.Motion (<code>uk.co.thefoundry.OfxImagePlaneForwardMotionVector</code>)
All Planes / processAllPlanes	Boolean	Off	When checked all planes in input will be processed and output to the same plane as in input. It is useful for example to apply a Transform effect on all planes.
Parts / partSplitting	Choice	Split Views,Layers	Defines whether to separate views/layers in different EXR parts or not. Note that multi-part files are only supported by OpenEXR >= 2 Single Part (single) : All views and layers will be in the same part, ensuring compatibility with OpenEXR 1.x Split Views (views) : All views will have its own part, and each part will contain all layers. This will produce an EXR optimized in size that can be opened only with applications supporting OpenEXR 2 Split Views,Layers (views_layers) : Each layer of each view will have its own part. This will produce an EXR optimized for decoding speed that can be opened only with applications supporting OpenEXR 2
Views / viewsSelector	Choice	All	Select the views to render. When choosing All, make sure the output filename does not have a %v or %V view pattern in which case each view would be written to a separate file. All Main
OpenImageIO Info... /libraryInfo	Button		Display information about the underlying library.

2.2.22 WritePFM node



This documentation is for version 1.0 of WritePFM.

Description

Write PFM (Portable Float Map) files.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Output Components / <code>outputComponents</code>	Choice	RGBA	Map the input layer to this type of components before writing it to the output file. Alpha RGB RGBA
File / <code>filename</code>	N/A		The output image sequence/video stream file(s). The string must match the following format: <code>path/sequenceName###.ext</code> where the number of # (hashes) will define the number of digits to append to each file. For example <code>path/mySequence###.jpg</code> will be translated to <code>path/mySequence000.jpg</code> , <code>path/mySequence001.jpg</code> , etc. <code>%d</code> printf-like notation can also be used instead of the hashes, for example <code>path/sequenceName%03d.ext</code> will achieve the same than the example aforementioned. there will be at least 2 digits). The file name may not contain any # (hash) in which case it will be overridden everytimes. Views can be specified using the “long” view notation <code>%V</code> or the “short” notation using <code>%v</code> .
Overwrite / <code>overwrite</code>	Boolean	On	Overwrite existing files when rendering.
Format Type / <code>formatType</code>	Choice	Project Format	Determines which rectangle of pixels will be written in output. Input Format (input) : Renders the pixels included in the input format Project Format (project) : Renders the pixels included in the project format Fixed Format (fixed) : Renders the pixels included in the format indicated by the Format parameter.

Continued on next page

Table 22 – continued from previous page

Parameter / script name	Type	Default	Function
Format / NatronParamFormatChoice	Choice	HD 1920x1080	<p>The output format to render</p> <p>PC_Video 640x480 (PC_Video)</p> <p>NTSC 720x486 0.91 (NTSC)</p> <p>PAL 720x576 1.09 (PAL)</p> <p>NTSC_16:9 720x486 1.21 (NTSC_16:9)</p> <p>PAL_16:9 720x576 1.46 (PAL_16:9)</p> <p>HD_720 1280x720 (HD_720)</p> <p>HD 1920x1080 (HD)</p> <p>UHD_4K 3840x2160 (UHD_4K)</p> <p>1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap))</p> <p>1K_Cinemascope 914x778 2.00 (1K_Cinemascope)</p> <p>2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap))</p> <p>2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope)</p> <p>2K_DCP 2048x1080 (2K_DCP)</p> <p>4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap))</p> <p>4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope)</p> <p>4K_DCP 4096x2160 (4K_DCP)</p> <p>square_256 256x256 (square_256)</p> <p>square_512 512x512 (square_512)</p> <p>square_1K 1024x1024 (square_1K)</p> <p>square_2K 2048x2048 (square_2K)</p>
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
Input Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
File Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation:</p> <p>http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation:</p> <p>http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 22 – continued from previous page

Parameter / script name	Type	Default	Function
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value3 / value3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key4 / key4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 22 – continued from previous page

Parameter / script name	Type	Default	Function
value4 / value4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.
Input Premult / inputPremult	Choice	PreMultiplied	Input is considered to have this premultiplication state. If it is Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion. This is set automatically from the input stream information, but can be adjusted if this information is wrong. Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point. PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”). UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).
Clip Info... / clipInfo	Button		Display information about the inputs
Frame Range / frameRange	Choice	Project frame range	What frame range should be rendered. Union of input ranges (union): The union of all inputs frame ranges will be rendered. Project frame range (project): The frame range delimited by the frame range of the project will be rendered. Manual (manual): The frame range will be the one defined by the first frame and last frame parameters.
First Frame / firstFrame	Integer	0	
Last Frame / lastFrame	Integer	0	

2.2.23 WritePNG node



This documentation is for version 1.0 of WritePNG.

Description

Write PNG files.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Output Components / outputComponents	Choice	RGBA	Map the input layer to this type of components before writing it to the output file. RGB RGBA
File / filename	N/A		The output image sequence/video stream file(s). The string must match the following format: path/sequenceName###.ext where the number of # (hashes) will define the number of digits to append to each file. For example path/mySequence###.jpg will be translated to path/mySequence000.jpg, path/mySequence001.jpg, etc. %d printf-like notation can also be used instead of the hashes, for example path/sequenceName%03d.ext will achieve the same than the example aforementioned. there will be at least 2 digits). The file name may not contain any # (hash) in which case it will be overridden everytimes. Views can be specified using the “long” view notation %V or the “short” notation using %v.
Overwrite / overwrite	Boolean	On	Overwrite existing files when rendering.
Format Type / formatType	Choice	Project Format	Determines which rectangle of pixels will be written in output. Input Format (input) : Renders the pixels included in the input format Project Format (project) : Renders the pixels included in the project format Fixed Format (fixed) : Renders the pixels included in the format indicated by the Format parameter.

Continued on next page

Table 23 – continued from previous page

Parameter / script name	Type	Default	Function
Format / NatronParamFormatChoice	Choice	HD 1920x1080	<p>The output format to render</p> <p>PC_Video 640x480 (PC_Video)</p> <p>NTSC 720x486 0.91 (NTSC)</p> <p>PAL 720x576 1.09 (PAL)</p> <p>NTSC_16:9 720x486 1.21 (NTSC_16:9)</p> <p>PAL_16:9 720x576 1.46 (PAL_16:9)</p> <p>HD_720 1280x720 (HD_720)</p> <p>HD 1920x1080 (HD)</p> <p>UHD_4K 3840x2160 (UHD_4K)</p> <p>1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap))</p> <p>1K_Cinemascope 914x778 2.00 (1K_Cinemascope)</p> <p>2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap))</p> <p>2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope)</p> <p>2K_DCP 2048x1080 (2K_DCP)</p> <p>4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap))</p> <p>4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope)</p> <p>4K_DCP 4096x2160 (4K_DCP)</p> <p>square_256 256x256 (square_256)</p> <p>square_512 512x512 (square_512)</p> <p>square_1K 1024x1024 (square_1K)</p> <p>square_2K 2048x2048 (square_2K)</p>
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
Input Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
File Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation:</p> <p>http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation:</p> <p>http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 23 – continued from previous page

Parameter / script name	Type	Default	Function
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value3 / value3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key4 / key4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 23 – continued from previous page

Parameter / script name	Type	Default	Function
value4 / value4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.
Input Premult / inputPremult	Choice	PreMultiplied	<p>Input is considered to have this premultiplication state.</p> <p>If it is Premultiplied, red, green and blue channels are divided by the alpha channel before applying the colorspace conversion.</p> <p>This is set automatically from the input stream information, but can be adjusted if this information is wrong.</p> <p>Opaque (opaque): The image is opaque and so has no premultiplication state, as if the alpha component in all pixels were set to the white point.</p> <p>PreMultiplied (premult): The image is premultiplied by its alpha (also called “associated alpha”).</p> <p>UnPreMultiplied (unpremult): The image is unpremultiplied (also called “unassociated alpha”).</p>
Clip Info... / clipInfo	Button		Display information about the inputs
Frame Range / frameRange	Choice	Project frame range	<p>What frame range should be rendered.</p> <p>Union of input ranges (union): The union of all inputs frame ranges will be rendered.</p> <p>Project frame range (project): The frame range delimited by the frame range of the project will be rendered.</p> <p>Manual (manual): The frame range will be the one defined by the first frame and last frame parameters.</p>
First Frame / firstFrame	Integer	0	
Last Frame / lastFrame	Integer	0	

Continued on next page

Table 23 – continued from previous page

Parameter / script name	Type	Default	Function
Compression / compression	Choice	Default	<p>Compression used by the internal zlib library when encoding the file. This parameter is used to tune the compression algorithm.</p> <p>Filtered data consists mostly of small values with a somewhat random distribution. In this case, the compression algorithm is tuned to compress them better. The effect of Filtered is to force more Huffman coding and less string matching; it is somewhat intermediate between Default and Huffman Only. RLE is designed to be almost as fast as Huffman Only, but give better compression for PNG image data. The strategy parameter only affects the compression ratio but not the correctness of the compressed output even if it is not set appropriately. Fixed prevents the use of dynamic Huffman codes, allowing for a simpler decoder for special applications.</p> <p>Default (default): Use this for normal data</p> <p>Filtered (filtered): Use this for data produced by a filter (or predictor)</p> <p>Huffman Only (huffman): Forces Huffman encoding only (nostring match)</p> <p>RLE (rle): Limit match distances to one (run-length encoding)</p> <p>Fixed (fixed): Prevents the use of dynamic Huffman codes, allowing for a simpler decoder for special applications</p>
Compression Level / compressionLevel	Integer	6	<p>Between 0 and 9:</p> <p>1 gives best speed, 9 gives best compression, 0 gives no compression at all (the input data is simply copied a block at a time). Default compromise between speed and compression is 6.</p>
Depth / bitDepth	Choice	8-bit	<p>The depth of the internal PNG. Only 8bit and 16bit are supported by this writer</p> <p>8-bit (8u)</p> <p>16-bit (16u)</p>
Dithering / enableDithering	Boolean	On	<p>When checked, conversion from float input buffers to 8-bit PNG will use a dithering algorithm to reduce quantization artifacts. This has no effect when writing to 16bit PNG</p>
libpng Info... / libraryInfo	Button		<p>Display information about the underlying library.</p>

2.3 Draw nodes

The following sections contain documentation about every node in the Draw group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.3.1 LightWrap node



This documentation is for version 1.0 of LightWrap.

Description

LightWrap helps composite objects onto a bright background by simulating reflections from the background light on the foreground, around its edges. Input A is the foreground image and its matte, and input B the the background to use for the wrapping effect.

The output of LightWrap should then be composited over the background to give the final composite.

Inputs

Input	Description	Optional
A		No
B		No

Controls

Parameter / script name	Type	Default	Function
Convert to Group / <code>convertToGroup</code>	Button		Converts this node to a Group: the internal node-graph and the user parameters will become editable
Diffuse / <code>diffuse</code>	Double	x: 15 y: 15	Size of the reflections from the background to the foreground element (Intensity controls their intensity). Start by setting Diffuse to zero and adjust intensity to see what colors from the background are being reflected. Then adjust Diffuse, come back to Intensity if necessary, and balance both parameters until the result is satisfactory.
Intensity / <code>intensity</code>	Double	0	Brightness of the reflections from the background to the foreground element (Diffuse controls their size). Start by setting Diffuse to zero and adjust intensity to see what colors from the background are being reflected. Then adjust Diffuse, come back to Intensity if necessary, and balance both parameters until the result is satisfactory.
Generate Wrap Only / <code>onlyWrap</code>	Boolean	Off	When checked, the LightWrap in generated but is not merged with the foreground object (disables the Highlight Merge).
Disable luminance-Based Wrap / <code>disableLuma</code>	Boolean	Off	When checked, the LightWrap effect is created uniformly around the edged, rather than being controled by the color of the background.
Enable Glow / <code>enableGlow</code>	Boolean	Off	When checked, the LightWrap is not masked by the foreground object, so that the objects seems to glow.
FGBLur / <code>fgblur</code>	Double	x: 1 y: 1	Size of the blur applied to the alpha channel of the foreground (i.e. the foreground matte). More blur causes more background to be added to the foreground.

Continued on next page

Table 24 – continued from previous page

Parameter / script name	Type	Default	Function
FGBLur Border Conditions / fgblurBoundary	Choice	Black	<p>Border conditions of the blur applied to the alpha channel of the foreground (i.e. the foreground matte). Use “Black” in most cases, and “Nearest” if the foreground matte should be extended beyond image borders when it touches them.</p> <p>Black (black): Dirichlet boundary condition: pixel values out of the image domain are zero.</p> <p>Nearest (nearest): Neumann boundary condition: pixel values out of the image domain are those of the closest pixel location in the image domain.</p>
BGBLur / bgblur	Double	x: 0 y: 0	Size of the blur applied to the background before merging it with the foreground element and applying the Diffuse blur.
Saturation / saturation	Double	1	Color saturation of the LightWrap effect. Advanced color correction parameters are available in the ColorCorrect tab.
Luma Tolerance / lumaTolerance	Double	0	Luminance threshold of the LightWrap effect. Luminance values below this do not generate a LightWrap.

Continued on next page

Table 24 – continued from previous page

Parameter / script name	Type	Default	Function
Highlight Merge / highlightmerge	Choice	plus	<p>Merge operation between the foreground object and the background. The default operation is “plus”, which produces a glow effect.</p> <p>atop: $Ab + B(1 - a)$ (a.k.a. src-atop)</p> <p>average: $(A + B) / 2$</p> <p>color: SetLum(A, Lum(B))</p> <p>color-burn: darken B towards A</p> <p>color-dodge: brighten B towards A</p> <p>conjoint-over: $A + B(1-a)/b$, A if $a > b$</p> <p>copy: A (a.k.a. src)</p> <p>difference: $\text{abs}(A-B)$ (a.k.a. absminus)</p> <p>disjoint-over: $A+B(1-a)/b$, A+B if $a+b < 1$</p> <p>divide: A/B, 0 if $A < 0$ and $B < 0$</p> <p>exclusion: $A+B-2AB$</p> <p>freeze: $1-\sqrt{1-A}/B$</p> <p>from: $B-A$ (a.k.a. subtract)</p> <p>geometric: $2AB/(A+B)$</p> <p>grain-extract: $B - A + 0.5$</p> <p>grain-merge: $B + A - 0.5$</p> <p>hard-light: multiply if $A < 0.5$, screen if $A > 0.5$</p> <p>hue: SetLum(SetSat(A, Sat(B)), Lum(B))</p> <p>hypot: $\sqrt{A^2+B^2}$</p> <p>in: Ab (a.k.a. src-in)</p> <p>luminosity: SetLum(B, Lum(A))</p> <p>mask: Ba (a.k.a. dst-in)</p> <p>matte: $Aa + B(1-a)$ (unpremultiplied over)</p> <p>max: $\max(A, B)$ (a.k.a. lighten only)</p> <p>min: $\min(A, B)$ (a.k.a. darken only)</p> <p>minus: $A-B$</p> <p>multiply: AB, 0 if $A < 0$ and $B < 0$</p> <p>out: $A(1-b)$ (a.k.a. src-out)</p> <p>over: $A+B(1-a)$ (a.k.a. src-over)</p> <p>overlay: multiply if $B < 0.5$, screen if $B > 0.5$</p> <p>pinlight: if $B \geq 0.5$ then $\max(A, 2*B - 1)$, $\min(A, B * 2.0)$ else</p> <p>plus: $A+B$ (a.k.a. add)</p> <p>reflect: $A*A / (1 - B)$</p> <p>saturation: SetLum(SetSat(B, Sat(A)), Lum(B))</p> <p>screen: $A+B-AB$ if A or $B \leq 1$, otherwise $\max(A, B)$</p> <p>soft-light: burn-in if $A < 0.5$, lighten if $A > 0.5$</p> <p>stencil: $B(1-a)$ (a.k.a. dst-out)</p> <p>under: $A(1-b)+B$ (a.k.a. dst-over)</p> <p>xor: $A(1-b)+B(1-a)$</p>
Use Constant Highlight / useConstant	Boolean	Off	When checked, use a constant color (specified by the Constant parameter) instead of the background for the LightWrap effect.

Continued on next page

Table 24 – continued from previous page

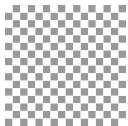
Parameter / script name	Type	Default	Function
Constant / constantcolor	Color	r: 1 g: 1 b: 1 a: 1	Color to use in the LightWrap effect when Use constant highlight is enabled.
Saturation / ColorCorrect1MasterSaturation	Color	r: 1 g: 1 b: 1 a: 1	
Contrast / ColorCorrect1MasterContrast	Color	r: 1 g: 1 b: 1 a: 1	
Gamma / ColorCorrect1MasterGamma	Color	r: 1 g: 1 b: 1 a: 1	
Gain / ColorCorrect1MasterGain	Color	r: 1 g: 1 b: 1 a: 1	
Offset / ColorCorrect1MasterOffset	Color	r: 0 g: 0 b: 0 a: 0	
Enable / ColorCorrect1ShadowsEnable	Boolean	On	
Saturation / ColorCorrect1ShadowsSaturation	Color	r: 1 g: 1 b: 1 a: 1	
Contrast / ColorCorrect1ShadowsContrast	Color	r: 1 g: 1 b: 1 a: 1	
Gamma / ColorCorrect1ShadowsGamma	Color	r: 1 g: 1 b: 1 a: 1	
Gain / ColorCorrect1ShadowsGain	Color	r: 1 g: 1 b: 1 a: 1	
Offset / ColorCorrect1ShadowsOffset	Color	r: 0 g: 0 b: 0 a: 0	
Enable / ColorCorrect1MidtonesEnable	Boolean	On	
Saturation / ColorCorrect1MidtonesSaturation	Color	r: 1 g: 1 b: 1 a: 1	
Contrast / ColorCorrect1MidtonesContrast	Color	r: 1 g: 1 b: 1 a: 1	
Gamma / ColorCorrect1MidtonesGamma	Color	r: 1 g: 1 b: 1 a: 1	
Gain / ColorCorrect1MidtonesGain	Color	r: 1 g: 1 b: 1 a: 1	
Offset / ColorCorrect1MidtonesOffset	Color	r: 0 g: 0 b: 0 a: 0	

Continued on next page

Table 24 – continued from previous page

Parameter / script name	Type	Default	Function
Enable / ColorCorrect1Highlights	Boolean	On	
Saturation / ColorCorrect1Highlights	Color	r: 1 g: 1 b: 1 a: 1	Saturation
Contrast / ColorCorrect1Highlights	Color	r: 1 g: 1 b: 1 a: 1	Contrast
Gamma / ColorCorrect1Highlights	Color	r: 1 g: 1 b: 1 a: 1	Gamma
Gain / ColorCorrect1Highlights	Color	r: 1 g: 1 b: 1 a: 1	Gain
Offset / ColorCorrect1Highlights	Color	r: 0 g: 0 b: 0 a: 0	Offset

2.3.2 Noise node



This documentation is for version 2.0 of Noise.

Description

Add random noise to input stream.

Uses the ‘noise’ function from the CImg library, modified so that noise is reproducible at each render.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		Yes
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Sigma / sigma	Double	0.01	Amplitude of the random additive noise.

Continued on next page

Table 25 – continued from previous page

Parameter / script name	Type	Default	Function
Type / type	Choice	Gaussian	Type of additive noise. Gaussian (gaussian) : Gaussian noise. Uniform (uniform) : Uniform noise. Salt & Pepper (saltnpapper) : Salt & pepper noise. Poisson (poisson) : Poisson noise. Image is divided by Sigma before computing noise, then remultiplied by Sigma. Rice (rice) : Rician noise.
Seed / seed	Integer	2000	Random seed: change this if you want different instances to have different noise.
Static Seed / staticSeed	Boolean	Off	When enabled, the dither pattern remains the same for every frame producing a constant noise effect.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.3.3 Plasma node



This documentation is for version 2.0 of Plasma.

Description

Draw a random plasma texture (using the mid-point algorithm).

Uses the ‘draw_plasma’ function from the CImg library, modified so that noise is reproducible at each render.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		Yes
Mask		Yes

Controls

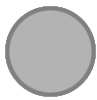
Parameter / script name	Type	Default	Function
Alpha / alpha	Double	0.002	Alpha-parameter, in intensity units (≥ 0).
Beta / beta	Double	0	Beta-parameter, in intensity units (≥ 0).
Scale / scale	Integer	8	Noise scale, as a power of two (≥ 0).

Continued on next page

Table 26 – continued from previous page

Parameter / script name	Type	Default	Function
Offset / offset	Double	0	Offset to add to the plasma noise.
Seed / seed	Integer	2000	Random seed: change this if you want different instances to have different noise.
Static Seed / staticSeed	Boolean	On	When enabled, the dither pattern remains the same for every frame producing a constant noise effect.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.3.4 Radial node



This documentation is for version 2.1 of Radial.

Description

Radial ramp.

The ramp is composited with the source image using the ‘over’ operator.

If no source is connected, this effect behaves like a generator. Its region of definition is:

- The selected format if the Extent parameter is a format.
- The project output format if Color0 is not black and transparent.
- The selected extent plus a one-pixel border if Color0 is black and transparent.

See also: <http://opticalenquiry.com/nuke/index.php?title=Radial>

Inputs

Input	Description	Optional
Source		Yes
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Extent / extent	Choice	Size	Extent (size and offset) of the output. Format (format): Use a pre-defined image format. Size (size): Use a specific extent (size and offset). Project (project): Use the project extent (size and offset). Default (default): Use the default extent (e.g. the source clip extent, if connected).
Center / recenter	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.
Reformat / reformat	Boolean	Off	Set the output format to the given extent, except if the Bottom Left or Size parameters is animated.
Format / NatronParamFormat	Choice	HD 1920x1080	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Frame Range / frameRange	Integer	min: 1 max: 1	Time domain.
Softness / softness	Double	1	Softness of the radial ramp. Draws an anti-aliased disc or ellipse if zero.
Perceptually Linear / plinear	Boolean	Off	Make the radial ramp look more linear to the eye.
Color 0 / color0	Color	r: 0 g: 0 b: 0 a: 0	

Continued on next page

Table 27 – continued from previous page

Parameter / script name	Type	Default	Function
Color 1 / <code>color1</code>	Color	r: 1 g: 1 b: 1 a: 1	
Expand RoD / <code>expandRoD</code>	Boolean	On	Expand the source region of definition by the shape RoD (if Source is connected and <code>color0=(0,0,0,0)</code>).
Invert Mask / <code>maskInvert</code>	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / <code>mix</code>	Double	1	Mix factor between the original and the transformed image.

2.3.5 Ramp node



This documentation is for version 2.0 of Ramp.

Description

Draw a ramp between 2 edges.

The ramp is composited with the source image using the ‘over’ operator.

See also: <http://opticalenquiry.com/nuke/index.php?title=Ramp>

Inputs

Input	Description	Optional
Source		Yes
Mask		Yes

Controls

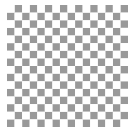
Parameter / script name	Type	Default	Function
Ramp Type / <code>type</code>	Choice	Linear	<p>The type of interpolation used to generate the ramp</p> <p>Linear (linear): Linear ramp.</p> <p>PLinear (plinear): Perceptually linear ramp in Rec.709.</p> <p>Ease-in (easein): Catmull-Rom spline, smooth start, linear end (a.k.a. <code>smooth0</code>).</p> <p>Ease-out (easeout): Catmull-Rom spline, linear start, smooth end (a.k.a. <code>smooth1</code>).</p> <p>Smooth (smooth): Traditional smoothstep ramp.</p> <p>None (none): No color gradient.</p>
Point 0 / <code>point0</code>	Double	x: 100 y: 100	

Continued on next page

Table 28 – continued from previous page

Parameter / script name	Type	Default	Function
Color 0 / <code>color0</code>	Color	r: 0 g: 0 b: 0 a: 0	
Point 1 / <code>point1</code>	Double	x: 100 y: 200	
Color 1 / <code>color1</code>	Color	r: 1 g: 1 b: 1 a: 1	
Interactive Update / <code>interactive</code>	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Invert Mask / <code>maskInvert</code>	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / <code>mix</code>	Double	1	Mix factor between the original and the transformed image.

2.3.6 Rand node



This documentation is for version 1.0 of Rand.

Description

Generate a random field of noise. The field does not resample if you change the resolution or density (you can animate the density without pixels randomly changing).

Inputs

Input	Description	Optional
Source		Yes

Controls

Parameter / script name	Type	Default	Function
Extent / <code>extent</code>	Choice	Default	Extent (size and offset) of the output. Format (format): Use a pre-defined image format. Size (size): Use a specific extent (size and offset). Project (project): Use the project extent (size and offset). Default (default): Use the default extent (e.g. the source clip extent, if connected).
Center / <code>recenter</code>	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.

Continued on next page

Table 29 – continued from previous page

Parameter / script name	Type	Default	Function
Reformat / reformat	Boolean	Off	Set the output format to the given extent, except if the Bottom Left or Size parameters is animated.
Format / NatronParamFormatChoice	Choice	HD 1920x1080	<p>The output format</p> <p>PC_Video 640x480 (PC_Video)</p> <p>NTSC 720x486 0.91 (NTSC)</p> <p>PAL 720x576 1.09 (PAL)</p> <p>NTSC_16:9 720x486 1.21 (NTSC_16:9)</p> <p>PAL_16:9 720x576 1.46 (PAL_16:9)</p> <p>HD_720 1280x720 (HD_720)</p> <p>HD 1920x1080 (HD)</p> <p>UHD_4K 3840x2160 (UHD_4K)</p> <p>1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap))</p> <p>1K_Cinemascope 914x778 2.00 (1K_Cinemascope)</p> <p>2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap))</p> <p>2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope)</p> <p>2K_DCP 2048x1080 (2K_DCP)</p> <p>4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap))</p> <p>4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope)</p> <p>4K_DCP 4096x2160 (4K_DCP)</p> <p>square_256 256x256 (square_256)</p> <p>square_512 512x512 (square_512)</p> <p>square_1K 1024x1024 (square_1K)</p> <p>square_2K 2048x2048 (square_2K)</p>
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Frame Range / frameRange	Integer	min: 1 max: 1	Time domain.
Output Components / outputComponents	Choice	RGB	<p>Components in the output</p> <p>RGBA</p> <p>RGB</p> <p>XY</p> <p>Alpha</p>
Noise / noise	Double	1	How much noise to make.
Density / density	Double	1	The density from 0 to 1 of the pixels. A lower density mean fewer random pixels.
seed / seed	Integer	2000	Random seed: change this if you want different instances to have different noise.
Static Seed / staticSeed	Boolean	Off	When enabled, the seed is not combined with the frame number, and thus the effect is the same for all frames for a given seed number.

2.3.7 Rectangle node



This documentation is for version 2.1 of Rectangle.

Description

Draw a rectangle.

The rectangle is composited with the source image using the ‘over’ operator.

If no source is connected, this effect behaves like a generator. Its region of definition is:

- The selected format if the Extent parameter is a format.
- The project output format if Color0 is not black and transparent.
- The selected extent plus a one-pixel border if Color0 is black and transparent.

See also: <http://opticalenquiry.com/nuke/index.php?title=Rectangle>

Inputs

Input	Description	Optional
Source		Yes
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Extent / extent	Choice	Size	Extent (size and offset) of the output. Format (format): Use a pre-defined image format. Size (size): Use a specific extent (size and offset). Project (project): Use the project extent (size and offset). Default (default): Use the default extent (e.g. the source clip extent, if connected).
Center / recenter	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.
Reformat / reformat	Boolean	Off	Set the output format to the given extent, except if the Bottom Left or Size parameters is animated.

Continued on next page

Table 30 – continued from previous page

Parameter / script name	Type	Default	Function
Format / NatronParamFormatChoice	Choice	HD 1920x1080	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Frame Range / frameRange	Integer	min: 1 max: 1	Time domain.
Corner Radius / cornerRadius	Double	x: 0 y: 0	If non-zero, this is the radius of the round corners.
Softness / softness	Double	0	Softness of the rectangle edges. Draws an anti-aliased rectangle if zero
Color 0 / color0	Color	r: 0 g: 0 b: 0 a: 0	
Color 1 / color1	Color	r: 1 g: 1 b: 1 a: 1	
Expand RoD / expandRoD	Boolean	On	Expand the source region of definition by the shape RoD (if Source is connected and color0=(0,0,0,0)).
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.3.8 Roto node

This documentation is for version 1.0 of Roto.

Description

Create masks and shapes

Inputs

Input	Description	Optional
Bg		Yes
Bg2		Yes
Bg3		Yes
Bg4		Yes

Controls

Parameter / script name	Type	Default	Function
Opacity / opacity	Double	1	Controls the opacity of the selected shape(s).
Color / color	Color	r: 1 g: 1 b: 1	The color of the shape. This parameter is used when the output components are set to RGBA.
Life Time / lifeTime	Choice	All	Controls the life-time of the shape/stroke All: All frames Single: Only for the specified frame From start: From the start of the sequence up to the specified frame To end: From the specified frame to the end of the sequence Custom: Use the Activated parameter animation to control the life-time of the shape/stroke using keyframes
Activated / activated	Boolean	On	Controls whether the selected shape(s) should be rendered or not. Note that you can animate this parameter so you can activate/deactivate the shape throughout the time.
Feather / feather	Double	1.5	Controls the distance of feather (in pixels) to add around the selected shape(s)
Feather fall-off / featherFalloff	Double	1	Controls the rate at which the feather is applied on the selected shape(s).
Source / sourceType	Choice	background	Source color used for painting the stroke when the Reveal/Clone tools are used. foreground: The painted result at this point in the hierarchy. background: The original image unpainted connected to bg. background 2: The original image unpainted connected to bg1. background 3: The original image unpainted connected to bg2. background 4: The original image unpainted connected to bg3. background 5: The original image unpainted connected to bg4. background 6: The original image unpainted connected to bg5. background 7: The original image unpainted connected to bg6. background 8: The original image unpainted connected to bg7. background 9: The original image unpainted connected to bg8. background 10: The original image unpainted connected to bg9.

Continued on next page

Table 31 – continued from previous page

Parameter / script name	Type	Default	Function
Translate / cloneTranslate	Double	x: 0 y: 0	
Rotate / cloneRotate	Double	0	
Scale / cloneScale	Double	x: 1 y: 1	
Uniform / cloneUniform	Boolean	On	
Skew X / cloneSkewx	Double	0	
Skew Y / cloneSkewy	Double	0	
Skew Order / cloneSkewOrder	Choice	XY	XY YX
Center / cloneCenter	Double	x: 0.5 y: 0.5	
Reset Center / resetCloneCenter	Button		Reset the clone transform center
Reset Transform / resetCloneTransform	Button		Reset the clone transform to an identity
Filter / cloneFilter	Choice	Cubic	Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+). Impulse: (nearest neighbor / box) Use original values. Bilinear: (tent / triangle) Bilinear interpolation between original values. Cubic: (cubic spline) Some smoothing. Keys: (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*). Simon: Some smoothing, plus medium sharpening (*). Rifman: Some smoothing, plus significant sharpening (*). Mitchell: Some smoothing, plus blurring to hide pixelation (*+). Parzen: (cubic B-spline) Greatest smoothing of all filters (+). Notch: Flat smoothing (which tends to hide moire' patterns) (+).
Black Outside / blackOutside	Boolean	On	Fill the area outside the source image with black
Clone time offset / timeOffset	Integer	0	When the Clone tool is used, this determines depending on the time offset mode the source frame to clone. When in absolute mode, this is the frame number of the source, when in relative mode, this is an offset relative to the current frame.
Mode / timeOffsetMode	Choice	Relative	Time offset mode: when in absolute mode, this is the frame number of the source, when in relative mode, this is an offset relative to the current frame. Relative Absolute

Continued on next page

Table 31 – continued from previous page

Parameter / script name	Type	Default	Function
Brush Size / <code>brushSize</code>	Double	25	This is the diameter of the brush in pixels. Shift + drag on the viewer to modify this value
Brush Spacing / <code>brushSpacing</code>	Double	0.1	Spacing between stamps of the paint brush
Brush Hardness / <code>brushHardness</code>	Double	0.2	Fall off of the brush effect from the center to the edge
Brush effect / <code>brushEffect</code>	Double	15	The strength of the effect
Opacity / <code>pressureOpacity</code>	Boolean	On	Alters the opacity of the paint brush proportionate to changes in pen pressure
Size / <code>pressureSize</code>	Boolean	Off	Alters the size of the paint brush proportionate to changes in pen pressure
Hardness / <code>pressureHardness</code>	Boolean	Off	Alters the hardness of the paint brush proportionate to changes in pen pressure
Build-up / <code>buildUp</code>	Boolean	Off	When checked, the paint stroke builds up when painted over itself
Visible portion / <code>strokeVisiblePortion</code>	Double	start: 0 end: 1	Defines the range of the stroke that should be visible: 0 is the start of the stroke and 1 the end.
Translate / <code>translate</code>	Double	x: 0 y: 0	
Rotate / <code>rotate</code>	Double	0	
Scale / <code>scale</code>	Double	x: 1 y: 1	
Uniform / <code>uniform</code>	Boolean	On	
Skew X / <code>skewx</code>	Double	0	
Skew Y / <code>skewy</code>	Double	0	
Skew Order / <code>skewOrder</code>	Choice	XY	XY YX
Center / <code>center</code>	Double	x: 0.5 y: 0.5	
Reset Center / <code>resetTransformCenter</code>	Button		Reset the transform center
Interactive / <code>RotoTransformInteractive</code>	Boolean	On	When check, modifying the transform will directly render the shape in the viewer. When unchecked, modifications are applied when releasing the mouse button.
Extra Matrix / <code>extraMatrix</code>	Double	x: 1 y: 0 z: 0 w: 0 : 1 : 0 : 0 : 0 : 1	This matrix gets concatenated to the transform resulting from the parameter above.
Reset Transform / <code>resetTransform</code>	Button		Reset the transform to an identity

2.3.9 RotoPaint node

This documentation is for version 1.0 of RotoPaint.

Description

RotoPaint is a vector based free-hand drawing node that helps for tasks such as rotoscoping, matting, etc. . .

Inputs

Input	Description	Optional
Bg		Yes
Bg2		Yes
Bg3		Yes
Bg4		Yes

Controls

Parameter / script name	Type	Default	Function
Opacity / opacity	Double	1	Controls the opacity of the selected shape(s).
Color / color	Color	r: 1 g: 1 b: 1	The color of the shape. This parameter is used when the output components are set to RGBA.
Life Time / lifeTime	Choice	Single	Controls the life-time of the shape/stroke All: All frames Single: Only for the specified frame From start: From the start of the sequence up to the specified frame To end: From the specified frame to the end of the sequence Custom: Use the Activated parameter animation to control the life-time of the shape/stroke using keyframes
Frame / lifeTimeFrame	Integer	0	Use this to specify the frame when in mode Single/From start/To end
Feather / feather	Double	1.5	Controls the distance of feather (in pixels) to add around the selected shape(s)
Feather fall-off / featherFalloff	Double	1	Controls the rate at which the feather is applied on the selected shape(s).
Source / sourceType	Choice	background	Source color used for painting the stroke when the Reveal/Clone tools are used. foreground: The painted result at this point in the hierarchy. background: The original image unpainted connected to bg. background 2: The original image unpainted connected to bg1. background 3: The original image unpainted connected to bg2. background 4: The original image unpainted connected to bg3. background 5: The original image unpainted connected to bg4. background 6: The original image unpainted connected to bg5. background 7: The original image unpainted connected to bg6. background 8: The original image unpainted connected to bg7. background 9: The original image unpainted connected to bg8. background 10: The original image unpainted connected to bg9.
Translate / cloneTranslate	Double	x: 0 y: 0	
Rotate / cloneRotate	Double	0	
Scale / cloneScale	Double	x: 1 y: 1	

Continued on next page

Table 32 – continued from previous page

Parameter / script name	Type	Default	Function
Uniform / cloneUniform	Boolean	On	
Skew X / cloneSkewx	Double	0	
Skew Y / cloneSkewy	Double	0	
Skew Order / cloneSkewOrder	Choice	XY	XY YX
Center / cloneCenter	Double	x: 0.5 y: 0.5	
Reset Center / resetCloneCenter	Button		Reset the clone transform center
Reset Transform / resetCloneTransform	Button		Reset the clone transform to an identity
Filter / cloneFilter	Choice	Cubic	<p>Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+).</p> <p>Impulse: (nearest neighbor / box) Use original values.</p> <p>Bilinear: (tent / triangle) Bilinear interpolation between original values.</p> <p>Cubic: (cubic spline) Some smoothing.</p> <p>Keys: (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*).</p> <p>Simon: Some smoothing, plus medium sharpening (*).</p> <p>Rifman: Some smoothing, plus significant sharpening (*).</p> <p>Mitchell: Some smoothing, plus blurring to hide pixelation (*+).</p> <p>Parzen: (cubic B-spline) Greatest smoothing of all filters (+).</p> <p>Notch: Flat smoothing (which tends to hide moire' patterns) (+).</p>
Black Outside / blackOutside	Boolean	On	Fill the area outside the source image with black
Clone time offset / timeOffset	Integer	0	When the Clone tool is used, this determines depending on the time offset mode the source frame to clone. When in absolute mode, this is the frame number of the source, when in relative mode, this is an offset relative to the current frame.
Mode / timeOffsetMode	Choice	Relative	<p>Time offset mode: when in absolute mode, this is the frame number of the source, when in relative mode, this is an offset relative to the current frame.</p> <p>Relative Absolute</p>
Brush Size / brushSize	Double	25	This is the diameter of the brush in pixels. Shift + drag on the viewer to modify this value
Brush Spacing / brushSpacing	Double	0.1	Spacing between stamps of the paint brush
Brush Hardness / brushHardness	Double	0.2	Fall off of the brush effect from the center to the edge

Continued on next page

Table 32 – continued from previous page

Parameter / script name	Type	Default	Function
Brush effect / <code>brushEffect</code>	Double	15	The strength of the effect
Opacity / <code>pressureOpacity</code>	Boolean	On	Alters the opacity of the paint brush proportionate to changes in pen pressure
Size / <code>pressureSize</code>	Boolean	Off	Alters the size of the paint brush proportionate to changes in pen pressure
Hardness / <code>pressureHardness</code>	Boolean	Off	Alters the hardness of the paint brush proportionate to changes in pen pressure
Build-up / <code>buildUp</code>	Boolean	Off	When checked, the paint stroke builds up when painted over itself
Visible portion / <code>strokeVisiblePortion</code>	Double	start: 0 end: 1	Defines the range of the stroke that should be visible: 0 is the start of the stroke and 1 the end.
Translate / <code>translate</code>	Double	x: 0 y: 0	
Rotate / <code>rotate</code>	Double	0	
Scale / <code>scale</code>	Double	x: 1 y: 1	
Uniform / <code>uniform</code>	Boolean	On	
Skew X / <code>skewx</code>	Double	0	
Skew Y / <code>skewy</code>	Double	0	
Skew Order / <code>skewOrder</code>	Choice	XY	XY YX
Center / <code>center</code>	Double	x: 0.5 y: 0.5	
Reset Center / <code>resetTransformCenter</code>	Button		Reset the transform center
Interactive / <code>RotoTransformInteractive</code>	Boolean	On	When check, modifying the transform will directly render the shape in the viewer. When unchecked, modifications are applied when releasing the mouse button.
Extra Matrix / <code>extraMatrix</code>	Double	x: 1 y: 0 z: 0 w: 0 : 1 : 0 : 0 : 0 : 0 : 1	This matrix gets concatenated to the transform resulting from the parameter above.
Reset Transform / <code>resetTransform</code>	Button		Reset the transform to an identity

2.3.10 SeGrain node

This documentation is for version 1.0 of SeGrain.

Description

Adds synthetic grain.

Push “presets” to get predefined types of grain, these are the correct size for 2K scans.

You can also adjust the sliders to match a sample piece of grain. Find a sample with a rather constant background, blur it to remove the grain, and use as input to this. View with a wipe in the viewer so you can make a match. It helps to view and match each of the red, green, blue separately.

See also http://opticalenquiry.com/nuke/index.php?title=Integration#Matching_grain

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Seed / grainSeed	Double	134	Change this value to make different instances of this operator produce different noise.
Static Seed / staticSeed	Boolean	Off	When enabled, the seed is not combined with the frame number, and thus the effect is the same for all frames for a given seed number.
Presets / grainPresets	Choice	Kodak 5248	Presets for common types of film. Kodak 5248 Kodak 5279 Kodak FX214 Kodak GT5274 Kodak 5217 Kodak 5218 Other
All / grainSizeAll	Double	1	Global factor on grain size. Useful if working with scans which are not 2K (the preset sizes are computed for 2K scans).
Red / grainSizeRed	Double	3.3	Red grain size (in pixels).
Green / grainSizeGreen	Double	2.9	Green grain size (in pixels).
Blue / grainSizeBlue	Double	2.5	Blue grain size (in pixels).
Red / grainIrregularityRed	Double	0.6	Red grain irregularity.
Green / grainIrregularityGreen	Double	0.6	Green grain irregularity.
Blue / grainIrregularityBlue	Double	0.6	Blue grain irregularity.
Red / grainIntensityRed	Double	0.42	Amount of red grain to add to a white pixel.
Green / grainIntensityGreen	Double	0.46	Amount of green grain to add to a white pixel.
Blue / grainIntensityBlue	Double	0.85	Amount of blue grain to add to a white pixel.
Correlation / colorCorr	Double	0	This parameter specifies the apparent colorfulness of the grain. The value represents how closely the grain in each channel overlaps. This means that negative color correlation values decrease the amount of overlap, which increases the apparent color of the grain, while positive values decrease its colorfulness.
Black / grainBlack	Color	r: 0 g: 0 b: 0	Amount of grain to add everywhere.
Minimum / grainMinimum	Color	r: 0 g: 0 b: 0	Minimum black level.

Continued on next page

Table 33 – continued from previous page

Parameter / script name	Type	Default	Function
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.3.11 SeNoise node

This documentation is for version 1.0 of SeNoise.

Description

Generate noise.

Inputs

Input	Description	Optional
Source		Yes
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Replace / replace	Boolean	Off	Clear the selected channel(s) before drawing into them.
Noise Size / noiseSize	Double	x: 350 y: 350	Size of noise in pixels, corresponding to its lowest frequency.
Z0 / noiseZ	Double	0	Z coordinate on the noise at frame=0. The noise pattern is different for every integer value of Z, so this can be used as a random seed.
Z Slope / noiseZSlope	Double	0	Z is computed as $Z = Z0 + \text{frame} * Z_slope$. 0 means a constant noise, 1 means a different noise pattern at every frame.

Continued on next page

Table 34 – continued from previous page

Parameter / script name	Type	Default	Function
Noise Type / noiseType	Choice	FBM	<p>Kind of noise.</p> <p>Cell Noise (cell): Cell noise generates a field of constant colored cubes based on the integer location. This is the same as the prman cellnoise function. You may want to set xRotate and yRotate to 0 in the Transform tab to get square cells.</p> <p>Noise (noise): Noise is a random function that smoothly blends between samples at integer locations. This is Ken Perlin’s original noise function.</p> <p>FBM (fbm): FBM (Fractal Brownian Motion) is a multi-frequency noise function. The base frequency is the same as the “Noise” function. The total number of frequencies is controlled by octaves. The lacunarity is the spacing between the frequencies - a value of 2 means each octave is twice the previous frequency. The gain controls how much each frequency is scaled relative to the previous frequency.</p> <p>Turbulence (turbulence): turbulence is a variant of fbm where the absolute value of each noise term is taken. This gives a more billowy appearance.</p> <p>Voronoi (voronoi): Voronoi is a cellular noise pattern. It is a jittered variant of cellnoise. The type parameter describes different variants of the noise function. The jitter param controls how irregular the pattern is (jitter = 0 is like ordinary cellnoise). The fbm* params can be used to distort the noise field. When fbmScale is zero (the default), there is no distortion. The remaining params are the same as for the fbm function. NOTE: This does not necessarily return [0,1] value, because it can return arbitrary distance.</p>
Voronoi Type / voronoiType	Choice	Cell	<p>Different variants of the Voronoi noise function.</p> <p>Cell (cell)</p> <p>Type 2 (type2)</p> <p>Type 3 (type3)</p> <p>Type 4 (type4)</p> <p>Type 5 (type5)</p>
Jitter / jitter	Double	0.5	The jitter param controls how irregular the pattern is (jitter = 0 is like ordinary cellnoise).
FBM Scale / fbmScale	Double	0	The fbm* params can be used to distort the noise field. When fbmScale is zero (the default), there is no distortion.
Octaves / fbmOctaves	Integer	6	The total number of frequencies is controlled by octaves.
Lacunarity / fbmLacunarity	Double	2	The lacunarity is the spacing between the frequencies - a value of 2 means each octave is twice the previous frequency.
Gain / fbmGain	Double	0.5	The gain controls how much each frequency is scaled relative to the previous frequency.
Translate / transformTranslate	Double	x: 0 y: 0	Translation along the x and y axes in pixels. Can also be adjusted by clicking and dragging the center handle in the Viewer.
Rotate / transformRotate	Double	0	Rotation angle in degrees around the Center. Can also be adjusted by clicking and dragging the rotation bar in the Viewer.
Scale / transformScale	Double	x: 1 y: 1	Scale factor along the x and y axes. Can also be adjusted by clicking and dragging the outer circle or the diameter handles in the Viewer.

Continued on next page

Table 34 – continued from previous page

Parameter / script name	Type	Default	Function
Uniform / transformScaleUniform	Boolean	Off	Use the X scale for both directions
Skew X / transformSkewX	Double	0	Skew along the x axis. Can also be adjusted by clicking and dragging the skew bar in the Viewer.
Skew Y / transformSkewY	Double	0	Skew along the y axis.
Skew Order / transformSkewOrder	Choice	XY	The order in which skew transforms are applied: X then Y, or Y then X. XY YX
Amount / transformAmount	Double	1	Amount of transform to apply. 0 means the transform is identity, 1 means to apply the full transform.
Center / transformCenter	Double	x: 0.5 y: 0.5	Center of rotation and scale.
Reset Center / transformResetCenter	Button		Reset the position of the center to the center of the input region of definition
Interactive Update / transformInteractive	Boolean	On	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
X Rotate / XRotate	Double	27	Rotation about the X axis in the 3D noise space (X,Y,Z). Noise artifacts may appear if it is 0 or a multiple of 90.
Y Rotate / YRotate	Double	37	Rotation about the Y axis in the 3D noise space (X,Y,Z). Noise artifacts may appear if it is 0 or a multiple of 90.
Ramp Type / rampType	Choice	None	The type of interpolation used to generate the ramp Linear (linear) : Linear ramp. PLinear (plinear) : Perceptually linear ramp in Rec.709. Ease-in (easein) : Catmull-Rom spline, smooth start, linear end (a.k.a. smooth0). Ease-out (easeout) : Catmull-Rom spline, linear start, smooth end (a.k.a. smooth1). Smooth (smooth) : Traditional smoothstep ramp. None (none) : No color gradient.
Point 0 / rampPoint0	Double	x: 100 y: 100	
Color 0 / rampColor0	Color	r: 0 g: 0 b: 0 a: 0	
Point 1 / rampPoint1	Double	x: 100 y: 200	
Color 1 / rampColor1	Color	r: 1 g: 1 b: 1 a: 1	
Interactive Update / rampInteractive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.3.12 Text node



This documentation is for version 6.9 of Text.

Description

Advanced text generator node using Pango and Cairo.

Inputs

Input	Description	Optional
Source		Yes

Controls

Parameter / script name	Type	Default	Function
Rotate / rotate	Double	0	Rotation angle in degrees around the Center. Can also be adjusted by clicking and dragging the rotation bar in the Viewer.
Scale / scale	Double	x: 1 y: 1	Scale factor along the x and y axes. Can also be adjusted by clicking and dragging the outer circle or the diameter handles in the Viewer.
Uniform / uniform	Boolean	Off	Use the X scale for both directions
Skew X / skewX	Double	0	Skew along the x axis. Can also be adjusted by clicking and dragging the skew bar in the Viewer.
Skew Y / skewY	Double	0	Skew along the y axis.
Skew Order / skewOrder	Choice	XY	The order in which skew transforms are applied: X then Y, or Y then X. XY YX
Amount / transformAmount	Double	1	Amount of transform to apply. 0 means the transform is identity, 1 means to apply the full transform.
Center / center	Double	x: 0.5 y: 0.5	Center of rotation and scale.
Reset Center / resetCenter	Button		Reset the position of the center to the center of the input region of definition
Interactive Update / interactive	Boolean	On	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Transform / transform	Boolean	On	Use transform overlay for text position.
Auto size / autoSize	Boolean	Off	Set canvas sized based on text. This will disable word wrap, custom canvas size and circle effect. Transform functions should also not be used in combination with this feature.
Center Interact / centerInteract	Boolean	Off	Center the text in the interact.
Canvas size / canvas	Integer	x: 0 y: 0	Set canvas size, default (0) is project format. Disabled if auto size is active.

Continued on next page

Table 35 – continued from previous page

Parameter / script name	Type	Default	Function
Markup / markup	Boolean	Off	Pango Text Attribute Markup Language, https://developer.gnome.org/pango/stable/PangoMarkupFormat.html . Colors don't work if Circle/Arc effect is used.
File / file	N/A		Use text from filename.
Text / text	String	Enter text	The text that will be drawn.
Justify / justify	Boolean	Off	Text justify.
Wrap / wrap	Choice	None	Word wrap. Disabled if auto size or custom (transform) position is enabled. None Word Char Word-Char
Horizontal align / align	Choice	Left	Horizontal text align. Disabled if custom position is enabled. Left Right Center
Vertical align / valign	Choice	Top	Vertical text align. Disabled if custom position is enabled. Top Center Bottom
Font family / name	Choice		The name of the font to be used.
Custom font / custom	N/A		Add custom font.
Font size / size	Integer	64	The height of the characters to render in pixels. Should not be used for animation, see the scale param.
Font color / color	Color	r: 1 g: 1 b: 1 a: 1	The fill color of the text to render.
Background Color / backgroundColor	Color	r: 0 g: 0 b: 0 a: 0	The fill color of the background.
Letter spacing / letterSpacing	Integer	0	Spacing between letters. Disabled if markup is used.
Hint style / hintStyle	Choice	Default	This controls whether to fit font outlines to the pixel grid, and if so, whether to optimize for fidelity or contrast. Default None Slight Medium Full

Continued on next page

Table 35 – continued from previous page

Parameter / script name	Type	Default	Function
Hint metrics / hintMetrics	Choice	Default	This controls whether metrics are quantized to integer values in device units. Default Off On
Antialiasing / antialiasing	Choice	Default	This specifies the type of antialiasing to do when rendering text. Default None Gray Subpixel
Subpixel / subpixel	Choice	Default	The subpixel order specifies the order of color elements within each pixel on the device. The antialiasing mode for the font display device when rendering with an antialiasing mode. Default RGB BGR VRGB VBGR
Style / style	Choice	Normal	Font style. Normal Bold Italic
Weight / weight	Choice	Normal	The weight field specifies how bold or light the font should be. Thin Ultra light Light Semi light Book Normal Medium Semi bold Bold Ultra bold Heavy Ultra heavy

Continued on next page

Table 35 – continued from previous page

Parameter / script name	Type	Default	Function
Stretch / stretch	Choice	Normal	Width of the font relative to other designs within a family. Ultra condensed Extra condensed Condensed Semi condensed Normal Semi expanded Expanded Extra expanded Ultra expanded
Stroke size / strokeSize	Double	0	Stroke size.
Stroke color / strokeColor	Color	r: 1 g: 0 b: 0 a: 1	The fill color of the stroke to render.
Stroke dash length / strokeDash	Integer	0	The length of the dashes.
Stroke dash pattern / strokeDashPattern	Double	x: 1 y: 0 z: 0	An array specifying alternate lengths of on and off stroke portions.
Circle radius / circleRadius	Double	0	Circle radius. Effect only works if auto size is disabled.
Circle Words / circleWords	Integer	10	X times text in circle.
Arc Radius / arcRadius	Double	100	Arc path radius (size of the path). The Arc effect is an experimental feature. Effect only works if auto size is disabled.
Arc Angle / arcAngle	Double	0	Arc Angle, set to 360 for a full circle. The Arc effect is an experimental feature. Effect only works if auto size is disabled.
Scroll X / scrollX	Double	0	Scroll canvas X. Only works if Transform, AutoSize, Circle and Arc is disabled/not used.
Scroll Y / scrollY	Double	0	Scroll canvas Y. Only works if Transform, AutoSize, Circle and Arc is disabled/not used.

2.4 Time nodes

The following sections contain documentation about every node in the Time group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.4.1 AppendClip node

This documentation is for version 1.0 of AppendClip.

Description

Append one clip to another.

See also: <http://opticalenquiry.com/nuke/index.php?title=AppendClip>

Inputs

Input	Description	Optional
1		Yes
2		Yes
3		Yes
4		Yes

Controls

Parameter / script name	Type	Default	Function
Fade In / <code>fadeIn</code>	Integer	0	Number of frames to fade in from black at the beginning of the first clip.
Fade Out / <code>fadeOut</code>	Integer	0	Number of frames to fade out to black at the end of the last clip.
Cross Dissolve / <code>crossDissolve</code>	Integer	0	Number of frames to cross-dissolve between clips.
First Frame / <code>firstFrame</code>	Integer	1	Frame to start the first clip at.
Last Frame / <code>lastFrame</code>	Integer	0	Last frame of the assembled clip (read-only).
Update / <code>updateLastFrame</code>	Button		Update lastFrame.

2.4.2 Deinterlace node



This documentation is for version 1.0 of Deinterlace.

Description

Deinterlace input stream.

The following deinterlacing algorithms are supported:

- Weave: This is what 100fps.com calls “do nothing”. Other names: “disabled” or “no deinterlacing”. Should be used for PsF content.
- Blend: Blender (full resolution). Each line of the picture is created as the average of a line from the odd and a line from the even half-pictures. This ignores the fact that they are supposed to be displayed at different times.
- Bob: Doubler. Display each half-picture like a full picture, by simply displaying each line twice. Preserves temporal resolution of interlaced video.
- Discard: Only display one of the half-pictures, discard the other. Other name: “single field”. Both temporal and vertical spatial resolutions are halved. Can be used for slower computers or to give interlaced video movie-like look with characteristic judder.
- Linear: Doubler. Bob with linear interpolation: instead of displaying each line twice, line 2 is created as the average of line 1 and 3, etc.
- Mean: Blender (half resolution). Display a half-picture that is created as the average of the two original half-pictures.

- Yadif: Interpolator (Yet Another DeInterlacing Filter) from MPlayer by Michael Niedermayer (<http://www.mplayerhq.hu>). It checks pixels of previous, current and next frames to re-create the missed field by some local adaptive method (edge-directed interpolation) and uses spatial check to prevent most artifacts.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Field Order / <code>fieldOrder</code>	Choice	HD=upper,SD=lower	Interlaced field order Lower field first (lower) : Lower field first. Upper field first (upper) : Upper field first HD=upper,SD=lower (auto) : Automatic.
Parity / <code>parity</code>	Choice	Lower	Field to interpolate. Lower (lower) : Interpolate lower field. Upper (upper) : Interpolate upper field.
Yadif Processing Mode / <code>yadifMode</code>	Choice	Temporal & spatial	Mode of checking fields Temporal & spatial (temporalspatial) : Temporal and spatial interlacing check (default). Temporal only (temporal) : Skips spatial interlacing check.

2.4.3 FrameBlend node



This documentation is for version 2.0 of FrameBlend.

Description

Blend frames of the input clip.

If a foreground matte is connected, only pixels with a negative or zero foreground value are taken into account, so that the foreground is not mixed with the background.

The number of values used to compute each pixel can be output to the alpha channel.

Inputs

Input	Description	Optional
Source		No
Mask		Yes
FgM		Yes

Controls

Parameter / script name	Type	Default	Function
Frame Range / frameRange	Integer	first: -5 last: 0	Range of frames which are to be blended together. Frame range is absolute if “absolute” is checked, else relative. The last frame is always included, and then one frame out of frameInterval within this interval.
Absolute / absolute	Boolean	Off	Use an absolute frame range. If the frame range is not animated or is not an expression, then all output images will be the same.
Input Range / inputRange	Button		Set the frame range to the input range. This can be used, combined with a foreground matte, to produce a clean background plate.
Frame Interval / frameInterval	Integer	1	Interval (in frames) between frames to process. 1 means to process every frame in the range. The first frame processed is the lower bound of the range. Can be used to reduce processing time or memory usage.
Operation / operation	Choice	Average	<p>The operation used to compute the output image.</p> <p>Average (average): Output is the average of selected frames.</p> <p>Min (min): Output is the minimum of selected frames.</p> <p>Max (max): Output is the maximum of selected frames.</p> <p>Sum (sum): Output is the sum/addition of selected frames.</p> <p>Product (product): Output is the product/multiplication of selected frames.</p> <p>Over (over): Output is the ‘over’ composition of selected frames.</p>
Decay / decay	Double	0	Before applying the blending operation, frame t is multiplied by $(1 - \text{decay})^{(\text{last} - t)}$.
Output Count to Alpha / outputCount	Boolean	Off	Output image count at each pixel to alpha (input must have an alpha channel).
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.4.4 FrameHold node



This documentation is for version 1.0 of FrameHold.

Description

Hold a given frame for the input clip indefinitely, or use a subsample of the input frames and hold them for several frames.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
First Frame / firstFrame	Integer	0	Reference input frame (the frame to hold if increment is 0).
Increment / increment	Integer	0	If increment is 0, only the “firstFrame” will be held. If it is positive, every multiple of “increment” plus “firstFrame” will be held for “increment” frames afterwards (before if it is negative).

2.4.5 FrameRange node



This documentation is for version 1.0 of FrameRange.

Description

Set the frame range for a clip. Useful in conjunction with AppendClipOFX.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Frame Range / frameRange	Integer	first: 1 last: 1	Output frame range.
Reset / reset	Button		Resets the frame range to its initial value.
Before / before	Choice	Black	What the plugin should return for frames before the first frame. Original (original): Return the original frame from the source, even if it is out of the frame range. Hold (hold): Return the nearest frame within the frame range. Black (black): Return an empty frame.

Continued on next page

Table 40 – continued from previous page

Parameter / script name	Type	Default	Function
After / after	Choice	Black	<p>What the plugin should return for frames after the last frame.</p> <p>Original (original): Return the original frame from the source, even if it is out of the frame range.</p> <p>Hold (hold): Return the nearest frame within the frame range.</p> <p>Black (black): Return an empty frame.</p>

2.4.6 NoTimeBlur node

This documentation is for version 1.0 of NoTimeBlur.

Description

Rounds fractional frame numbers to integers. This can be used to avoid computing non-integer frame numbers, and to discretize motion (useful for animated objects). This plug-in is usually inserted upstream from TimeBlur.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Rounding / rounding	Choice	rint	<p>Rounding type/operation to use when blocking fractional frames.</p> <p>rint: Round to the nearest integer value.</p> <p>floor: Round down to the nearest integer value.</p> <p>ceil: Round up to the nearest integer value.</p> <p>none: Do not round.</p>

2.4.7 Retime node



This documentation is for version 1.0 of Retime.

Description

Change the timing of the input clip.

See also: <http://opticalenquiry.com/nuke/index.php?title=Retime>

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Reverse input / <code>reverseInput</code>	Boolean	Off	Reverse the order of the input frames so that last one is first
Speed / <code>speed</code>	Double	1	How much to change the speed of the input clip. To determine which input frame is taken at a given time, the speed is integrated from the beginning of the source frame range to the given time, so that speed can be animated to locally accelerate ($\text{speed} > 1$), decelerate ($\text{speed} < 1$) or reverse ($\text{speed} < 0$) the source clip. Note that this is not the same as the speed parameter of the Nuke Retime node, which just multiplies the speed value at the current time by the time to obtain the source frame number.
Warp / <code>warp</code>	Parametric		Curve that maps input range (after applying speed) to the output range. A low positive slope slows down the input clip, and a negative slope plays it backwards.
Filter / <code>filter</code>	Choice	Linear	How input images are combined to compute the output image. None (none) : Do not interpolate, ask for images with fractional time to the input effect. Useful if the input effect can interpolate itself. Nearest (nearest) : Pick input image with nearest integer time. Linear (linear) : Blend the two nearest images with linear interpolation.

2.4.8 SlitScan node



This documentation is for version 1.0 of SlitScan.

Description

Apply per-pixel retiming: the time is computed for each pixel from the retime function, which can be either a horizontal ramp, a vertical ramp, or a retime map.

The default retime function corresponds to a horizontal slit: it is a vertical ramp, which is a linear function of y , which is 0 at the center of the bottom image line, and 1 at the center of the top image line. Optionally, a vertical slit may be used (0 at the center of the leftmost image column, 1 at the center of the rightmost image column), or the optional single-channel “Retime Map” input may also be used.

This plugin requires to render many frames on input, which may require a lot of memory.

Note that the results may be on higher quality if the video is slowed fown (e.g. using `slowmoVideo`)

The parameters are:

- retime function (default = horizontal slit)
- offset for the retime function (default = 0)
- gain for the retime function (default = -10)
- absolute, a boolean indicating that the time map gives absolute frames rather than relative frames
- frame range, only used if the retime function is given by a retime map, because the actual frame range cannot be guessed without inspecting the retime map content (default = -10..0). If “absolute” is checked, this frame range is absolute, else it is relative to the current frame
- filter to handle time offsets that “fall between” frames. They can be mapped to the nearest frame, or interpolated between the nearest frames (corresponding to a shutter of 1 frame).

References:

- An Informal Catalogue of Slit-Scan Video Artworks and Research, Golan Levin, http://www.flong.com/texts/lists/slit_scan/

Inputs

Input	Description	Optional
Source		No
Retime Map		Yes

Controls

Parameter / script name	Type	Default	Function
Retime Function / <code>retimeFunction</code>	Choice	Horizontal Slit	<p>The function that gives, for each pixel in the image, its time. The default retime function corresponds to a horizontal slit: it is a vertical ramp (a linear function of y) which is 0 at the center of the bottom image line, and 1 at the center of the top image line. Optionally, a vertical slit may be used (0 at the center of the leftmost image column, 1 at the center of the rightmost image column), or the optional single-channel “Retime Map” input may also be used.</p> <p>Horizontal Slit (<code>horizontalslit</code>): A vertical ramp (a linear function of y) which is 0 at the center of the bottom image line, and 1 at the center of the top image line.</p> <p>Vertical Slit (<code>verticalslit</code>): A horizontal ramp (a linear function of x) which is 0 at the center of the leftmost image line, and 1 at the center of the rightmost image line.</p> <p>Retime Map (<code>retimeMap</code>): The single-channel image from the “Retime Map” input (zero if not connected).</p>
Retime Offset / <code>retimeOffset</code>	Double	0	Offset to the retime map.
Retime Gain / <code>retimeGain</code>	Double	-10	Gain applied to the retime map (after offset). With the horizontal or vertical slits, to get one line or column per frame you should use respectively (height-1) or (width-1).
Absolute / <code>retimeAbsolute</code>	Boolean	Off	If checked, the retime map contains absolute time, if not it is relative to the current frame.
Max. Frame Range / <code>frameRange</code>	Integer	min: -10 max: 0	Maximum input frame range to fetch images from (may be relative or absolute, depending on the “absolute” parameter). Only used if the Retime Map is used and connected.

Continued on next page

Table 43 – continued from previous page

Parameter / script name	Type	Default	Function
Filter / filter	Choice	Nearest	How input images are combined to compute the output image. Nearest (nearest) : Pick input image with nearest integer time. Linear (linear) : Blend the two nearest images with linear interpolation.

2.4.9 TimeBlur node

This documentation is for version 1.0 of TimeBlur.

Description

Blend frames of the input clip over the shutter range.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Divisions / division	Integer	10	Number of time samples along the shutter time. The first frame is always at the start of the shutter range, and the shutter range is divided by divisions. The frame corresponding to the end of the shutter range is not included. If divisions=4, Shutter=1, Shutter Offset=Centered, this leads to blending the frames at t-0.5, t-0.25, t, t+0.25.
Shutter / shutter	Double	0.5	Controls how long (in frames) the shutter should remain open.
Shutter Offset / shutterOffset	Choice	Start	Controls when the shutter should be open/closed. Ignored if there is no motion blur (i.e. shutter=0 or motionBlur=0). Centered (centered) : Centers the shutter around the frame (from t-shutter/2 to t+shutter/2) Start (start) : Open the shutter at the frame (from t to t+shutter) End (end) : Close the shutter at the frame (from t-shutter to t) Custom (custom) : Open the shutter at t+shuttercustomoffset (from t+shuttercustomoffset to t+shuttercustomoffset+shutter)
Custom Offset / shutterCustomOffset	Double	0	When custom is selected, the shutter is open at current time plus this offset (in frames). Ignored if there is no motion blur (i.e. shutter=0 or motionBlur=0).

2.4.10 TimeOffset node



This documentation is for version 1.0 of TimeOffset.

Description

Move the input clip forward or backward in time. This can also reverse the order of the input frames so that last one is first.

See also <http://opticalenquiry.com/nuke/index.php?title=TimeOffset>

Inputs

Input	Description	Optional
Source		No

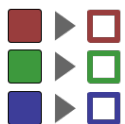
Controls

Parameter / script name	Type	Default	Function
Time Offset (Frames) / <code>timeOffset</code>	Integer	0	Offset in frames (frame <i>f</i> from the input will be at <i>f</i> +offset)
Reverse Input / <code>reverseInput</code>	Boolean	Off	Reverse the order of the input frames so that last one is first
Clip to Input Range / <code>clipToInputRange</code>	Boolean	Off	Never ask for frames outside of the input frame range.

2.5 Channel nodes

The following sections contain documentation about every node in the Channel group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.5.1 Shuffle node



This documentation is for version 3.0 of Shuffle.

Description

Rearrange channels from one or two inputs and/or convert to different bit depth or components. No colorspace conversion is done (mapping is linear, even for 8-bit and 16-bit types).

Inputs

Input	Description	Optional
B		Yes
A		Yes

Controls

Parameter / script name	Type	Default	Function
Output Layer / <code>outputLayer</code>	Choice	Color.RGBA	<p>The layer where the result of the Shuffle operation is output.</p> <p>Color.RGBA (<code>uk.co.thefoundry.OfxImagePlaneColour</code>)</p> <p>DisparityLeft.Disparity (<code>uk.co.thefoundry.OfxImagePlaneStereoDisparityLeft</code>)</p> <p>DisparityRight.Disparity (<code>uk.co.thefoundry.OfxImagePlaneStereoDisparityRight</code>)</p> <p>Backward.Motion (<code>uk.co.thefoundry.OfxImagePlaneBackMotionVector</code>)</p> <p>Forward.Motion (<code>uk.co.thefoundry.OfxImagePlaneForwardMotionVector</code>)</p>
Output Components / <code>outputComponents</code>	Choice	RGBA	<p>Select what types of components the plug-in should output, this has an effect only when the Output Layer is set to the Color layer. This controls what should be the components for the Color Layer: Alpha, RGB or RGBA.</p> <p>RGBA (rgba): Output RGBA components.</p> <p>RGB (rgb): Output RGB components.</p> <p>Alpha (alpha): Output Alpha component.</p>
Output Premult / <code>outputPremult</code>	Choice	Unpremultiplied	<p>Set the premultiplication metadata on the output. This does not modify the data itself. The premultiplication metadata will flow downstream so that further down effects know what kind of data to expect. By default it should be set to Unpremultiplied and you should always provide the Shuffle node unpremultiplied data. Providing alpha-premultiplied data in input of the Shuffle may produce wrong results because of the potential loss of the associated alpha channel.</p> <p>Opaque</p> <p>Premultiplied</p> <p>Unpremultiplied</p>

Continued on next page

Table 46 – continued from previous page

Parameter / script name	Type	Default	Function
R / outputR	Choice	B.Color.R	<p>Input channel for the output red channel.</p> <p>A.Color.R (A.uk.co.thefoundry.OfxImagePlaneColour.R): R channel from input A</p> <p>A.Color.G (A.uk.co.thefoundry.OfxImagePlaneColour.G): G channel from input A</p> <p>A.Color.B (A.uk.co.thefoundry.OfxImagePlaneColour.B): B channel from input A</p> <p>A.Color.A (A.uk.co.thefoundry.OfxImagePlaneColour.A): A channel from input A</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p> <p>B.Color.R (B.uk.co.thefoundry.OfxImagePlaneColour.R): R channel from input B</p> <p>B.Color.G (B.uk.co.thefoundry.OfxImagePlaneColour.G): G channel from input B</p> <p>B.Color.B (B.uk.co.thefoundry.OfxImagePlaneColour.B): B channel from input B</p> <p>B.Color.A (B.uk.co.thefoundry.OfxImagePlaneColour.A): A channel from input B</p>
G / outputG	Choice	B.Color.G	<p>Input channel for the output green channel.</p> <p>A.Color.R (A.uk.co.thefoundry.OfxImagePlaneColour.R): R channel from input A</p> <p>A.Color.G (A.uk.co.thefoundry.OfxImagePlaneColour.G): G channel from input A</p> <p>A.Color.B (A.uk.co.thefoundry.OfxImagePlaneColour.B): B channel from input A</p> <p>A.Color.A (A.uk.co.thefoundry.OfxImagePlaneColour.A): A channel from input A</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p> <p>B.Color.R (B.uk.co.thefoundry.OfxImagePlaneColour.R): R channel from input B</p> <p>B.Color.G (B.uk.co.thefoundry.OfxImagePlaneColour.G): G channel from input B</p> <p>B.Color.B (B.uk.co.thefoundry.OfxImagePlaneColour.B): B channel from input B</p> <p>B.Color.A (B.uk.co.thefoundry.OfxImagePlaneColour.A): A channel from input B</p>

Continued on next page

Table 46 – continued from previous page

Parameter / script name	Type	Default	Function
B / outputB	Choice	B.Color.B	<p>Input channel for the output blue channel.</p> <p>A.Color.R (A.uk.co.thefoundry.OfxImagePlaneColour.R): R channel from input A</p> <p>A.Color.G (A.uk.co.thefoundry.OfxImagePlaneColour.G): G channel from input A</p> <p>A.Color.B (A.uk.co.thefoundry.OfxImagePlaneColour.B): B channel from input A</p> <p>A.Color.A (A.uk.co.thefoundry.OfxImagePlaneColour.A): A channel from input A</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p> <p>B.Color.R (B.uk.co.thefoundry.OfxImagePlaneColour.R): R channel from input B</p> <p>B.Color.G (B.uk.co.thefoundry.OfxImagePlaneColour.G): G channel from input B</p> <p>B.Color.B (B.uk.co.thefoundry.OfxImagePlaneColour.B): B channel from input B</p> <p>B.Color.A (B.uk.co.thefoundry.OfxImagePlaneColour.A): A channel from input B</p>
A / outputA	Choice	B.Color.A	<p>Input channel for the output alpha channel.</p> <p>A.Color.R (A.uk.co.thefoundry.OfxImagePlaneColour.R): R channel from input A</p> <p>A.Color.G (A.uk.co.thefoundry.OfxImagePlaneColour.G): G channel from input A</p> <p>A.Color.B (A.uk.co.thefoundry.OfxImagePlaneColour.B): B channel from input A</p> <p>A.Color.A (A.uk.co.thefoundry.OfxImagePlaneColour.A): A channel from input A</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p> <p>B.Color.R (B.uk.co.thefoundry.OfxImagePlaneColour.R): R channel from input B</p> <p>B.Color.G (B.uk.co.thefoundry.OfxImagePlaneColour.G): G channel from input B</p> <p>B.Color.B (B.uk.co.thefoundry.OfxImagePlaneColour.B): B channel from input B</p> <p>B.Color.A (B.uk.co.thefoundry.OfxImagePlaneColour.A): A channel from input B</p>
Set GBA From R / setGBAFromR	Boolean	On	If checked, setting the R output channel from the GUI to the R channel of an input also sets the G, B and A output channels from the same plane.
Clip Info... / clipInfo	Button		Display information about the inputs.

2.6 Color nodes

The following sections contain documentation about every node in the Color group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.6.1 Add node



This documentation is for version 2.0 of Add.

Description

Add a constant to the selected channels.

See also: <http://opticalenquiry.com/nuke/index.php?title=Add>

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Value / value	Color	r: 0 g: 0 b: 0 a: 0	Constant to add to the selected channels.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.2 Clamp node



This documentation is for version 2.0 of Clamp.

Description

Clamp the values of the selected channels.

A special use case for the Clamp plugin is to generate a binary mask image (i.e. each pixel is either 0 or 1) by thresholding an image. Let us say one wants all input pixels whose value is above or equal to some threshold value to become 1, and all values below this threshold to become 0. Set the “Minimum” value to the threshold, set the “Maximum” to any value strictly below the threshold (e.g. 0 if the threshold is positive), and check “Enable MinClampTo” and “Enable MaxClampTo” while keeping the default values for “MinClampTo” (0.0) and “MaxClampTo” (1.0). The result is a binary mask image. To create a non-binary mask, with softer edges, either blur the output of Clamp, or use the Grade plugin instead, setting the “Black Point” and “White Point” to values close to the threshold, and checking the “Clamp Black” and “Clamp White” options.

See also: <http://opticalenquiry.com/nuke/index.php?title=Clamp>

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Minimum / minimum	Color	r: 0 g: 0 b: 0 a: 0	If enabled, all values that are lower than this number are set to this value, or to the minClampTo value if minClampTo is enabled.
Enable Minimum / minimumEnable	Boolean	On	Whether to clamp selected channels to a minimum value.
Maximum / maximum	Color	r: 1 g: 1 b: 1 a: 1	If enabled, all values that are higher than this number are set to this value, or to the maxClampTo value if maxClampTo is enabled.
Enable Maximum / maximumEnable	Boolean	On	Whether to clamp selected channels to a maximum value.
MinClampTo / minClampTo	Color	r: 0 g: 0 b: 0 a: 0	The value to which values below minimum are clamped when minClampTo is enabled. Setting this to a custom color helps visualizing the clamped areas or create graphic effects.
Enable MinClampTo / minClampToEnable	Boolean	Off	When enabled, all values below minimum are set to the minClampTo value. When disabled, all values below minimum are clamped to the minimum value.
MaxClampTo / maxClampTo	Color	r: 1 g: 1 b: 1 a: 1	The value to which values above maximum are clamped when maxClampTo is enabled. Setting this to a custom color helps visualizing the clamped areas or create graphic effects.
Enable MaxClampTo / maxClampToEnable	Boolean	Off	When enabled, all values above maximum are set to the maxClampTo value. When disabled, all values above maximum are clamped to the maximum value.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.3 ClipTest node



This documentation is for version 2.0 of ClipTest.

Description

Draw zebra stripes on all pixels outside of the specified range.

See also: http://opticalenquiry.com/nuke/index.php?title=Evaluating_Color#The_ClipTest_node

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Lower / lower	Color	r: 0 g: 0 b: 0 a: 0	Highlight pixels lower than this value.
Upper / upper	Color	r: 1 g: 1 b: 1 a: 1	Highlight pixels higher than this value.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.4 ColorCorrect node



This documentation is for version 2.1 of ColorCorrect.

Description

Adjusts the saturation, contrast, gamma, gain and offset of an image.

The ranges of the shadows, midtones and highlights are controlled by the curves in the “Ranges” tab.

The Contrast adjustment works using the formula: $\text{Output} = (\text{Input}/0.18)^{\text{Contrast}} \times 0.18$.

See also:

- <http://opticalenquiry.com/nuke/index.php?title=ColorCorrect>
- <https://compositormathematic.wordpress.com/2013/07/06/gamma-contrast/>

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Saturation / MasterSaturation	Color	r: 1 g: 1 b: 1 a: 1	
Contrast / MasterContrast	Color	r: 1 g: 1 b: 1 a: 1	
Gamma / MasterGamma	Color	r: 1 g: 1 b: 1 a: 1	
Gain / MasterGain	Color	r: 1 g: 1 b: 1 a: 1	
Offset / MasterOffset	Color	r: 0 g: 0 b: 0 a: 0	
Enable / ShadowsEnable	Boolean	On	When checked, Shadows correction is enabled.
Saturation / ShadowsSaturation	Color	r: 1 g: 1 b: 1 a: 1	
Contrast / ShadowsContrast	Color	r: 1 g: 1 b: 1 a: 1	
Gamma / ShadowsGamma	Color	r: 1 g: 1 b: 1 a: 1	
Gain / ShadowsGain	Color	r: 1 g: 1 b: 1 a: 1	
Offset / ShadowsOffset	Color	r: 0 g: 0 b: 0 a: 0	
Enable / MidtonesEnable	Boolean	On	When checked, Midtones correction is enabled.
Saturation / MidtonesSaturation	Color	r: 1 g: 1 b: 1 a: 1	
Contrast / MidtonesContrast	Color	r: 1 g: 1 b: 1 a: 1	

Continued on next page

Table 50 – continued from previous page

Parameter / script name	Type	Default	Function
Gamma / MidtonesGamma	Color	r: 1 g: 1 b: 1 a: 1	
Gain / MidtonesGain	Color	r: 1 g: 1 b: 1 a: 1	
Offset / MidtonesOffset	Color	r: 0 g: 0 b: 0 a: 0	
Enable / HighlightsEnable	Boolean	On	When checked, Highlights correction is enabled.
Saturation / HighlightsSaturation	Color	r: 1 g: 1 b: 1 a: 1	
Contrast / HighlightsContrast	Color	r: 1 g: 1 b: 1 a: 1	
Gamma / HighlightsGamma	Color	r: 1 g: 1 b: 1 a: 1	
Gain / HighlightsGain	Color	r: 1 g: 1 b: 1 a: 1	
Offset / HighlightsOffset	Color	r: 0 g: 0 b: 0 a: 0	
Range / range	Double	min: 0 max: 1	Expected range for input values. Within this range, a lookup table is used for faster computation.
Tone Ranges / toneRanges	Parameter	Shadow: High-light:	Tone ranges lookup table
Luminance Math / luminanceMath	Choice	Rec. 709	<p>Formula used to compute luminance from RGB values (used for saturation adjustments).</p> <p>Rec. 709 (rec709): Use Rec. 709 (0.2126r + 0.7152g + 0.0722b).</p> <p>Rec. 2020 (rec2020): Use Rec. 2020 (0.2627r + 0.6780g + 0.0593b).</p> <p>ACES AP0 (acesap0): Use ACES AP0 (0.3439664498r + 0.7281660966g + -0.0721325464b).</p> <p>ACES AP1 (acesap1): Use ACES AP1 (0.2722287168r + 0.6740817658g + 0.0536895174b).</p> <p>CCIR 601 (ccir601): Use CCIR 601 (0.2989r + 0.5866g + 0.1145b).</p> <p>Average (average): Use average of r, g, b.</p> <p>Max (max): Use max of r, g, b.</p>
Clamp Black / clampBlack	Boolean	On	All colors below 0 on output are set to 0.
Clamp White / clampWhite	Boolean	Off	All colors above 1 on output are set to 1.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.

Continued on next page

Table 50 – continued from previous page

Parameter / script name	Type	Default	Function
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.5 ColorLookup node



This documentation is for version 1.1 of ColorLookup.

Description

Apply a parametric lookup curve with the possibility to adjust each channel separately.

The master curve is combined with the red, green and blue curves, but not with the alpha curve.

Different algorithms are available when applying the master curve, which are selectable using the “Master Curve Mode” parameter.

Computation is faster for values that are within the given range, so it is recommended to set the Range parameter if the input range goes beyond [0,1].

Note that you can easily do color remapping by setting Source and Target colors and clicking “Set RGB” or “Set RGBA” below.

This will add control points on the curve to match the target from the source. You can add as many point as you like.

This is very useful for matching color of one shot to another, or adding custom colors to a black and white ramp.

Optionally, the RGB histogram or a color ramp can be displayed in the background of the lookup curves.

See also: <http://opticalenquiry.com/nuke/index.php?title=ColorLookup>

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Range / range	Double	min: 0 max: 1	Expected range for input values. Within this range, a lookup table is used for faster computation.
Lookup Table / lookupTable	Parametric	master: red: green: blue: alpha:	Colour lookup table. The master curve is combined with the red, green and blue curves, but not with the alpha curve.

Continued on next page

Table 51 – continued from previous page

Parameter / script name	Type	Default	Function
Display / backgroundDisplay	Choice	Color Ramp	Display a color ramp or a histogram behind the curves. None (none) : No background display. Color Ramp (colorramp) : Display a color ramp. RGB Histogram (histogram) : Display the input histogram. Press “Refresh Histogram” to recompute the histogram.
Update Histogram / updateHistogram	Button		Update the histogram from the input at current time.
Source / source	Color	r: 0 g: 0 b: 0 a: 0	Source color for newly added points (x coordinate on the curve).
Target / target	Color	r: 0 g: 0 b: 0 a: 0	Target color for newly added points (y coordinate on the curve).
Set Master / setMaster	Button		Add a new control point mapping source to target to the master curve (the relative luminance is computed using the ‘Luminance Math’ parameter).
Set RGB / setRGB	Button		Add a new control point mapping source to target to the red, green, and blue curves.
Set RGBA / setRGBA	Button		Add a new control point mapping source to target to the red, green, blue and alpha curves.
Set A / setA	Button		Add a new control point mapping source to target to the alpha curve

Continued on next page

Table 51 – continued from previous page

Parameter / script name	Type	Default	Function
Master Curve Mode / <code>masterCurveMode</code>	Choice	Standard	<p>Algorithm that will be used for the master curve. The curve mode will have a strong effect on the appearance of colors, especially if you use a contrast-enhancing curve (S-curve). This can be used for creative effect, but can for some purposes or styles cause undesired color changes depending which mode you choose. Choose a mode that suits your specific taste and needs for the photo at hand. More information can be found at http://rawpedia.rawtherapee.com/Exposure</p> <p>Standard (standard): The master curve is applied independently to R, G and B channels. The drawback of this mode is that e.g. considering an S-curve shape to get more contrast, an orange color with a high value of red and green and a low value of blue will tend to shift toward yellow, because the red and green component will be raised, while the blue one will be lowered.</p> <p>Weighted Standard (weightedstandard): You can use this method to limit the color shift of the standard curve, even if it won't suppress it entirely.</p> <p>Film-Like (filmlike): The film-like curve provides a result highly similar to the standard type (that is strong saturation increase with increased contrast), but the RGB-HSV hue is kept constant - that is, there are less color-shift problems. This curve type was designed by Adobe as a part of DNG and is thus the one used by Adobe Camera Raw and Lightroom.</p> <p>Luminance (luminance): Each component of the pixel is boosted by the same factor so color and saturation is kept stable, that is the result is very true to the original color. However contrast-increasing curves can still lead to a slightly desaturated look. First the relative luminance value of a pixel is obtained, then the curve is applied to that value, the multiplication factor between before and after luminance is calculated, and then this factor is applied to each R, G and B component. The formula used to compute the luminance can be selected using the "luminanceMath" parameter.</p>
Luminance Math / <code>luminanceMath</code>	Choice	Rec. 709	<p>Formula used to compute luminance from RGB values (only used by 'Set Master').</p> <p>Rec. 709 (rec709): Use Rec. 709 ($0.2126r + 0.7152g + 0.0722b$).</p> <p>Rec. 2020 (rec2020): Use Rec. 2020 ($0.2627r + 0.6780g + 0.0593b$).</p> <p>ACES AP0 (acesap0): Use ACES AP0 ($0.3439664498r + 0.7281660966g + -0.0721325464b$).</p> <p>ACES AP1 (acesap1): Use ACES AP1 ($0.2722287168r + 0.6740817658g + 0.0536895174b$).</p> <p>CCIR 601 (ccir601): Use CCIR 601 ($0.2989r + 0.5866g + 0.1145b$).</p> <p>Average (average): Use average of r, g, b.</p> <p>Max (max): Use max of r, g, b.</p>
Clamp Black / <code>clampBlack</code>	Boolean	Off	All colors below 0 on output are set to 0.
Clamp White / <code>clampWhite</code>	Boolean	Off	All colors above 1 on output are set to 1.
(Un)premult / <code>premult</code>	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.

Continued on next page

Table 51 – continued from previous page

Parameter / script name	Type	Default	Function
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.6 ColorMatrix node

```
1 0 0
0 1 0
0 0 1
```

This documentation is for version 2.0 of ColorMatrix.

Description

Multiply the RGBA channels by an arbitrary 4x4 matrix.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Output Red / outputRed	Color	r: 1 g: 0 b: 0 a: 0	values for red output component.
Output Green / outputGreen	Color	r: 0 g: 1 b: 0 a: 0	values for green output component.
Output Blue / outputBlue	Color	r: 0 g: 0 b: 1 a: 0	values for blue output component.
Output Alpha / outputAlpha	Color	r: 0 g: 0 b: 0 a: 1	values for alpha output component.
Clamp Black / clampBlack	Boolean	On	All colors below 0 on output are set to 0.
Clamp White / clampWhite	Boolean	Off	All colors above 1 on output are set to 1.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.7 ColorSuppress node

This documentation is for version 1.0 of ColorSuppress.

Description

Remove a color or tint from an image.

The effect can either modify the color and/or extract the amount of color and store it in the alpha channel. It can be used to fix the despill or extract a mask from a color.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Red / redSuppress	Double	0	Fraction of red to suppress.
Green / greenSuppress	Double	0	Fraction of green to suppress.
Blue / blueSuppress	Double	0	Fraction of blue to suppress.
Cyan / cyanSuppress	Double	0	Fraction of cyan to suppress.
Magenta / magentaSuppress	Double	0	Fraction of magenta to suppress.
Yellow / yellowSuppress	Double	0	Fraction of yellow to suppress.
Output / outputMode	Choice	Image	<p>Suppress mode.</p> <p>Image (image): Suppress color from the image.</p> <p>Alpha (alpha): Only store the suppress mask in the Alpha channel.</p> <p>Image and Alpha (both): Suppress the color from the image and store the suppress mask in the Alpha channel.</p>
Preserve Luminance / preserveLuma	Boolean	Off	Preserve image luminosity.

Continued on next page

Table 53 – continued from previous page

Parameter / script name	Type	Default	Function
Luminance Math / luminanceMath	Choice	Rec. 709	Formula used to compute luminance from RGB values. Rec. 709 (rec709) : Use Rec. 709 ($0.2126r + 0.7152g + 0.0722b$). Rec. 2020 (rec2020) : Use Rec. 2020 ($0.2627r + 0.6780g + 0.0593b$). ACES AP0 (acesap0) : Use ACES AP0 ($0.3439664498r + 0.7281660966g + -0.0721325464b$). ACES AP1 (acesap1) : Use ACES AP1 ($0.2722287168r + 0.6740817658g + 0.0536895174b$). CCIR 601 (ccir601) : Use CCIR 601 ($0.2989r + 0.5866g + 0.1145b$). Average (average) : Use average of r, g, b. Max (max) : Use max of r, g, b.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.8 Equalize node



This documentation is for version 2.0 of Equalize.

Description

Equalize histogram of pixel values.

To equalize image brightness only, use the HistEQCImg plugin.

Uses the 'equalize' function from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
NbLevels / nb_levels	Integer	4096	Number of histogram levels used for the equalization.

Continued on next page

Table 54 – continued from previous page

Parameter / script name	Type	Default	Function
Min Value / min_value	Double	0	Minimum pixel value considered for the histogram computation. All pixel values lower than min_value will not be counted.
Max Value / max_value	Double	1	Maximum pixel value considered for the histogram computation. All pixel values higher than max_value will not be counted.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.9 Gamma node



This documentation is for version 2.0 of Gamma.

Description

Apply gamma function to the selected channels. The actual function is $\text{pow}(x, 1/\max(1e-8, \text{value}))$.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Value / value	Color	r: 1 g: 1 b: 1 a: 1	Gamma value to apply to the selected channels.
Invert / invert	Boolean	Off	Invert the gamma transform.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.10 Grade node



This documentation is for version 2.0 of Grade.

Description

Modify the tonal spread of an image from the white and black points.

This node can also be used to match colors of 2 images: The darkest and lightest points of the target image are converted to black and white using the blackpoint and whitepoint values. These 2 values are then moved to new values using the black(for dark point) and white(for white point). You can also apply multiply/offset/gamma for other color fixing you may need.

Here is the formula used:

$A = \text{multiply} * (\text{white} - \text{black}) / (\text{whitepoint} - \text{blackpoint})$

$B = \text{offset} + \text{black} - A * \text{blackpoint}$

$\text{output} = \text{pow}(A * \text{input} + B, 1 / \text{gamma}).$

A special use for Grade is to generate a mask image with soft edges by thresholding an input image. Set the “Black Point” and “White Point” to values just below and just above the threshold, and check the “Clamp Black” and “Clamp White” options. If a binary mask containing only 0 and 1 is preferred, the Clamp plugin can be used instead.

See also: <http://opticalenquiry.com/nuke/index.php?title=Grade> and http://opticalenquiry.com/nuke/index.php?title=Integration#Matching_color

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Black Point / blackPoint	Color	r: 0 g: 0 b: 0 a: 0	Set the color of the darkest pixels in the image.
White Point / whitePoint	Color	r: 1 g: 1 b: 1 a: 1	Set the color of the brightest pixels in the image.
Black / black	Color	r: 0 g: 0 b: 0 a: 0	Colors corresponding to the blackpoint are set to this value.
White / white	Color	r: 1 g: 1 b: 1 a: 1	Colors corresponding to the whitepoint are set to this value.
Multiply / multiply	Color	r: 1 g: 1 b: 1 a: 1	Multiplies the result by this value.
Offset / offset	Color	r: 0 g: 0 b: 0 a: 0	Adds this value to the result (this applies to black and white).

Continued on next page

Table 56 – continued from previous page

Parameter / script name	Type	Default	Function
Gamma / gamma	Color	r: 1 g: 1 b: 1 a: 1	Final gamma correction. Negative values are not affected by gamma.
Normalize / normalize	Button		Normalize the image by setting the white point and black point from the minimum and maximum values of the input.
Reverse / reverse	Boolean	Off	Apply the inverse correction. Useful to apply the inverse of a Grade downstream: copy-and-paste or clone the upstream node, and invert the downstream one.
Clamp Black / clampBlack	Boolean	On	All colors below 0 on output are set to 0.
Clamp White / clampWhite	Boolean	Off	All colors above 1 on output are set to 1.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.11 HSIToRGB node

This documentation is for version 1.0 of HSIToRGB.

Description

Convert from HSI color model (hue, saturation, intensity, as defined by Gonzalez and Woods in 1992) to linear RGB. H is in degrees, S and I are in the same units as RGB. No gamma correction is applied to RGB after conversion.

The HSI colour space (hue, saturation and intensity) attempts to produce a more intuitive representation of colour. The I axis represents the luminance information. The H and S axes are polar coordinates on the plane orthogonal to I. H is the angle, specified such that red is at zero, green at 120 degrees, and blue at 240 degrees. Hue thus represents what humans implicitly understand as colour. S is the magnitude of the colour vector projected in the plane orthogonal to I, and so represents the difference between pastel colours (low saturation) and vibrant colours (high saturation). The main drawback of this colour space is that hue is undefined if saturation is zero, making error propagation in transformations from the RGB colour space more complicated.

It should also be noted that, although the HSI colour space may be more intuitive, is not “perceptual”, in the sense that small displacements of equal size in different parts of the colour space will be perceived by human observers as changes of different magnitude. Attempts have been made to define such colour spaces: CIE-LAB and CIE-LUV are two examples.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Premult / premult	Boolean	Off	Multiply the image by the alpha channel after processing. Use to get premultiplied output images.

2.6.12 HSLToRGB node

This documentation is for version 1.0 of HSLToRGB.

Description

Convert from HSL color model (hue, saturation, lightness, as defined by Joblove and Greenberg in 1978) to linear RGB. H is in degrees, S and L are in the same units as RGB. No gamma correction is applied to RGB after conversion.

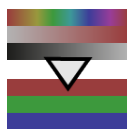
Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Premult / premult	Boolean	Off	Multiply the image by the alpha channel after processing. Use to get premultiplied output images.

2.6.13 HSVToRGB node



This documentation is for version 1.0 of HSVToRGB.

Description

Convert from HSV color model (hue, saturation, value, as defined by A. R. Smith in 1978) to linear RGB. H is in degrees, S and V are in the same units as RGB. No gamma correction is applied to RGB after conversion.

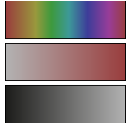
Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Premult / premult	Boolean	Off	Multiply the image by the alpha channel after processing. Use to get premultiplied output images.

2.6.14 HSVTool node



This documentation is for version 1.0 of HSVTool.

Description

Adjust hue, saturation and brightness, or perform color replacement.

Color replacement:

Set the srcColor and dstColor parameters. The range of the replacement is determined by the three groups of parameters: Hue, Saturation and Brightness.

Color adjust:

Use the Rotation of the Hue parameter and the Adjustment of the Saturation and Lightness. The ranges and falloff parameters allow for more complex adjustments.

Hue keyer:

Set the outputAlpha parameter (the last one) to All (the default is Hue), and use a viewer to display the Alpha channel. First, set the Range parameter of the Hue parameter set and then work down the other Ranges parameters, tuning with the range Falloff and Adjustment parameters.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Src Analysis Rectangle / enableRectangle	Boolean	Off	Enable the rectangle interact for analysis of Src and Dst colors and ranges.
Bottom Left / bottomLeft	Double	x: 0.25 y: 0.25	Coordinates of the bottom left corner of the rectangle
Size / size	Double	w: 0.5 h: 0.5	Width and height of the rectangle
Set Src from Rectangle / setSrcFromRectangle	Button		Set the Src color and ranges and the adjustments from the colors of the source image within the selection rectangle and the Dst Color.

Continued on next page

Table 60 – continued from previous page

Parameter / script name	Type	Default	Function
Src Color / <code>srcColor</code>	Color	r: 0 g: 0 b: 0	Source color for replacement. Changing this parameter sets the hue, saturation and brightness ranges for this color, and sets the falloffs to default values.
Dst Color / <code>dstColor</code>	Color	r: 0 g: 0 b: 0	Destination color for replacement. Changing this parameter sets the hue rotation, and saturation and brightness adjustments. Should be set after Src Color.
Hue Range / <code>hueRange</code>	Double	: 0 : 360	Range of color hues that are modified (in degrees). Red is 0, green is 120, blue is 240. The affected hue range is the smallest interval. For example, if the range is (12, 348), then the selected range is red plus or minus 12 degrees. Exception: if the range width is exactly 360, then all hues are modified.
Hue Rotation / <code>hueRotation</code>	Double	0	Rotation of color hues (in degrees) within the range.
Hue Rotation Gain / <code>hueRotationGain</code>	Double	1	Factor to be applied to the rotation of color hues (in degrees) within the range. A value of 0 will set all values within range to a constant (computed at the center of the range), and a value of 1 will add <code>hueRotation</code> to all values within range.
Hue Range Rolloff / <code>hueRangeRolloff</code>	Double	0	Interval (in degrees) around Hue Range, where hue rotation decreases progressively to zero.
Saturation Range / <code>saturationRange</code>	Double	: 0 : 1	Range of color saturations that are modified.
Saturation Adjustment / <code>saturationAdjustment</code>	Double	0	Adjustment of color saturations within the range. Saturation is clamped to zero to avoid color inversions.
Saturation Adjustment Gain / <code>saturationAdjustmentGain</code>	Double	1	Factor to be applied to the saturation adjustment within the range. A value of 0 will set all values within range to a constant (computed at the center of the range), and a value of 1 will add <code>saturationAdjustment</code> to all values within range.
Saturation Range Rolloff / <code>saturationRangeRolloff</code>	Double	0	Interval (in degrees) around Saturation Range, where saturation rotation decreases progressively to zero.
Brightness Range / <code>brightnessRange</code>	Double	: 0 : 1	Range of color brightnesss that are modified.
Brightness Adjustment / <code>brightnessAdjustment</code>	Double	0	Adjustment of color brightnesss within the range.
Brightness Adjustment Gain / <code>brightnessAdjustmentGain</code>	Double	1	Factor to be applied to the brightness adjustment within the range. A value of 0 will set all values within range to a constant (computed at the center of the range), and a value of 1 will add <code>brightnessAdjustment</code> to all values within range.
Brightness Range Rolloff / <code>brightnessRangeRolloff</code>	Double	0	Interval (in degrees) around Brightness Range, where brightness rotation decreases progressively to zero.
Clamp Black / <code>clampBlack</code>	Boolean	On	All colors below 0 on output are set to 0.
Clamp White / <code>clampWhite</code>	Boolean	Off	All colors above 1 on output are set to 1.

Continued on next page

Table 60 – continued from previous page

Parameter / script name	Type	Default	Function
Output Alpha / outputAlpha	Choice	Hue	<p>Output alpha channel. This can either be the source alpha, one of the coefficients for hue, saturation, brightness, or a combination of those. If it is not source alpha, the image on output are unpremultiplied, even if input is premultiplied.</p> <p>Source (source): Alpha channel is kept unmodified.</p> <p>Hue (hue): Set Alpha to the Hue modification mask.</p> <p>Saturation (saturation): Set Alpha to the Saturation modification mask.</p> <p>Brightness (brightness): Alpha is set to the Brightness mask.</p> <p>min(Hue,Saturation) (minhuesaturation): Alpha is set to min(Hue mask,Saturation mask)</p> <p>min(Hue,Brightness) (minhuebrightness): Alpha is set to min(Hue mask,Brightness mask)</p> <p>min(Saturation,Brightness) (minsaturationbrightness): Alpha is set to min(Saturation mask,Brightness mask)</p> <p>min(all) (min): Alpha is set to min(Hue mask,Saturation mask,Brightness mask)</p>
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.15 HistEQ node



This documentation is for version 2.0 of HistEQ.

Description

Equalize histogram of brightness values.

Uses the ‘equalize’ function from the CImg library on the ‘V’ channel of the HSV decomposition of the image.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

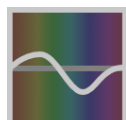
Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
NbLevels / nb_levels	Integer	4096	Number of histogram levels used for the equalization.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.16 HueCorrect node



This documentation is for version 1.0 of HueCorrect.

Description

Apply hue-dependent color adjustments using lookup curves.

Hue and saturation are computed from the the source RGB values. Depending on the hue value, the various adjustment values are computed, and then applied:

sat: saturation gain. This modification is applied last.

lum: luminance gain

red: red gain

green: green gain

blue: blue gain

r_sup: red suppression. If $r > \min(g,b)$, $r = \min(g,b) + r_sup * (r - \min(g,b))$

g_sup: green suppression

b_sup: blue suppression

sat_thrsh: if source saturation is below this value, do not apply the lum, red, green, blue gains. Above this value, apply gain progressively.

The 'Luminance Mix' parameter may be used to restore partially or fully the original luminance (luminance is computed using the 'Luminance Math' parameter).

See also: <http://opticalenquiry.com/nuke/index.php?title=HueCorrect>

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Hue Curves / hue	Parameter	sat: lum: red: green: blue: r_sup: g_sup: b_sup: sat_thrsh:	Hue-dependent adjustment lookup curves: sat: saturation gain. This modification is applied last. lum: luminance gain red: red gain green: green gain blue: blue gain r_sup: red suppression. If $r > \min(g,b)$, $r = \min(g,b) + r_sup * (r - \min(g,b))$ g_sup: green suppression b_sup: blue suppression sat_thrsh: if source saturation is below this value, do not apply the lum, red, green, blue gains. Above this value, apply gain progressively.
Luminance Math / luminanceMath	Choice	Rec. 709	Formula used to compute luminance from RGB values (only used by 'Set Master'). Rec. 709 (rec709) : Use Rec. 709 ($0.2126r + 0.7152g + 0.0722b$). Rec. 2020 (rec2020) : Use Rec. 2020 ($0.2627r + 0.6780g + 0.0593b$). ACES AP0 (acesap0) : Use ACES AP0 ($0.3439664498r + 0.7281660966g + -0.0721325464b$). ACES AP1 (acesap1) : Use ACES AP1 ($0.2722287168r + 0.6740817658g + 0.0536895174b$). CCIR 601 (ccir601) : Use CCIR 601 ($0.2989r + 0.5866g + 0.1145b$). Average (average) : Use average of r, g, b. Max (max) : Use max of r, g, b.
Clamp Black / clampBlack	Boolean	Off	All colors below 0 on output are set to 0.
Clamp White / clampWhite	Boolean	Off	All colors above 1 on output are set to 1.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Mix Luminance / mixLuminanceEnable	Boolean	On	Mix luminance
/mixLuminance	Double	0	Mix luminance
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.17 Invert node



This documentation is for version 2.0 of Invert.

Description

Inverse the selected channels

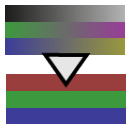
Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.18 LabToRGB709 node



This documentation is for version 1.0 of LabToRGB709.

Description

Convert from L*a*b color model to RGB (Rec.709 with D65 illuminant). L*a*b coordinates are divided by 100 for better visualization.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Premult / premult	Boolean	Off	Multiply the image by the alpha channel after processing. Use to get premultiplied output images.

2.6.19 LabToXYZ node

This documentation is for version 1.0 of LabToXYZ.

Description

Convert from CIE L*a*b color space to CIE XYZ color space. L*a*b coordinates are divided by 100 for better visualization.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function

2.6.20 Log2Lin node

This documentation is for version 1.0 of Log2Lin.

Description

Convert between the logarithmic encoding used in Cineon files and linear encoding.

This plugin may be used to customize the conversion between the linear and the logarithmic space, using different parameters than the Kodak-recommended settings.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Operation / operation	Choice	Log to Lin	The operation to perform. Log to Lin (log2lin) : Convert the input from logarithmic to linear colorspace (usually after a Read node). Lin to Log (lin2log) : Convert the input from linear to logarithmic colorspace (usually before a Write node).
Black / black	Color	r: 95 g: 95 b: 95	Value in the Cineon file that corresponds to black.
White / white	Color	r: 685 g: 685 b: 685	Value in the Cineon file that corresponds to white.
Gamma / gamma	Color	r: 0.6 g: 0.6 b: 0.6	The film response gamma value.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.

Continued on next page

Table 66 – continued from previous page

Parameter / script name	Type	Default	Function
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.21 Multiply node



This documentation is for version 2.0 of Multiply.

Description

Multiply the selected channels by a constant.

See also: <http://opticalenquiry.com/nuke/index.php?title=Multiply>

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Value / value	Color	r: 1 g: 1 b: 1 a: 1	Constant to multiply with the selected channels.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.22 OCIOCDLTransform node



This documentation is for version 1.0 of OCIOCDLTransform.

Description

Use OpenColorIO to apply an ASC Color Decision List (CDL) grade.

The formula applied for each channel is:

$$\text{out} = (\text{in} * \text{slope} + \text{offset})^{\text{power}}.$$

The saturation is then applied to all channel using the standard rec709 saturation coefficients:

$$\text{luma} = 0.2126 * \text{inR} + 0.7152 * \text{inG} + 0.0722 * \text{inB}$$

$$\text{outR} = \text{Clamp}(\text{luma} + \text{sat} * (\text{inR} - \text{luma}))$$

$$\text{outG} = \text{Clamp}(\text{luma} + \text{sat} * (\text{inG} - \text{luma}))$$

$$\text{outB} = \text{Clamp}(\text{luma} + \text{sat} * (\text{inB} - \text{luma})).$$

The grade can be loaded from an ASC .ccc (Color Correction Collection) or .cc (Color Correction) file.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Slope / slope	Color	r: 1 g: 1 b: 1	ASC CDL slope
Offset / offset	Color	r: 0 g: 0 b: 0	ASC CDL offset
Power / power	Color	r: 1 g: 1 b: 1	ASC CDL power
Saturation / saturation	Double	1	ASC CDL saturation
Direction / direction	Choice	Forward	Transform direction. Forward (forward) Inverse (inverse)
Read from file / readFromFile	Boolean	Off	Load color correction information from the .cc or .ccc file.
File / file	N/A		Specify the src ASC CDL file, on disk, to use for this transform. This can be either a .cc or .ccc file. If .ccc is specified, the cccid is required.
Reload / reload	Button		Reloads specified files
CCC Id / cccId	String		If the source file is an ASC CDL CCC (color correction collection), this specifies the id to lookup. OpenColorIO::Contexts (envvars) are obeyed.
Export / export	N/A		Export this grade as a ColorCorrection XML file (.cc), which can be loaded with the OCIOFileTransform, or using a FileTransform in an OCIO config. The file must not already exist.

Continued on next page

Table 68 – continued from previous page

Parameter / script name	Type	Default	Function
Enable GPU Render / enableGPU	Boolean	Off	Enable GPU-based OpenGL render. Note that GPU render is not as accurate as CPU render, so this should be enabled with care. If the checkbox is checked but is not enabled (i.e. it cannot be unchecked), GPU render can not be enabled or disabled from the plugin and is probably part of the host options. If the checkbox is not checked and is not enabled (i.e. it cannot be checked), GPU render is not available on this host.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.23 OCIOColorSpace node



This documentation is for version 1.0 of OCIOColorSpace.

Description

ColorSpace transformation using OpenColorIO configuration file.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
Input Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Output Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.

Continued on next page

Table 69 – continued from previous page

Parameter / script name	Type	Default	Function
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 69 – continued from previous page

Parameter / script name	Type	Default	Function
value3 / value3	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
key4 / key4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
value4 / value4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.
Enable GPU Render / enableGPU	Boolean	Off	Enable GPU-based OpenGL render. Note that GPU render is not as accurate as CPU render, so this should be enabled with care. If the checkbox is checked but is not enabled (i.e. it cannot be unchecked), GPU render can not be enabled or disabled from the plugin and is probably part of the host options. If the checkbox is not checked and is not enabled (i.e. it cannot be checked), GPU render is not available on this host.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.24 OCIODisplay node



This documentation is for version 1.0 of OCIODisplay.

Description

Uses the OpenColorIO library to apply a colorspace conversion to an image sequence, so that it can be accurately represented on a specific display device.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
Input Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Display Device / displayIndex	Choice		Specifies the display device that will be used to view the sequence.
View Transform / viewIndex	Choice		Specifies the display transform to apply to the scene or image.
Gain / gain	Double	1	Exposure adjustment, in scene-linear, prior to the display transform.
Gamma / gamma	Double	1	Gamma correction applied after the display transform.
Channel View / channelSelector	Choice	RGB	Specify which channels to view (prior to the display transform). RGB (rgb) : Color. R (r) : Red. G (g) : Green. B (b) : Blue. A (a) : Alpha. Luminance (l) : Luma
Enable GPU Render / enableGPU	Boolean	Off	Enable GPU-based OpenGL render. Note that GPU render is not as accurate as CPU render, so this should be enabled with care. If the checkbox is checked but is not enabled (i.e. it cannot be unchecked), GPU render can not be enabled or disabled from the plugin and is probably part of the host options. If the checkbox is not checked and is not enabled (i.e. it cannot be checked), GPU render is not available on this host.
key1 / key1	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html

Continued on next page

Table 70 – continued from previous page

Parameter / script name	Type	Default	Function
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value3 / value3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 70 – continued from previous page

Parameter / script name	Type	Default	Function
key4 / key4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
value4 / value4	String		OCIO Contexts allow you to apply specific LUTs or grades to different shots. Here you can specify the context name (key) and its corresponding value. Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html
OCIO config help... / ocioHelpDisplays	Button		Help about the OpenColorIO configuration.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.

2.6.25 OCIOFileTransform node



This documentation is for version 1.0 of OCIOFileTransform.

Description

Use OpenColorIO to apply a transform loaded from the given file.

This is usually a 1D or 3D LUT file, but can be other file-based transform, for example an ASC ColorCorrection XML file.

Note that the file's transform is applied with no special input/output colorspace handling - so if the file expects log-encoded pixels, but you apply the node to a linear image, you will get incorrect results.

Supported formats:

- .3dl (flame)
- .3dl (lustre)
- .ccc (ColorCorrectionCollection)
- .cdl (ColorDecisionList)
- .cc (ColorCorrection)
- .csp (cinespace)
- .lut (houdini)
- .itx (iridas_itx)

.cube (iridas_cube)
 .look (iridas_look)
 .mga (pandora_mga)
 .m3d (pandora_m3d)
 .spi1d (spi1d)
 .spi3d (spi3d)
 .spimtx (spimtx)
 .cub (truelight)
 .vf (nukevf)

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
File / file	N/A		<p>File containing the transform.</p> <p>Supported formats:</p> <p>.3dl (flame) .3dl (lustre) .ccc (ColorCorrectionCollection) .cdl (ColorDecisionList) .cc (ColorCorrection) .csp (cinespace) .lut (houdini) .itx (iridas_itx) .cube (iridas_cube) .look (iridas_look) .mga (pandora_mga) .m3d (pandora_m3d) .spi1d (spi1d) .spi3d (spi3d) .spimtx (spimtx) .cub (truelight) .vf (nukevf)</p>
Reload / reload	Button		Reloads specified files

Continued on next page

Table 71 – continued from previous page

Parameter / script name	Type	Default	Function
Direction / direction	Choice	Forward	Transform direction. Forward (forward) Inverse (inverse)
Interpolation / interpolation	Choice	Linear	Interpolation method. For files that are not LUTs (mtx, etc) this is ignored. Nearest (nearest) Linear (linear) Tetrahedral (tetrahedral) Best (best)
Enable GPU Render / enableGPU	Boolean	Off	Enable GPU-based OpenGL render. Note that GPU render is not as accurate as CPU render, so this should be enabled with care. If the checkbox is checked but is not enabled (i.e. it cannot be unchecked), GPU render can not be enabled or disabled from the plugin and is probably part of the host options. If the checkbox is not checked and is not enabled (i.e. it cannot be checked), GPU render is not available on this host.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.26 OCIOLogConvert node



This documentation is for version 1.0 of OCIOLogConvert.

Description

Use OpenColorIO to convert from SCENE_LINEAR to COMPOSITING_LOG (or back).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
OCIO config help... / ocioHelp	Button		Help about the OpenColorIO configuration.
Operation / operation	Choice	Log to Lin	Operation to perform. Lin is the SCENE_LINEAR profile and Log is the COMPOSITING_LOG profile of the OCIO configuration. Log to Lin (log2lin) Lin to Log (lin2log)
Enable GPU Render / enableGPU	Boolean	Off	Enable GPU-based OpenGL render. Note that GPU render is not as accurate as CPU render, so this should be enabled with care. If the checkbox is checked but is not enabled (i.e. it cannot be unchecked), GPU render can not be enabled or disabled from the plugin and is probably part of the host options. If the checkbox is not checked and is not enabled (i.e. it cannot be checked), GPU render is not available on this host.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.27 OCIOLookTransform node



This documentation is for version 1.0 of OCIOLookTransform.

Description

OpenColorIO LookTransform

A ‘look’ is a named color transform, intended to modify the look of an image in a ‘creative’ manner (as opposed to a colorspace definition which tends to be technically/mathematically defined).

Examples of looks may be a neutral grade, to be applied to film scans prior to VFX work, or a per-shot DI grade decided on by the director, to be applied just before the viewing transform.

OCIOLooks must be predefined in the OpenColorIO configuration before usage, and often reference per-shot/sequence LUTs/CCs.

See the ‘Look Combination’ parameter for further syntax details.

See opencolorio.org for look configuration customization examples.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
OCIO Config File / ocioConfigFile	N/A		OpenColorIO configuration file
Input Colorspace / ocioInputSpaceIndex	Choice		Input data is taken to be in this colorspace.
Single Look / singleLook	Boolean	On	When checked, only the selected Look is applied. When not checked, the Look Combination is applied.

Continued on next page

Table 73 – continued from previous page

Parameter / script name	Type	Default	Function
Look / lookChoice	Choice	Filmic - Very High Contrast	<p>Look to apply (if “Single Look” is checked) or append to the Look Combination (when the “Append” button is pressed).</p> <p>Filmic - Very High Contrast</p> <p>Filmic - High Contrast</p> <p>Filmic - Medium High Contrast</p> <p>Filmic - Base Contrast</p> <p>Filmic - Medium Low Contrast</p> <p>Filmic - Low Contrast</p> <p>Filmic - Very Low Contrast</p> <p>Agfa Agfacolor Futura 100</p> <p>Agfa Agfacolor Futura 200</p> <p>Agfa Agfacolor Futura 400</p> <p>Agfa Agfacolor Futura II 100</p> <p>Agfa Agfacolor Futura II 200</p> <p>Agfa Agfacolor Futura II 400</p> <p>Agfa Agfacolor HDC 100 plus</p> <p>Agfa Agfacolor HDC 400 plus</p> <p>Agfa Agfacolor HDC 200 plus</p> <p>Agfa Agfacolor Optima II 100</p> <p>Agfa Agfacolor Optima II 200</p> <p>Agfa Agfacolor Ultra 050</p> <p>Agfa Agfacolor Vista 100</p> <p>Agfa Agfacolor Vista 200</p> <p>Agfa Agfacolor Vista 400</p> <p>Agfa Agfacolor Vista 800</p> <p>Agfa Agfachrome CT Precisa 100</p> <p>Agfa Agfachrome CT Precisa 200</p> <p>Agfa Agfachrome RSX2 050</p> <p>Agfa Agfachrome RSX2 100</p> <p>Agfa Agfachrome RSX2 200</p> <p>Agfa Advantix 100</p> <p>Agfa Advantix 200</p> <p>Agfa Advantix 400</p> <p>Kodak Gold 100</p> <p>Kodak Gold 200</p> <p>Kodak Max Zoom 800</p> <p>Kodak Portra 100T</p> <p>Kodak Portra 160NC</p> <p>Kodak Portra 160VC</p> <p>Kodak Portra 800</p> <p>Kodak Portra 400VC</p> <p>Kodak Portra 400NC</p> <p>Kodak Ektachrome 100 plus</p> <p>Kodak Ektachrome 320T</p> <p>Kodak Ektachrome 400X</p> <p>Kodak Ektachrome 64</p> <p>Kodak Ektachrome 64T</p> <p>Kodak Ektachrome E100S</p> <p>Kodak Ektachrome 100</p>
238			<p>Kodak Kodachrome 200</p> <p>Kodak Kodachrome 25</p> <p>Kodak Kodachrome 64</p> <p>Kodak DSCS 3151</p>

Table 73 – continued from previous page

Parameter / script name	Type	Default	Function
Append Look to Combination / append	Button		Append the selected Look to the Look Combination
Look Combination / lookCombination	String		<p>Specify the look(s) to apply.</p> <p>This may be empty, the name of a single look, or a combination of looks using the ‘look syntax’.</p> <p>If it is empty, no look is applied.</p> <p>Look Syntax:</p> <p>Multiple looks are combined with commas: ‘firstlook, secondlook’</p> <p>Direction is specified with +/- prefixes: ‘+firstlook, -secondlook’</p> <p>Missing look ‘fallbacks’ specified with : ‘firstlook, -secondlook -secondlook’</p>
Direction / direction	Choice	Forward	<p>Transform direction.</p> <p>Forward (forward)</p> <p>Inverse (inverse)</p>
Output Colorspace / ocioOutputSpaceIndex	Choice		Output data is taken to be in this colorspace.
key1 / key1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation:</p> <p>http://opencolorio.org/userguide/contexts.html</p>
value1 / value1	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation:</p> <p>http://opencolorio.org/userguide/contexts.html</p>
key2 / key2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation:</p> <p>http://opencolorio.org/userguide/contexts.html</p>

Continued on next page

Table 73 – continued from previous page

Parameter / script name	Type	Default	Function
value2 / value2	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key3 / key3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value3 / value3	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
key4 / key4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
value4 / value4	String		<p>OCIO Contexts allow you to apply specific LUTs or grades to different shots.</p> <p>Here you can specify the context name (key) and its corresponding value.</p> <p>Full details of how to set up contexts and add them to your config can be found in the OpenColorIO documentation: http://opencolorio.org/userguide/contexts.html</p>
OCIO config help... / ocioHelpLooks	Button		Help about the OpenColorIO configuration.

Continued on next page

Table 73 – continued from previous page

Parameter / script name	Type	Default	Function
Enable GPU Render / enableGPU	Boolean	Off	Enable GPU-based OpenGL render. Note that GPU render is not as accurate as CPU render, so this should be enabled with care. If the checkbox is checked but is not enabled (i.e. it cannot be unchecked), GPU render can not be enabled or disabled from the plugin and is probably part of the host options. If the checkbox is not checked and is not enabled (i.e. it cannot be checked), GPU render is not available on this host.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.28 PLogLin node

This documentation is for version 1.0 of PLogLin.

Description

Convert between logarithmic and linear encoding.

This method uses the so-called “Josh Pines log conversion” or “printing density transform” (as described in <http://lists.gnu.org/archive/html/openexr-devel/2005-03/msg00006.html>), which is based on a single gray point, rather than the white and black points in the Cineon formula (as implemented in the Log2Lin plugin).

Log to Lin conversion: $x_{Lin} = linRef * pow(10.0, (x_{Log} * 1023. - logRef) * density / nGamma)$

Lin to Log conversion: $x_{Log} = (logRef + log10(max(x_{Lin}, 1e-10) / linRef) * nGamma / density) / 1023.$

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Operation / operation	Choice	Log to Lin	The operation to perform. Log to Lin (log2lin) : Convert the input from logarithmic to linear colorspace (usually after a Read node). Lin to Log (lin2log) : Convert the input from linear to logarithmic colorspace (usually before a Write node).

Continued on next page

Table 74 – continued from previous page

Parameter / script name	Type	Default	Function
Linear Reference / <code>linRef</code>	Color	r: 0.18 g: 0.18 b: 0.18	Linear value of the reference gray point. Set this to the linear value that corresponds with the log reference value.
Log Reference / <code>logRef</code>	Color	r: 445 g: 445 b: 445	Log value of the reference gray point. Set this to the log value that corresponds with the lin reference value.
Negative Gamma / <code>nGamma</code>	Color	r: 0.6 g: 0.6 b: 0.6	The film response gamma value.
Density / <code>density</code>	Color	r: 0.002 g: 0.002 b: 0.002	Density per code value. The change in the negative gamma for each log space code value. This is usually left to the default value of 0.002.
(Un)premult / <code>premult</code>	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / <code>maskInvert</code>	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / <code>mix</code>	Double	1	Mix factor between the original and the transformed image.

2.6.29 Quantize node

This documentation is for version 1.0 of Quantize.

Description

Reduce the number of color levels per channel.

See also: http://opticalenquiry.com/nuke/index.php?title=Color_Operation

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

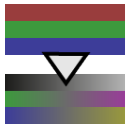
Parameter / script name	Type	Default	Function
Colors / <code>colors</code>	Double	16	Number of color levels to use per channel.

Continued on next page

Table 75 – continued from previous page

Parameter / script name	Type	Default	Function
Dither / dither	Choice	None	<p>Dithering method to apply in order to avoid the banding effect.</p> <p>None (none): No dithering (posterize), creating abrupt changes.</p> <p>Ordered (Bayer 2x2) (bayer2x2): Ordered dithering using a 2x2 Bayer matrix.</p> <p>Ordered (Bayer 4x4) (bayer4x4): Ordered dithering using a 4x4 Bayer matrix.</p> <p>Ordered (Bayer 8x8) (bayer8x8): Ordered dithering using a 8x8 Bayer matrix.</p> <p>Ordered (void-and-cluster 14x14) (vac14x14): Ordered dithering using a void-and-cluster 14x14 matrix.</p> <p>Ordered (void-and-cluster 25x25) (vac25x25): Ordered dithering using a void-and-cluster 25x25 matrix.</p> <p>Random (random): Random dithering.</p>
Seed / seed	Integer	2000	Random seed: change this if you want different instances to have different dithering (only for random dithering).
Static Seed / staticSeed	Boolean	Off	When enabled, the dither pattern remains the same for every frame producing a constant dither effect.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.30 RGB709ToLab node



This documentation is for version 1.0 of RGB709ToLab.

Description

Convert from RGB (Rec.709 with D65 illuminant) to L*a*b color model. L*a*b coordinates are divided by 100 for better visualization.

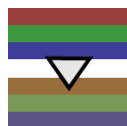
Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Unpremult / premult	Boolean	Off	Divide the image by the alpha channel before processing. Use if the input images are premultiplied.

2.6.31 RGB709ToXYZ node



This documentation is for version 1.0 of RGB709ToXYZ.

Description

Convert from RGB (Rec.709 with D65 illuminant) to XYZ color model. X, Y and Z are in the same units as RGB.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Unpremult / premult	Boolean	Off	Divide the image by the alpha channel before processing. Use if the input images are premultiplied.

2.6.32 RGBToHSI node

This documentation is for version 1.0 of RGBToHSI.

Description

Convert from linear RGB to HSI color model (hue, saturation, intensity, as defined by Gonzalez and Woods in 1992). H is in degrees, S and I are in the same units as RGB. No gamma correction is applied to RGB before conversion.

The HSI colour space (hue, saturation and intensity) attempts to produce a more intuitive representation of colour. The I axis represents the luminance information. The H and S axes are polar coordinates on the plane orthogonal to I. H is the angle, specified such that red is at zero, green at 120 degrees, and blue at 240 degrees. Hue thus represents what humans implicitly understand as colour. S is the magnitude of the colour vector projected in the plane orthogonal to I, and so represents the difference between pastel colours (low saturation) and vibrant colours (high saturation). The main drawback of this colour space is that hue is undefined if saturation is zero, making error propagation in transformations from the RGB colour space more complicated.

It should also be noted that, although the HSI colour space may be more intuitive, is not “perceptual”, in the sense that small displacements of equal size in different parts of the colour space will be perceived by human observers as changes of different magnitude. Attempts have been made to define such colour spaces: CIE-LAB and CIE-LUV are two examples.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Unpremult / premult	Boolean	Off	Divide the image by the alpha channel before processing. Use if the input images are premultiplied.

2.6.33 RGBToHSL node

This documentation is for version 1.0 of RGBToHSL.

Description

Convert from RGB to HSL color model (hue, saturation, lightness, as defined by Joblove and Greenberg in 1978). H is in degrees, S and L are in the same units as RGB. No gamma correction is applied to RGB before conversion.

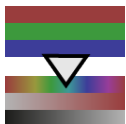
Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Unpremult / premult	Boolean	Off	Divide the image by the alpha channel before processing. Use if the input images are premultiplied.

2.6.34 RGBToHSV node



This documentation is for version 1.0 of RGBToHSV.

Description

Convert from linear RGB to HSV color model (hue, saturation, value, as defined by A. R. Smith in 1978). H is in degrees, S and V are in the same units as RGB. No gamma correction is applied to RGB before conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Unpremult / premult	Boolean	Off	Divide the image by the alpha channel before processing. Use if the input images are premultiplied.

2.6.35 RGBToYCbCr601 node

This documentation is for version 1.0 of RGBToYCbCr601.

Description

Convert from linear RGB to YCbCr color model (ITU.BT-601). RGB is gamma-compressed using the sRGB Opto-Electronic Transfer Function (OETF) before conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Unpremult / premult	Boolean	Off	Divide the image by the alpha channel before processing. Use if the input images are premultiplied.

2.6.36 RGBToYCbCr709 node

This documentation is for version 1.0 of RGBToYCbCr709.

Description

Convert from linear RGB to YCbCr color model (ITU.BT-709). RGB is gamma-compressed using the Rec.709 Opto-Electronic Transfer Function (OETF) before conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Unpremult / premult	Boolean	Off	Divide the image by the alpha channel before processing. Use if the input images are premultiplied.

2.6.37 RGBToYPbPr601 node

This documentation is for version 1.0 of RGBToYPbPr601.

Description

Convert from RGB to YPbPr color model (ITU.BT-601). RGB is gamma-compressed using the sRGB Opto-Electronic Transfer Function (OETF) before conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Unpremult / premult	Boolean	Off	Divide the image by the alpha channel before processing. Use if the input images are premultiplied.

2.6.38 RGBToYPbPr709 node

This documentation is for version 1.0 of RGBToYPbPr709.

Description

Convert from RGB to YPbPr color model (ITU.BT-709). RGB is gamma-compressed using the Rec.709 Opto-Electronic Transfer Function (OETF) before conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Unpremult / premult	Boolean	Off	Divide the image by the alpha channel before processing. Use if the input images are premultiplied.

2.6.39 RGBToYUV601 node

This documentation is for version 1.0 of RGBToYUV601.

Description

Convert from RGB to YUV color model (ITU.BT-601). RGB is gamma-compressed using the sRGB Opto-Electronic Transfer Function (OETF) before conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Unpremult / premult	Boolean	Off	Divide the image by the alpha channel before processing. Use if the input images are premultiplied.

2.6.40 RGBToYUV709 node

This documentation is for version 1.0 of RGBToYUV709.

Description

Convert from RGB to YUV color model (ITU.BT-709). RGB is gamma-compressed using the Rec.709 Opto-Electronic Transfer Function (OETF) before conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Unpremult / premult	Boolean	Off	Divide the image by the alpha channel before processing. Use if the input images are premultiplied.

2.6.41 Saturation node



This documentation is for version 2.0 of Saturation.

Description

Modify the color saturation of an image.

See also: <http://opticalenquiry.com/nuke/index.php?title=Saturation>

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Saturation / saturation	Double	1	Color saturation factor to apply. 0 produces grayscale.
Luminance Math / luminanceMath	Choice	Rec. 709	Formula used to compute luminance from RGB values. Rec. 709 (rec709) : Use Rec. 709 ($0.2126r + 0.7152g + 0.0722b$). Rec. 2020 (rec2020) : Use Rec. 2020 ($0.2627r + 0.6780g + 0.0593b$). ACES AP0 (acesap0) : Use ACES AP0 ($0.3439664498r + 0.7281660966g + -0.0721325464b$). ACES AP1 (acesap1) : Use ACES AP1 ($0.2722287168r + 0.6740817658g + 0.0536895174b$). CCIR 601 (ccir601) : Use CCIR 601 ($0.2989r + 0.5866g + 0.1145b$). Average (average) : Use average of r, g, b. Max (max) : Use max of r, g, b.
Clamp Black / clampBlack	Boolean	On	All colors below 0 on output are set to 0.
Clamp White / clampWhite	Boolean	Off	All colors above 1 on output are set to 1.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.6.42 VectorToColor node



This documentation is for version 1.0 of VectorToColor.

Description

Convert x and y vector components to a color representation.

H (hue) gives the direction, S (saturation) is set to the amplitude/norm, and V is 1. The role of S and V can be switched. Output can be RGB or HSV, with H in degrees.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X channel / xChannel	Choice	r	Selects the X component of vectors r : R channel from input. g : G channel from input. b : B channel from input. a : A channel from input.
Y channel / yChannel	Choice	g	Selects the Y component of vectors r : R channel from input. g : G channel from input. b : B channel from input. a : A channel from input.
Opposite / opposite	Boolean	Off	If checked, opposite of X and Y are used.
Inverse Y / inverseY	Boolean	On	If checked, opposite of Y is used (on by default, because most optical flow results are shown using a downward Y axis).
Modulate V / modulateV	Boolean	Off	If checked, modulate V using the vector amplitude, instead of S.
HSV Output / hsvOutput	Boolean	Off	If checked, output is in the HSV color model.

2.6.43 XYZToLab node

This documentation is for version 1.0 of XYZToLab.

Description

Convert from CIE XYZ color space to CIE L*a*b color space. L*a*b coordinates are divided by 100 for better visualization.

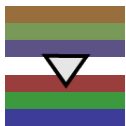
Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function

2.6.44 XYZToRGB709 node



This documentation is for version 1.0 of XYZToRGB709.

Description

Convert from XYZ color model to RGB (Rec.709 with D65 illuminant). X, Y and Z are in the same units as RGB.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Premult / premult	Boolean	Off	Multiply the image by the alpha channel after processing. Use to get premultiplied output images.

2.6.45 XYZToxyY node

This documentation is for version 1.0 of XYZToxyY.

Description

Convert from CIE XYZ color space to CIE xyY color space.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function

2.6.46 YCbCrToRGB601 node

This documentation is for version 1.0 of YCbCrToRGB601.

Description

Convert from YCbCr color model (ITU.BT-601) to linear RGB. RGB is gamma-decompressed using the sRGB Electro-Optical Transfer Function (EOTF) after conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Premult / premult	Boolean	Off	Multiply the image by the alpha channel after processing. Use to get premultiplied output images.

2.6.47 YCbCrToRGB709 node

This documentation is for version 1.0 of YCbCrToRGB709.

Description

Convert from YCbCr color model (ITU.BT-709) to linear RGB. RGB is gamma-decompressed using the Rec.709 Electro-Optical Transfer Function (EOTF) after conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Premult / premult	Boolean	Off	Multiply the image by the alpha channel after processing. Use to get premultiplied output images.

2.6.48 YPbPrToRGB601 node

This documentation is for version 1.0 of YPbPrToRGB601.

Description

Convert from YPbPr color model (ITU.BT-601) to RGB. RGB is gamma-decompressed using the sRGB Electro-Optical Transfer Function (EOTF) after conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Premult / premult	Boolean	Off	Multiply the image by the alpha channel after processing. Use to get premultiplied output images.

2.6.49 YPbPrToRGB709 node

This documentation is for version 1.0 of YPbPrToRGB709.

Description

Convert from YPbPr color model (ITU.BT-709) to RGB. RGB is gamma-decompressed using the Rec.709 Electro-Optical Transfer Function (EOTF) after conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Premult / <code>premult</code>	Boolean	Off	Multiply the image by the alpha channel after processing. Use to get premultiplied output images.

2.6.50 YUVToRGB601 node

This documentation is for version 1.0 of YUVToRGB601.

Description

Convert from YUV color model (ITU.BT-601) to RGB. RGB is gamma-decompressed using the sRGB Electro-Optical Transfer Function (EOTF) after conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Premult / <code>premult</code>	Boolean	Off	Multiply the image by the alpha channel after processing. Use to get premultiplied output images.

2.6.51 YUVToRGB709 node

This documentation is for version 1.0 of YUVToRGB709.

Description

Convert from YUV color model (ITU.BT-709) to RGB. RGB is gamma-decompressed using the Rec.709 Electro-Optical Transfer Function (EOTF) after conversion.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Premult / <code>premult</code>	Boolean	Off	Multiply the image by the alpha channel after processing. Use to get premultiplied output images.

2.6.52 xyYToXYZ node

This documentation is for version 1.0 of xyYToXYZ.

Description

Convert from CIE xyY color space to CIE XYZ color space.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function

2.7 Filter nodes

The following sections contain documentation about every node in the Filter group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.7.1 AngleBlur node

This documentation is for version 1.0 of AngleBlur.

Description

The Angle Blur effect gives the illusion of motion in a given direction.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Convert to Group / <code>convertToGroup</code>	Button		Converts this node to a Group: the internal node-graph and the user parameters will become editable
Angle / <code>angleBlur_angle</code>	Double	0	Determines the direction into which the image is blurred. This is an angle in degrees.
Distance / <code>angleBlur_distance</code>	Double	0	Determines how much the image will be blurred

2.7.2 Bloom node



This documentation is for version 4.0 of Bloom.

Description

Apply a Bloom filter (Kawase 2004) that sums multiple blur filters of different radii, resulting in a larger but sharper glare than a simple blur.

It is similar to applying ‘Count’ separate Blur filters to the same input image with sizes ‘Size’, ‘Size’*‘Ratio’, ‘Size’*‘Ratio’^2, etc., and averaging the results.

The blur radii follow a geometric progression (of common ratio 2 in the original implementation, bloomRatio in this implementation), and a total of bloomCount blur kernels are summed up (bloomCount=5 in the original implementation, and the kernels are Gaussian).

The blur filter can be a quasi-Gaussian, a Gaussian, a box, a triangle or a quadratic filter.

Ref.: Masaki Kawase, “Practical Implementation of High Dynamic Range Rendering”, GDC 2004.

Uses the ‘vanvliet’ and ‘deriche’ functions from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Size / size	Double	x: 0 y: 0	Size (diameter) of the filter kernel, in pixel units (≥ 0). The standard deviation of the corresponding Gaussian is $\text{size}/2.4$. No filter is applied if $\text{size} < 1.2$.
Uniform / uniform	Boolean	Off	Apply the same amount of blur on X and Y.
Ratio / bloomRatio	Double	2	Ratio between successive kernel sizes of the bloom filter. A ratio of 1 gives no Bloom effect, just the original blur. A higher ratio gives a blur kernel with a heavier tail. The original implementation uses a value of 2.
Count / bloomCount	Integer	5	Number of blur kernels of the bloom filter. The original implementation uses a value of 5. Higher values give a wider of heavier tail (the size of the largest blur kernel is $2^{**}\text{bloomCount} * \text{size}$). A count of 1 is just the original blur.

Continued on next page

Table 100 – continued from previous page

Parameter / script name	Type	Default	Function
Border Conditions / boundary	Choice	Nearest	<p>Specifies how pixel values are computed out of the image domain. This mostly affects values at the boundary of the image. If the image represents intensities, Nearest (Neumann) conditions should be used. If the image represents gradients or derivatives, Black (Dirichlet) boundary conditions should be used.</p> <p>Black (black): Dirichlet boundary condition: pixel values out of the image domain are zero.</p> <p>Nearest (nearest): Neumann boundary condition: pixel values out of the image domain are those of the closest pixel location in the image domain.</p>
Filter / filter	Choice	Quasi-Gaussian	<p>Blurring filter. The quasi-Gaussian filter should be appropriate in most cases. The Gaussian filter is more isotropic (its impulse response has rotational symmetry), but slower.</p> <p>Quasi-Gaussian (quasigaussian): Quasi-Gaussian filter (0-order recursive Deriche filter, faster) - IIR (infinite support / impulsional response).</p> <p>Gaussian (gaussian): Gaussian filter (Van Vliet recursive Gaussian filter, more isotropic, slower) - IIR (infinite support / impulsional response).</p> <p>Box (box): Box filter - FIR (finite support / impulsional response).</p> <p>Triangle (triangle): Triangle/tent filter - FIR (finite support / impulsional response).</p> <p>Quadratic (quadratic): Quadratic filter - FIR (finite support / impulsional response).</p>
Expand RoD / expandRoD	Boolean	Off	Expand the source region of definition by 1.5*size (3.6*sigma).
Crop To Format / cropToFormat	Boolean	On	If the source is inside the format and the effect extends it outside of the format, crop it to avoid unnecessary calculations. To avoid unwanted crops, only the borders that were inside of the format in the source clip will be cropped.
Alpha Threshold / alphaThreshold	Double	0	If this value is non-zero, any alpha value below this is set to zero. This is only useful for IIR filters (Gaussian and Quasi-Gaussian), which may produce alpha values very close to zero due to arithmetic precision. Remind that, in theory, a black image with a single white pixel should produce non-zero values everywhere, but a few VFX tricks rely on the fact that alpha should be zero far from the alpha edges (e.g. the premult-blur-unpremult trick to fill holes)). A threshold value of 0.003 is reasonable, and values between 0.001 and 0.01 are usually enough to remove these artifacts.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.3 Blur node



This documentation is for version 4.0 of Blur.

Description

Blur input stream or compute derivatives.

The blur filter can be a quasi-Gaussian, a Gaussian, a box, a triangle or a quadratic filter.

Note that the Gaussian filter [1] is implemented as an IIR (infinite impulse response) filter [2][3], whereas most compositing software implement the Gaussian as a FIR (finite impulse response) filter by cropping the Gaussian impulse response. Consequently, when blurring a white dot on black background, it produces very small values very far away from the dot. The quasi-Gaussian filter is also IIR.

A very common process in compositing to expand colors on the edge of a matte is to use the premult-blur-unpremult combination [4][5]. The very small values produced by the IIR Gaussian filter produce undesirable artifacts after unpremult. For this process, the FIR quadratic filter (or the faster triangle or box filters) should be preferred over the IIR Gaussian filter.

References:

[1] https://en.wikipedia.org/wiki/Gaussian_filter

[2] I.T. Young, L.J. van Vliet, M. van Ginkel, Recursive Gabor filtering. IEEE Trans. Sig. Proc., vol. 50, pp. 2799-2805, 2002. (this is an improvement over Young-Van Vliet, Sig. Proc. 44, 1995)

[3] B. Triggs and M. Sdika. Boundary conditions for Young-van Vliet recursive filtering. IEEE Trans. Signal Processing, vol. 54, pp. 2365-2367, 2006.

[4] Nuke Expand Edges or how to get rid of outlines. <http://franzbrandstaetter.com/?p=452>

[5] Colour Smear for Nuke. <http://richardfrazier.com/tools-tutorials/colour-smear-for-nuke/>

Uses the ‘vanvliet’ and ‘deriche’ functions from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

This plugin was compiled without debug, without assertions, with optimizations, with OpenMP 201107, using Clang version 6.0.0 (tags/RELEASE_600/final).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Size / size	Double	x: 0 y: 0	Size (diameter) of the filter kernel, in pixel units (≥ 0). The standard deviation of the corresponding Gaussian is $\text{size}/2.4$. No filter is applied if $\text{size} < 1.2$.
Uniform / uniform	Boolean	Off	Apply the same amount of blur on X and Y.
X derivation order / orderX	Integer	0	Derivation order in the X direction. (orderX=0,orderY=0) does smoothing, (orderX=1,orderY=0) computes the X component of the image gradient.
Y derivation order / orderY	Integer	0	Derivation order in the Y direction. (orderX=0,orderY=0) does smoothing, (orderX=0,orderY=1) computes the Y component of the image gradient.
Border Conditions / boundary	Choice	Black	<p>Specifies how pixel values are computed out of the image domain. This mostly affects values at the boundary of the image. If the image represents intensities, Nearest (Neumann) conditions should be used. If the image represents gradients or derivatives, Black (Dirichlet) boundary conditions should be used.</p> <p>Black (black): Dirichlet boundary condition: pixel values out of the image domain are zero.</p> <p>Nearest (nearest): Neumann boundary condition: pixel values out of the image domain are those of the closest pixel location in the image domain.</p>
Filter / filter	Choice	Gaussian	<p>Blurring filter. The quasi-Gaussian filter should be appropriate in most cases. The Gaussian filter is more isotropic (its impulse response has rotational symmetry), but slower.</p> <p>Quasi-Gaussian (quasigaussian): Quasi-Gaussian filter (0-order recursive Deriche filter, faster) - IIR (infinite support / impulsional response).</p> <p>Gaussian (gaussian): Gaussian filter (Van Vliet recursive Gaussian filter, more isotropic, slower) - IIR (infinite support / impulsional response).</p> <p>Box (box): Box filter - FIR (finite support / impulsional response).</p> <p>Triangle (triangle): Triangle/tent filter - FIR (finite support / impulsional response).</p> <p>Quadratic (quadratic): Quadratic filter - FIR (finite support / impulsional response).</p>
Expand RoD / expandRoD	Boolean	On	Expand the source region of definition by $1.5 * \text{size}$ ($3.6 * \text{sigma}$).
Crop To Format / cropToFormat	Boolean	On	If the source is inside the format and the effect extends it outside of the format, crop it to avoid unnecessary calculations. To avoid unwanted crops, only the borders that were inside of the format in the source clip will be cropped.
Alpha Threshold / alphaThreshold	Double	0	If this value is non-zero, any alpha value below this is set to zero. This is only useful for IIR filters (Gaussian and Quasi-Gaussian), which may produce alpha values very close to zero due to arithmetic precision. Remind that, in theory, a black image with a single white pixel should produce non-zero values everywhere, but a few VFX tricks rely on the fact that alpha should be zero far from the alpha edges (e.g. the premult-blur-unpremult trick to fill holes)). A threshold value of 0.003 is reasonable, and values between 0.001 and 0.01 are usually enough to remove these artifacts.

Continued on next page

Table 101 – continued from previous page

Parameter / script name	Type	Default	Function
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.4 ChromaBlur node



This documentation is for version 4.0 of ChromaBlur.

Description

Blur the chrominance of an input stream. Smoothing is done on the x and y components in the CIE xyY color space. Used to prep strongly compressed and chroma subsampled footage for keying.

The blur filter can be a quasi-Gaussian, a Gaussian, a box, a triangle or a quadratic filter.

Uses the ‘vanvliet’ and ‘deriche’ functions from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Size / size	Double	x: 0 y: 0	Size (diameter) of the filter kernel, in pixel units (≥ 0). The standard deviation of the corresponding Gaussian is $\text{size}/2.4$. No filter is applied if $\text{size} < 1.2$.
Uniform / uniform	Boolean	Off	Apply the same amount of blur on X and Y.
Colorspace / colorspace	Choice	Rec. 709	Formula used to compute chrominance from RGB values. Rec. 709 (rec709) : Use Rec. 709 with D65 illuminant. Rec. 2020 (rec2020) : Use Rec. 2020 with D65 illuminant. ACES AP0 (acesap0) : Use ACES AP0 with ACES (approx. D60) illuminant. ACES AP1 (acesap1) : Use ACES AP1 with ACES (approx. D60) illuminant.

Continued on next page

Table 102 – continued from previous page

Parameter / script name	Type	Default	Function
Filter / <code>filter</code>	Choice	Gaussian	<p>Blurring filter. The quasi-Gaussian filter should be appropriate in most cases. The Gaussian filter is more isotropic (its impulse response has rotational symmetry), but slower.</p> <p>Quasi-Gaussian (quasigaussian): Quasi-Gaussian filter (0-order recursive Deriche filter, faster) - IIR (infinite support / impulsional response).</p> <p>Gaussian (gaussian): Gaussian filter (Van Vliet recursive Gaussian filter, more isotropic, slower) - IIR (infinite support / impulsional response).</p> <p>Box (box): Box filter - FIR (finite support / impulsional response).</p> <p>Triangle (triangle): Triangle/tent filter - FIR (finite support / impulsional response).</p> <p>Quadratic (quadratic): Quadratic filter - FIR (finite support / impulsional response).</p>
(Un)premult / <code>premult</code>	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / <code>maskInvert</code>	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / <code>mix</code>	Double	1	Mix factor between the original and the transformed image.

2.7.5 DenoiseSharpen node

This documentation is for version 1.0 of DenoiseSharpen.

Description

Denoise and/or sharpen images using wavelet-based algorithms.

Description

This plugin allows the separate denoising of image channels in multiple color spaces using wavelets, using the BayesShrink algorithm, and can also sharpen the image details.

Noise levels for each channel may be either set manually, or analyzed from the image data in each wavelet subband using the MAD (median absolute deviation) estimator. Noise analysis is based on the assumption that the noise is Gaussian and additive (it is not intensity-dependent). If there is speckle or salt-and-pepper noise in the images, the Median or SmoothPatchBased filters may be more appropriate. The color model specifies the channels and the transforms used. Noise levels have to be re-adjusted or re-analyzed when changing the color model.

Basic Usage

The input image should be in linear RGB.

For most footage, the effect works best by keeping the default Y'CbCr color model. The color models are made to work with Rec.709 data, but DenoiseSharpen will still work if the input is in another colorspace, as long as the input is linear RGB:

- The Y'CbCr color model uses the Rec.709 opto-electronic transfer function to convert from RGB to R'G'B' and the the Rec.709 primaries to convert from R'G'B' to Y'CbCr.
- The L * a * b color model uses the Rec.709 RGB primaries to convert from RGB to L * a * b.

- The R'G'B' color model uses the Rec.709 opto-electronic transfer function to convert from RGB to R'G'B'.
- The RGB color model (linear) makes no assumption about the RGB color space, and works directly on the RGB components, assuming additive noise. This is the only option if the noisy source contains negative values. If, say, the noise is known to be multiplicative, one can convert the images to Log before denoising, use this option, and convert back to linear after denoising.
- The Alpha channel, if processed, is always considered to be linear.

The simplest way to use this plugin is to leave the noise analysis area to the whole image, and click “Analyze Noise Levels”. Once the analysis is done, “Lock Noise Analysis” is checked in order to avoid modifying the essential parameters by mistake.

If the image has many textured areas, it may be preferable to select an analysis area with flat colors, free from any details, shadows or highlights, to avoid considering texture as noise. The AnalysisMask input can be used to mask the analysis, if the rectangular area is not appropriate. Any non-zero pixels in the mask are taken into account. A good option for the AnalysisMask would be to take the inverse of the output of an edge detector and clamp it correctly so that all pixels near the edges have a value of zero..

If the sequence to be denoised does not have enough flat areas, you can also connect a reference footage with the same kind of noise to the AnalysisSource input: that source will be used for the analysis only. If no source with flat areas is available, and noise analysis can only be performed on areas which also contain details, it is often preferable to disable very low, low, and sometimes medium frequencies in the “Frequency Tuning” parameters group, or at least to lower their gain, since they may be misestimated by the noise analysis process. If the noise is IID (independent and identically distributed), such as digital sensor noise, only “Denoise High Frequencies” should be checked. If the noise has some grain (i.e. it comes from lossy compression of noisy images by a camera, or it is scanned film), then you may want to enable medium frequencies as well. If low and very low frequencies are enabled, but the analysis area is not a flat zone, the signal itself (i.e. the noise-free image) could be considered as noise, and the result may exhibit low contrast and blur.

To check what details have been kept after denoising, you can raise the Sharpen Amount to something like 10, and then adjust the Noise Level Gain to get the desired denoising amount, until no noise is left and only image details remain in the sharpened image. You can then reset the Sharpen Amount to zero, unless you actually want to enhance the contrast of your denoised footage.

You can also check what was actually removed from the original image by selecting the “Noise” Output mode (instead of “Result”). If too many image details are visible in the noise, noise parameters may need to be tuned.

Inputs

Input	Description	Optional
Source	The footage to be denoised. If nothing is connected to the AnalysisSource input, this is also used for noise analysis.	No
Mask	An optional image to use as a mask. By default, the effect is limited to the non-black areas of the mask.	Yes
Analysis-Source	An optional noise source. If connected, this is used instead of the Source input for the noise analysis. This is used to analyse noise from some footage by apply it on another footage, in case the footage to be denoised does not have enough flat areas.	Yes
Analysis-Mask	An optional mask for the analysis area. This mask is intersected with the Analysis Rectangle. Non-zero pixels are taken into account in the noise analysis phase.	Yes

Controls

Parameter / script name	Type	Default	Function
Output / <code>outputMode</code>	Choice	Result	<p>Select which image is output when analysis is locked. When analysis is not locked, the effect does nothing (the output is the source image).</p> <p>Result (result): The result of denoising and sharpening the Source image.</p> <p>Noise (noise): An image containing what would be added to the image to denoise it. If 'Denoise Amount' is zero, this image should be black. Only noise should be visible in this image. If you can see a lot of picture detail in the noise output, it means the current settings are denoising too hard and remove too much of the image, which leads to a smoothed result. Try to lower the noise levels or the noise level gain.</p> <p>Sharpen (sharpen): An image containing what would be added to the image to sharpen it. If 'Sharpen Amount' is zero, this image should be black. Only image details should be visible in this image. If you can see a lot of noise in the sharpen output, it means the current settings are denoising not enough, which leads to a noisy result. Try to raise the noise levels or the noise level gain.</p>
Color Model / <code>colorModel</code>	Choice	Y'CbCr(A)	<p>The colorspace where denoising is performed. These colorspace assume that input and output use the Rec.709/sRGB chromaticities and the D65 illuminant, but should tolerate other input colorspace (the output colorspace will always be the same as the input colorspace). Noise levels are reset when the color model is changed.</p> <p>Y'CbCr(A) (ycbcr): The YCbCr color model has one luminance channel (Y) which contains most of the detail information of an image (such as brightness and contrast) and two chroma channels (Cb = blueness, Cr = redness) that hold the color information. Note that this choice drastically affects the result. Uses the Rec.709 opto-electronic transfer function to convert from RGB to R'G'B' and the the Rec.709 primaries to convert from R'G'B' to Y'CbCr.</p> <p>CIE L*a*b(A) (cielab): CIE L*a*b* is a color model in which chrominance is separated from lightness and color distances are perceptually uniform. Note that this choice drastically affects the result. Uses the Rec.709 primaries to convert from RGB to L*a*b.</p> <p>R'G'B'(A) (gammargb): The R'G'B' color model (gamma-corrected RGB) separates an image into channels of red, green, and blue. Note that this choice drastically affects the result. Uses the Rec.709 opto-electronic transfer function to convert from RGB to R'G'B'.</p> <p>RGB(A) (linearrgb): The Linear RGB color model processes the raw linear components. Usually a bad choice, except when denoising non-color data (e.g. depth or motion vectors). No assumption is made about the RGB color space.</p>
Lock Analysis and Apply / <code>analysisLock</code>	Boolean	Off	Lock all noise analysis parameters and apply denoising. When the analysis is not locked, the source image is output.
Bottom Left / <code>bottomLeft</code>	Double	x: 0.1 y: 0.1	Coordinates of the bottom left corner of the analysis rectangle. This rectangle is intersected with the AnalysisMask input, if connected.
Size / <code>size</code>	Double	w: 0.8 h: 0.8	Width and height of the analysis rectangle. This rectangle is intersected with the AnalysisMask input, if connected.

Continued on next page

Table 103 – continued from previous page

Parameter / script name	Type	Default	Function
B3 Spline Interpolation / useB3Spline	Boolean	On	For wavelet decomposition, use a 5x5 filter based on B3 spline interpolation rather than a 3x3 Lagrange linear filter. Noise levels are reset when this setting is changed. The influence of this parameter is minimal, and it should not be changed.
Analysis Frame / analysisFrame	Integer	-1	The frame number where the noise levels were analyzed.
Analyze Noise Levels / analyzeNoiseLevels	Button		Computes the noise levels from the current frame and current color model. To use the same settings for the whole sequence, analyze a frame that is representative of the sequence. If a mask is set, it is used to compute the noise levels from areas where the mask is non-zero. If there are keyframes on the noise level parameters, this sets a keyframe at the current frame. The noise levels can then be fine-tuned.
Y Level (High) / ylrNoiseLevelHigh	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Cb Level (High) / cbagNoiseLevelHigh	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Cr Level (High) / crbbNoiseLevelHigh	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Alpha Level (High) / alphaNoiseLevelHigh	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Y Level (Medium) / ylrNoiseLevelMedium	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Cb Level (Medium) / cbagNoiseLevelMedium	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Cr Level (Medium) / crbbNoiseLevelMedium	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Alpha Level (Medium) / alphaNoiseLevelMedium	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Y Level (Low) / ylrNoiseLevelLow	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Cb Level (Low) / cbagNoiseLevelLow	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Cr Level (Low) / crbbNoiseLevelLow	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Alpha Level (Low) / alphaNoiseLevelLow	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Y Level (Very Low) / ylrNoiseLevelVeryLow	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Cb Level (Very Low) / cbagNoiseLevelVeryLow	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.

Continued on next page

Table 103 – continued from previous page

Parameter / script name	Type	Default	Function
Cr Level (Very Low) / <code>crbbNoiseLevelVeryLow</code>	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Alpha Level (Very Low) / <code>alphaNoiseLevelVeryLow</code>	Double	0	Adjusts the noise variance of the selected channel for the given noise frequency. May be estimated from image data by pressing the “Analyze Noise” button.
Noise Level Gain / <code>noiseLevelGain</code>	Double	1	Global gain to apply to the noise level thresholds. 0 means no denoising, 1 means use the estimated thresholds multiplied by the per-frequency gain and the channel gain. The default value (1.0) is rather conservative (it does not destroy any kind of signal). Values around 1.1 or 1.2 usually give more pleasing results.
Denoise Amount / <code>denoiseAmount</code>	Double	1	The amount of denoising to apply. 0 means no denoising (which may be useful to sharpen without denoising), between 0 and 1 does a soft thresholding of below the thresholds, thus keeping some noise, and 1 applies the threshold strictly and removes everything below the thresholds. This should be used only if you want to keep some noise, for example for noise matching. This value is multiplied by the per-channel amount set in the ‘Channel Tuning’ group. Remember that the thresholds are multiplied by the per-frequency gain, the channel gain, and the Noise Level Gain first.
Denoise High Frequencies / <code>enableFreqHigh</code>	Boolean	On	Check to enable the high frequency noise level thresholds. It is recommended to always leave this checked.
High Gain / <code>gainFreqHigh</code>	Double	1	Gain to apply to the high frequency noise level thresholds. 0 means no denoising, 1 means use the estimated thresholds multiplied by the channel Gain and the Noise Level Gain.
Denoise Medium Frequencies / <code>enableFreqMedium</code>	Boolean	On	Check to enable the medium frequency noise level thresholds. Can be disabled if the analysis area contains high frequency texture, or if the noise is known to be IID (independent and identically distributed), for example if this is only sensor noise and lossless compression is used, and not grain or compression noise.
Medium Gain / <code>gainFreqMedium</code>	Double	1	Gain to apply to the medium frequency noise level thresholds. 0 means no denoising, 1 means use the estimated thresholds multiplied by the channel Gain and the Noise Level Gain.
Denoise Low Frequencies / <code>enableFreqLow</code>	Boolean	On	Check to enable the low frequency noise level thresholds. Must be disabled if the analysis area contains texture, or if the noise is known to be IID (independent and identically distributed), for example if this is only sensor noise and lossless compression is used, and not grain or compression noise.
Low Gain / <code>gainFreqLow</code>	Double	1	Gain to apply to the low frequency noise level thresholds. 0 means no denoising, 1 means use the estimated thresholds multiplied by the channel Gain and the Noise Level Gain.
Denoise Very Low Frequencies / <code>enableFreqVeryLow</code>	Boolean	On	Check to enable the very low frequency noise level thresholds. Can be disabled in most cases. Must be disabled if the analysis area contains texture, or if the noise is known to be IID (independent and identically distributed), for example if this is only sensor noise and lossless compression is used, and not grain or compression noise.
Very Low Gain / <code>gainFreqVeryLow</code>	Double	1	Gain to apply to the very low frequency noise level thresholds. 0 means no denoising, 1 means use the estimated thresholds multiplied by the channel Gain and the global Noise Level Gain.
Adaptive Radius / <code>adaptiveRadius</code>	Integer	4	Radius of the window where the signal level is analyzed at each scale. If zero, the signal level is computed from the whole image, which may excessively blur the edges if the image has many flat color areas. A reasonable value should be in the range 2-4.

Continued on next page

Table 103 – continued from previous page

Parameter / script name	Type	Default	Function
Y Gain / ylrGain	Double	1	Gain to apply to the thresholds for this channel. 0 means no denoising, 1 means use the estimated thresholds multiplied by the per-frequency gain and the global Noise Level Gain.
Y Amount / ylrAmount	Double	1	The amount of denoising to apply to the specified channel. 0 means no denoising, between 0 and 1 does a soft thresholding of below the thresholds, thus keeping some noise, and 1 applies the threshold strictly and removes everything below the thresholds. This should be used only if you want to keep some noise, for example for noise matching. This value is multiplied by the global Denoise Amount. Remember that the thresholds are multiplied by the per-frequency gain, the channel gain, and the Noise Level Gain first.
Cb Gain / cbagGain	Double	1	Gain to apply to the thresholds for this channel. 0 means no denoising, 1 means use the estimated thresholds multiplied by the per-frequency gain and the global Noise Level Gain.
Cb Amount / cbagAmount	Double	1	The amount of denoising to apply to the specified channel. 0 means no denoising, between 0 and 1 does a soft thresholding of below the thresholds, thus keeping some noise, and 1 applies the threshold strictly and removes everything below the thresholds. This should be used only if you want to keep some noise, for example for noise matching. This value is multiplied by the global Denoise Amount. Remember that the thresholds are multiplied by the per-frequency gain, the channel gain, and the Noise Level Gain first.
Cr Gain / crbbGain	Double	1	Gain to apply to the thresholds for this channel. 0 means no denoising, 1 means use the estimated thresholds multiplied by the per-frequency gain and the global Noise Level Gain.
Cr Amount / crbbAmount	Double	1	The amount of denoising to apply to the specified channel. 0 means no denoising, between 0 and 1 does a soft thresholding of below the thresholds, thus keeping some noise, and 1 applies the threshold strictly and removes everything below the thresholds. This should be used only if you want to keep some noise, for example for noise matching. This value is multiplied by the global Denoise Amount. Remember that the thresholds are multiplied by the per-frequency gain, the channel gain, and the Noise Level Gain first.
Alpha Gain / alphaGain	Double	1	Gain to apply to the thresholds for this channel. 0 means no denoising, 1 means use the estimated thresholds multiplied by the per-frequency gain and the global Noise Level Gain.
Alpha Amount / alphaAmount	Double	1	The amount of denoising to apply to the specified channel. 0 means no denoising, between 0 and 1 does a soft thresholding of below the thresholds, thus keeping some noise, and 1 applies the threshold strictly and removes everything below the thresholds. This should be used only if you want to keep some noise, for example for noise matching. This value is multiplied by the global Denoise Amount. Remember that the thresholds are multiplied by the per-frequency gain, the channel gain, and the Noise Level Gain first.
Sharpen Amount / sharpenAmount	Double	0	Adjusts the amount of sharpening applied. Be careful that only components that are above the noise levels are enhanced, so the noise level gain parameters are very important for proper sharpening. For example, if ‘Noise Level Gain’ is set to zero (0), then noise is sharpened as well as signal. If the ‘Noise Level Gain’ is set to one (1), only signal is sharpened. In order to sharpen without denoising, set the ‘Denoise Amount’ parameter to zero (0).
Sharpen Size / sharpenSize	Double	10	Adjusts the size of the sharpening. For very unsharp images it is recommended to use higher values. Default is 10.

Continued on next page

Table 103 – continued from previous page

Parameter / script name	Type	Default	Function
Sharpen Y Only / sharpenLuminance	Boolean	On	Sharpens luminance only (if colormodel is R'G'B', sharpen only RGB). This avoids color artifacts to appear. Colour sharpness in natural images is not critical for the human eye.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.6 Dilate node



This documentation is for version 2.1 of Dilate.

Description

Dilate (or erode) input stream by a rectangular structuring element of specified size and Neumann boundary conditions (pixels out of the image get the value of the nearest pixel).

A negative size will perform an erosion instead of a dilation.

Different sizes can be given for the x and y axis.

Uses the 'dilate' and 'erode' functions from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Size / size	Integer	x: 1 y: 1	Width/height of the rectangular structuring element is 2*size+1, in pixel units (>=0).
Expand RoD / expandRoD	Boolean	On	Expand the source region of definition by 2*size pixels if size is positive
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.7 DirBlur node



This documentation is for version 1.0 of DirBlur.

Description

Apply directional blur to an image.

This plugin concatenates transforms upstream.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Translate / translate	Double	x: 0 y: 0	Translation along the x and y axes in pixels. Can also be adjusted by clicking and dragging the center handle in the Viewer.
Rotate / rotate	Double	0	Rotation angle in degrees around the Center. Can also be adjusted by clicking and dragging the rotation bar in the Viewer.
Scale / scale	Double	x: 1 y: 1	Scale factor along the x and y axes. Can also be adjusted by clicking and dragging the outer circle or the diameter handles in the Viewer.
Uniform / uniform	Boolean	Off	Use the X scale for both directions
Skew X / skewX	Double	0	Skew along the x axis. Can also be adjusted by clicking and dragging the skew bar in the Viewer.
Skew Y / skewY	Double	0	Skew along the y axis.
Skew Order / skewOrder	Choice	XY	The order in which skew transforms are applied: X then Y, or Y then X. XY YX
Amount / transformAmount	Double	1	Amount of transform to apply. 0 means the transform is identity, 1 means to apply the full transform.
Center / center	Double	x: 0.5 y: 0.5	Center of rotation and scale.
Reset Center / resetCenter	Button		Reset the position of the center to the center of the input region of definition
Interactive Update / interactive	Boolean	On	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Invert / invert	Boolean	Off	Invert the transform.

Continued on next page

Table 105 – continued from previous page

Parameter / script name	Type	Default	Function
Filter / filter	Choice	Cubic	<p>Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+).</p> <p>Impulse (impulse): (nearest neighbor / box) Use original values.</p> <p>Box (box): Integrate the source image over the bounding box of the back-transformed pixel.</p> <p>Bilinear (bilinear): (tent / triangle) Bilinear interpolation between original values.</p> <p>Cubic (cubic): (cubic spline) Some smoothing.</p> <p>Keys (keys): (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*).</p> <p>Simon (simon): Some smoothing, plus medium sharpening (*).</p> <p>Rifman (rifman): Some smoothing, plus significant sharpening (*).</p> <p>Mitchell (mitchell): Some smoothing, plus blurring to hide pixelation (*+).</p> <p>Parzen (parzen): (cubic B-spline) Greatest smoothing of all filters (+).</p> <p>Notch (notch): Flat smoothing (which tends to hide moire' patterns) (+).</p>
Clamp / clamp	Boolean	Off	Clamp filter output within the original range - useful to avoid negative values in mattes
Black outside / black_outside	Boolean	Off	Fill the area outside the source image with black
Motion Blur / motionBlur	Double	1	Quality of motion blur rendering. 0 disables motion blur, 1 is a good value. Increasing this slows down rendering.
Amount / amount	Double	1	Amount of blur transform to apply. A value of 1 means to apply the full transform range. A value of 0 means to apply no blur at all. Default is 1.
Centered / centered	Boolean	Off	When checked, apply directional blur symmetrically around the neutral position.
Fading / fading	Double	0	Controls the fading function. A value of 1 corresponds to linear fading. A value of 0 disables fading. Default is 0.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.8 Distance node

This documentation is for version 1.0 of Distance.

Description

Compute at each pixel the distance to pixels that have a value of zero.

The distance is normalized with respect to the largest image dimension, so that it is between 0 and 1.

Optionally, a signed distance to the frontier between zero and nonzero values can be computed.

The distance transform can then be thresholded using the Threshold effect, or transformed using the ColorLookup effect, in order to generate a mask for another effect.

See also https://en.wikipedia.org/wiki/Distance_transform

Uses the 'distance' function from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Metric / <code>metric</code>	Choice	Euclidean	Type of metric. Chebyshev (chebyshev) : $\max(\text{abs}(x-x\text{border}), \text{abs}(y-y\text{border}))$ Manhattan (manhattan) : $\text{abs}(x-x\text{border}) + \text{abs}(y-y\text{border})$ Euclidean (euclidean) : $\sqrt{\text{sqr}(x-x\text{border}) + \text{sqr}(y-y\text{border})}$
Signed Distance / <code>signed</code>	Boolean	Off	Instead of computing the distance to pixels with a value of zero, compute the signed distance to the contour between zero and non-zero pixels. On output, non-zero-valued pixels have a positive signed distance, zero-valued pixels have a negative signed distance.
(Un)premult / <code>premult</code>	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / <code>maskInvert</code>	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / <code>mix</code>	Double	1	Mix factor between the original and the transformed image.

2.7.9 DropShadow node



This documentation is for version 1.0 of DropShadow.

Description

Creates a drop shadow on the source image using its alpha channel.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Convert to Group / <code>convertToGroup</code>	Button		Converts this node to a Group: the internal node-graph and the user parameters will become editable
Angle / <code>shadowAngle</code>	Double	-45	
Distance / <code>shadowDist</code>	Double	20	
Bluriness / <code>shadowBlur</code>	Double	x: 0 y: 0	
Opacity / <code>shadowOpacity</code>	Color	r: 0.5 g: 0.5 b: 0.5 a: 0.5	
Color / <code>shadowColor</code>	Color	r: 0 g: 0 b: 0	
Color from source / <code>shadowCFS</code>	Boolean	Off	
Shadow only / <code>shadowOnly</code>	Boolean	Off	

2.7.10 EdgeBlur node

This documentation is for version 1.0 of EdgeBlur.

Description

Blur the image where there are edges in the alpha/matte channel.

Inputs

Input	Description	Optional
Source		No
Mask		Yes
Matte		Yes

Controls

Parameter / script name	Type	Default	Function
Convert to Group / <code>convertToGroup</code>	Button		Converts this node to a Group: the internal node-graph and the user parameters will become editable
R / <code>Blur1NatronOfxParamProcessR</code>	Boolean	On	
G / <code>Blur1NatronOfxParamProcessG</code>	Boolean	On	
B / <code>Blur1NatronOfxParamProcessB</code>	Boolean	On	
A / <code>Blur1NatronOfxParamProcessA</code>	Boolean	On	
External Matte / <code>externalMatte</code>	Boolean	Off	Use the edges from the Matte input instead of the alpha channel of the source image.

Continued on next page

Table 108 – continued from previous page

Parameter / script name	Type	Default	Function
Size / size	Double	3	
Filter / filter	Choice	Gaussian	<p>Simple (simple): Gradient is estimated by centered finite differences.</p> <p>Sobel (sobel): Compute gradient using the Sobel 3x3 filter.</p> <p>Rotation Invariant (rotinvariant): Compute gradient using a 3x3 rotation-invariant filter.</p> <p>Quasi-Gaussian (quasigaussian): Quasi-Gaussian filter (0-order recursive Deriche filter, faster) - IIR (infinite support / impulsional response).</p> <p>Gaussian (gaussian): Gaussian filter (Van Vliet recursive Gaussian filter, more isotropic, slower) - IIR (infinite support / impulsional response).</p> <p>Box (box): Box filter - FIR (finite support / impulsional response).</p> <p>Triangle (triangle): Triangle/tent filter - FIR (finite support / impulsional response).</p> <p>Quadratic (quadratic): Quadratic filter - FIR (finite support / impulsional response).</p>
Crop To Format / cropToFormat	Boolean	On	
Edge Mult / edgeMult	Double	2	Sharpness of the borders of the blur area.
Invert Mask / MergelmaskInvert	Boolean	Off	
Mix / BlurImix	Double	1	

2.7.11 EdgeDetect node



This documentation is for version 4.0 of EdgeDetect.

Description

Perform edge detection by computing the image gradient magnitude. Optionally, edge detection can be preceded by blurring, and followed by erosion and thresholding. In most cases, EdgeDetect is followed a Grade node to extract the proper edges and generate a mask from these.

For color or multi-channel images, several edge detection algorithms are proposed to combine the gradients computed in each channel:

- Separate: the gradient magnitude is computed in each channel separately, and the output is a color edge image.
- RMS: the RMS of per-channel gradients magnitudes is computed.
- Max: the maximum per-channel gradient magnitude is computed.
- Tensor: the tensor gradient norm [1].

References:

- [1] Silvano Di Zenzo, A note on the gradient of a multi-image, CVGIP 33, 116-125 (1986). <http://people.csail.mit.edu/tieu/notebook/imageproc/dizenzo86.pdf>

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Filter / <code>filter</code>	Choice	Gaussian	<p>Edge detection filter. If the blur size is not zero, it is used as the kernel size for quasi-Gaussian, Gaussian, box, triangle and quadratic filters. For the simple, rotation-invariant and Sobel filters, the image is pre-blurred with a Gaussian filter.</p> <p>Simple (simple): Gradient is estimated by centered finite differences.</p> <p>Sobel (sobel): Compute gradient using the Sobel 3x3 filter.</p> <p>Rotation Invariant (rotinvariant): Compute gradient using a 3x3 rotation-invariant filter.</p> <p>Quasi-Gaussian (quasigaussian): Quasi-Gaussian filter (0-order recursive Deriche filter, faster) - IIR (infinite support / impulsional response).</p> <p>Gaussian (gaussian): Gaussian filter (Van Vliet recursive Gaussian filter, more isotropic, slower) - IIR (infinite support / impulsional response).</p> <p>Box (box): Box filter - FIR (finite support / impulsional response).</p> <p>Triangle (triangle): Triangle/tent filter - FIR (finite support / impulsional response).</p> <p>Quadratic (quadratic): Quadratic filter - FIR (finite support / impulsional response).</p>
Multi-Channel / <code>multiChannel</code>	Choice	Tensor	<p>Operation used to combine multi-channel (e.g. color) gradients into an edge detector. This parameter has no effect if a single channel (e.g. alpha) is processed.</p> <p>Separate (separate): The gradient magnitude is computed in each channel separately, and the output is a color edge image.</p> <p>RMS (rms): The RMS of per-channel gradients magnitudes is computed.</p> <p>Max (max): The maximum per-channel gradient magnitude is computed.</p> <p>Tensor (tensor): The tensor gradient norm is computed. See Silvano Di Zenzo, A note on the gradient of a multi-image, CVGIP 33, 116-125 (1986).</p>

Continued on next page

Table 109 – continued from previous page

Parameter / script name	Type	Default	Function
Blur Size / <code>blurSize</code>	Double	0	Size of the blur kernel applied before edge detection.
Erode Size / <code>erodeSize</code>	Double	0	Size of the erosion performed after edge detection.
Non-Maxima Suppression / <code>nms</code>	Boolean	Off	Perform non-maxima suppression (after edge detection and erosion): only values that are maximal in the direction orthogonal to the contour are kept. For multi-channel images, the contour direction estimation depends on the multi-channel operation.
Expand RoD / <code>expandRoD</code>	Boolean	On	Expand the source region of definition by $1.5 \times \text{size}$ ($3.6 \times \text{sigma}$).
Crop To Format / <code>cropToFormat</code>	Boolean	On	If the source is inside the format and the effect extends it outside of the format, crop it to avoid unnecessary calculations. To avoid unwanted crops, only the borders that were inside of the format in the source clip will be cropped.
(Un)premult / <code>premult</code>	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / <code>maskInvert</code>	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / <code>mix</code>	Double	1	Mix factor between the original and the transformed image.

2.7.12 EdgeExtend node



This documentation is for version 4.0 of EdgeExtend.

Description

Fill a matte (i.e. a non-opaque color image with an alpha channel) by extending the edges of the matte. This effect does nothing on an opaque image.

If the input matte comes from a keyer, the alpha channel of the matte should be first eroded by a small amount to remove pixels containing mixed foreground/background colors. If not, these mixed colors may be extended instead of the pure foreground colors.

The filling process works by iteratively blurring the image, and merging the non-blurred image over the image to get to the next iteration. There are exactly ‘Slices’ such operations. The blur size at each iteration is linearly increasing.

‘Size’ is thus the total size of the edge extension, and ‘Slices’ is an indicator of the precision: the more slices there are, the sharper is the final image near the original edges.

Optionally, the image can be multiplied by the alpha channel on input (premultiplied), and divided by the alpha channel on output (unpremultiplied), so that if RGB contain an image and Alpha contains a mask, the output is an image where the RGB is smeared from the non-zero areas of the mask to the zero areas of the same mask.

The ‘Size’ parameter gives the size of the largest blur kernel, ‘Count’ gives the number of blur kernels, and ‘Ratio’ gives the ratio between consecutive blur kernel sizes. The size of the smallest blur kernel is thus $\text{‘Size’} / \text{‘Ratio’}^{(\text{‘Count’} - 1)}$

To get the classical single unpremult-blur-premult, use ‘Count’=1 and set the size to the size of the blur kernel. However, near the mask borders, a frontier can be seen between the non-blurred area (this inside of the mask) and the blurred area. Using more blur sizes will give a much smoother transition.

The idea for the buildup blurs to expand RGB comes from the EdgeExtend effect for Nuke by Frank Rueter (except the blurs were merged from the smallest to the largest, and here it is done the other way round), with suggestions by Lucas Pfaff.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Premult Source / <code>edgeExtendPremult</code>	Boolean	Off	Premultiply the source image by its alpha channel before processing. Do not check if the source matte is already premultiplied
Size / <code>edgeExtendSize</code>	Double	20	Maximum blur kernel size applied in the ExtendSlices filter. Raise to extend the edges further.
Slices / <code>edgeExtendSlices</code>	Integer	5	Number of blur kernels applied in the ExtendSlices filter. A count of 1 just merges the source image over the source image blurred by a kernel of size Size.
Unpremult Result / <code>edgeExtendUnpremult</code>	Boolean	Off	Unpremultiply the result image by its alpha channel after processing.
Filter / <code>filter</code>	Choice	Quasi-Gaussian	<p>Blurring filter. The quasi-Gaussian filter should be appropriate in most cases. The Gaussian filter is more isotropic (its impulse response has rotational symmetry), but slower.</p> <p>Quasi-Gaussian (quasigaussian): Quasi-Gaussian filter (0-order recursive Deriche filter, faster) - IIR (infinite support / impulsional response).</p> <p>Gaussian (gaussian): Gaussian filter (Van Vliet recursive Gaussian filter, more isotropic, slower) - IIR (infinite support / impulsional response).</p> <p>Box (box): Box filter - FIR (finite support / impulsional response).</p> <p>Triangle (triangle): Triangle/tent filter - FIR (finite support / impulsional response).</p> <p>Quadratic (quadratic): Quadratic filter - FIR (finite support / impulsional response).</p>
Expand RoD / <code>expandRoD</code>	Boolean	On	Expand the source region of definition by 1.5*size (3.6*sigma).
Crop To Format / <code>cropToFormat</code>	Boolean	On	If the source is inside the format and the effect extends it outside of the format, crop it to avoid unnecessary calculations. To avoid unwanted crops, only the borders that were inside of the format in the source clip will be cropped.
Invert Mask / <code>maskInvert</code>	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / <code>mix</code>	Double	1	Mix factor between the original and the transformed image.

2.7.13 Erode node



This documentation is for version 2.1 of Erode.

Description

Erode (or dilate) input stream by a rectangular structuring element of specified size and Neumann boundary conditions (pixels out of the image get the value of the nearest pixel).

A negative size will perform a dilation instead of an erosion.

Different sizes can be given for the x and y axis.

Uses the ‘erode’ and ‘dilate’ functions from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Size / size	Integer	x: 1 y: 1	Width/height of the rectangular structuring element is 2*size+1, in pixel units (≥ 0).
Expand RoD / expandRoD	Boolean	On	Expand the source region of definition by 2*size pixels if size is negative
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.14 ErodeBlur node

This documentation is for version 4.0 of ErodeBlur.

Description

Performs an operation that looks like an erosion or a dilation by smoothing the image and then remapping the values of the result.

The image is first smoothed by a triangle filter of width $2 \cdot \text{abs}(\text{size})$.

Now suppose the image is a 0-1 step edge ($I=0$ for x less than 0, $I=1$ for x greater than 0). The intensities are linearly remapped so that the value at $x=\text{size}-0.5$ is mapped to 0 and the value at $x=\text{size}+0.5$ is mapped to 1.

This process usually works well for mask images (i.e. images which are either 0 or 1), but may give strange results on images with real intensities, where another Erode filter has to be used.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Size / size	Double	-1	How much to shrink the black and white mask, in pixels (can be negative to dilate).
Blur / blur	Double	0	Soften the borders of the generated mask.
Expand RoD / expandRoD	Boolean	On	Expand the source region of definition by 1.5*size (3.6*sigma).
Crop To Format / cropToFormat	Boolean	On	If the source is inside the format and the effect extends it outside of the format, crop it to avoid unnecessary calculations. To avoid unwanted crops, only the borders that were inside of the format in the source clip will be cropped.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.15 ErodeSmooth node



This documentation is for version 2.0 of ErodeSmooth.

Description

Erode or dilate input stream using a normalized power-weighted filter.

This gives a smoother result than the Erode or Dilate node.

See “Robust local max-min filters by normalized power-weighted filtering” by L.J. van Vliet, <http://dx.doi.org/10.1109/ICPR.2004.1334273>

Uses the ‘vanvliet’ and ‘deriche’ functions from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Range / range	Double	min: 0 max: 1	Expected range for input values.
Size / size	Double	x: 0 y: 0	Size (diameter) of the filter kernel, in pixel units (≥ 0). The standard deviation of the corresponding Gaussian is $\text{size}/2.4$. No filter is applied if $\text{size} < 1.2$. Negative values correspond to dilation, positive values to erosion. Both values should have the same sign.
Uniform / uniform	Boolean	Off	Apply the same amount of blur on X and Y.
Exponent / exponent	Integer	5	Exponent of the normalized power-weighted filter. Lower values give a smoother result. Default is 5.
Border Conditions / boundary	Choice	Nearest	<p>Specifies how pixel values are computed out of the image domain. This mostly affects values at the boundary of the image. If the image represents intensities, Nearest (Neumann) conditions should be used. If the image represents gradients or derivatives, Black (Dirichlet) boundary conditions should be used.</p> <p>Black (black): Dirichlet boundary condition: pixel values out of the image domain are zero.</p> <p>Nearest (nearest): Neumann boundary condition: pixel values out of the image domain are those of the closest pixel location in the image domain.</p>
Filter / filter	Choice	Quadratic	<p>Blurring filter. The quasi-Gaussian filter should be appropriate in most cases. The Gaussian filter is more isotropic (its impulse response has rotational symmetry), but slower.</p> <p>Quasi-Gaussian (quasigaussian): Quasi-Gaussian filter (0-order recursive Deriche filter, faster).</p> <p>Gaussian (gaussian): Gaussian filter (Van Vliet recursive Gaussian filter, more isotropic, slower).</p> <p>Box (box): Box filter - FIR (finite support / impulsional response).</p> <p>Triangle (triangle): Triangle/tent filter - FIR (finite support / impulsional response).</p> <p>Quadratic (quadratic): Quadratic filter - FIR (finite support / impulsional response).</p>
Expand RoD / expandRoD	Boolean	On	Expand the source region of definition by $1.5 \times \text{size}$ ($3.6 \times \text{sigma}$).
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.16 Fill node



This documentation is for version 1.0 of Fill.

Description

Add a constant color on the source image where the alpha channel not 0. You can control the blending between the original image and the constant color with the operator and the mix factor.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Convert to Group / <code>convertToGroup</code>	Button		Converts this node to a Group: the internal node-graph and the user parameters will become editable
Color / <code>Solid1color</code>	Color	r: 0 g: 0 b: 0	

Continued on next page

Table 114 – continued from previous page

Parameter / script name	Type	Default	Function
Operation / Mergeloperation	Choice	over	<p>atop: $Ab + B(1 - a)$ (a.k.a. src-atop)</p> <p>average: $(A + B) / 2$</p> <p>color: $\text{SetLum}(A, \text{Lum}(B))$</p> <p>color-burn: darken B towards A</p> <p>color-dodge: brighten B towards A</p> <p>conjoint-over: $A + B(1-a)/b$, A if $a > b$</p> <p>copy: A (a.k.a. src)</p> <p>difference: $\text{abs}(A-B)$ (a.k.a. absminus)</p> <p>disjoint-over: $A+B(1-a)/b$, $A+B$ if $a+b < 1$</p> <p>divide: A/B, 0 if $A < 0$ and $B < 0$</p> <p>exclusion: $A+B-2AB$</p> <p>freeze: $1-\sqrt{1-A}/B$</p> <p>from: $B-A$ (a.k.a. subtract)</p> <p>geometric: $2AB/(A+B)$</p> <p>grain-extract: $B - A + 0.5$</p> <p>grain-merge: $B + A - 0.5$</p> <p>hard-light: multiply if $A < 0.5$, screen if $A > 0.5$</p> <p>hue: $\text{SetLum}(\text{SetSat}(A, \text{Sat}(B)), \text{Lum}(B))$</p> <p>hypot: $\sqrt{A*A+B*B}$</p> <p>in: Ab (a.k.a. src-in)</p> <p>luminosity: $\text{SetLum}(B, \text{Lum}(A))$</p> <p>mask: Ba (a.k.a. dst-in)</p> <p>matte: $Aa + B(1-a)$ (unpremultiplied over)</p> <p>max: $\max(A, B)$ (a.k.a. lighten only)</p> <p>min: $\min(A, B)$ (a.k.a. darken only)</p> <p>minus: $A-B$</p> <p>multiply: AB, 0 if $A < 0$ and $B < 0$</p> <p>out: $A(1-b)$ (a.k.a. src-out)</p> <p>over: $A+B(1-a)$ (a.k.a. src-over)</p> <p>overlay: multiply if $B < 0.5$, screen if $B > 0.5$</p> <p>pinlight: if $B \geq 0.5$ then $\max(A, 2*B - 1)$, $\min(A, B * 2.0)$ else</p> <p>plus: $A+B$ (a.k.a. add)</p> <p>reflect: $A*A / (1 - B)$</p> <p>saturation: $\text{SetLum}(\text{SetSat}(B, \text{Sat}(A)), \text{Lum}(B))$</p> <p>screen: $A+B-AB$ if A or $B \leq 1$, otherwise $\max(A, B)$</p> <p>soft-light: burn-in if $A < 0.5$, lighten if $A > 0.5$</p> <p>stencil: $B(1-a)$ (a.k.a. dst-out)</p> <p>under: $A(1-b)+B$ (a.k.a. dst-over)</p> <p>xor: $A(1-b)+B(1-a)$</p>
Mask / MergelenableMask	Boolean	Off	
/ MergelmaskChannel	Choice		None

Continued on next page

Table 114 – continued from previous page

Parameter / script name	Type	Default	Function
Invert Mask / MergelmaskInvert	Boolean	Off	
Mix / Mergelmix	Double	1	

2.7.17 GMICExpr node



This documentation is for version 2.1 of GMICExpr.

Description

Quickly generate or process image from mathematical formula evaluated for each pixel. Full documentation for **G'MIC/CImg** expressions is reproduced below and available online from the **G'MIC help**. The only additions of this plugin are the predefined variables **T** (current time) and **K** (render scale).

Uses the 'fill' function from the CImg library. CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Sample expressions

- `'j(sin(y/100/K+T/10)*20*K, sin(x/100/K+T/10)*20*K)'` distorts the image with time-varying waves.
- `'0.5*(j(1)-j(-1))'` estimates the X-derivative of an image with a classical finite difference scheme.
- `'if(x%10==0,1,i)'` draws blank vertical lines on every 10th column of an image.
- `'sqrt(zr=-1.2+2.4*x/w; zi=-1.2+2.4*y/h; for(i=0, zr*zr+zi*zi<=4&&i<256, t=zr*zr-zi*zi+0.4; zi=2*zr*zi+0.2; zr=t; i=i+1))/255'` draws the Mandelbrot fractal (give it a 1024x1024 image as input).

Expression language

- The expression is evaluated for each pixel of the selected images.
- The mathematical parser understands the following set of functions, operators and variables:
 - **Usual operators:** `||` (logical or), `&&` (logical and), `|` (bitwise or), `&` (bitwise and), `!=`, `==`, `<=`, `>=`, `<`, `>`, `<<` (left bitwise shift), `>>` (right bitwise shift), `-`, `+`, `*`, `/`, `%` (modulo), `^` (power), `!` (logical not), `~` (bitwise not), `++`, `--`, `+=`, `-=`, `*=`, `/=`, `%=`, `&=`, `|=`, `^=`, `>>=`, `<<=` (in-place operators).
 - **Usual math functions:** `abs()`, `acos()`, `arg()`, `argkth()`, `argmax()`, `argmin()`, `asin()`, `atan()`, `atan2()`, `avg()`, `bool()`, `cbrt()`, `ceil()`, `cos()`, `cosh()`, `cut()`, `exp()`, `fact()`, `fibonacci()`, `floor()`, `gauss()`, `int()`, `isval()`, `isnan()`, `isinf()`, `isint()`, `isbool()`, `isfile()`, `isdir()`, `isin()`, `kth()`, `log()`, `log2()`, `log10()`, `max()`, `mean()`, `med()`, `min()`, `narg()`, `prod()`, `rol()` (left bit rotation), `ror()` (right bit rotation), `round()`, `sign()`, `sin()`, `sinc()`, `sinh()`, `sqrt()`, `std()`, `srand(_seed)`, `sum()`, `tan()`, `tanh()`, `variance()`, `xor()`. * `'atan2(y,x)'` is the version of `'atan()'` with two arguments `__y__` and `__x__` (as in C/C++). * `'permut(k,n,with_order)'` computes the number of permutations of `__k__` objects from a set

of `__n__` objects. * `'gauss(x,sigma)'` returns `'exp(-x\^2/(2\s\^2))/sqrt(2*pi*sigma\^2)'`. * `'cut(value,min,max)'` returns value if it is in range `__[min,max]__`, or `__min__` or `__max__` otherwise. * `'narg(a_1,...,a_N)'` returns the number of specified arguments (here, `__N__`). * `'arg(i,a_1,...,a_N)'` returns the `__ith__` argument `__a_i__`. * `'isval()'`, `'isnans()'`, `'isinf()'`, `'isint()'`, `'isbool()'` test the type of the given number or expression, and return `__0 (false)__` or `__1 (true)__`. * `'isfile()'` (resp. `'isdir()'`) returns `__0 (false)__` or `__1 (true)__` whether its argument is a path to an existing file (resp. to a directory) or not. * `'isin(v,a_1,...,a_n)'` returns `__0 (false)__` or `__1 (true)__` whether the first value `__v__` appears in the set of other values `__a_i__`. * `'argmin()'`, `'argmax()'`, `'kth()'`, `'max()'`, `'mean()'`, `'med()'`, `'min()'`, `'std()'`, `'sum()'` and `'variance()'` can be called with an arbitrary number of scalar/vector arguments. * `'round(value,rounding_value,direction)'` returns a rounded value. **'direction'** can be { **-1=to-lowest** | **0=to-nearest** | **1=to-highest** }.

– **Variable names** below are pre-defined. They can be overridden.

- * `'l'`: length of the associated list of images.
- * `'w'`: width of the associated image, if any (**0** otherwise).
- * `'h'`: height of the associated image, if any (**0** otherwise).
- * `'d'`: depth of the associated image, if any (**0** otherwise).
- * `'s'`: spectrum of the associated image, if any (**0** otherwise).
- * `'r'`: shared state of the associated image, if any (**0** otherwise).
- * `'wh'`: shortcut for width x height.
- * `'whd'`: shortcut for width x height x depth.
- * `'whds'`: shortcut for width x height x depth x spectrum (i.e. number of image values).
- * `'im','iM','ia','iv','is','ip','ic'`: Respectively the minimum, maximum, average, variance, sum, product and median value of the associated image, if any (**0** otherwise).
- * `'xm','ym','zm','cm'`: The pixel coordinates of the minimum value in the associated image, if any (**0** otherwise).
- * `'xM','yM','zM','cM'`: The pixel coordinates of the maximum value in the associated image, if any (**0** otherwise).
- * All these variables are considered as **constant values** by the math parser (for optimization purposes) which is indeed the case most of the time. Anyway, this might not be the case, if function `'resize(#ind, ...)'` is used in the math expression. If so, it is safer to invoke functions `'l()''`, `'w(_#ind)'`, `'h(_#ind)'`,... `'s(_#ind)'` and `'ic(_#ind)'` instead of the corresponding named variables.
- * `'i'`: current processed pixel value (i.e. value located at **(x,y,z,c)**) in the associated image, if any (**0** otherwise).
- * `'iN'`: Nth channel value of current processed pixel (i.e. value located at **(x,y,z,N)**) in the associated image, if any (**0** otherwise). **'N'** must be an integer in range **[0,9]**.
- * `'R','G','B'` and `'A'` are equivalent to `'i0'`, `'i1'`, `'i2'` and `'i3'` respectively.
- * `'I'`: current vector-valued processed pixel in the associated image, if any (**0** otherwise). The number of vector components is equal to the number of image channels (e.g. **I = [R,G,B]** for a **RGB** image).
- * You may add `'#ind'` to any of the variable name above to retrieve the information for any numbered image **[ind]** of the list (when this makes sense). For instance `'ia#0'` denotes the average value of the first image of the list).
- * `'x'`: current processed column of the associated image, if any (**0** otherwise).

- * 'y': current processed row of the associated image, if any (**0** otherwise).
- * 'z': current processed slice of the associated image, if any (**0** otherwise).
- * 'c': current processed channel of the associated image, if any (**0** otherwise).
- * 't': thread id when an expression is evaluated with multiple threads (**0** means 'master thread').
- * 'T': current time [OpenFX-only].
- * 'K': render scale (1 means full scale, 0.5 means half scale) [OpenFX-only].
- * 'e': value of e, i.e. **2.71828...**
- * 'pi': value of pi, i.e. **3.1415926...**
- * 'u': a random value between [**0,1**], following a uniform distribution.
- * 'g': a random value, following a gaussian distribution of variance 1 (roughly in [**-6,6**]).
- * 'interpolation': value of the default interpolation mode used when reading pixel values with the pixel access operators (i.e. when the interpolation argument is not explicitly specified, see below for more details on pixel access operators). Its initial default value is **0**.
- * 'boundary': value of the default boundary conditions used when reading pixel values with the pixel access operators (i.e. when the boundary condition argument is not explicitly specified, see below for more details on pixel access operators). Its initial default value is **0**.

– **Vector calculus:** Most operators are also able to work with vector-valued elements.

- * '[a0, a1, . . . , aN]' defines a (**N+1**)-dimensional vector with scalar coefficients **ak**.
- * 'vectorN(a0, a1, . . . ,)' does the same, with the **ak** being repeated periodically if only a few are specified.
- * In both previous expressions, the **ak** can be vectors themselves, to be concatenated into a single vector.
- * The scalar element **ak** of a vector **X** is retrieved by 'X[k]'.
 - * The sub-vector [X[p]..X[p+q-1]] (of size **q**) of a vector **X** is retrieved by 'X[p, q]'.
- * Equality/inequality comparisons between two vectors is done with operators '==' and '!='.
- * Some vector-specific functions can be used on vector values: 'cross(X, Y)' (cross product), 'dot(X, Y)' (dot product), 'size(X)' (vector dimension), 'sort(X, _is_increasing, _chunk_size)' (sorting values), 'reverse(A)' (reverse order of components), 'shift(A, _length, _boundary_conditions)' and 'same(A, B, _nb_vals, _is_case_sensitive)' (vector equality test).
- * Function 'normP(u1, . . . , un)' computes the LP-norm of the specified vector (**P** being an **unsigned integer** constant or 'inf'). If **P** is omitted, the L2 norm is used.
- * Function 'resize(A, size, _interpolation, _boundary_conditions)' returns a resized version of a vector '**A**' with specified interpolation mode. '**interpolation**' can be { **-1=none (memory content) | 0=none | 1=nearest | 2=average | 3=linear | 4=grid | 5=bicubic | 6=lanczos** }, and '**boundary_conditions**' can be { **0=dirichlet | 1=neumann | 2=periodic | 3=mirror** }.
- * Function 'find(A, B, _is_forward, _starting_indice)' returns the index where sub-vector **B** appears in vector **A**, (or **-1** if **B** is not found in **A**). Argument **A** can be also replaced by an image indice **#ind**.
- * A **2-dimensional** vector may be seen as a complex number and used in those particular functions/operators: '**' (complex multiplication), '/' (complex division), '^' (complex exponentiation), '**=' (complex self-multiplication), '/=' (complex self-division), '^=' (complex self-exponentiation), 'cabs()' (complex modulus), 'carg()' (complex argument), 'cconj()' (complex conjugate), 'cexp()' (complex exponential) and 'clog()' (complex logarithm).

- * A MN -dimensional vector may be seen as a $M \times N$ matrix and used in those particular functions/operators: `'*` (matrix-vector multiplication), `'det(A)'` (determinant), `'diag(V)'` (diagonal matrix from a vector), `'eig(A)'` (eigenvalues/eigenvectors), `'eye(n)'` ($n \times n$ identity matrix), `'inv(A)'` (matrix inverse), `'mul(A,B,_nb_colsB)'` (matrix-matrix multiplication), `'pseudoinv(A,_nb_colsA)'`, `'rot(u,v,w,angle)'` (3d rotation matrix), `'rot(angle)'` (2d rotation matrix), `'solve(A,B,_nb_colsB)'` (least-square solver of linear system $A.X = B$), `'svd(A,_nb_colsA)'` (singular value decomposition), `'trace(A)'` (matrix trace) and `'transp(A,_nb_colsA)'` (matrix transpose). Argument `'nb_colsB'` may be omitted if it is equal to **1**.
- * Specifying a vector-valued math expression as an argument of a command that operates on image values (e.g. `'fill'`) modifies the whole spectrum range of the processed image(s), for each spatial coordinates (**x,y,z**). The command does not loop over the **C**-axis in this case.
- **String manipulation:** Character strings are defined and managed as vectors objects. Dedicated functions and initializers to manage strings are
 - * `['string']` and `'string'` define a vector whose values are the ascii codes of the specified **character string** (e.g. `'foo'` is equal to `[102,111,111]`).
 - * `_'character'` returns the (scalar) ascii code of the specified character (e.g. `_'A'` is equal to **65**).
 - * A special case happens for **empty** strings: Values of both expressions `['']` and `''` are **0**.
 - * Functions `'lowercase()'` and `'uppercase()'` return string with all string characters lowercased or uppercased.
 - * Function `'stov(str,_starting_indice,_is_strict)'` parses specified string `'str'` and returns the value contained in it.
 - * Function `'vtos(expr,_nb_digits,_siz)'` returns a vector of size `'siz'` which contains the ascii representation of values described by expression `'expr'`. `'nb_digits'` can be `{ -1=auto-reduced | 0=all | >0=max number of digits }`.
 - * Function `'echo(str1,str2,...,strN)'` prints the concatenation of given string arguments on the console.
 - * Function `'cats(str1,str2,...,strN,siz)'` returns the concatenation of given string arguments as a new vector of size `'siz'`.
- **Special operators** can be used:
 - * `';`: expression separator. The returned value is always the last encountered expression. For instance expression `'1;2;pi'` is evaluated as `'pi'`.
 - * `'=`: variable assignment. Variables in mathematical parser can only refer to numerical values (vectors or scalars). Variable names are case-sensitive. Use this operator in conjunction with `';` to define more complex evaluable expressions, such as `'t=cos(x);3*t^2+2*t+1'`. These variables remain **local** to the mathematical parser and cannot be accessed outside the evaluated expression.
 - * Variables defined in math parser may have a **constant** property, by specifying keyword `const` before the variable name (e.g. `const foo = pi/4;`). The value set to such a variable must be indeed a `constant scalar`. Constant variables allows certain types of optimizations in the math JIT compiler.
- The following **specific functions** are also defined:
 - * `'u(max)'` or `'u(min,max)'`: return a random value between `[0,max]` or `[min,max]`, following a uniform distribution.
 - * `'i(_a,_b,_c,_d,_interpolation_type,_boundary_conditions)'`: return the value of the pixel located at position (**a,b,c,d**) in the associated image, if any (**0** otherwise). `'interpolation_type'` can be `{ 0=nearest neighbor | other=linear }`. `'boundary_conditions'` can be `{ 0=dirichlet | 1=neumann | 2=periodic | 3=mirror }`. Omitted coordinates are replaced by their default values which are respectively `x`, `y`, `z`, `c`, `interpolation` and `boundary`.

For instance command `'fill 0.5*(i(x+1)-i(x-1))'` will estimate the X-derivative of an image with a classical finite difference scheme.

- * `'j(dx,dy,dz,dc,interpolation_type,boundary_conditions)'` does the same for the pixel located at position `(x+dx,y+dy,z+dz,c+dc)` (pixel access relative to the current coordinates).
- * `'i[offset,boundary_conditions]'` returns the value of the pixel located at specified **'offset'** in the associated image buffer (or **0** if offset is out-of-bounds).
- * `'j[offset,boundary_conditions]'` does the same for an offset relative to the current pixel coordinates `(x,y,z,c)`.
- * `'i(#ind,_x,_y,_z,_c,interpolation,boundary_conditions)'`, `'j(#ind,_dx,_dy,_dz,_dc,interpolation,boundary_conditions)'`, `'i[#ind,offset,boundary_conditions]'` and `'i[offset,boundary_conditions]'` are similar expressions used to access pixel values for any numbered image **[ind]** of the list.
- * `'I/J[offset,boundary_conditions]'` and `'I/J(#ind,_x,_y,_z,_c,interpolation,boundary_conditions)'` do the same as `'i/j[offset,boundary_conditions]'` and `'i/j(#ind,_x,_y,_z,_c,interpolation,boundary_conditions)'` but return a vector instead of a scalar (e.g. a vector `[R,G,B]` for a pixel at `(a,b,c)` in a color image).
- * `'sort(#ind,is_increasing,_axis)'` sorts the values in the specified image **[ind]**.
- * `'crop(_#ind,_x,_y,_z,_c,_dx,_dy,_dz,_dc,boundary_conditions)'` returns a vector whose values come from the cropped region of image **[ind]** (or from default image selected if 'ind' is not specified). Cropped region starts from point `(x,y,z,c)` and has a size of `dx x dy x dz x dc`. Arguments for coordinates and sizes can be omitted if they are not ambiguous (e.g. `'crop(#ind,x,y,dx,dy)'` is a valid invocation of this function).
- * `'draw(_#ind,S,x,y,z,c,dx,dy,dz,dc,opacity,_M,_max_M)'` draws a sprite **S** in image **[ind]** (or in default image selected if 'ind' is not specified) at coordinates `(x,y,z,c)`. The size of the sprite `dx x dy x dz x dc` must be specified. You can also specify a corresponding opacity mask **M** if its size matches **S**.
- * `'resize(#ind,w,_h,_d,_s,_interp,boundary_conditions,cx,_cy,_cz,_cc)'` resizes an image of the associated list with specified dimension and interpolation method. When using this, function, you should consider retrieving the (non-constant) image dimensions using the dynamic functions `'w(_#ind)'`, `'h(_#ind)'`, `'d(_#ind)'`, `'s(_#ind)'`, `'wh(_#ind)'`, `'whd(_#ind)'` and `'whds(_#ind)'` instead of the corresponding constant variables.
- * `'if(condition,expr_then,expr_else)'`: return value of `'expr_then'` or `'expr_else'`, depending on the value of `'condition'` (**0=false, other=true**). `'expr_else'` can be omitted in which case **0** is returned if the condition does not hold. Using the ternary operator `'condition?expr_then[:expr_else]'` gives an equivalent expression. For instance, expressions `'if(x%10==0,255,i)'` and `'x%10?i:255'` both draw blank vertical lines on every 10th column of an image.
- * `'dowhile(expression,_condition)'` repeats the evaluation of `'expression'` until `'condition'` vanishes (or until `'expression'` vanishes if no `'condition'` is specified). For instance, the expression: `'if(N<2,N,n=N-1;F0=0;F1=1;dowhile(F2=F0+F1;F0=F1;F1=F2,n=n-1))'` returns the Nth value of the Fibonacci sequence, for **N>=0** (e.g., **46368** for **N=24**). `'dowhile(expression,condition)'` always evaluates the specified expression at least once, then check for the loop condition. When done, it returns the last value of `'expression'`.
- * `'for(init,condition,_procedure,body)'` first evaluates the expression `'init'`, then iteratively evaluates `'body'` (followed by `'procedure'` if specified) while `'condition'` is verified (i.e. not zero). It may happen that no iteration is done, in which case the function returns **nan**. Otherwise, it returns the last value of `'body'`. For instance, the expression: `'if(N<2,`

`N, for (n=N; F0=0; F1=1, n=n-1, F2=F0+F1; F0=F1; F1=F2))` returns the **Nth** value of the Fibonacci sequence, for **N>=0** (e.g., **46368** for **N=24**).

- * `'whiledo(condition, expression)'` is exactly the same as `'for(init, condition, expression)'` without the specification of an initializing expression.
- * `'break()'` and `'continue()'` respectively breaks and continues the current running bloc (loop, init or main environment).
- * `'date(attr, path)'` returns the date attribute for the given 'path' (file or directory), with **'attr'** being { **0=year** | **1=month** | **2=day** | **3=day of week** | **4=hour** | **5=minute** | **6=second** }, or a vector of those values.
- * `'date(_attr)'` returns the specified attribute for the current (locale) date.
- * `'print(expr1, expr2, ...)'` or `'print(#ind)'` prints the value of the specified expressions (or image information) on the console, and returns the value of the last expression (or **nan** in case of an image). Function `'prints(expr)'` also prints the string composed of the ascii characters defined by the vector-valued expression (e.g. `'prints('Hello')'`).
- * `'debug(expression)'` prints detailed debug information about the sequence of operations done by the math parser to evaluate the expression (and returns its value).
- * `'display(_X, _w, _h, _d, _s)'` or `'display(#ind)'` display the contents of the vector 'X' (or specified image) and wait for user events. if no arguments are provided, a memory snapshot of the math parser environment is displayed instead.
- * `'init(expression)'` and `'end(expression)'` evaluates the specified expressions only once, respectively at the beginning and end of the evaluation procedure, and this, even when multiple evaluations are required (e.g. in `'fill init(foo=0); ++foo'`).
- * `'copy(dest, src, _nb_elts, _inc_d, _inc_s, _opacity)'` copies an entire memory block of `'nb_elts'` elements starting from a source value `'src'` to a specified destination `'dest'`, with increments defined by `'inc_d'` and `'inc_s'` respectively for the destination and source pointers.
- * `'unref(a, b, ...)'` destroys references to the named variable given as arguments.
- * `'stats(_#ind)'` returns the statistics vector of the running image [**ind**], i.e the vector [**im, iM, ia, iv, xm, ym, zm, cm, xM, yM, zM, cM, is, ip**] (14 values).
- * `'_(expr)'` just ignores its arguments (mainly useful for debugging).

– User-defined macros:

- * Custom macro functions can be defined in a math expression, using the assignment operator '=', e.g. `'foo(x, y) = cos(x + y); result = foo(1, 2) + foo(2, 3)'`.
- * Trying to override a built-in function (e.g. `'abs()'`) has no effect.
- * Overloading macros with different number of arguments is possible. Re-defining a previously defined macro with the same number of arguments discards its previous definition.
- * Macro functions are indeed processed as **macros** by the mathematical evaluator. You should avoid invoking them with arguments that are themselves results of assignments or self-operations. For instance, `'foo(x) = x + x; z = 0; foo(++z)'` returns **'4'** rather than expected value **'2'**.
- * When substituted, macro arguments are placed inside parentheses, except if a number sign '#' is located just before or after the argument name. For instance, expression `'foo(x, y) = x*y; foo(1+2, 3)'` returns **'9'** (being substituted as `'(1+2)*(3)'`), while expression `'foo(x, y) = x*y#; foo(1+2, 3)'` returns **'7'** (being substituted as `'1+2*3'`).
- * Number signs appearing between macro arguments function actually count for 'empty' separators. They may be used to force the substitution of macro arguments in unusual places, e.g. as in `'str(N) = ['I like N#']'`.

– Multi-threaded and in-place evaluation:

- * If your image data are large enough and you have several CPUs available, it is likely that the math expression passed to a 'fill' or 'input' command is evaluated in parallel, using multiple computation threads.
 - * Starting an expression with ':' or '*' forces the evaluations required for an image to be run in parallel, even if the amount of data to process is small (beware, it may be slower to evaluate in this case!). Specify ':' (instead of '*') to avoid possible image copy done before evaluating the expression (this saves memory, but do this only if you are sure this step is not required!)
 - * If the specified expression starts with '>' or '<', the pixel access operators 'i()', 'i[]', 'j()' and 'j[]' return values of the image being currently modified, in forward ('>') or backward ('<') order. The multi-threading evaluation of the expression is also disabled in this case.
 - * Function 'critical(operands)' forces the execution of the given operands in a single thread at a time.
- Expressions 'i(_#ind,x,_y,_z,_c)=value', 'j(_#ind,x,_y,_z,_c)=value', 'i[_#ind,offset]=value' and 'j[_#ind,offset]=value' set a pixel value at a different location than the running one in the image **[ind]** (or in the associated image if argument '#ind' is omitted), either with global coordinates/offsets (with 'i(...)' and 'i[...]'), or relatively to the current position (**x,y,z,c**) (with 'j(...)' and 'j[...]'). These expressions always return 'value'.

Inputs

Input	Description	Optional
Source		Yes
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Expression / expression	String	i	G'MIC/CImg expression, see the plugin description/help, or http://gmic.eu/reference.shtml#section9
Help... / help	Button		Display help for writing GMIC expressions.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.18 Glow node



This documentation is for version 1.0 of Glow.

Description

A glow effect based on the bloom filter node. The mask input limits the area where the glowing elements are. It does not cut off the shine produced by the glow.

For more interesting looks there are some additional features like stretch, rotation and postgrade.

Written by PostPollux

Inputs

Input	Description	Optional
1		No
mask		Yes

Controls

Parameter / script name	Type	Default	Function
Convert to Group / <code>convertToGroup</code>	Button		Converts this node to a Group: the internal node-graph and the user parameters will become editable
Isolate Highlights / <code>isolateHighlights</code>	Double	0.25	<p>“Isolate Highlights” controls on which parts of the image the glow is applied.</p> <p>high value -> only bright areas</p>
Isolate Hue / <code>isolateHue</code>	Boolean	Off	If checked, the picked hue will additionally limit the areas of the glow.
pick Hue / <code>HSVTool1srcColor</code>	Color	r: 0 g: 0 b: 0	<p>Pick a color in the viewport to isolate it’s hue.</p> <p>The glow will only be applied to those areas, if ‘Isolate Hue’ is checked.</p>
GlowMap Preview / <code>preview</code>	Boolean	Off	When checked, the output will be the glowmap. Use this as a preview to adjust the highlight and hue isolation.
Size / <code>size</code>	Double	3	This will change the size of the glow.
Ratio / <code>BloombloomRatio</code>	Double	2	
Iterations / <code>iterations</code>	Integer	7	Number of blur kernels of the bloom filter. The original implementation uses a value of 5. Higher values give a wider of heavier tail (the size of the largest blur kernel is $2^{**}bloomCount*size$). A count of 1 is just the original blur.

Continued on next page

Table 116 – continued from previous page

Parameter / script name	Type	Default	Function
Filter / Bloomfilter	Choice	Quasi-Gaussian	<p>Blurring filter. The quasi-Gaussian filter should be appropriate in most cases. The Gaussian filter is more isotropic (its impulse response has rotational symmetry), but slower.</p> <p>Quadratic might also look a bit better, but it is slower, too.</p> <p>Quasi-Gaussian (quasigaussian): Quasi-Gaussian filter (0-order recursive Deriche filter, faster) - IIR (infinite support / impulsional response).</p> <p>Gaussian (gaussian): Gaussian filter (Van Vliet recursive Gaussian filter, more isotropic, slower) - IIR (infinite support / impulsional response).</p> <p>Box (box): Box filter - FIR (finite support / impulsional response).</p> <p>Triangle (triangle): Triangle/tent filter - FIR (finite support / impulsional response).</p> <p>Quadratic (quadratic): Quadratic filter - FIR (finite support / impulsional response).</p>
Stretch / stretch	Double	0	<p>Stretch the glow!</p> <p>0 -> uniform proportions 1 -> blur only in one direction</p>
Rotate / rotate	Double	0	This will rotate your stretched glow. If “Stretch” is 0 it won’t have any effect.
Gain / PostGradeMasterGain	Color	r: 1 g: 1 b: 1 a: 1	
Gamma / PostGradeMasterGamma	Color	r: 1 g: 1 b: 1 a: 1	
Saturation / PostGradeMasterSaturation	Color	r: 1 g: 1 b: 1 a: 1	
Screen / screen	Boolean	Off	<p>If checked, the bloomed image will be screened on top of the input image. This helps to preserve the highlights in your image.</p> <p>By default it is added on top of the input image. (plus)</p>
Add Input / addInput	Boolean	Off	Add the input image to the bloomed one.
Glow Only / glowOnly	Boolean	Off	<p>The output will only be the glow effect itself.</p> <p>You can add it on top of your source with a plus-merge or a screen-merge, later.</p>

Continued on next page

Table 116 – continued from previous page

Parameter / script name	Type	Default	Function
expand RoD / expRoD	Boolean	Off	By default the Region of Definition (RoD) will be cropped to the input RoD. Use this option, if you want the glow effect to be available even outside your input format, and thus being not cropped. As the blur sizes of the bloom node can get very big, this may lead to a very big RoD! Especially, if you use a lot of iterations.
apply on alpha / alpha	Boolean	Off	Controls if the glow is applied to the alpha channel, too.

2.7.19 GodRays node



This documentation is for version 1.0 of GodRays.

Description

Average an image over a range of transforms.

This can be used to create crepuscular rays (also called God rays) by setting the scale and center parameters: scale governs the length of rays, and center should be set to the Sun or light position (which may be outside of the image).

Setting toColor to black and gamma to 1 causes an exponential decay which is very similar to the real crepuscular rays.

This can also be used to create directional blur using a fixed number of samples (as opposed to DirBlur, which uses an adaptive sampling method).

This plugin concatenates transforms upstream.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Rotate / rotate	Double	0	Rotation angle in degrees around the Center. Can also be adjusted by clicking and dragging the rotation bar in the Viewer.
Scale / scale	Double	x: 1 y: 1	Scale factor along the x and y axes. Can also be adjusted by clicking and dragging the outer circle or the diameter handles in the Viewer.
Uniform / uniform	Boolean	Off	Use the X scale for both directions

Continued on next page

Table 117 – continued from previous page

Parameter / script name	Type	Default	Function
Skew X / skewX	Double	0	Skew along the x axis. Can also be adjusted by clicking and dragging the skew bar in the Viewer.
Skew Y / skewY	Double	0	Skew along the y axis.
Skew Order / skewOrder	Choice	XY	The order in which skew transforms are applied: X then Y, or Y then X. XY YX
Amount / transformAmount	Double	1	Amount of transform to apply. 0 means the transform is identity, 1 means to apply the full transform.
Center / center	Double	x: 0.5 y: 0.5	Center of rotation and scale.
Reset Center / resetCenter	Button		Reset the position of the center to the center of the input region of definition
Interactive Update / interactive	Boolean	On	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Invert / invert	Boolean	Off	Invert the transform.
Filter / filter	Choice	Cubic	Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+). Impulse (impulse) : (nearest neighbor / box) Use original values. Box (box) : Integrate the source image over the bounding box of the back-transformed pixel. Bilinear (bilinear) : (tent / triangle) Bilinear interpolation between original values. Cubic (cubic) : (cubic spline) Some smoothing. Keys (keys) : (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*). Simon (simon) : Some smoothing, plus medium sharpening (*). Rifman (rifman) : Some smoothing, plus significant sharpening (*). Mitchell (mitchell) : Some smoothing, plus blurring to hide pixelation (*+). Parzen (parzen) : (cubic B-spline) Greatest smoothing of all filters (+). Notch (notch) : Flat smoothing (which tends to hide moire' patterns) (+).
Clamp / clamp	Boolean	Off	Clamp filter output within the original range - useful to avoid negative values in mattes
Black outside / black_outside	Boolean	Off	Fill the area outside the source image with black
From Color / fromColor	Color	r: 1 g: 1 b: 1 a: 1	Color by which the initial image is multiplied.
To Color / toColor	Color	r: 1 g: 1 b: 1 a: 1	Color by which the final image is multiplied.
Gamma / gamma	Color	r: 1 g: 1 b: 1 a: 1	Gamma space in which the colors are interpolated. Higher values yield brighter intermediate images.
Steps / steps	Integer	5	The number of intermediate images is 2^steps, i.e. 32 for steps=5.
Max / max	Boolean	Off	Output the brightest value at each pixel rather than the average.

Continued on next page

Table 117 – continued from previous page

Parameter / script name	Type	Default	Function
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.20 Inpaint node



This documentation is for version 1.0 of Inpaint.

Description

Inpaint (a.k.a. content-aware fill) the areas indicated by the Mask input using patch-based inpainting.

Be aware that this filter may produce different results on each frame of a video, even if there is little change in the video content. To inpaint areas with lots of details, it may be better to inpaint on a single frame and paste the inpainted area on other frames (if a transform is also required to match the other frames, it may be computed by tracking).

A tutorial on using this filter can be found at <http://blog.patdavid.net/2014/02/getting-around-in-gimp-gmic-inpainting.html>

The algorithm is described in the two following publications:

“A Smarter Exemplar-based Inpainting Algorithm using Local and Global Heuristics for more Geometric Coherence.” (M. Daisy, P. Buysens, D. Tschumperlé, O. Lezoray). IEEE International Conference on Image Processing (ICIP’14), Paris/France, Oct. 2014

and

“A Fast Spatial Patch Blending Algorithm for Artefact Reduction in Pattern-based Image Inpainting.” (M. Daisy, D. Tschumperlé, O. Lezoray). SIGGRAPH Asia 2013 Technical Briefs, Hong-Kong, November 2013.

Uses the ‘inpaint’ plugin from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>). The ‘inpaint’ CImg plugin is distributed under the CeCILL (compatible with the GNU GPL) license.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Patch Size / patchSize	Integer	7	
Lookup Size / lookupSize	Double	16	
Lookup Factor / lookupFactor	Double	0.1	
Blend Size / blendSize	Double	1.2	
Blend Threshold / blendThreshold	Double	0	
Blend Decay / blendDecay	Double	0.05	
Blend Scales / blendScales	Integer	10	
Allow Outer Blending / isBlendOuter	Boolean	On	
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.21 Laplacian node



This documentation is for version 4.0 of Laplacian.

Description

Blur input stream, and subtract the result from the input image. This is not a mathematically correct Laplacian (which would be the sum of second derivatives over X and Y).

Uses the ‘vanvliet’ and ‘deriche’ functions from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Size / size	Double	x: 3 y: 3	Size (diameter) of the filter kernel, in pixel units (≥ 0). The standard deviation of the corresponding Gaussian is $\text{size}/2.4$. No filter is applied if $\text{size} < 1.2$.
Uniform / uniform	Boolean	Off	Apply the same amount of blur on X and Y.
Filter / filter	Choice	Gaussian	<p>Blurring filter. The quasi-Gaussian filter should be appropriate in most cases. The Gaussian filter is more isotropic (its impulse response has rotational symmetry), but slower.</p> <p>Quasi-Gaussian (quasigaussian): Quasi-Gaussian filter (0-order recursive Deriche filter, faster) - IIR (infinite support / impulsional response).</p> <p>Gaussian (gaussian): Gaussian filter (Van Vliet recursive Gaussian filter, more isotropic, slower) - IIR (infinite support / impulsional response).</p> <p>Box (box): Box filter - FIR (finite support / impulsional response).</p> <p>Triangle (triangle): Triangle/tent filter - FIR (finite support / impulsional response).</p> <p>Quadratic (quadratic): Quadratic filter - FIR (finite support / impulsional response).</p>
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.22 Matrix3x3 node

This documentation is for version 1.0 of Matrix3x3.

Description

Compute the convolution of the input image with the specified matrix.

This works by multiplying each surrounding pixel of the input image with the corresponding matrix coefficient (the current pixel is at the center of the matrix), and summing up the results.

For example $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ produces an edge detection filter (which is an approximation of the Laplacian filter) by multiplying the center pixel by 8 and the surrounding pixels by -1, and then adding the nine values together to calculate the new value of the center pixel.

Uses the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
/matrix31	Double	0	Matrix coefficient.
/matrix32	Double	0	Matrix coefficient.
/matrix33	Double	0	Matrix coefficient.
/matrix21	Double	0	Matrix coefficient.
/matrix22	Double	0	Matrix coefficient.
/matrix23	Double	0	Matrix coefficient.
/matrix11	Double	0	Matrix coefficient.
/matrix12	Double	0	Matrix coefficient.
/matrix13	Double	0	Matrix coefficient.
Normalize / normalize	Boolean	Off	Normalize the matrix coefficients so that their sum is 1.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.23 Matrix5x5 node

This documentation is for version 1.0 of Matrix5x5.

Description

Compute the convolution of the input image with the specified matrix.

This works by multiplying each surrounding pixel of the input image with the corresponding matrix coefficient (the current pixel is at the center of the matrix), and summing up the results.

For example $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ produces an edge detection filter (which is an approximation of the Laplacian filter) by multiplying the center pixel by 8 and the surrounding pixels by -1, and then adding the nine values together to calculate the new value of the center pixel.

Uses the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
/matrix51	Double	0	Matrix coefficient.
/matrix52	Double	0	Matrix coefficient.

Continued on next page

Table 121 – continued from previous page

Parameter / script name	Type	Default	Function
/matrix53	Double	0	Matrix coefficient.
/matrix54	Double	0	Matrix coefficient.
/matrix55	Double	0	Matrix coefficient.
/matrix41	Double	0	Matrix coefficient.
/matrix42	Double	0	Matrix coefficient.
/matrix43	Double	0	Matrix coefficient.
/matrix44	Double	0	Matrix coefficient.
/matrix45	Double	0	Matrix coefficient.
/matrix31	Double	0	Matrix coefficient.
/matrix32	Double	0	Matrix coefficient.
/matrix33	Double	0	Matrix coefficient.
/matrix34	Double	0	Matrix coefficient.
/matrix35	Double	0	Matrix coefficient.
/matrix21	Double	0	Matrix coefficient.
/matrix22	Double	0	Matrix coefficient.
/matrix23	Double	0	Matrix coefficient.
/matrix24	Double	0	Matrix coefficient.
/matrix25	Double	0	Matrix coefficient.
/matrix11	Double	0	Matrix coefficient.
/matrix12	Double	0	Matrix coefficient.
/matrix13	Double	0	Matrix coefficient.
/matrix14	Double	0	Matrix coefficient.
/matrix15	Double	0	Matrix coefficient.
Normalize / normalize	Boolean	Off	Normalize the matrix coefficients so that their sum is 1.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.24 Median node

This documentation is for version 2.0 of Median.

Description

Apply a median filter to input images. Pixel values within a square box of the given size around the current pixel are sorted, and the median value is output if it does not differ from the current value by more than the given. Median filtering is performed per-channel.

Uses the ‘blur_median’ function from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Size / size	Integer	1	Width and height of the structuring element is 2*size+1, in pixel units (≥ 0).
Threshold / threshold	Double	0	Threshold used to discard pixels too far from the current pixel value in the median computation. A threshold value of zero disables the threshold.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.25 Shadertoy node



This documentation is for version 1.0 of Shadertoy.

Description

Apply a [Shadertoy](#) fragment shader.

This plugin implements [Shadertoy 0.8.8](#), but multipass shaders and sound are not supported. Some multipass shaders can still be implemented by chaining several Shadertoy nodes, one for each pass.

[Shadertoy 0.8.8](#) uses WebGL 1.0 (a.k.a. [GLSL ES 1.0](#) from GLES 2.0), based on [GLSL 1.20](#)

Note that the more recent [Shadertoy 0.9.1](#) uses WebGL 2.0 (a.k.a. [GLSL ES 3.0](#) from GLES 3.0), based on [GLSL 3.3](#)

This help only covers the parts of GLSL ES that are relevant for Shadertoy. For the complete specification please have a look at [GLSL ES 1.0 specification](#) or pages 3 and 4 of the [OpenGL ES 2.0 quick reference card](#). See also the [Shadertoy/GLSL tutorial](#).

Image shaders

Image shaders implement the `mainImage()` function in order to generate the procedural images by computing a color for each pixel. This function is expected to be called once per pixel, and it is responsibility of the host application to provide the right inputs to it and get the output color from it and assign it to the screen pixel. The prototype is:

```
void mainImage( out vec4 fragColor, in vec2 fragCoord );
```

where `fragCoord` contains the pixel coordinates for which the shader needs to compute a color. The coordinates are in pixel units, ranging from 0.5 to resolution-0.5, over the rendering surface, where the resolution is passed to the shader through the `iResolution` uniform (see below).

The resulting color is gathered in `fragColor` as a four component vector.

Language:

- **Preprocessor:** # #define #undef #if #ifdef #ifndef #else #elif #endif #error #pragma #extension #version #line
- **Operators:** () + - ! * / % < > <= >= == != && ||
- **Comments:** // /* */
- **Types:** void bool int float vec2 vec3 vec4 bvec2 bvec3 bvec4 ivec2 ivec3 ivec4 mat2 mat3 mat4 sampler2D
- **Function Parameter Qualifiers:** [STRIKEOUT:none], in, out, inout
- **Global Variable Qualifiers:** const
- **Vector Components:** .xyzw .rgba .stpq
- **Flow Control:** if else for return break continue
- **Output:** vec4 fragColor
- **Input:** vec2 fragCoord

Built-in Functions (details)

Angle and Trigonometry Functions

- *type* radians (*type* degrees)
- *type* degrees (*type* radians)
- *type* sin (*type* angle)
- *type* cos (*type* angle)
- *type* tan (*type* angle)
- *type* asin (*type* x)
- *type* acos (*type* x)
- *type* atan (*type* y, *type* x)
- *type* atan (*type* y_over_x)

Exponential Functions

- *type* pow (*type* x, *type* y)
- *type* exp (*type* x)
- *type* log (*type* x)
- *type* exp2 (*type* x)
- *type* log2 (*type* x)
- *type* sqrt (*type* x)
- *type* inversesqrt (*type* x)

Common Functions

- *type* abs (*type* x)
- *type* sign (*type* x)
- *type* floor (*type* x)
- *type* ceil (*type* x)
- *type* fract (*type* x)
- *type* mod (*type* x, float y)
- *type* mod (*type* x, *type* y)
- *type* min (*type* x, *type* y)
- *type* min (*type* x, float y)
- *type* max (*type* x, *type* y)
- *type* max (*type* x, float y)
- *type* clamp (*type* x, *type* minV, *type* maxV)
- *type* clamp (*type* x, float minV, float maxV)
- *type* mix (*type* x, *type* y, *type* a)
- *type* mix (*type* x, *type* y, float a)
- *type* step (*type* edge, *type* x)
- *type* step (float edge, *type* x)
- *type* smoothstep (*type* a, *type* b, *type* x)
- *type* smoothstep (float a, float b, *type* x)

Geometric Functions

- float length (*type* x)
- float distance (*type* p0, *type* p1)
- float dot (*type* x, *type* y)
- vec3 cross (vec3 x, vec3 y)
- *type* normalize (*type* x)
- *type* faceforward (*type* N, *type* I, *type* Nref)
- *type* reflect (*type* I, *type* N)
- *type* refract (*type* I, *type* N, float eta)

Matrix Functions

- mat matrixCompMult (mat x, mat y)

Vector Relational Functions

- `bvec lessThan(vec x, vec y)`
- `bvec lessThan(ivec x, ivec y)`
- `bvec lessThanEqual(vec x, vec y)`
- `bvec lessThanEqual(ivec x, ivec y)`
- `bvec greaterThan(vec x, vec y)`
- `bvec greaterThan(ivec x, ivec y)`
- `bvec greaterThanEqual(vec x, vec y)`
- `bvec greaterThanEqual(ivec x, ivec y)`
- `bvec equal(vec x, vec y)`
- `bvec equal(ivec x, ivec y)`
- `bvec equal(bvec x, bvec y)`
- `bvec notEqual(vec x, vec y)`
- `bvec notEqual(ivec x, ivec y)`
- `bvec notEqual(bvec x, bvec y)`
- `bool any(bvec x)`
- `bool all(bvec x)`
- `bvec not(bvec x)`

Texture Lookup Functions

- `vec4 texture2D(sampler2D sampler, vec2 coord)`
- `vec4 texture2D(sampler2D sampler, vec2 coord, float bias)`
- `vec4 textureCube(samplerCube sampler, vec3 coord)`
- `vec4 texture2DProj(sampler2D sampler, vec3 coord)`
- `vec4 texture2DProj(sampler2D sampler, vec3 coord, float bias)`
- `vec4 texture2DProj(sampler2D sampler, vec4 coord)`
- `vec4 texture2DProj(sampler2D sampler, vec4 coord, float bias)`
- `vec4 texture2DLodEXT(sampler2D sampler, vec2 coord, float lod)`
- `vec4 texture2DProjLodEXT(sampler2D sampler, vec3 coord, float lod)`
- `vec4 texture2DProjLodEXT(sampler2D sampler, vec4 coord, float lod)`
- `vec4 textureCubeLodEXT(samplerCube sampler, vec3 coord, float lod)`
- `vec4 texture2DGradEXT(sampler2D sampler, vec2 P, vec2 dPdx, vec2 dPdy)`
- `vec4 texture2DProjGradEXT(sampler2D sampler, vec3 P, vec2 dPdx, vec2 dPdy)`
- `vec4 texture2DProjGradEXT(sampler2D sampler, vec4 P, vec2 dPdx, vec2 dPdy)`
- `vec4 textureCubeGradEXT(samplerCube sampler, vec3 P, vec3 dPdx, vec3 dPdy)`

Function Derivatives

- *type* dFdx(*type* x), dFdy(*type* x)
- *type* fwidth(*type* p)

How-to

- **Use structs:** `struct myDataType { float occlusion; vec3 color; }; myDataType myData = myDataType(0.7, vec3(1.0, 2.0, 3.0));`
- **Initialize arrays:** arrays cannot be initialized in WebGL.
- **Do conversions:** `int a = 3; float b = float(a);`
- **Do component swizzling:** `vec4 a = vec4(1.0,2.0,3.0,4.0); vec4 b = a.zyyw;`
- **Access matrix components:** `mat4 m; m[1] = vec4(2.0); m[0][0] = 1.0; m[2][3] = 2.0;`

Be careful!

- **the f suffix for floating point numbers:** 1.0f is illegal in GLSL. You must use 1.0
- **saturate():** saturate(x) doesn't exist in GLSL. Use clamp(x,0.0,1.0) instead
- **pow/sqrt:** please don't feed sqrt() and pow() with negative numbers. Add an abs() or max(0.0,) to the argument
- **mod:** please don't do mod(x,0.0). This is undefined in some platforms
- **variables:** initialize your variables! Don't assume they'll be set to zero by default
- **functions:** don't call your functions the same as some of your variables

Shadertoy Inputs

Type	Name	Function	Description
vec3	iResolution	image	The viewport resolution (z is pixel aspect ratio, usually 1.0)
float	iTime	image/sound	Current time in seconds
float	iTimeDelta	image	Time it takes to render a frame, in seconds
int	iFrame	image	Current frame
float	iFrameRate	image	Number of frames rendered per second
float	iChannelTime[4]	image	Time for channel (if video or sound), in seconds
vec3	iChannelResolution[4]	image/sound	Input texture resolution for each channel
vec2	iChannelOffset[4]	image	Input texture offset in pixel coords for each channel
vec4	iMouse	image	xy = current pixel coords (if LMB is down). zw = click pixel
sampler2D	iChannel{i}	image/sound	Sampler for input textures i
vec4	iDate	image/sound	Year, month, day, time in seconds in .xyzw
float	iSampleRate	image/sound	The sound sample rate (typically 44100)
vec2	iRenderScale	image	The OpenFX render scale (e.g. 0.5,0.5 when rendering half-size) [OFX plugin only]

Shadertoy Outputs

For image shaders, `fragColor` is used as output channel. It is not, for now, mandatory but recommended to leave the alpha channel to 1.0.

For sound shaders, the `mainSound()` function returns a `vec2` containing the left and right (stereo) sound channel wave data.

OpenFX extensions to Shadertoy

Shadertoy was extended to:

- Expose shader parameters as uniforms, which are presented as OpenFX parameters.
- Provide the description and help for these parameters directly in the GLSL code.
- Add a default uniform containing the render scale. In OpenFX, a render scale of 1 means that the image is rendered at full resolution, 0.5 at half resolution, etc. This can be used to scale parameter values so that the final aspect does not depend on the render scale. For example, a blur size parameter given in pixels at full resolution would have to be multiplied by the render scale.
- Add a default uniform containing the offset of the processed texture with respect to the position of the origin.

The extensions are:

- The pre-defined `iRenderScale` uniform contains the current render scale. Basically all pixel sizes must be multiplied by the `renderscale` to get a scale-independent effect. For compatibility with Shadertoy, the first line that starts with `const vec2 iRenderScale` is ignored (the full line should be `const vec2 iRenderScale = vec2(1.,1.);`).
- The pre-defined `iChannelOffset` uniform contains the texture offset for each channel relative to channel 0. For compatibility with Shadertoy, the first line that starts with `const vec2 iChannelOffset` is ignored (the full line should be `const vec2 iChannelOffset[4] = vec2[4](vec2(0.,0.), vec2(0.,0.), vec2(0.,0.), vec2(0.,0.));`).
- The shader may define additional uniforms, which should have a default value, as in `uniform vec2 blurSize = vec2(5., 5.);`. These uniforms can be made available as OpenFX parameters using settings in the ‘Extra parameters’ group, which can be set automatically using the ‘Auto. Params’ button (automatic parameters are only updated if the node is connected to a Viewer). A parameter label and help string can be given in the comment on the same line. The help string must be in parenthesis. `uniform vec2 blurSize = vec2(5., 5.); // Blur Size (The blur size in pixels.) min/max values can also be given after a comma. The strings must be exactly min= and max=, without additional spaces, separated by a comma, and the values must have the same dimension as the uniform: uniform vec2 blurSize = vec2(5., 5.); // Blur Size (The blur size in pixels.), min=(0.,0.), max=(1000.,1000.)`
- The following comment line placed in the shader gives a label and help string to input 1 (the comment must be the only thing on the line): `// iChannel1: Noise (A noise texture to be used for random number calculations. The texture should not be frame-varying.)`
- This one also sets the filter and wrap parameters: `// iChannel0: Source (Source image.), filter=linear, wrap=clamp`
- And this one sets the output bounding box (possible values are Default, Union, Intersection, and `iChannel0` to `iChannel3`): `// BBox: iChannel0`

Converting a Shadertoy for use in OpenFX

To better understand how to modify a Shadertoy for OpenFX, let us take the simple [Gaussian blur](#) example, which is also available as a preset in the Shadertoy node.

In Natron, create a new project, create a Shadertoy node, connect the input 1 of the Viewer to the output of the Shadertoy node. This should give you a blurry color image that corresponds to the default Shadertoy source code. The Shadertoy node should have four inputs, named “iChannel0” to “iChannel3”.

In the Shadertoy node parameters, open the “Image Shader” group. You should see the GLSL source code. Now in the “Load from Preset” choice, select “Blur/Gaussian Blur”. The viewer should display a black image, but you should also notice that the Shadertoy node now has two visible inputs: “Source” and “Modulate” (in Nuke, these inputs are still called iChannel0 and iChannel1). Create a Read node that reads a still image or a video, and connect it to the “Source” input. A blurred version of the image should now appear in the viewer. You should also notice that two parameters appeared at the top of the parameters for the Shadertoy node: “Size” and “Modulate”. Play with the “Size” parameter and see how it affects the blur size (you may have to zoom on the image to see precisely the effect).

Now let us examine the modifications that were brought to the [original GLSL code](#):

These three comment lines describe the label, filter, and wrap parameters for each input, as well as the size of the output bounding box (also called “region of definition”):

```
// iChannel0: Source, filter=linear, wrap=clamp
// iChannel1: Modulate (Image containing a factor to be applied to the Blur size_
↳in the first channel), filter=linear, wrap=clamp
// BBox: iChannel0
```

Two constant global variables were added, which are ignored by the Shadertoy plugin, so that you can still copy-and-paste the source code in Shadertoy 0.8.8 and it still works (unfortunately, it does not work anymore with later versions of Shadertoy). You can safely ignore these:

```
const vec2 iRenderScale = vec2(1.,1.);
const vec2 iChannelOffset[4] = vec2[4]( vec2(0.,0.), vec2(0.,0.), vec2(0.,0.),_
↳vec2(0.,0.) );
```

Then the uniform section gives the list of what will appear as OpenFX parameters, together with their default value, label, help string, and default range. Note that in the original Shadertoy code, the blur size was a constant hidden inside the code. Finding out the parameters of a Shadertoy requires precise code inspection. If you modify this part of the code, pressing the “Auto. Params” button will apply these changes to the OpenFX parameters:

```
uniform float size = 10.; // Size (Size of the filter kernel in pixel units. The_
↳standard deviation of the corresponding Gaussian is size/2.4.), min=0., max=21.
uniform bool perpixel_size = false; // Modulate (Modulate the blur size by_
↳multiplying it by the first channel of the Modulate input)
```

In the mainImage function, which does the processing, we compute the mSize and kSize variables, which are the kernel size and mask size for that particular algorithm, from the “Size” parameter, multiplied by the render scale to get a scale-invariant effect. If the “Modulate” check box is on, we also multiply the size by the value found in the first channel (which is red, not alpha) of the “Modulate” input, which is in the iChannel1 texture according to the comments at the beginning of the source code. This can be used to modulate the blur size depending on the position in the image. The “Modulate” input may be for example connected to the output of a Roto node (with the “R” checkbox checked in the Roto node). Since the Roto output may not have the same size and origin as the Source image, we take care of these by using the iChannelOffset and iChannelResolution values for input 1.

```
float fSize = size * iRenderScale.x;
if (perpixel_size) {
    fSize *= texture2D(iChannel1, (fragCoord.xy-iChannelOffset[1].xy)/
↳iChannelResolution[1].xy).x;
}
int kSize = int(min(int((fSize-1)/2), KSIZE_MAX));
int mSize = kSize*2+1;
```

In the rest of the code, the only difference is that the blur size is not constant and equal to 7, but comes from the fSize variable:

```
float sigma = fSize / 2.4;
```

Issues with Gamma correction

OpenGL processing supposes all textures are linear, i.e. not gamma-compressed. This for example about bilinear interpolation on textures: this only works if the intensities are represented linearly. So a proper OpenGL rendering pipe should in principle:

1. Convert all textures to a linear representation (many 8-bit textures are gamma-compressed)
2. Render with OpenGL
3. Gamma-compress the linear framebuffer for display

When processing floating-point buffers in OpenFX, the color representation is usually linear, which means that the OpenFX host usually performs steps 1 and 3 anyway (that includes Natron and Nuke): the images given to an OpenFX plugins are in linear color space, and their output is also supposed to be linear.

However, many OpenGL applications, including Shadertoy and most games, skip steps 1 and 3 (mainly for performance issue): they process gamma-compressed textures as if they were linear, and sometimes have to boost their output by gamma compression so that it looks nice on a standard display (which usually accepts a sRGB-compressed framebuffer).

This is why many shaders from Shadertoy convert their outout from linear to sRGB or gamma=2.2, see for example the `srgb2lin` and `lin2srgb` functions in <https://www.shadertoy.com/view/XsfXzf> . These conversions *must* be removed when using the shader in OpenFX.

An alternative solution would be to convert all Shadertoy inputs from linear to sRGB, and convert back all outputs to linear, either inside the Shadertoy node, or using external conversion nodes (such as OCIOColorSpace). But this is a bad option, because this adds useless processing. Removing the `srgb2lin` and `lin2srgb` conversions from the shader source is a much better option (these functions may have different names, or there may simply be operations like `pow(c, vec3(2.2))` and/or `pow(c, vec3(1./2.2))` in the GLSL code).

As an example, take a look at the changes made to the [Barrel Blur Chroma](#) Shadertoy: the OpenFX version is available as a preset in the Shadertoy node as “Effects/Barrel Blur Chroma”. When it was converted to OpenFX, all gamma compression and decompression operations were identified and removed.

Multipass shaders

Most multipass shaders (those using BufA, BufB, BufC, or BufD) can be implemented using the Shadertoy plugin.

The shader sources for two sample multipass shadertoys are available as Natron PyPlugs (but the shader sources are also available separately next to the PyPlugs if you want to use these in another OpenFX host:

- a 3-pass circular bokeh blur (available as [Community/GLSL/BokehCircular_GL](#) in natron-plugins)
- a 4-pass octagonal bokeh blur (available as [Community/GLSL/BokehOctagon_GL](#) in natron-plugins)

The principle is very simple: since multipass cannot be done using a single Shadertoy, use several Shadertoy nodes, route the textures between them, and link the parameters. You can learn from these two examples. To figure out the route between textures, click on the tab for each shader in [shadertoy.com](#), and check which shader output is connected to the input textures (iChannel0, etc.) for this shader. The connections between nodes should follow these rules.

The only multipass effects that can not be implemented are the shaders that read back the content of a buffer to compute that same buffer, because compositing graphs cannot have loops (the execution of such a graph would cause an infinite recursion). One example is [this progressive lightmap render](#), where BufB from the previous render is read back as iChannel1 in the BufB shader.

Default textures and videos

The default shadertoy textures and videos are available from the [Shadertoy](#) web site. In order to mimic the behavior of each shader, download the corresponding textures or videos and connect them to the proper input.

- Textures: `tex00`, `tex01`, `tex02`, `tex03`, `tex04`, `tex05`, `tex06`, `tex07`, `tex08`, `tex09`, `tex10`, `tex11`, `tex12`, `tex14`, `tex15`, `tex16`, `tex17`, `tex18`, `tex19`, `tex20`, `tex21`.
- Videos: `vid00`, `vid01`, `vid02`, `vid03`.
- Cubemaps: `cube00_0`, `cube01_0`, `cube02_0`, `cube03_0`, `cube04_0`, `cube05`

Inputs

Input	Description	Optional
<code>iChannel0</code>		Yes
<code>iChannel1</code>		Yes
<code>iChannel2</code>		Yes
<code>iChannel3</code>		Yes

Controls

Parameter / script name	Type	Default	Function
Mouse Pos. / <code>mousePosition</code>	Double	x: 0 y: 0	Mouse position, in pixels. Gets mapped to the xy components of the <code>iMouse</code> input. Note that in the web version of Shadertoy, the y coordinate goes from 1 to height.
Click Pos. / <code>mouseClick</code>	Double	x: 1 y: 1	Mouse click position, in pixels. The zw components of the <code>iMouse</code> input contain <code>mouseClick</code> if <code>mousePressed</code> is checked, else <code>-mouseClick</code> . The default is (1.,1.)
Mouse Pressed / <code>mousePressed</code>	Boolean	Off	When checked, the zw components of the <code>iMouse</code> input contain <code>mouseClick</code> , else they contain <code>-mouseClick</code> . If the host does not support animating this parameter, use negative values for <code>mouseClick</code> to emulate a released mouse button.
Value0 / <code>paramValueBool0</code>	Boolean	Off	Value of the parameter.
Value0 / <code>paramValueInt0</code>	Integer	0	Value of the parameter.
Value0 / <code>paramValueFloat0</code>	Double	0	Value of the parameter.
Value0 / <code>paramValueVec20</code>	Double	x: 0 y: 0	Value of the parameter.
Value0 / <code>paramValueVec30</code>	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value0 / <code>paramValueVec40</code>	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value1 / <code>paramValueBool1</code>	Boolean	Off	Value of the parameter.
Value1 / <code>paramValueInt1</code>	Integer	0	Value of the parameter.
Value1 / <code>paramValueFloat1</code>	Double	0	Value of the parameter.
Value1 / <code>paramValueVec21</code>	Double	x: 0 y: 0	Value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Value1 / paramValueVec31	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value1 / paramValueVec41	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value2 / paramValueBool2	Boolean	Off	Value of the parameter.
Value2 / paramValueInt2	Integer	0	Value of the parameter.
Value2 / paramValueFloat2	Double	0	Value of the parameter.
Value2 / paramValueVec22	Double	x: 0 y: 0	Value of the parameter.
Value2 / paramValueVec32	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value2 / paramValueVec42	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value3 / paramValueBool3	Boolean	Off	Value of the parameter.
Value3 / paramValueInt3	Integer	0	Value of the parameter.
Value3 / paramValueFloat3	Double	0	Value of the parameter.
Value3 / paramValueVec23	Double	x: 0 y: 0	Value of the parameter.
Value3 / paramValueVec33	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value3 / paramValueVec43	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value4 / paramValueBool4	Boolean	Off	Value of the parameter.
Value4 / paramValueInt4	Integer	0	Value of the parameter.
Value4 / paramValueFloat4	Double	0	Value of the parameter.
Value4 / paramValueVec24	Double	x: 0 y: 0	Value of the parameter.
Value4 / paramValueVec34	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value4 / paramValueVec44	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value5 / paramValueBool5	Boolean	Off	Value of the parameter.
Value5 / paramValueInt5	Integer	0	Value of the parameter.
Value5 / paramValueFloat5	Double	0	Value of the parameter.
Value5 / paramValueVec25	Double	x: 0 y: 0	Value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Value5 / paramValueVec35	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value5 / paramValueVec45	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value6 / paramValueBool6	Boolean	Off	Value of the parameter.
Value6 / paramValueInt6	Integer	0	Value of the parameter.
Value6 / paramValueFloat6	Double	0	Value of the parameter.
Value6 / paramValueVec26	Double	x: 0 y: 0	Value of the parameter.
Value6 / paramValueVec36	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value6 / paramValueVec46	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value7 / paramValueBool7	Boolean	Off	Value of the parameter.
Value7 / paramValueInt7	Integer	0	Value of the parameter.
Value7 / paramValueFloat7	Double	0	Value of the parameter.
Value7 / paramValueVec27	Double	x: 0 y: 0	Value of the parameter.
Value7 / paramValueVec37	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value7 / paramValueVec47	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value8 / paramValueBool8	Boolean	Off	Value of the parameter.
Value8 / paramValueInt8	Integer	0	Value of the parameter.
Value8 / paramValueFloat8	Double	0	Value of the parameter.
Value8 / paramValueVec28	Double	x: 0 y: 0	Value of the parameter.
Value8 / paramValueVec38	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value8 / paramValueVec48	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value9 / paramValueBool9	Boolean	Off	Value of the parameter.
Value9 / paramValueInt9	Integer	0	Value of the parameter.
Value9 / paramValueFloat9	Double	0	Value of the parameter.
Value9 / paramValueVec29	Double	x: 0 y: 0	Value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Value9 / paramValueVec39	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value9 / paramValueVec49	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value10 / paramValueBool10	Boolean	Off	Value of the parameter.
Value10 / paramValueInt10	Integer	0	Value of the parameter.
Value10 / paramValueFloat10	Double	0	Value of the parameter.
Value10 / paramValueVec210	Double	x: 0 y: 0	Value of the parameter.
Value10 / paramValueVec310	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value10 / paramValueVec410	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value11 / paramValueBool11	Boolean	Off	Value of the parameter.
Value11 / paramValueInt11	Integer	0	Value of the parameter.
Value11 / paramValueFloat11	Double	0	Value of the parameter.
Value11 / paramValueVec211	Double	x: 0 y: 0	Value of the parameter.
Value11 / paramValueVec311	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value11 / paramValueVec411	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value12 / paramValueBool12	Boolean	Off	Value of the parameter.
Value12 / paramValueInt12	Integer	0	Value of the parameter.
Value12 / paramValueFloat12	Double	0	Value of the parameter.
Value12 / paramValueVec212	Double	x: 0 y: 0	Value of the parameter.
Value12 / paramValueVec312	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value12 / paramValueVec412	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value13 / paramValueBool13	Boolean	Off	Value of the parameter.
Value13 / paramValueInt13	Integer	0	Value of the parameter.
Value13 / paramValueFloat13	Double	0	Value of the parameter.
Value13 / paramValueVec213	Double	x: 0 y: 0	Value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Value13 / paramValueVec313	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value13 / paramValueVec413	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value14 / paramValueBool14	Boolean	Off	Value of the parameter.
Value14 / paramValueInt14	Integer	0	Value of the parameter.
Value14 / paramValueFloat14	Double	0	Value of the parameter.
Value14 / paramValueVec214	Double	x: 0 y: 0	Value of the parameter.
Value14 / paramValueVec314	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value14 / paramValueVec414	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value15 / paramValueBool15	Boolean	Off	Value of the parameter.
Value15 / paramValueInt15	Integer	0	Value of the parameter.
Value15 / paramValueFloat15	Double	0	Value of the parameter.
Value15 / paramValueVec215	Double	x: 0 y: 0	Value of the parameter.
Value15 / paramValueVec315	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value15 / paramValueVec415	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value16 / paramValueBool16	Boolean	Off	Value of the parameter.
Value16 / paramValueInt16	Integer	0	Value of the parameter.
Value16 / paramValueFloat16	Double	0	Value of the parameter.
Value16 / paramValueVec216	Double	x: 0 y: 0	Value of the parameter.
Value16 / paramValueVec316	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value16 / paramValueVec416	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value17 / paramValueBool17	Boolean	Off	Value of the parameter.
Value17 / paramValueInt17	Integer	0	Value of the parameter.
Value17 / paramValueFloat17	Double	0	Value of the parameter.
Value17 / paramValueVec217	Double	x: 0 y: 0	Value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Value17 / paramValueVec317	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value17 / paramValueVec417	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value18 / paramValueBool18	Boolean	Off	Value of the parameter.
Value18 / paramValueInt18	Integer	0	Value of the parameter.
Value18 / paramValueFloat18	Double	0	Value of the parameter.
Value18 / paramValueVec218	Double	x: 0 y: 0	Value of the parameter.
Value18 / paramValueVec318	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value18 / paramValueVec418	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Value19 / paramValueBool19	Boolean	Off	Value of the parameter.
Value19 / paramValueInt19	Integer	0	Value of the parameter.
Value19 / paramValueFloat19	Double	0	Value of the parameter.
Value19 / paramValueVec219	Double	x: 0 y: 0	Value of the parameter.
Value19 / paramValueVec319	Color	r: 0 g: 0 b: 0	Value of the parameter.
Value19 / paramValueVec419	Color	r: 0 g: 0 b: 0 a: 0	Value of the parameter.
Load from File / imageShaderFileName	N/A		Load the source from the given file. The file contents is only loaded once. Press the “Reload” button to load again the same file.
Reload / imageShaderReload	Button		Reload the source from the given file.
Presets Directory / imageShaderPresetDir	N/A		The directory where presets are located. There must be a “Shadertoy.txt” file in this directory to give the list of presets (see the default presets directory for an example). The default textures are located in “/Applications/Natron.app/Contents/Plugins/OFX/Natron/Shadertoy.ofx.bundle/Contents/Re

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Load from Preset / imageShaderPreset	Choice	No preset	<p>Load the source from the preset. The default textures are located in “/Applications/Natron.app/Contents/Plugins/OFX/Natron/Shadertoy.ofx.bundle/Contents/Resources/” and more presets can be added by editing “Shadertoy.txt” in the Presets Directory.</p> <p>No preset</p> <p>Blur/Bilateral</p> <p>Blur/Bloom</p> <p>Blur/Bokeh Disc</p> <p>Blur/Circular Blur</p> <p>Blur/Fast Blur</p> <p>Blur/Gaussian Blur</p> <p>Blur/HDR Bloom</p> <p>Blur/Mipmap Blur</p> <p>Blur/Monte-Carlo Blur</p> <p>Blur/Poisson Disc</p> <p>Blur/Simple Radial Blur</p> <p>Effect/Anaglyphic</p> <p>Effect/Ball</p> <p>Effect/Barrel Blur Chroma</p> <p>Effect/Bloom Paint</p> <p>Effect/C64</p> <p>Effect/Chromatic Aberration</p> <p>Effect/CMYK Halftone</p> <p>Effect/CRT</p> <p>Effect/DawnBringer 4bit</p> <p>Effect/Film Grain</p> <p>Effect/Fisheye</p> <p>Effect/Glitch 01</p> <p>Effect/Glitch 02</p> <p>Effect/Glitch A</p> <p>Effect/Glitch B</p> <p>Effect/Image Cel Shade</p> <p>Effect/Kaleidoscope</p> <p>Effect/Median Filter</p> <p>Effect/Money Filter</p> <p>Effect/Noisy Distortion</p> <p>Effect/Old Video</p> <p>Effect/Quad Mirror</p> <p>Effect/Postprocessing</p> <p>Effect/Q*Bert-ify</p> <p>Effect/Sharpen</p> <p>Effect/Stripes</p> <p>Effect/TV Snow</p> <p>Effect/Van Gogh</p> <p>Effect/Vignette</p> <p>Noise Blur</p> <p>Notebook Drawings</p> <p>Plasma2</p>
2.7. Filter nodes			<p>Source/Bleepy Blocks</p> <p>Source/Bubbles</p> <p>Source/Cellular</p> <p>Source/Cloud</p>

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Source / imageShaderSource	String	<pre>// iChannel0: Source (Source image.), filter=linear, wrap=clamp, // BBox: iChannel0 const vec2 iRenderScale = vec2(1.,1.); // Render Scale (The size of a full- resolution pixel). uniform float amplitude = 0.5; // Amplitude (The amplitude of the xy sine wave), min=0., max=1. uniform float size = 50.; // Size (The period of the xy sine wave), min = 0., max = 200.</pre>	<p>Image shader.</p> <p>Shader Inputs:</p> <p>uniform vec3 iResolution; // viewport resolution (in pixels)</p> <p>uniform float iTime; // shader playback time (in seconds)</p> <p>uniform float iTimeDelta; // render time (in seconds)</p> <p>uniform int iFrame; // shader playback frame</p> <p>uniform float iChannelTime[4]; // channel playback time (in seconds)</p> <p>uniform vec3 iChannelResolution[4]; // channel resolution (in pixels)</p> <p>uniform vec2 iChannelOffset[4]; // channel texture offset relative to iChannel0 (in pixels)</p> <p>uniform vec4 iMouse; // mouse pixel coords. xy: current (if MLB down), zw: click</p> <p>uniform samplerXX iChannel0..3; // input channel. XX = 2D/Cube</p> <p>uniform vec4 iDate; // (year, month, day, time in seconds)</p> <p>uniform float iSampleRate; // sound sample rate (i.e., 44100)</p>
312			Chapter 2. Reference Guide

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Compile / imageShaderCompile	Button		Compile the image shader.
Auto. Params / autoParams	Button		Automatically set the parameters from the shader source next time image is rendered. May require clicking twice, depending on the OpenFX host. Also reset these parameters to their default value.
Reset Params Values / resetParams	Button		Set all the extra parameters to their default values, as set automatically by the “Auto. Params”, or in the “Extra Parameters” group.
Enable / inputEnable0	Boolean	On	Enable this input.
Filter / mipmap0	Choice	Mipmap	Texture filter for this input. Nearest (nearest): MIN/MAG = GL_NEAREST/GL_NEAREST Linear (linear): MIN/MAG = GL_LINEAR/GL_LINEAR Mipmap (mipmap): MIN/MAG = GL_LINEAR_MIPMAP_LINEAR/GL_LINEAR Anisotropic (anisotropic): Mipmap with anisotropic filtering. Available with GPU if supported (check for the presence of the GL_EXT_texture_filter_anisotropic extension in the Renderer Info) and with “softpipe” CPU driver.
Wrap / wrap0	Choice	Repeat	Texture wrap parameter for this input. Repeat (repeat): WRAP_S/T = GL_REPEAT Clamp (clamp): WRAP_S/T = GL_CLAMP_TO_EDGE Mirror (mirror): WRAP_S/T = GL_MIRRORED_REPEAT
Label / inputLabel0	String		Label for this input in the user interface.
Hint / inputHint0	String		
Enable / inputEnable1	Boolean	On	Enable this input.
Filter / mipmap1	Choice	Mipmap	Texture filter for this input. Nearest (nearest): MIN/MAG = GL_NEAREST/GL_NEAREST Linear (linear): MIN/MAG = GL_LINEAR/GL_LINEAR Mipmap (mipmap): MIN/MAG = GL_LINEAR_MIPMAP_LINEAR/GL_LINEAR Anisotropic (anisotropic): Mipmap with anisotropic filtering. Available with GPU if supported (check for the presence of the GL_EXT_texture_filter_anisotropic extension in the Renderer Info) and with “softpipe” CPU driver.
Wrap / wrap1	Choice	Repeat	Texture wrap parameter for this input. Repeat (repeat): WRAP_S/T = GL_REPEAT Clamp (clamp): WRAP_S/T = GL_CLAMP_TO_EDGE Mirror (mirror): WRAP_S/T = GL_MIRRORED_REPEAT
Label / inputLabel1	String		Label for this input in the user interface.
Hint / inputHint1	String		

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Enable / <code>inputEnable2</code>	Boolean	On	Enable this input.
Filter / <code>mipmap2</code>	Choice	Mipmap	Texture filter for this input. Nearest (nearest): MIN/MAG = GL_NEAREST/GL_NEAREST Linear (linear): MIN/MAG = GL_LINEAR/GL_LINEAR Mipmap (mipmap): MIN/MAG = GL_LINEAR_MIPMAP_LINEAR/GL_LINEAR Anisotropic (anisotropic): Mipmap with anisotropic filtering. Available with GPU if supported (check for the presence of the GL_EXT_texture_filter_anisotropic extension in the Renderer Info) and with “softpipe” CPU driver.
Wrap / <code>wrap2</code>	Choice	Repeat	Texture wrap parameter for this input. Repeat (repeat): WRAP_S/T = GL_REPEAT Clamp (clamp): WRAP_S/T = GL_CLAMP_TO_EDGE Mirror (mirror): WRAP_S/T = GL_MIRRORED_REPEAT
Label / <code>inputLabel2</code>	String		Label for this input in the user interface.
Hint / <code>inputHint2</code>	String		
Enable / <code>inputEnable3</code>	Boolean	On	Enable this input.
Filter / <code>mipmap3</code>	Choice	Mipmap	Texture filter for this input. Nearest (nearest): MIN/MAG = GL_NEAREST/GL_NEAREST Linear (linear): MIN/MAG = GL_LINEAR/GL_LINEAR Mipmap (mipmap): MIN/MAG = GL_LINEAR_MIPMAP_LINEAR/GL_LINEAR Anisotropic (anisotropic): Mipmap with anisotropic filtering. Available with GPU if supported (check for the presence of the GL_EXT_texture_filter_anisotropic extension in the Renderer Info) and with “softpipe” CPU driver.
Wrap / <code>wrap3</code>	Choice	Repeat	Texture wrap parameter for this input. Repeat (repeat): WRAP_S/T = GL_REPEAT Clamp (clamp): WRAP_S/T = GL_CLAMP_TO_EDGE Mirror (mirror): WRAP_S/T = GL_MIRRORED_REPEAT
Label / <code>inputLabel3</code>	String		Label for this input in the user interface.
Hint / <code>inputHint3</code>	String		

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Output Bounding Box / bbox	Choice	Default	<p>What to use to produce the output image's bounding box. If no selected input is connected, use the project size.</p> <p>Default (default): Default bounding box (project size).</p> <p>Format (format): Use a pre-defined image format.</p> <p>Union (union): Union of all connected inputs.</p> <p>Intersect (intersection): Intersection of all connected inputs.</p> <p>iChannel0: Bounding box of iChannel0.</p> <p>iChannel1: Bounding box of iChannel1.</p> <p>iChannel2: Bounding box of iChannel2.</p> <p>iChannel3: Bounding box of iChannel3.</p>
Format / NatronParamFormat	Choice	HD 1920x1080	<p>The output format.</p> <p>PC_Video 640x480 (PC_Video)</p> <p>NTSC 720x486 0.91 (NTSC)</p> <p>PAL 720x576 1.09 (PAL)</p> <p>NTSC_16:9 720x486 1.21 (NTSC_16:9)</p> <p>PAL_16:9 720x576 1.46 (PAL_16:9)</p> <p>HD_720 1280x720 (HD_720)</p> <p>HD 1920x1080 (HD)</p> <p>UHD_4K 3840x2160 (UHD_4K)</p> <p>1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap))</p> <p>1K_Cinemascope 914x778 2.00 (1K_Cinemascope)</p> <p>2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap))</p> <p>2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope)</p> <p>2K_DCP 2048x1080 (2K_DCP)</p> <p>4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap))</p> <p>4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope)</p> <p>4K_DCP 4096x2160 (4K_DCP)</p> <p>square_256 256x256 (square_256)</p> <p>square_512 512x512 (square_512)</p> <p>square_1K 1024x1024 (square_1K)</p> <p>square_2K 2048x2048 (square_2K)</p>
Mouse Params. / mouseParams	Boolean	On	Enable mouse parameters.
Start Date / startDate	Color	y: 1970 m: 0 d: 1 s: 0	The date (yyyy,mm,dd,s) corresponding to frame 0. The month starts at 0 for january, the day starts at 1, and the seconds start from 0 at midnight and should be at most 24*60*60=86400. December 28, 1895 at 10:30 would thus be (1895,11,28,37800).
No. of Params / paramCount	Integer	0	Number of extra parameters.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType0	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName0	String		Name of the parameter, as used in the shader.
Label / paramLabel0	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint0	String		Help for the parameter.
Default0 / paramDefaultBool0	Boolean	Off	Default value of the parameter.
Default0 / paramDefaultInt0	Integer	0	Default value of the parameter.
Min0 / paramMinInt0	Integer	- 2147483648	Min value of the parameter.
Max0 / paramMaxInt0	Integer	2147483647	Max value of the parameter.
Default0 / paramDefaultFloat0	Double	0	Default value of the parameter.
Min0 / paramMinFloat0	Double	- 1.79769e+308	Min value of the parameter.
Max0 / paramMaxFloat0	Double	1.79769e+308	Max value of the parameter.
Default0 / paramDefaultVec20	Double	x: 0 y: 0	Default value of the parameter.
Min0 / paramMinVec20	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max0 / paramMaxVec20	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default0 / paramDefaultVec30	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default0 / paramDefaultVec40	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType1	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName1	String		Name of the parameter, as used in the shader.
Label / paramLabel1	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint1	String		Help for the parameter.
Default1 / paramDefaultBool1	Boolean	Off	Default value of the parameter.
Default1 / paramDefaultInt1	Integer	0	Default value of the parameter.
Min1 / paramMinInt1	Integer	-2147483648	Min value of the parameter.
Max1 / paramMaxInt1	Integer	2147483647	Max value of the parameter.
Default1 / paramDefaultFloat1	Double	0	Default value of the parameter.
Min1 / paramMinFloat1	Double	-1.79769e+308	Min value of the parameter.
Max1 / paramMaxFloat1	Double	1.79769e+308	Max value of the parameter.
Default1 / paramDefaultVec21	Double	x: 0 y: 0	Default value of the parameter.
Min1 / paramMinVec21	Double	x: -1.79769e+308 y: -1.79769e+308	Min value of the parameter.
Max1 / paramMaxVec21	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default1 / paramDefaultVec31	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default1 / paramDefaultVec41	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType2	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName2	String		Name of the parameter, as used in the shader.
Label / paramLabel2	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint2	String		Help for the parameter.
Default2 / paramDefaultBool2	Boolean	Off	Default value of the parameter.
Default2 / paramDefaultInt2	Integer	0	Default value of the parameter.
Min2 / paramMinInt2	Integer	-2147483648	Min value of the parameter.
Max2 / paramMaxInt2	Integer	2147483647	Max value of the parameter.
Default2 / paramDefaultFloat2	Double	0	Default value of the parameter.
Min2 / paramMinFloat2	Double	-1.79769e+308	Min value of the parameter.
Max2 / paramMaxFloat2	Double	1.79769e+308	Max value of the parameter.
Default2 / paramDefaultVec22	Double	x: 0 y: 0	Default value of the parameter.
Min2 / paramMinVec22	Double	x: -1.79769e+308 y: -1.79769e+308	Min value of the parameter.
Max2 / paramMaxVec22	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default2 / paramDefaultVec32	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default2 / paramDefaultVec42	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType3	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName3	String		Name of the parameter, as used in the shader.
Label / paramLabel3	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint3	String		Help for the parameter.
Default3 / paramDefaultBool3	Boolean	Off	Default value of the parameter.
Default3 / paramDefaultInt3	Integer	0	Default value of the parameter.
Min3 / paramMinInt3	Integer	- 2147483648	Min value of the parameter.
Max3 / paramMaxInt3	Integer	2147483647	Max value of the parameter.
Default3 / paramDefaultFloat3	Double	0	Default value of the parameter.
Min3 / paramMinFloat3	Double	- 1.79769e+308	Min value of the parameter.
Max3 / paramMaxFloat3	Double	1.79769e+308	Max value of the parameter.
Default3 / paramDefaultVec23	Double	x: 0 y: 0	Default value of the parameter.
Min3 / paramMinVec23	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max3 / paramMaxVec23	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default3 / paramDefaultVec33	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default3 / paramDefaultVec43	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType4	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName4	String		Name of the parameter, as used in the shader.
Label / paramLabel4	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint4	String		Help for the parameter.
Default4 / paramDefaultBool4	Boolean	Off	Default value of the parameter.
Default4 / paramDefaultInt4	Integer	0	Default value of the parameter.
Min4 / paramMinInt4	Integer	-2147483648	Min value of the parameter.
Max4 / paramMaxInt4	Integer	2147483647	Max value of the parameter.
Default4 / paramDefaultFloat4	Double	0	Default value of the parameter.
Min4 / paramMinFloat4	Double	-1.79769e+308	Min value of the parameter.
Max4 / paramMaxFloat4	Double	1.79769e+308	Max value of the parameter.
Default4 / paramDefaultVec24	Double	x: 0 y: 0	Default value of the parameter.
Min4 / paramMinVec24	Double	x: -1.79769e+308 y: -1.79769e+308	Min value of the parameter.
Max4 / paramMaxVec24	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default4 / paramDefaultVec34	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default4 / paramDefaultVec44	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType5	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName5	String		Name of the parameter, as used in the shader.
Label / paramLabel5	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint5	String		Help for the parameter.
Default5 / paramDefaultBool5	Boolean	Off	Default value of the parameter.
Default5 / paramDefaultInt5	Integer	0	Default value of the parameter.
Min5 / paramMinInt5	Integer	- 2147483648	Min value of the parameter.
Max5 / paramMaxInt5	Integer	2147483647	Max value of the parameter.
Default5 / paramDefaultFloat5	Double	0	Default value of the parameter.
Min5 / paramMinFloat5	Double	- 1.79769e+308	Min value of the parameter.
Max5 / paramMaxFloat5	Double	1.79769e+308	Max value of the parameter.
Default5 / paramDefaultVec25	Double	x: 0 y: 0	Default value of the parameter.
Min5 / paramMinVec25	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max5 / paramMaxVec25	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default5 / paramDefaultVec35	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default5 / paramDefaultVec45	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType6	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName6	String		Name of the parameter, as used in the shader.
Label / paramLabel6	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint6	String		Help for the parameter.
Default6 / paramDefaultBool6	Boolean	Off	Default value of the parameter.
Default6 / paramDefaultInt6	Integer	0	Default value of the parameter.
Min6 / paramMinInt6	Integer	-2147483648	Min value of the parameter.
Max6 / paramMaxInt6	Integer	2147483647	Max value of the parameter.
Default6 / paramDefaultFloat6	Double	0	Default value of the parameter.
Min6 / paramMinFloat6	Double	-1.79769e+308	Min value of the parameter.
Max6 / paramMaxFloat6	Double	1.79769e+308	Max value of the parameter.
Default6 / paramDefaultVec26	Double	x: 0 y: 0	Default value of the parameter.
Min6 / paramMinVec26	Double	x: -1.79769e+308 y: -1.79769e+308	Min value of the parameter.
Max6 / paramMaxVec26	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default6 / paramDefaultVec36	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default6 / paramDefaultVec46	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType7	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName7	String		Name of the parameter, as used in the shader.
Label / paramLabel7	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint7	String		Help for the parameter.
Default7 / paramDefaultBool7	Boolean	Off	Default value of the parameter.
Default7 / paramDefaultInt7	Integer	0	Default value of the parameter.
Min7 / paramMinInt7	Integer	- 2147483648	Min value of the parameter.
Max7 / paramMaxInt7	Integer	2147483647	Max value of the parameter.
Default7 / paramDefaultFloat7	Double	0	Default value of the parameter.
Min7 / paramMinFloat7	Double	- 1.79769e+308	Min value of the parameter.
Max7 / paramMaxFloat7	Double	1.79769e+308	Max value of the parameter.
Default7 / paramDefaultVec27	Double	x: 0 y: 0	Default value of the parameter.
Min7 / paramMinVec27	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max7 / paramMaxVec27	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default7 / paramDefaultVec37	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default7 / paramDefaultVec47	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType8	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName8	String		Name of the parameter, as used in the shader.
Label / paramLabel8	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint8	String		Help for the parameter.
Default8 / paramDefaultBool8	Boolean	Off	Default value of the parameter.
Default8 / paramDefaultInt8	Integer	0	Default value of the parameter.
Min8 / paramMinInt8	Integer	-2147483648	Min value of the parameter.
Max8 / paramMaxInt8	Integer	2147483647	Max value of the parameter.
Default8 / paramDefaultFloat8	Double	0	Default value of the parameter.
Min8 / paramMinFloat8	Double	-1.79769e+308	Min value of the parameter.
Max8 / paramMaxFloat8	Double	1.79769e+308	Max value of the parameter.
Default8 / paramDefaultVec28	Double	x: 0 y: 0	Default value of the parameter.
Min8 / paramMinVec28	Double	x: -1.79769e+308 y: -1.79769e+308	Min value of the parameter.
Max8 / paramMaxVec28	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default8 / paramDefaultVec38	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default8 / paramDefaultVec48	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType9	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName9	String		Name of the parameter, as used in the shader.
Label / paramLabel9	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint9	String		Help for the parameter.
Default9 / paramDefaultBool9	Boolean	Off	Default value of the parameter.
Default9 / paramDefaultInt9	Integer	0	Default value of the parameter.
Min9 / paramMinInt9	Integer	- 2147483648	Min value of the parameter.
Max9 / paramMaxInt9	Integer	2147483647	Max value of the parameter.
Default9 / paramDefaultFloat9	Double	0	Default value of the parameter.
Min9 / paramMinFloat9	Double	- 1.79769e+308	Min value of the parameter.
Max9 / paramMaxFloat9	Double	1.79769e+308	Max value of the parameter.
Default9 / paramDefaultVec29	Double	x: 0 y: 0	Default value of the parameter.
Min9 / paramMinVec29	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max9 / paramMaxVec29	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default9 / paramDefaultVec39	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default9 / paramDefaultVec49	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType10	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName10	String		Name of the parameter, as used in the shader.
Label / paramLabel10	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint10	String		Help for the parameter.
Default10 / paramDefaultBool10	Boolean	Off	Default value of the parameter.
Default10 / paramDefaultInt10	Integer	0	Default value of the parameter.
Min10 / paramMinInt10	Integer	- 2147483648	Min value of the parameter.
Max10 / paramMaxInt10	Integer	2147483647	Max value of the parameter.
Default10 / paramDefaultFloat10	Double	0	Default value of the parameter.
Min10 / paramMinFloat10	Double	- 1.79769e+308	Min value of the parameter.
Max10 / paramMaxFloat10	Double	1.79769e+308	Max value of the parameter.
Default10 / paramDefaultVec210	Double	x: 0 y: 0	Default value of the parameter.
Min10 / paramMinVec210	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max10 / paramMaxVec210	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default10 / paramDefaultVec310	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default10 / paramDefaultVec410	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType11	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName11	String		Name of the parameter, as used in the shader.
Label / paramLabel11	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint11	String		Help for the parameter.
Default11 / paramDefaultBool11	Boolean	Off	Default value of the parameter.
Default11 / paramDefaultInt11	Integer	0	Default value of the parameter.
Min11 / paramMinInt11	Integer	- 2147483648	Min value of the parameter.
Max11 / paramMaxInt11	Integer	2147483647	Max value of the parameter.
Default11 / paramDefaultFloat11	Double	0	Default value of the parameter.
Min11 / paramMinFloat11	Double	- 1.79769e+308	Min value of the parameter.
Max11 / paramMaxFloat11	Double	1.79769e+308	Max value of the parameter.
Default11 / paramDefaultVec211	Double	x: 0 y: 0	Default value of the parameter.
Min11 / paramMinVec211	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max11 / paramMaxVec211	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default11 / paramDefaultVec311	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default11 / paramDefaultVec411	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType12	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName12	String		Name of the parameter, as used in the shader.
Label / paramLabel12	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint12	String		Help for the parameter.
Default12 / paramDefaultBool12	Boolean	Off	Default value of the parameter.
Default12 / paramDefaultInt12	Integer	0	Default value of the parameter.
Min12 / paramMinInt12	Integer	- 2147483648	Min value of the parameter.
Max12 / paramMaxInt12	Integer	2147483647	Max value of the parameter.
Default12 / paramDefaultFloat12	Double	0	Default value of the parameter.
Min12 / paramMinFloat12	Double	- 1.79769e+308	Min value of the parameter.
Max12 / paramMaxFloat12	Double	1.79769e+308	Max value of the parameter.
Default12 / paramDefaultVec212	Double	x: 0 y: 0	Default value of the parameter.
Min12 / paramMinVec212	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max12 / paramMaxVec212	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default12 / paramDefaultVec312	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default12 / paramDefaultVec412	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType13	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName13	String		Name of the parameter, as used in the shader.
Label / paramLabel13	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint13	String		Help for the parameter.
Default13 / paramDefaultBool13	Boolean	Off	Default value of the parameter.
Default13 / paramDefaultInt13	Integer	0	Default value of the parameter.
Min13 / paramMinInt13	Integer	- 2147483648	Min value of the parameter.
Max13 / paramMaxInt13	Integer	2147483647	Max value of the parameter.
Default13 / paramDefaultFloat13	Double	0	Default value of the parameter.
Min13 / paramMinFloat13	Double	- 1.79769e+308	Min value of the parameter.
Max13 / paramMaxFloat13	Double	1.79769e+308	Max value of the parameter.
Default13 / paramDefaultVec213	Double	x: 0 y: 0	Default value of the parameter.
Min13 / paramMinVec213	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max13 / paramMaxVec213	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default13 / paramDefaultVec313	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default13 / paramDefaultVec413	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType14	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName14	String		Name of the parameter, as used in the shader.
Label / paramLabel14	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint14	String		Help for the parameter.
Default14 / paramDefaultBool14	Boolean	Off	Default value of the parameter.
Default14 / paramDefaultInt14	Integer	0	Default value of the parameter.
Min14 / paramMinInt14	Integer	-2147483648	Min value of the parameter.
Max14 / paramMaxInt14	Integer	2147483647	Max value of the parameter.
Default14 / paramDefaultFloat14	Double	0	Default value of the parameter.
Min14 / paramMinFloat14	Double	-1.79769e+308	Min value of the parameter.
Max14 / paramMaxFloat14	Double	1.79769e+308	Max value of the parameter.
Default14 / paramDefaultVec214	Double	x: 0 y: 0	Default value of the parameter.
Min14 / paramMinVec214	Double	x: -1.79769e+308 y: -1.79769e+308	Min value of the parameter.
Max14 / paramMaxVec214	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default14 / paramDefaultVec314	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default14 / paramDefaultVec414	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType15	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName15	String		Name of the parameter, as used in the shader.
Label / paramLabel15	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint15	String		Help for the parameter.
Default15 / paramDefaultBool15	Boolean	Off	Default value of the parameter.
Default15 / paramDefaultInt15	Integer	0	Default value of the parameter.
Min15 / paramMinInt15	Integer	- 2147483648	Min value of the parameter.
Max15 / paramMaxInt15	Integer	2147483647	Max value of the parameter.
Default15 / paramDefaultFloat15	Double	0	Default value of the parameter.
Min15 / paramMinFloat15	Double	- 1.79769e+308	Min value of the parameter.
Max15 / paramMaxFloat15	Double	1.79769e+308	Max value of the parameter.
Default15 / paramDefaultVec215	Double	x: 0 y: 0	Default value of the parameter.
Min15 / paramMinVec215	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max15 / paramMaxVec215	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default15 / paramDefaultVec315	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default15 / paramDefaultVec415	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType16	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName16	String		Name of the parameter, as used in the shader.
Label / paramLabel16	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint16	String		Help for the parameter.
Default16 / paramDefaultBool16	Boolean	Off	Default value of the parameter.
Default16 / paramDefaultInt16	Integer	0	Default value of the parameter.
Min16 / paramMinInt16	Integer	- 2147483648	Min value of the parameter.
Max16 / paramMaxInt16	Integer	2147483647	Max value of the parameter.
Default16 / paramDefaultFloat16	Double	0	Default value of the parameter.
Min16 / paramMinFloat16	Double	- 1.79769e+308	Min value of the parameter.
Max16 / paramMaxFloat16	Double	1.79769e+308	Max value of the parameter.
Default16 / paramDefaultVec216	Double	x: 0 y: 0	Default value of the parameter.
Min16 / paramMinVec216	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max16 / paramMaxVec216	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default16 / paramDefaultVec316	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default16 / paramDefaultVec416	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType17	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName17	String		Name of the parameter, as used in the shader.
Label / paramLabel17	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint17	String		Help for the parameter.
Default17 / paramDefaultBool17	Boolean	Off	Default value of the parameter.
Default17 / paramDefaultInt17	Integer	0	Default value of the parameter.
Min17 / paramMinInt17	Integer	- 2147483648	Min value of the parameter.
Max17 / paramMaxInt17	Integer	2147483647	Max value of the parameter.
Default17 / paramDefaultFloat17	Double	0	Default value of the parameter.
Min17 / paramMinFloat17	Double	- 1.79769e+308	Min value of the parameter.
Max17 / paramMaxFloat17	Double	1.79769e+308	Max value of the parameter.
Default17 / paramDefaultVec217	Double	x: 0 y: 0	Default value of the parameter.
Min17 / paramMinVec217	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max17 / paramMaxVec217	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default17 / paramDefaultVec317	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default17 / paramDefaultVec417	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType18	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName18	String		Name of the parameter, as used in the shader.
Label / paramLabel18	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint18	String		Help for the parameter.
Default18 / paramDefaultBool18	Boolean	Off	Default value of the parameter.
Default18 / paramDefaultInt18	Integer	0	Default value of the parameter.
Min18 / paramMinInt18	Integer	- 2147483648	Min value of the parameter.
Max18 / paramMaxInt18	Integer	2147483647	Max value of the parameter.
Default18 / paramDefaultFloat18	Double	0	Default value of the parameter.
Min18 / paramMinFloat18	Double	- 1.79769e+308	Min value of the parameter.
Max18 / paramMaxFloat18	Double	1.79769e+308	Max value of the parameter.
Default18 / paramDefaultVec218	Double	x: 0 y: 0	Default value of the parameter.
Min18 / paramMinVec218	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max18 / paramMaxVec218	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default18 / paramDefaultVec318	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default18 / paramDefaultVec418	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Type / paramType19	Choice	none	Type of the parameter. none : No parameter. bool : Boolean parameter (checkbox). int : Integer parameter. float : Floating-point parameter. vec2 : 2D floating-point parameter (e.g. position). vec3 : 3D floating-point parameter (e.g. 3D position or RGB color). vec4 : 4D floating-point parameter (e.g. RGBA color).
Name / paramName19	String		Name of the parameter, as used in the shader.
Label / paramLabel19	String		Label of the parameter, as displayed in the user interface.
Hint / paramHint19	String		Help for the parameter.
Default19 / paramDefaultBool19	Boolean	Off	Default value of the parameter.
Default19 / paramDefaultInt19	Integer	0	Default value of the parameter.
Min19 / paramMinInt19	Integer	- 2147483648	Min value of the parameter.
Max19 / paramMaxInt19	Integer	2147483647	Max value of the parameter.
Default19 / paramDefaultFloat19	Double	0	Default value of the parameter.
Min19 / paramMinFloat19	Double	- 1.79769e+308	Min value of the parameter.
Max19 / paramMaxFloat19	Double	1.79769e+308	Max value of the parameter.
Default19 / paramDefaultVec219	Double	x: 0 y: 0	Default value of the parameter.
Min19 / paramMinVec219	Double	x: - 1.79769e+308 y: - 1.79769e+308	Min value of the parameter.
Max19 / paramMaxVec219	Double	x: 1.79769e+308 y: 1.79769e+308	Max value of the parameter.
Default19 / paramDefaultVec319	Color	r: 0 g: 0 b: 0	Default value of the parameter.
Default19 / paramDefaultVec419	Color	r: 0 g: 0 b: 0 a: 0	Default value of the parameter.

Continued on next page

Table 123 – continued from previous page

Parameter / script name	Type	Default	Function
Enable GPU Render / <code>enableGPU</code>	Boolean	On	Enable GPU-based OpenGL render. If the checkbox is checked but is not enabled (i.e. it cannot be unchecked), GPU render can not be enabled or disabled from the plugin and is probably part of the host options. If the checkbox is not checked and is not enabled (i.e. it cannot be checked), GPU render is not available on this host.
CPU Driver / <code>cpuDriver</code>	Choice	llvmpipe	Driver for CPU rendering. May be “softpipe”, “llvmpipe” or “swr” (OpenSWR, not always available). softpipe : Gallium softpipe driver from Mesa. A reference single-threaded driver (slower, has GL_EXT_texture_filter_anisotropic GL_ARB_texture_query_lod GL_ARB_pipeline_statistics_query). llvmpipe : Gallium llvmpipe driver from Mesa, if available. Uses LLVM for x86 JIT code generation and is multi-threaded (faster, has GL_ARB_buffer_storage GL_EXT_polygon_offset_clamp). swr : OpenSWR driver from Mesa, if available. Fully utilizes modern instruction sets like AVX and AVX2 to achieve high rendering performance.
Renderer Info... / <code>rendererInfo</code>	Button		Retrieve information about the current OpenGL renderer.
Help... / <code>helpButton</code>	Button		Display help about using Shadertoy.

2.7.26 Sharpen node



This documentation is for version 4.0 of Sharpen.

Description

Sharpen the input stream by enhancing its Laplacian.

The effects adds the Laplacian (as computed by the Laplacian plugin) times the ‘Amount’ parameter to the input stream.

Uses the ‘vanvliet’ and ‘deriche’ functions from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Amount / amount	Double	1	Amount of sharpening to apply.
Size / size	Double	x: 3 y: 3	Size (diameter) of the filter kernel, in pixel units (≥ 0). The standard deviation of the corresponding Gaussian is $\text{size}/2.4$. No filter is applied if $\text{size} < 1.2$.
Uniform / uniform	Boolean	Off	Apply the same amount of blur on X and Y.
Filter / filter	Choice	Gaussian	<p>Blurring filter. The quasi-Gaussian filter should be appropriate in most cases. The Gaussian filter is more isotropic (its impulse response has rotational symmetry), but slower.</p> <p>Quasi-Gaussian (quasigaussian): Quasi-Gaussian filter (0-order recursive Deriche filter, faster) - IIR (infinite support / impulsional response).</p> <p>Gaussian (gaussian): Gaussian filter (Van Vliet recursive Gaussian filter, more isotropic, slower) - IIR (infinite support / impulsional response).</p> <p>Box (box): Box filter - FIR (finite support / impulsional response).</p> <p>Triangle (triangle): Triangle/tent filter - FIR (finite support / impulsional response).</p> <p>Quadratic (quadratic): Quadratic filter - FIR (finite support / impulsional response).</p>
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.27 SharpenInvDiff node



This documentation is for version 2.0 of SharpenInvDiff.

Description

Sharpen selected images by inverse diffusion.

Uses 'sharpen' function from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Amplitude / amplitude	Double	0.2	Standard deviation of the spatial kernel, in pixel units (≥ 0). Details smaller than this size are filtered out.
Iterations / iterations	Integer	2	Number of iterations. A reasonable value is 2.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.28 SharpenShock node



This documentation is for version 2.0 of SharpenShock.

Description

Sharpen selected images by shock filters.

Uses 'sharpen' function from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Amplitude / amplitude	Double	0.6	Standard deviation of the spatial kernel, in pixel units (≥ 0). Details smaller than this size are filtered out.
Edge Threshold / edgeThreshold	Double	0.1	Edge threshold.
Gradient Smoothness / alpha	Double	0.8	Gradient smoothness (in pixels).
Tensor Smoothness / sigma	Double	1.1	Tensor smoothness (in pixels).
Iterations / iterations	Integer	1	Number of iterations. A reasonable value is 1.

Continued on next page

Table 126 – continued from previous page

Parameter / script name	Type	Default	Function
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.29 SmoothAnisotropic node



This documentation is for version 2.0 of SmoothAnisotropic.

Description

Smooth/Denoise input stream using anisotropic PDE-based smoothing.

Uses the ‘blur_anisotropic’ function from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Amplitude / amplitude	Double	60	Amplitude of the smoothing, in pixel units (≥ 0). This is the maximum length of streamlines used to smooth the data.
Sharpness / sharpness	Double	0.7	
Anisotropy / anisotropy	Double	0.3	Smoothing anisotropy ($0 \leq a \leq 1$)
Gradient Smoothness / alpha	Double	0.6	
Tensor Smoothness / sigma	Double	1.1	Geometry regularity, in pixels units (≥ 0)
Spatial Precision / dl	Double	0.8	Spatial discretization, in pixel units ($0 \leq dl \leq 1$)
Angular Precision / da	Double	30	Angular integration step, in degrees ($0 \leq da \leq 90$). If $da=0$, Iterated oriented Laplacians is used instead of LIC-based smoothing.
Value Precision / prec	Double	2	Precision of the diffusion process (> 0).

Continued on next page

Table 127 – continued from previous page

Parameter / script name	Type	Default	Function
Interpolation / interpolation	Choice	Nearest-neighbor	Interpolation type Nearest-neighbor (nearest) : Nearest-neighbor. Linear (linear) : Linear interpolation. Runge-Kutta (rungekutta) : Runge-Kutta interpolation.
Fast Approximation / is_fast_approximation	Boolean	On	Tells if a fast approximation of the gaussian function is used or not
Iterations / iterations	Integer	1	Number of iterations.
Set Thin Brush Defaults / thinBrush	Button		Set the defaults to the value of the Thin Brush filter by PhotoComiX, as featured in the G'MIC Gimp plugin.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.30 SmoothBilateral node



This documentation is for version 2.0 of SmoothBilateral.

Description

Blur input stream by bilateral filtering.

Uses the ‘blur_bilateral’ function from the CImg library.

See also: <http://opticalenquiry.com/nuke/index.php?title=Bilateral>

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Spatial Std Dev / <code>sigma_s</code>	Double	10	Standard deviation of the spatial kernel (positional sigma), in pixel units (≥ 0). A reasonable value is 1/16 of the image dimension. Small values (1 pixel and below) will slow down filtering.
Value Std Dev / <code>sigma_r</code>	Double	0.3	Standard deviation of the range kernel (color sigma), in intensity units (≥ 0). A reasonable value is 1/10 of the intensity range. In the context of denoising, Liu et al. ("Noise estimation from a single image", CVPR2006) recommend a value of $1.95 \times \text{sigma}_n$, where <code>sigma_n</code> is the local image noise. Small values (1/256 of the intensity range and below) will slow down filtering.
Iterations / <code>iterations</code>	Integer	2	Number of iterations.
(Un)premult / <code>premult</code>	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / <code>maskInvert</code>	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / <code>mix</code>	Double	1	Mix factor between the original and the transformed image.

2.7.31 SmoothBilateralGuided node

This documentation is for version 2.0 of SmoothBilateralGuided.

Description

Apply joint/cross bilateral filtering on image A, guided by the intensity differences of image B. Uses the 'blur_bilateral' function from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

In-put	Description	Op-tional
Guide	The guide image indicates where similar pixels are located in each neighborhood. The neighborhood of a pixel consists of pixels that are within a neighborhood of side <code>sigma_s</code> , which have an intensity/value in the Guide image that is within a range of size <code>sigma_r</code> around the intensity of the considered pixel.	No
Source		No

Controls

Parameter / script name	Type	Default	Function
Spatial Std Dev / <code>sigma_s</code>	Double	10	Standard deviation of the spatial kernel (positional sigma), in pixel units (≥ 0). A reasonable value is 1/16 of the image dimension. Small values (1 pixel and below) will slow down filtering.

Continued on next page

Table 129 – continued from previous page

Parameter / script name	Type	Default	Function
Value Std Dev / sigma_r	Double	0.3	Standard deviation of the range kernel (color sigma), in intensity units (≥ 0). A reasonable value is 1/10 of the intensity range. In the context of denoising, Liu et al. (“Noise estimation from a single image”, CVPR2006) recommend a value of $1.95 \cdot \sigma_n$, where σ_n is the local image noise. Small values (1/256 of the intensity range and below) will slow down filtering.
Iterations / iterations	Integer	2	Number of iterations.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.32 SmoothGuided node



This documentation is for version 2.0 of SmoothGuided.

Description

Blur image, with the Guided Image filter.

The algorithm is described in: He et al., “Guided Image Filtering,” <http://research.microsoft.com/en-us/um/people/kahe/publications/pami12guidedfilter.pdf>

Uses the ‘blur_guided’ function from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Radius / radius	Integer	5	Radius of the spatial kernel (positional sigma), in pixel units (≥ 0).
Smoothness / epsilon	Double	0.2	Regularization parameter. The actual guided filter parameter is $\epsilon \cdot \sigma^2$.
Iterations / iterations	Integer	1	Number of iterations.

Continued on next page

Table 130 – continued from previous page

Parameter / script name	Type	Default	Function
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.33 SmoothPatchBased node



This documentation is for version 2.0 of SmoothPatchBased.

Description

Denoise selected images by non-local patch averaging.

This uses the method described in: Non-Local Image Smoothing by Applying Anisotropic Diffusion PDE's in the Space of Patches (D. Tschumperlé, L. Brun), ICIP'09 (https://tschumperle.users.greyc.fr/publications/tschumperle_icip09.pdf).

Uses the 'blur_patch' function from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Spatial Std Dev / sigma_s	Double	10	Standard deviation of the spatial kernel, in pixel units (≥ 0).
Value Std Dev / sigma_r	Double	0.05	Standard deviation of the range kernel, in intensity units (≥ 0). In the context of denoising, Liu et al. ("Noise estimation from a single image", CVPR2006) recommend a value of $1.95 \cdot \sigma_n$, where σ_n is the local image noise.
Patch Size / psize	Integer	5	Size of the patches, in pixels (≥ 0).
Lookup Size / lsize	Integer	6	Size of the window to search similar patches, in pixels (≥ 0).
Smoothness / smoothness	Double	1	Smoothness for the patch comparison, in pixels (≥ 0).
fast Approximation / is_fast_approximation	Boolean	On	Tells if a fast approximation of the gaussian function is used or not

Continued on next page

Table 131 – continued from previous page

Parameter / script name	Type	Default	Function
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.34 SmoothRollingGuidance node



This documentation is for version 2.0 of SmoothRollingGuidance.

Description

Filter out details under a given scale using the Rolling Guidance filter.

Rolling Guidance is described fully in <http://www.cse.cuhk.edu.hk/~leojia/projects/rollguidance/>

Iterates the ‘blur_bilateral’ function from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Spatial Std Dev / sigma_s	Double	10	Standard deviation of the spatial kernel, in pixel units (≥ 0). Details smaller than this size are filtered out.
Value Std Dev / sigma_r	Double	0.1	Standard deviation of the range kernel, in intensity units (≥ 0). A reasonable value is 1/10 of the intensity range. In the context of denoising, Liu et al. (“Noise estimation from a single image”, CVPR2006) recommend a value of $1.95 \cdot \sigma_n$, where σ_n is the local image noise.
Iterations / iterations	Integer	4	Number of iterations of the rolling guidance filter. 1 corresponds to Gaussian smoothing. A reasonable value is 4.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.35 Soften node

This documentation is for version 4.0 of Soften.

Description

Soften the input stream by reducing its Laplacian.

The effects subtracts the Laplacian (as computed by the Laplacian plugin) times the ‘Amount’ parameter from the input stream.

Uses the ‘vanvliet’ and ‘deriche’ functions from the CImg library.

CImg is a free, open-source library distributed under the CeCILL-C (close to the GNU LGPL) or CeCILL (compatible with the GNU GPL) licenses. It can be used in commercial applications (see <http://cimg.eu>).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Amount / amount	Double	0.5	Amount of softening to apply.
Size / size	Double	x: 3 y: 3	Size (diameter) of the filter kernel, in pixel units (≥ 0). The standard deviation of the corresponding Gaussian is $\text{size}/2.4$. No filter is applied if $\text{size} < 1.2$.
Uniform / uniform	Boolean	Off	Apply the same amount of blur on X and Y.
Filter / filter	Choice	Gaussian	<p>Blurring filter. The quasi-Gaussian filter should be appropriate in most cases. The Gaussian filter is more isotropic (its impulse response has rotational symmetry), but slower.</p> <p>Quasi-Gaussian (quasigaussian): Quasi-Gaussian filter (0-order recursive Deriche filter, faster) - IIR (infinite support / impulsional response).</p> <p>Gaussian (gaussian): Gaussian filter (Van Vliet recursive Gaussian filter, more isotropic, slower) - IIR (infinite support / impulsional response).</p> <p>Box (box): Box filter - FIR (finite support / impulsional response).</p> <p>Triangle (triangle): Triangle/tent filter - FIR (finite support / impulsional response).</p> <p>Quadratic (quadratic): Quadratic filter - FIR (finite support / impulsional response).</p>
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.7.36 ZMask node

This documentation is for version 1.0 of ZMask.

Description

Creates a mask from a depth buffer by specifying the center value and the amplitude of the range around it with the tightness parameter

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Convert to Group / <code>convertToGroup</code>	Button		Converts this node to a Group: the internal node-graph and the user parameters will become editable
Z picking / <code>zPicking</code>	Boolean	Off	
Center Value / <code>centerValue</code>	Color	r: 1 g: 1 b: 1	
Tightness / <code>tightness</code>	Color	r: 1 g: 1 b: 1 a: 1	
Contrast / <code>contrast</code>	Color	r: 1 g: 1 b: 1 a: 1	
Offset / <code>offset</code>	Color	r: 0 g: 0 b: 0 a: 0	
Gamma / <code>gamma</code>	Color	r: 1 g: 1 b: 1 a: 1	
Invert Gradient / <code>invertGradient</code>	Boolean	Off	
Source Layer / <code>Source_channels</code>	Choice		None

Continued on next page

Table 134 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / channels	Choice	Disparity	<p>Left.Disparity</p> <p>Color.RGBA (uk.co.thefoundry.OfxImagePlaneColour)</p> <p>DisparityLeft.Disparity (uk.co.thefoundry.OfxImagePlaneStereoDisparityLeft)</p> <p>DisparityRight.Disparity (uk.co.thefoundry.OfxImagePlaneStereoDisparityRight)</p> <p>Backward.Motion (uk.co.thefoundry.OfxImagePlaneBackMotionVector)</p> <p>Forward.Motion (uk.co.thefoundry.OfxImagePlaneForwardMotionVector)</p>

2.7.37 ZRemap node



This documentation is for version 1.0 of ZRemap.

Description

Remap Z-Depth pass according to a close limit value and a far limit

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Convert to Group / <code>convertToGroup</code>	Button		Converts this node to a Group: the internal node-graph and the user parameters will become editable
Z Picking mode / <code>zPicking</code>	Boolean	Off	
Close Limit / <code>closeLimit</code>	Color	r: 0 g: 0 b: 0 a: 0	Define the Z value remapped to white.
Far Limit / <code>farLimit</code>	Color	r: 100 g: 100 b: 100 a: 100	Define the Z value remapped to black.
Gamma / <code>gamma</code>	Color	r: 1 g: 1 b: 1 a: 1	

Continued on next page

Table 135 – continued from previous page

Parameter / script name	Type	Default	Function
Invert Gradient / <code>invertGradient</code>	Boolean	Off	
Source Layer / <code>Source_channels</code>	Choice		None
Output Layer / <code>channels</code>	Choice	DisparityLeft	DisparityLeft.Disparity Color.RGBA (uk.co.thefoundry.OfxImagePlaneColour) DisparityLeft.Disparity (uk.co.thefoundry.OfxImagePlaneStereoDisparityLeft) DisparityRight.Disparity (uk.co.thefoundry.OfxImagePlaneStereoDisparityRight) Backward.Motion (uk.co.thefoundry.OfxImagePlaneBackMotionVector) Forward.Motion (uk.co.thefoundry.OfxImagePlaneForwardMotionVector)

2.8 Keyer nodes

The following sections contain documentation about every node in the Keyer group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.8.1 ChromaKeyer node



This documentation is for version 1.1 of ChromaKeyer.

Description

Simple chroma Keyer.

Algorithm description:

Keith Jack, “Video Demystified”, Independent Pub Group (Computer), 1996, pp. 214-222, <http://www.ee-techs.com/circuit/video-demy5.pdf>

A simplified version is described in:

[2] High Quality Chroma Key, Michael Ashikhmin, <http://www.cs.utah.edu/~michael/chroma/>

Inputs

In-put	Description	Op-tional
Source	The foreground image to key.	No
InM	The Inside Mask, or holdout matte, or core matte, used to confirm areas that are definitely foreground.	Yes
OutM	The Outside Mask, or garbage matte, used to remove unwanted objects (lighting rigs, and so on) from the foreground. The Outside Mask has priority over the Inside Mask, so that areas where both are one are considered to be outside.	Yes
Bg	The background image to replace the blue/green screen in the foreground.	Yes

Controls

Parameter / script name	Type	Default	Function
Key Color / <code>keyColor</code>	Color	r: 0 g: 0 b: 0	Foreground key color; foreground areas containing the key color are replaced with the background image.
YCbCr Colorspace / <code>colorspace</code>	Choice	Rec. 709	Formula used to compute YCbCr from RGB values. CCIR 601 (ccir601) : Use CCIR 601 (SD footage). Rec. 709 (rec709) : Use Rec. 709 (HD footage). Rec. 2020 (rec2020) : Use Rec. 2020 (UltraHD/4K footage).
Linear Processing / <code>linearProcessing</code>	Boolean	Off	Do not delinearize RGB values to compute the key value.
Acceptance Angle / <code>acceptanceAngle</code>	Double	120	Foreground colors are only suppressed inside the acceptance angle (alpha).
Suppression Angle / <code>suppressionAngle</code>	Double	40	The chrominance of foreground colors inside the suppression angle (beta) is set to zero on output, to deal with noise. Use no more than one third of acceptance angle. This has no effect on the alpha channel, or if the output is in Intermediate mode.
Key Lift / <code>keyLift</code>	Double	0	Raise it so that less pixels are classified as background. Makes a sharper transition between foreground and background. Defaults to 0.
Key Gain / <code>keyGain</code>	Double	1	Lower it to classify more colors as background. Defaults to 1.
Output Mode / <code>show</code>	Choice	Composite	What image to output. Intermediate (intermediate) : Color is the source color. Alpha is the foreground key. Use for multi-pass keying. Premultiplied (premultiplied) : Color is the Source color after key color suppression, multiplied by alpha. Alpha is the foreground key. Unpremultiplied (unpremultiplied) : Color is the Source color after key color suppression. Alpha is the foreground key. Composite (composite) : Color is the composite of Source and Bg. Alpha is the foreground key.

Continued on next page

Table 136 – continued from previous page

Parameter / script name	Type	Default	Function
Source Alpha / sourceAlphaHandling	Choice	Ignore	<p>How the alpha embedded in the Source input should be used</p> <p>Ignore (ignore): Ignore the source alpha.</p> <p>Add to Inside Mask (insidemask): Source alpha is added to the inside mask. Use for multi-pass keying.</p> <p>Normal (normal): Foreground key is multiplied by source alpha when compositing.</p>

2.8.2 Despill node



This documentation is for version 1.0 of Despill.

Description

Remove the unwanted color contamination of the foreground (spill) caused by the reflected color of the blue-screen/greenscreen.

While a despill operation often only removes green (for greenscreens) this despill also enables adding red and blue to the spill area. A lot of Keyers already have implemented their own despill methods. However, in a lot of cases it is useful to separate the keying process in 2 tasks to get more control over the final result. Normally these tasks are the generation of the alpha mask and the spill correction. The generated alpha Mask (Key) is then used to merge the despill foreground over the new background.

This effect is based on the unspill operations described in section 4.5 of “Digital Compositing for Film and Video” by Steve Wright (Focal Press).

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Screen Type / screenType	Choice	Greenscreen	<p>Select the screen type according to your footage</p> <p>Greenscreen (green): The background screen has a green tint.</p> <p>Bluescreen (blue): The background screen has a blue tint.</p>

Continued on next page

Table 137 – continued from previous page

Parameter / script name	Type	Default	Function
Spillmap Mix / spillmapMix	Double	0.5	This value controls the generation of the spillmap. The spillmap decides in which areas the spill will be removed. To calculate this map the two none screen colors are combined according to this value and then subtracted from the screen color. Greenscreen: 0: limit green by blue 0,5: limit green by the average of red and blue 1: limit green by red Bluescreen: 0: limit blue by green 0,5: limit blue by the average of red and green 1: limit blue by red
Expand Spillmap / expandSpillmap	Double	0	This will expand the spillmap to get rid of still remaining spill. It works by lowering the values that will be subtracted from green or blue.
Spillmap to Alpha / outputSpillMap	Boolean	Off	If checked, this will output the spillmap in the alpha channel.
Red Scale / scaleRed	Double	0	Controls the amount of Red in the spill area
Green Scale / scaleGreen	Double	-1	Controls the amount of Green in the spill area. This value should be negative for greenscreen footage.
Blue Scale / scaleBlue	Double	0	Controls the amount of Blue in the spill area. This value should be negative for bluescreen footage.
Brightness / brightness	Double	0	Controls the brightness of the spill while trying to preserve the colors.
Clamp Black / clampBlack	Boolean	On	All colors below 0 on output are set to 0.
Clamp White / clampWhite	Boolean	Off	All colors above 1 on output are set to 1.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.8.3 Difference node



This documentation is for version 1.0 of Difference.

Description

Produce a rough matte from the difference of two input images.

A is the background without the subject (clean plate). B is the subject with the background. RGB is copied from B, the difference is output to alpha, after applying offset and gain.

See also: http://opticalenquiry.com/nuke/index.php?title=The_Keyer_Nodes#Difference and http://opticalenquiry.com/nuke/index.php?title=Keying_Tips

Inputs

Input	Description	Optional
B	The subject with the background.	No
A	The background without the subject (a clean plate).	No

Controls

Parameter / script name	Type	Default	Function
Offset / offset	Double	0	Value subtracted to each pixel of the output
Gain / gain	Double	1	Multiply each pixel of the output by this value

2.8.4 HueKeyer node



This documentation is for version 1.0 of HueKeyer.

Description

Compute a key depending on hue value.

Hue and saturation are computed from the the source RGB values. Depending on the hue value, the various adjustment values are computed, and then applied:

amount: output transparency for the given hue (amount=1 means alpha=0).

sat_thrsh: if source saturation is below this value, the output transparency is gradually decreased.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Hue Curves / hue	Parameter	amount: sat_thrsh:	Hue-dependent alpha lookup curves: amount: transparency (1-alpha) amount for the given hue sat_thrsh: if source saturation is below this value, transparency is decreased progressively.

2.8.5 Keyer node



This documentation is for version 1.0 of Keyer.

Description

A collection of simple keyers. These work by computing a foreground key from the RGB values of the input image (see the keyerMode parameter).

This foreground key is a scalar from 0 to 1. From the foreground key, a background key (or transparency) is computed.

The function that maps the foreground key to the background key is piecewise linear:

- it is 0 below $A = (\text{center} + \text{toleranceLower} + \text{softnessLower})$
- it is linear between $A = (\text{center} + \text{toleranceLower} + \text{softnessLower})$ and $B = (\text{center} + \text{toleranceLower})$
- it is 1 between $B = (\text{center} + \text{toleranceLower})$ and $C = (\text{center} + \text{toleranceUpper})$
- it is linear between $C = (\text{center} + \text{toleranceUpper})$ and $D = (\text{center} + \text{toleranceUpper} + \text{softnessUpper})$
- it is 0 above $D = (\text{center} + \text{toleranceUpper} + \text{softnessUpper})$

Keyer can pull mattes that correspond to the RGB channels, the luminance and the red, green and blue colors. One very useful application for a luminance mask is to mask out a sky (almost always it is the brightest thing in a landscape).

Conversion from A, B, C, D to Keyer parameters is:

$\text{softnessLower} = (A - B)$

$\text{toleranceLower} = (B - C) / 2$

$\text{center} = (B + C) / 2$

$\text{toleranceUpper} = (C - B) / 2$

$\text{softnessUpper} = (D - C)$

See also:

- http://opticalenquiry.com/nuke/index.php?title=The_Keyer_Nodes#Keyer
- http://opticalenquiry.com/nuke/index.php?title=Green_Screen
- http://opticalenquiry.com/nuke/index.php?title=Keying_Tips

Inputs

In-put	Description	Op-tional
Source	The foreground image to key.	No
InM	The Inside Mask, or holdout matte, or core matte, used to confirm areas that are definitely foreground.	Yes
OutM	The Outside Mask, or garbage matte, used to remove unwanted objects (lighting rigs, and so on) from the foreground. The Outside Mask has priority over the Inside Mask, so that areas where both are one are considered to be outside.	Yes
Bg	The background image to replace the blue/green screen in the foreground.	Yes

Controls

Parameter / script name	Type	Default	Function
Key Color / <code>keyColor</code>	Color	r: 0 g: 0 b: 0	Foreground key color. foreground areas containing the key color are replaced with the background image.
Keyer Mode / <code>mode</code>	Choice	Luminance	<p>The operation used to compute the foreground key.</p> <p>Luminance (luminance): Use the luminance for keying. The foreground key value is in luminance.</p> <p>Color (color): Use the color for keying. If the key color is pure green, this corresponds a green keyer, etc.</p> <p>Screen (screen): Use the color minus the other components for keying. If the key color is pure green, this corresponds a greenscreen, etc. When in screen mode, the upper tolerance should be set to 1.</p> <p>None (none): No keying, just despill color values. You can control despill areas using either set the inside mask, or use with 'Source Alpha' set to 'Add to Inside Mask'. If 'Output Mode' is set to 'Unpremultiplied', this despill the image even if no mask is present.</p>
Luminance Math / <code>luminanceMath</code>	Choice	Rec. 709	<p>Formula used to compute luminance from RGB values.</p> <p>Rec. 709 (rec709): Use Rec. 709 ($0.2126r + 0.7152g + 0.0722b$).</p> <p>Rec. 2020 (rec2020): Use Rec. 2020 ($0.2627r + 0.6780g + 0.0593b$).</p> <p>ACES AP0 (acesap0): Use ACES AP0 ($0.3439664498r + 0.7281660966g + -0.0721325464b$).</p> <p>ACES AP1 (acesap1): Use ACES AP1 ($0.2722287168r + 0.6740817658g + 0.0536895174b$).</p> <p>CCIR 601 (ccir601): Use CCIR 601 ($0.2989r + 0.5866g + 0.1145b$).</p> <p>Average (average): Use average of r, g, b.</p> <p>Max (max): Use max of r, g, b.</p>
Softness (lower) / <code>softnessLower</code>	Double	-0.5	Width of the lower softness range [key-tolerance-softness, key-tolerance]. Background key value goes from 0 to 1 when foreground key is over this range.
Tolerance (lower) / <code>toleranceLower</code>	Double	0	Width of the lower tolerance range [key-tolerance, key]. Background key value is 1 when foreground key is over this range.
Center / <code>center</code>	Double	1	Foreground key value corresponding to the key color, where the background key should be 1.

Continued on next page

Table 140 – continued from previous page

Parameter / script name	Type	Default	Function
Tolerance (upper) / toleranceUpper	Double	0	Width of the upper tolerance range [key,key+tolerance]. Background key value is 1 when foreground key is over this range. Ignored in Screen keyer mode.
Softness (upper) / softnessUpper	Double	0.5	Width of the upper softness range [key+tolerance,key+tolerance+softness]. Background key value goes from 1 to 0 when foreground key is over this range. Ignored in Screen keyer mode.
Despill / despill	Double	1	Reduces color spill on the foreground object (Screen mode only). Between 0 and 1, only mixed foreground/background regions are despilled. Above 1, foreground regions are despilled too.
Despill Angle / despillAngle	Double	120	Opening of the cone centered around the keyColor where colors are despilled. A larger angle means that more colors are modified.
Output Mode / show	Choice	Intermediate	<p>What image to output.</p> <p>Intermediate (intermediate): Color is the source color. Alpha is the foreground key. Use for multi-pass keying.</p> <p>Premultiplied (premultiplied): Color is the Source color after key color suppression, multiplied by alpha. Alpha is the foreground key.</p> <p>Unpremultiplied (unpremultiplied): Color is the Source color after key color suppression. Alpha is the foreground key.</p> <p>Composite (composite): Color is the composite of Source and Bg. Alpha is the foreground key.</p>
Source Alpha / sourceAlphaHandling	Choice	Ignore	<p>How the alpha embedded in the Source input should be used</p> <p>Ignore (ignore): Ignore the source alpha.</p> <p>Add to Inside Mask (inside): Source alpha is added to the inside mask. Use for multi-pass keying.</p> <p>Normal (normal): Foreground key is multiplied by source alpha when compositing.</p>

2.8.6 MatteMonitor node

This documentation is for version 1.0 of MatteMonitor.

Description

A Matte Monitor: make alpha values that are strictly between 0 and 1 more visible.

After applying a Keyer, a scaling operation is usually applied to clean the matte. However, it is difficult to visualize on the output values that are very close to 0 or 1, but not equal. This plugin can be used to better visualize these values: connect it to the output of the scaling operator, then to a viewer, and visualize the alpha channel.

Alpha values lower or equal to 0 and greater or equal to 1 are leaved untouched, and alpha values in between are stretched towards 0.5 (using the slope parameter), making them more visible.

The output of this plugin should not be used for firther processing, but only for viewing.

The Matte Monitor is described in “Digital Compositing for Film and Video” by Steve Wright (Sec. 3.1).

See also the video at http://www.vfxio.com/images/movies/Comp_Tip_2.mov

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Slope / <code>slope</code>	Double	0.5	Slope applied to alpha values strictly between 0 and 1.

2.8.7 PIK node



This documentation is for version 1.0 of PIK.

Description

A keyer that works by generating a clean plate from the green/blue screen sequences. Inspired by Nuke's IBK by Paul Lambert and Fusion's KAK by Pieter Van Houte.

There are 2 options to pull a key with PIK. One is to use PIKColor to automatically extract a clean plate from the foreground image and use it as the C input, and the other is to pick a color which best represents the area you are trying to key.

The blue- or greenscreen image should be used as the Fg input, which is used to compute the output color. If that image contains significant noise, a denoised version should be used as the PFg input, which is used to pull the key. The C input should either be a clean plate or the output of PIKColor, and is used as the screen color if the 'Screen Type' is not 'Pick'. The Bg image is used in calculating fine edge detail when either 'Use Bg Luminance' or 'Use Bg Chroma' is checked. Optionally, an inside mask (a.k.a. holdout matte or core matte) and an outside mask (a.k.a. garbage matte) can be connected to inputs InM and OutM. Note that the outside mask takes precedence over the inside mask.

If PIKcolor is used to build the clean plate, the PIKColor Source input should be the same as the PFg input to PIK, e.g. the denoised footage, and the inside mask of PIK can also be fed into the InM input of PIKColor.

The color weights deal with the hardness of the matte. When viewing the output (with screen subtraction checked), one may notice areas where edges have a slight discoloration due to the background not being fully removed from the original plate. This is not spill but a result of the matte being too strong. Lowering one of the weights will correct that particular edge. For example, if it is a red foreground image with an edge problem, lower the red weight. This may affect other edges so the use of multiple PIKs with different weights, split with KeyMixes, is recommended.

The Alpha Bias setting may be used either if there is a strong global color cast on the scene (e.g. the green or blue screen color is not pure), or if parts of the foreground are transparent in the output. This color is considered by the algorithm as being a grey reference: all colors from the PFg input are first normalized by this color before computation.

If the Alpha Bias is set, but the screen subtraction has a strong color bias (e.g. the despill areas show the screen color), uncheck 'Use Alpha for Despill' and set the Despill Bias to the color of the foreground elements that are most affected by the color bias.

‘Screen Subtraction’ (a.k.a. despill) removes the background color from the output via a subtraction process (1-alpha times the screen color is subtracted at each pixel). When unchecked, the output is simply the original Fg premultiplied with the generated matte.

‘Use Bkg Luminance’ and ‘Use Bkg Chroma’ affect the output color by the new background. This feature can also sometimes really help with screens that exhibit some form of fringing artifact - usually a darkening or lightening of an edge on one of the color channels on the screen. The effect can be offset by grading the Bg input up or down with a grade node just before input. If it is just an area which needs help then just rotoscope that area and locally grade the Bg input up or down to remove the artifact.

The output of PIK is controlled by the “Output Mode” option. For example, if the output is “Premultiplied”, it should be composited with the background using a Merge-over operation.

The basic equation used to extract the key in PIK is (in the case of “green” keying):

$$\alpha = 0 \text{ if } (Ag - Ar * rw - Ab * gbw) \text{ is negative, else } 1 - (Ag - Ar * rw - Ab * gbw) / (Bg - Br * rw - Bb * gbw)$$

A is input PFg and B is input C, rw is the value of “Red Weight” and gbw is the value of “Green/Blue Weight”.

See also:

- http://opticalenquiry.com/nuke/index.php?title=The_Keyer_Nodes#IBK
- <https://compositingmentor.com/2014/07/19/advanced-keying-breakdown-alpha-1-4-ibk-stacked-technique/>

Inputs

In-put	Description	Optional
Fg	The blue- or greenscreen image. Used to compute the output color.	No
PFg	(optional) The preprocessed/denoised blue- or greenscreen image. Used to compute the output key (alpha). A denoised image usually gives a less noisy key. If not connected, the Fg input is used instead.	Yes
C	(optional) A clean plate if available, or the output of PIKColor to generate the clean plate at each frame.	Yes
Bg	(optional) The background image. This is used in calculating fine edge detail when the ‘Use Bg Luminance’ or ‘Use Bg Chroma’ options are checked.	Yes

Controls

Parameter / script name	Type	Default	Function
Screen Type / screenType	Choice	C-Blue	The type of background screen used for the key. C-Green (green): Background screen with a green tint. C-Blue (blue): Background screen with a blue tint. Pick (pick): The background screen color is selected by the “color” parameter, and the type of screen (green or blue) is set automatically from this color.
Color / color	Color	r: 0 g: 0 b: 1	The screen color in case ‘Pick’ was chosen as the ‘Screen Type’.
No Key / noKey	Boolean	Off	Apply despill, background luminance and chroma to Fg rgba input using the Fg alpha channel as the key - no key is pulled, but Inside Mask and Outside Mask are applied if connected.

Continued on next page

Table 142 – continued from previous page

Parameter / script name	Type	Default	Function
Red Weight / redWeight	Double	0.5	Determines how the red channel and complement channel (blue for a green screen, green for a blue screen) are weighted in the keying calculation.
Blue/Green Weight / blueGreenWeight	Double	0.5	Determines how the red channel and complement channel (blue for a green screen, green for a blue screen) are weighted in the keying calculation.
Alpha Bias / alphaBias	Color	r: 0.5 g: 0.5 b: 0.5	Divide C and PFg colors by this color before computing alpha. This may be used when the whole scene, including the background, has a strong color cast.
Despill Bias / despillBias	Color	r: 0.5 g: 0.5 b: 0.5	Divide C color by this color before despill.
Use Alpha Bias for Despill / despillBiasIsAlphaBias	Boolean	On	Use alpha bias color for despill instead of despill bias color.
Screen Subtraction / ss	Boolean	On	Have the keyer subtract the foreground or just premult.
Clamp / clampAlpha	Boolean	On	Clamp matte to 0-1.
Clip Black / screenClipMin	Double	0	Any alpha below this value is set to 0.
Clip White / screenClipMax	Double	1	Any alpha above this value is set to 1.
Screen Replace / screenReplace	Choice	Soft Color	<p>What to do with the color of the pixels for which alpha was modified by the screen matte settings.</p> <p>None (none): Subtracted image is not affected by alpha modifications.</p> <p>Source (source): When alpha is modified, a corresponding amount of the Fg color is added.</p> <p>Hard Color (hardcolor): When alpha is modified, a corresponding amount of the replace color is added.</p> <p>Soft Color (softcolor): When alpha is modified, a corresponding amount of the replace color is added, but the resulting luminance is matched with Fg.</p>
Screen Replace Color / screenReplaceColor	Color	r: 0.5 g: 0.5 b: 0.5	The color to use when the Screen Replace parameter is set to Soft or Hard Color.
Source Alpha / sourceAlphaHandling	Choice	Ignore	<p>How the alpha embedded in the Source input should be used</p> <p>Ignore (ignore): Ignore the source alpha.</p> <p>Add to Inside Mask (inside): Source alpha is added to the inside mask. Use for multi-pass keying.</p>

Continued on next page

Table 142 – continued from previous page

Parameter / script name	Type	Default	Function
Inside Replace / <code>insideReplace</code>	Choice	Soft Color	<p>What to do with the color of the pixels for which alpha was modified by the inside mask.</p> <p>None (none): Subtracted image is not affected by alpha modifications.</p> <p>Source (source): When alpha is modified, a corresponding amount of the Fg color is added.</p> <p>Hard Color (hardcolor): When alpha is modified, a corresponding amount of the replace color is added.</p> <p>Soft Color (softcolor): When alpha is modified, a corresponding amount of the replace color is added, but the resulting luminance is matched with Fg.</p>
Inside Replace Color / <code>insideReplaceColor</code>	Color	r: 0.5 g: 0.5 b: 0.5	The color to use when the Inside Replace parameter is set to Soft or Hard Color.
Use Bg Luminance / <code>ubl</code>	Boolean	Off	Have the output RGB be biased by the difference between the Bg luminance and the C luminance). Luminance is computed using the given Colorspace.
Use Bg Chroma / <code>ubc</code>	Boolean	Off	Have the output RGB be biased by the Bg chroma. Chroma is computed using the given Colorspace
Colorspace / <code>colorspace</code>	Choice	Rec. 709	<p>Formula used to compute luminance and chrominance from RGB values for the “Use Bg Luminance” and “Use Bg Choma” options.</p> <p>Rec. 709 (rec709): Use Rec. 709 with D65 illuminant.</p> <p>Rec. 2020 (rec2020): Use Rec. 2020 with D65 illuminant.</p> <p>ACES AP0 (acesap0): Use ACES AP0 with ACES (approx. D60) illuminant.</p> <p>ACES AP1 (acesap1): Use ACES AP1 with ACES (approx. D60) illuminant.</p>

Continued on next page

Table 142 – continued from previous page

Parameter / script name	Type	Default	Function
Output Mode / <code>show</code>	Choice	Premultiplied	<p>What image to output.</p> <p>Source (source): The PFg input (or Fg input, if PFg is not connected).</p> <p>Source Alpha (sourcealpha): The Alpha channel from the PFg input (or Fg input, if PFg is not connected), displayed as luminance.</p> <p>Clean Plate (cleanplate): The clean plate from the C input (or the screen color, if C is not connected).</p> <p>Screen Matte (screenmatte): The screen matte after keying and screen matte processing, but before applying the inside and outside mask, displayed as luminance.</p> <p>Inside Mask (insidemask): The inside mask, displayed as luminance.</p> <p>Outside Mask (outsidemask): The outside mask, displayed as luminance.</p> <p>Combined Matte (matte): The final matte, after applying inside and outside mask, displayed as luminance.</p> <p>Status (status): An image showing which pixels are pure background (black), pure foreground (white), partially transparent (grey), affected by Screen Replace (green), affected by Inside Replace (blue), or affected by Outside Mask (red).</p> <p>Intermediate (intermediate): Color is the source color. Alpha is the foreground key. Use for multi-pass keying.</p> <p>Premultiplied (premultiplied): Color is the Source color after key color suppression, multiplied by alpha. Alpha is the foreground key.</p> <p>Unpremultiplied (unpremultiplied): Color is the Source color after key color suppression. Alpha is the foreground key.</p> <p>Composite (composite): Color is the composite of Source and Bg. Alpha is the foreground key.</p>

2.8.8 PIKColor node



This documentation is for version 1.0 of PIKColor.

Description

This node provides the PIK per-pixel keyer a pseudo clean-plate to be used as color reference.

The idea is to remove the foreground image and only leave the shades and hues of the original blue/greenscreen.

Attach the output of this node to the ‘C’ input of a PIK node. Attach the input of this node and the ‘PFg’ input of PIK to the original screen, or preferably the denoised screen.

Pick which color your screen type is in both nodes and then while viewing the alpha output from PIK lower the darks.b (if a bluescreen - adjust darks.g if a greenscreen) in this node until you see a change in the garbage area of the matte. Once you see a change then you have gone too far -back off a step. If you are still left with discolored edges you can use the other colors in the lights and darks to eliminate them. Remember the idea is to be left with the original shades of the screen and the foreground blacked out. While swapping between viewing the matte from the PIK and the rgb output of PIKColor adjust the other colors until you see a change in the garbage area of the matte. Simple rule of thumb - if you have a light red discolored area increase the lights.r - if you have a dark green

discolored area increase darks.g. If your screen does not have a very saturated hue you may still be left with areas of discoloration after the above process. The 'erode' slider can help with this - while viewing the rgb output adjust the erode until those areas disappear.

The 'Patch Black' slider allows you to fill in the black areas with screen color. This is not always necessary but if you see blue squares in your composite increase this value and it'll fix it.

The optional 'InM' input can be used to provide an inside mask (a.k.a. core matte or holdout matte), which is excluded from the clean plate. If an inside mask is fed into the Keyer (PIK or another Keyer), the same inside mask should be fed inside PIKColor.

The above is the only real workflow for this node - working from the top parameter to the bottom parameter- going back to tweak darks/lights with 'erode' and 'patch black' activated is not really going to work.

Inputs

Input	Description	Optional
Source		No
InM		Yes

Controls

Parameter / script name	Type	Default	Function
Convert to Group / <code>convertToGroup</code>	Button		Converts this node to a Group: the internal node-graph and the user parameters will become editable
Screen Type / <code>screenType</code>	Choice	Blue	Green Blue
Size / <code>size</code>	Double	10	Size of color expansion.
Darks / <code>off</code>	Color	r: 0 g: 0 b: 0	adjust the color values to get the best separation between black and the screen type color. You want to be left with only shades of the screen color and black. If a green screen is selected start by bringing down darks->green If a blue screen is selected start by bringing down darks->blue
Lights / <code>mult</code>	Color	r: 1 g: 1 b: 1	adjust the color values to get the best separation between black and the screen type color. You want to be left with only shades of the screen color and black. If a green screen is selected start by bringing down darks->green If a blue screen is selected start by bringing down darks->blue
Erode / <code>erode</code>	Double	0	increase this value if you still see traces of the foreground edge color in the output
Patch Black / <code>multi</code>	Double	0	Increase this to optionally remove the black from the output. This should only be used once the the above darks/lights have been set.
Filter / <code>filt</code>	Boolean	On	
Level / <code>level</code>	Double	1	multiply the rgb output. Helps remove noise from main key

2.9 Merge nodes

The following sections contain documentation about every node in the Merge group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.9.1 Absminus node

The *Absminus* node is a convenience node identical to the *Merge node*, except that the operator is set to *difference* (a.k.a. *absminus*) by default.

2.9.2 ContactSheet node

This documentation is for version 1.0 of ContactSheet.

Description

Make a contact sheet from several inputs or frames.

Inputs

Input	Description	Optional
0		Yes
1		Yes
2		Yes
3		Yes

Controls

Parameter / script name	Type	Default	Function
Resolution / resolution	Integer	x: 3072 y: 2048	Resolution of the output image, in pixels.
Rows/Columns / rowsColumns	Integer	x: 3 y: 4	How many rows and columns in the grid where the input images or frames are arranged.
Gap / gap	Integer	0	Gap in pixels around each input or frame.
Center / center	Boolean	Off	Center each input/frame within its cell.
Row Order / rowOrder	Choice	BottomTop	How image rows are populated. TopBottom (topbottom) : From top to bottom row. BottomTop (bottomtop) : From bottom to top row.
Column Order / colOrder	Choice	LeftRight	How image columns are populated. LeftRight : From left to right column. RightLeft : From right to left column.
Frame Range / frameRange	Integer	x: 0 y: 0	Frames that are taken from each input. For example, if there are 4 inputs, 'frameRange' is 0-1, and 'absolute' is not checked, the current frame and the next frame is taken from each input, and the contact sheet will contain 8 frames in total.

Continued on next page

Table 144 – continued from previous page

Parameter / script name	Type	Default	Function
Absolute / <code>frameRangeAbsolute</code>	Boolean	Off	If checked, the ‘frameRange’ parameter contains absolute frame numbers.
Enable Selection / <code>selection</code>	Boolean	Off	If checked, the mouse can be used to select an input or frame, and ‘selectionInput’ and ‘selectionFrame’ are set to the selected frame. At at least one keyframe to ‘selectionInput’ and ‘selectionFrame’ to enable time-varying selection.
Selection Input / <code>selectionInput</code>	Integer	0	The selected input. Can be used as the ‘which’ parameter of a Switch effect. At at least one keyframe to this parameter to enable time-varying selection.
Selection Frame / <code>selectionFrame</code>	Integer	0	The selected frame (if <code>frameRangeAbsolute</code> is checked, this is an absolute frame number). Can be used as the ‘firstFrame’ parameter of a FrameHold effect. At at least one keyframe to this parameter to enable time-varying selection.

2.9.3 CopyRectangle node

This documentation is for version 2.0 of CopyRectangle.

Description

Copies a rectangle from the input A to the input B in output.

It can be used to limit an effect to a rectangle of the original image by plugging the original image into the input B.

See also <http://opticalenquiry.com/nuke/index.php?title=CopyRectangle>

Inputs

Input	Description	Optional
B	The image from which the rectangle is copied.	No
A	The image from which the rectangle is copied.	No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Bottom Left / <code>bottomLeft</code>	Double	x: 0 y: 0	Coordinates of the bottom left corner of the rectangle
Size / <code>size</code>	Double	w: 1 h: 1	Width and height of the rectangle
Interactive Update / <code>interactive</code>	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Softness / <code>softness</code>	Double	0	Size of the fade around edges of the rectangle to apply
Invert Mask / <code>maskInvert</code>	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / <code>mix</code>	Double	1	Mix factor between the original and the transformed image.

2.9.4 Dissolve node



This documentation is for version 1.0 of Dissolve.

Description

Weighted average of two inputs.

Inputs

Input	Description	Optional
0		Yes
1		Yes
Mask		Yes
2		Yes

Controls

Parameter / script name	Type	Default	Function
Which / which	Double	0	Mix factor between the inputs.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.

2.9.5 In node

The *In* node is a convenience node identical to the *Merge node*, except that the operator is set to *in* by default.

2.9.6 KeyMix node

This documentation is for version 1.0 of KeyMix.

Description

KeyMix takes two images and layers them together according to a third input. It can be used to lay a foreground over a background using the output of a keyer. The only disadvantage to this method is that it outputs an image with no alpha.

It copies the pixel from A to B only where the Mask is non-zero. It is the same as the Matte operation, but alpha for input A is taken from an external mask, and the output alpha is mixed between A and B. The output bounding box is the union of A and B.

As well as functioning as a layering node, it can also be used to integrate two color operations with one mask. This guards against ‘recycled masks’, where two consecutive color filters are masked using the same mask, which may generate strange artifacts.

See also: <http://opticalenquiry.com/nuke/index.php?title=KeyMix>

Inputs

Input	Description	Optional
B	The main input. This input is passed through when the KeyMix node is disabled.	Yes
A	The image sequence to mix with input B.	Yes
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.9.7 LayerContactSheet node

This documentation is for version 1.0 of LayerContactSheet.

Description

Make a contact sheet from all layers.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Resolution / resolution	Integer	x: 3072 y: 2048	Resolution of the output image, in pixels.
Rows/Columns / rowsColumns	Integer	x: 3 y: 4	How many rows and columns in the grid where the input images or frames are arranged.
Automatic Rows/Columns / autoDims	Boolean	On	Automatically sets the number of rows/columns to display all layers.
Gap / gap	Integer	0	Gap in pixels around each input or frame.
Center / center	Boolean	Off	Center each input/frame within its cell.
Row Order / rowOrder	Choice	TopBottom	How image rows are populated. TopBottom (topbottom) : From top to bottom row. BottomTop (bottomtop) : From bottom to top row.

Continued on next page

Table 148 – continued from previous page

Parameter / script name	Type	Default	Function
Column Order / <code>colOrder</code>	Choice	LeftRight	How image columns are populated. LeftRight (leftright) : From left to right column. RightLeft (rightleft) : From right to left column.
Show Layer Names / <code>showLayerNames</code>	Boolean	Off	Display the layer name in the bottom left of each frame.

2.9.8 Matte node

The *Matte* node is a convenience node identical to the *Merge node*, except that the operator is set to *matte* by default.

2.9.9 Max node

The *Max* node is a convenience node identical to the *Merge node*, except that the operator is set to *max* by default.

2.9.10 Merge node



This documentation is for version 1.0 of Merge.

Description

Pixel-by-pixel merge operation between two or more inputs. Input A is first merged with B (or with a black and transparent background if B is not connected), then A2, if connected, is merged with the intermediary result, then A3, etc.

A complete explanation of the Porter-Duff compositing operators can be found in “Compositing Digital Images”, by T. Porter and T. Duff (Proc. SIGGRAPH 1984) <http://keithp.com/~keithp/porterduff/p253-porter.pdf>

Note that if an input with only RGB components is connected to A or B, its alpha channel is considered to be transparent (zero) by default, and the “A” checkbox for the given input is automatically unchecked, unless it is set explicitly by the user. In fact, most of the time, RGB images without an alpha channel are only used as background images in the B input, and should be considered as transparent, since they should not occlude anything. That way, the alpha channel on output only contains the opacity of elements that are merged with this background. In some rare cases, though, one may want the RGB image to actually be opaque, and can check the “A” checkbox for the given input to do so.

Operators

The following operators are available.

Porter-Duff compositing operators

- copy: A (a.k.a. src)
- over: $A+B(1-a)$ (a.k.a. src-over)

- under: $A(1-b)+B$ (a.k.a. dst-over)
- in: Ab (a.k.a. src-in)
- mask: Ba (a.k.a. dst-in)
- out: $A(1-b)$ (a.k.a. src-out)
- stencil: $B(1-a)$ (a.k.a. dst-out)
- atop: $Ab + B(1 - a)$ (a.k.a. src-atop)
- xor: $A(1-b)+B(1-a)$

Blend modes, see https://en.wikipedia.org/wiki/Blend_modes

Multiply and Screen

- multiply: AB , 0 if $A < 0$ and $B < 0$
- screen: $A+B-AB$ if A or $B \leq 1$, otherwise $\max(A, B)$
- overlay: multiply if $B < 0.5$, screen if $B > 0.5$
- hard-light: multiply if $A < 0.5$, screen if $A > 0.5$
- soft-light: burn-in if $A < 0.5$, lighten if $A > 0.5$

Dodge and burn

- color-dodge: brighten B towards A
- color-burn: darken B towards A
- pinlight: if $B \geq 0.5$ then $\max(A, 2*B - 1)$, $\min(A, B * 2.0)$ else
- difference: $\text{abs}(A-B)$ (a.k.a. absminus)
- exclusion: $A+B-2AB$
- divide: A/B , 0 if $A < 0$ and $B < 0$

Simple arithmetic blend modes

- divide: A/B , 0 if $A < 0$ and $B < 0$
- plus: $A+B$ (a.k.a. add)
- from: $B-A$ (a.k.a. subtract)
- minus: $A-B$
- difference: $\text{abs}(A-B)$ (a.k.a. absminus)
- min: $\min(A, B)$ (a.k.a. darken only)
- max: $\max(A, B)$ (a.k.a. lighten only)

Hue, saturation and luminosity

- hue: $\text{SetLum}(\text{SetSat}(A, \text{Sat}(B)), \text{Lum}(B))$
- saturation: $\text{SetLum}(\text{SetSat}(B, \text{Sat}(A)), \text{Lum}(B))$
- color: $\text{SetLum}(A, \text{Lum}(B))$

- luminosity: $\text{SetLum}(B, \text{Lum}(A))$

Other

- average: $(A + B) / 2$
- conjoint-over: $A + B(1-a)/b$, A if $a > b$
- disjoint-over: $A+B(1-a)/b$, $A+B$ if $a+b < 1$
- freeze: $1-\text{sqrt}(1-A)/B$
- geometric: $2AB/(A+B)$
- grain-extract: $B - A + 0.5$
- grain-merge: $B + A - 0.5$
- hypot: $\text{sqrt}(A*A+B*B)$
- matte: $Aa + B(1-a)$ (unpremultiplied over)
- reflect: $A*A / (1 - B)$

See also:

- “Digital Image Compositing” by Marc Levoy <https://graphics.stanford.edu/courses/cs248-06/comp/comp.html>
- “SVG Compositing Specification” <https://www.w3.org/TR/SVGCompositing/>
- “ISO 32000-1:2008: Portable Document Format (July 2008)”, Sec. 11.3 “Basic Compositing Operations” http://www.adobe.com/devnet/pdf/pdf_reference.html
- “Merge” by Martin Constable <http://opticalenquiry.com/nuke/index.php?title=Merge>
- “Merge Blend Modes” by Martin Constable http://opticalenquiry.com/nuke/index.php?title=Merge_Blend_Modes
- “Primacy of the B Feed” by Martin Constable http://opticalenquiry.com/nuke/index.php?title=Primacy_of_the_B_Feed
- grain-extract and grain-merge are described in <http://docs.gimp.org/en/gimp-concepts-layer-modes.html>

Inputs

Input	Description	Optional
B	The main input. This input is passed through when the merge node is disabled.	Yes
A	The image sequence to merge with input B.	Yes
Mask		Yes
A2		Yes

Controls

Parameter / script name	Type	Default	Function
Operation / operation	Choice	over	<p>The operation used to merge the input A and B images.</p> <p>The operator formula is applied to each component: A and B represent the input component (Red, Green, Blue, or Alpha) of each input, and a and b represent the Alpha component of each input.</p> <p>If Alpha masking is checked, the output alpha is computed using a different formula ($a+b - a*b$).</p> <p>Alpha masking is always enabled for HSL modes (hue, saturation, color, luminosity).</p> <p>atop: $Ab + B(1 - a)$ (a.k.a. src-atop)</p> <p>average: $(A + B) / 2$</p> <p>color: SetLum(A, Lum(B))</p> <p>color-burn: darken B towards A</p> <p>color-dodge: brighten B towards A</p> <p>conjoint-over: $A + B(1-a)/b$, A if $a > b$</p> <p>copy: A (a.k.a. src)</p> <p>difference: $\text{abs}(A-B)$ (a.k.a. absminus)</p> <p>disjoint-over: $A+B(1-a)/b$, $A+B$ if $a+b < 1$</p> <p>divide: A/B, 0 if $A < 0$ and $B < 0$</p> <p>exclusion: $A+B-2AB$</p> <p>freeze: $1-\text{sqrt}(1-A)/B$</p> <p>from: $B-A$ (a.k.a. subtract)</p> <p>geometric: $2AB/(A+B)$</p> <p>grain-extract: $B - A + 0.5$</p> <p>grain-merge: $B + A - 0.5$</p> <p>hard-light: multiply if $A < 0.5$, screen if $A > 0.5$</p> <p>hue: SetLum(SetSat(A, Sat(B)), Lum(B))</p> <p>hypot: $\text{sqrt}(A*A+B*B)$</p> <p>in: Ab (a.k.a. src-in)</p> <p>luminosity: SetLum(B, Lum(A))</p> <p>mask: Ba (a.k.a. dst-in)</p> <p>matte: $Aa + B(1-a)$ (unpremultiplied over)</p> <p>max: $\max(A, B)$ (a.k.a. lighten only)</p> <p>min: $\min(A, B)$ (a.k.a. darken only)</p> <p>minus: $A-B$</p> <p>multiply: AB, 0 if $A < 0$ and $B < 0$</p> <p>out: $A(1-b)$ (a.k.a. src-out)</p> <p>over: $A+B(1-a)$ (a.k.a. src-over)</p> <p>overlay: multiply if $B < 0.5$, screen if $B > 0.5$</p> <p>pinlight: if $B \geq 0.5$ then $\max(A, 2*B - 1)$, $\min(A, B * 2.0)$ else</p> <p>plus: $A+B$ (a.k.a. add)</p> <p>reflect: $A*A / (1 - B)$</p> <p>saturation: SetLum(SetSat(B, Sat(A)), Lum(B))</p> <p>screen: $A+B-AB$ if A or $B \leq 1$, otherwise $\max(A, B)$</p> <p>soft-light: burn-in if $A < 0.5$, lighten if $A > 0.5$</p> <p>stencil: $B(1-a)$ (a.k.a. dst-out)</p> <p>under: $A(1-b)+B$ (a.k.a. dst-over)</p> <p>xor: $A(1-b)+B(1-a)$</p>

Continued on next page

Table 149 – continued from previous page

Parameter / script name	Type	Default	Function
Bounding Box / bbox	Choice	Union	What to use to produce the output image's bounding box. Union (union) : Union of all connected inputs. Intersection (intersection) : Intersection of all connected inputs. A (a) : Bounding box of input A. B (b) : Bounding box of input B.
Alpha masking / screenAlpha	Boolean	Off	When enabled, the input images are unchanged where the other image has 0 alpha, and the output alpha is set to $a+b - a*b$. When disabled the alpha channel is processed as any other channel. Option is disabled for operations where it does not apply or makes no difference.
R / AChannelsR	Boolean	On	Use red component from A input(s).
G / AChannelsG	Boolean	On	Use green component from A input(s).
B / AChannelsB	Boolean	On	Use blue component from A input(s).
A / AChannelsA	Boolean	On	Use alpha component from A input(s).
R / BChannelsR	Boolean	On	Use red component from B input.
G / BChannelsG	Boolean	On	Use green component from B input.
B / BChannelsB	Boolean	On	Use blue component from B input.
A / BChannelsA	Boolean	On	Use alpha component from B input.
R / OutputChannelsR	Boolean	On	Write red component to output.
G / OutputChannelsG	Boolean	On	Write green component to output.
B / OutputChannelsB	Boolean	On	Write blue component to output.
A / OutputChannelsA	Boolean	On	Write alpha component to output.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.9.11 Min node

The *Min* node is a convenience node identical to the *Merge node*, except that the operator is set to *min* by default.

2.9.12 Multiply node

The *Multiply* node is a convenience node identical to the *Merge node*, except that the operator is set to *multiply* by default.

2.9.13 Out node

The *Out* node is a convenience node identical to the *Merge node*, except that the operator is set to *out* by default.

2.9.14 Plus node

The *Plus* node is a convenience node identical to the *Merge node*, except that the operator is set to *plus* by default.

2.9.15 Premult node



This documentation is for version 2.0 of Premult.

Description

Multiply the selected channels by alpha (or another channel).

If no channel is selected, or the premultChannel is set to None, the image data is left untouched, but its premultiplication state is set to PreMultiplied.

See also: <http://opticalenquiry.com/nuke/index.php?title=Premultiplication>

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Plane / inputPlane	Choice	Color.RGBA	<p>The plane channels to premult</p> <p>Color.RGBA (uk.co.thefoundry.OfxImagePlaneColour)</p> <p>DisparityLeft.Disparity (uk.co.thefoundry.OfxImagePlaneStereoDisparityLeft)</p> <p>DisparityRight.Disparity (uk.co.thefoundry.OfxImagePlaneStereoDisparityRight)</p> <p>Backward.Motion (uk.co.thefoundry.OfxImagePlaneBackMotionVector)</p> <p>Forward.Motion (uk.co.thefoundry.OfxImagePlaneForwardMotionVector)</p>
By / premultChannel	Choice	Color.A	<p>The channel to use for (un)premult.</p> <p>Color.R (uk.co.thefoundry.OfxImagePlaneColour.R): R channel from input Source</p> <p>Color.G (uk.co.thefoundry.OfxImagePlaneColour.G): G channel from input Source</p> <p>Color.B (uk.co.thefoundry.OfxImagePlaneColour.B): B channel from input Source</p> <p>Color.A (uk.co.thefoundry.OfxImagePlaneColour.A): A channel from input Source</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p>

Continued on next page

Table 150 – continued from previous page

Parameter / script name	Type	Default	Function
All Planes / processAllPlanes	Boolean	Off	When checked all planes in input will be processed and output to the same plane as in input. It is useful for example to apply a Transform effect on all planes.
Clip Info... / clipInfo	Button		Display information about the inputs

2.9.16 RotoMerge node

This documentation is for version 1.0 of RotoMerge.

Description

Pixel-by-pixel merge operation between two inputs using and external alpha component for input A. All channels from input A are merged with those from B, using RotoMask as the alpha component for input A: the alpha channel from A is thus merged onto the alpha channel from B using the RotoMask as the alpha value (“a” in the formulas). This may be useful, for example, to “paint” alpha values from A onto the alpha channel of B using a given operation with an external alpha mask (which may be opaque even where the alpha channel of A is zero).

A complete explanation of the Porter-Duff compositing operators can be found in “Compositing Digital Images”, by T. Porter and T. Duff (Proc. SIGGRAPH 1984) <http://keithp.com/~keithp/porterduff/p253-porter.pdf>

Note that if an input with only RGB components is connected to A or B, its alpha channel is considered to be transparent (zero) by default, and the “A” checkbox for the given input is automatically unchecked, unless it is set explicitly by the user. In fact, most of the time, RGB images without an alpha channel are only used as background images in the B input, and should be considered as transparent, since they should not occlude anything. That way, the alpha channel on output only contains the opacity of elements that are merged with this background. In some rare cases, though, one may want the RGB image to actually be opaque, and can check the “A” checkbox for the given input to do so.

Operators

The following operators are available.

Porter-Duff compositing operators

- copy: A (a.k.a. src)
- over: $A+B(1-a)$ (a.k.a. src-over)
- under: $A(1-b)+B$ (a.k.a. dst-over)
- in: Ab (a.k.a. src-in)
- mask: Ba (a.k.a. dst-in)
- out: $A(1-b)$ (a.k.a. src-out)
- stencil: $B(1-a)$ (a.k.a. dst-out)
- atop: $Ab + B(1 - a)$ (a.k.a. src-atop)
- xor: $A(1-b)+B(1-a)$

Blend modes, see https://en.wikipedia.org/wiki/Blend_modes

Multiply and Screen

- multiply: AB , 0 if $A < 0$ and $B < 0$
- screen: $A+B-AB$ if A or $B \leq 1$, otherwise $\max(A, B)$
- overlay: multiply if $B < 0.5$, screen if $B > 0.5$
- hard-light: multiply if $A < 0.5$, screen if $A > 0.5$
- soft-light: burn-in if $A < 0.5$, lighten if $A > 0.5$

Dodge and burn

- color-dodge: brighten B towards A
- color-burn: darken B towards A
- pinlight: if $B \geq 0.5$ then $\max(A, 2*B - 1)$, $\min(A, B * 2.0)$ else
- difference: $\text{abs}(A-B)$ (a.k.a. absminus)
- exclusion: $A+B-2AB$
- divide: A/B , 0 if $A < 0$ and $B < 0$

Simple arithmetic blend modes

- divide: A/B , 0 if $A < 0$ and $B < 0$
- plus: $A+B$ (a.k.a. add)
- from: $B-A$ (a.k.a. subtract)
- minus: $A-B$
- difference: $\text{abs}(A-B)$ (a.k.a. absminus)
- min: $\min(A, B)$ (a.k.a. darken only)
- max: $\max(A, B)$ (a.k.a. lighten only)

Hue, saturation and luminosity

- hue: $\text{SetLum}(\text{SetSat}(A, \text{Sat}(B)), \text{Lum}(B))$
- saturation: $\text{SetLum}(\text{SetSat}(B, \text{Sat}(A)), \text{Lum}(B))$
- color: $\text{SetLum}(A, \text{Lum}(B))$
- luminosity: $\text{SetLum}(B, \text{Lum}(A))$

Other

- average: $(A + B) / 2$
- conjoint-over: $A + B(1-a)/b$, A if $a > b$
- disjoint-over: $A+B(1-a)/b$, $A+B$ if $a+b < 1$
- freeze: $1-\sqrt{1-A}/B$

- geometric: $2AB/(A+B)$
- grain-extract: $B - A + 0.5$
- grain-merge: $B + A - 0.5$
- hypot: $\sqrt{A^2+B^2}$
- matte: $Aa + B(1-a)$ (unpremultiplied over)
- reflect: $A*A / (1 - B)$

See also:

- “Digital Image Compositing” by Marc Levoy <https://graphics.stanford.edu/courses/cs248-06/comp/comp.html>
- “SVG Compositing Specification” <https://www.w3.org/TR/SVGCompositing/>
- “ISO 32000-1:2008: Portable Document Format (July 2008)”, Sec. 11.3 “Basic Compositing Operations” http://www.adobe.com/devnet/pdf/pdf_reference.html
- “Merge” by Martin Constable <http://opticalenquiry.com/nuke/index.php?title=Merge>
- “Merge Blend Modes” by Martin Constable http://opticalenquiry.com/nuke/index.php?title=Merge_Blend_Modes
- “Primacy of the B Feed” by Martin Constable http://opticalenquiry.com/nuke/index.php?title=Primacy_of_the_B_Feed
- grain-extract and grain-merge are described in <http://docs.gimp.org/en/gimp-concepts-layer-modes.html>

Inputs

Input	Description	Optional
B	The main input. This input is passed through when the merge node is disabled.	Yes
A	The image sequence to merge with input B.	Yes
Mask		Yes
A2		Yes

Controls

Parameter / script name	Type	Default	Function
Operation / operation	Choice	over	<p>The operation used to merge the input A and B images.</p> <p>The operator formula is applied to each component: A and B represent the input component (Red, Green, Blue, or Alpha) of each input, and a and b represent the Alpha component of each input.</p> <p>If Alpha masking is checked, the output alpha is computed using a different formula ($a+b - a*b$).</p> <p>Alpha masking is always enabled for HSL modes (hue, saturation, color, luminosity).</p> <p>atop: $Ab + B(1 - a)$ (a.k.a. src-atop)</p> <p>average: $(A + B) / 2$</p> <p>color: SetLum(A, Lum(B))</p> <p>color-burn: darken B towards A</p> <p>color-dodge: brighten B towards A</p> <p>conjoint-over: $A + B(1-a)/b$, A if $a > b$</p> <p>copy: A (a.k.a. src)</p> <p>difference: $\text{abs}(A-B)$ (a.k.a. absminus)</p> <p>disjoint-over: $A+B(1-a)/b$, A+B if $a+b < 1$</p> <p>divide: A/B, 0 if $A < 0$ and $B < 0$</p> <p>exclusion: $A+B-2AB$</p> <p>freeze: $1-\text{sqrt}(1-A)/B$</p> <p>from: $B-A$ (a.k.a. subtract)</p> <p>geometric: $2AB/(A+B)$</p> <p>grain-extract: $B - A + 0.5$</p> <p>grain-merge: $B + A - 0.5$</p> <p>hard-light: multiply if $A < 0.5$, screen if $A > 0.5$</p> <p>hue: SetLum(SetSat(A, Sat(B)), Lum(B))</p> <p>hypot: $\text{sqrt}(A*A+B*B)$</p> <p>in: Ab (a.k.a. src-in)</p> <p>luminosity: SetLum(B, Lum(A))</p> <p>mask: Ba (a.k.a. dst-in)</p> <p>matte: $Aa + B(1-a)$ (unpremultiplied over)</p> <p>max: $\max(A, B)$ (a.k.a. lighten only)</p> <p>min: $\min(A, B)$ (a.k.a. darken only)</p> <p>minus: $A-B$</p> <p>multiply: AB, 0 if $A < 0$ and $B < 0$</p> <p>out: $A(1-b)$ (a.k.a. src-out)</p> <p>over: $A+B(1-a)$ (a.k.a. src-over)</p> <p>overlay: multiply if $B < 0.5$, screen if $B > 0.5$</p> <p>pinlight: if $B \geq 0.5$ then $\max(A, 2*B - 1)$, $\min(A, B * 2.0)$ else</p> <p>plus: $A+B$ (a.k.a. add)</p> <p>reflect: $A*A / (1 - B)$</p> <p>saturation: SetLum(SetSat(B, Sat(A)), Lum(B))</p> <p>screen: $A+B-AB$ if A or B ≤ 1, otherwise $\max(A, B)$</p> <p>soft-light: burn-in if $A < 0.5$, lighten if $A > 0.5$</p> <p>stencil: $B(1-a)$ (a.k.a. dst-out)</p> <p>under: $A(1-b)+B$ (a.k.a. dst-over)</p> <p>xor: $A(1-b)+B(1-a)$</p>

Continued on next page

Table 151 – continued from previous page

Parameter / script name	Type	Default	Function
Bounding Box / bbox	Choice	Union	What to use to produce the output image's bounding box. Union (union) : Union of all connected inputs. Intersection (intersection) : Intersection of all connected inputs. A (a) : Bounding box of input A. B (b) : Bounding box of input B.
Alpha masking / screenAlpha	Boolean	Off	When enabled, the input images are unchanged where the other image has 0 alpha, and the output alpha is set to $a+b - a*b$. When disabled the alpha channel is processed as any other channel. Option is disabled for operations where it does not apply or makes no difference.
R / AChannelsR	Boolean	On	Use red component from A input(s).
G / AChannelsG	Boolean	On	Use green component from A input(s).
B / AChannelsB	Boolean	On	Use blue component from A input(s).
A / AChannelsA	Boolean	On	Use alpha component from A input(s).
R / BChannelsR	Boolean	On	Use red component from B input.
G / BChannelsG	Boolean	On	Use green component from B input.
B / BChannelsB	Boolean	On	Use blue component from B input.
A / BChannelsA	Boolean	On	Use alpha component from B input.
R / OutputChannelsR	Boolean	On	Write red component to output.
G / OutputChannelsG	Boolean	On	Write green component to output.
B / OutputChannelsB	Boolean	On	Write blue component to output.
A / OutputChannelsA	Boolean	On	Write alpha component to output.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.9.17 Screen node

The *Screen* node is a convenience node identical to the *Merge node*, except that the operator is set to *screen* by default.

2.9.18 SeExpr node

s(E)

This documentation is for version 2.0 of SeExpr.

Description

Use the SeExpr expression language (by Walt Disney Animation Studios) to process images.

What is SeExpr?

SeExpr is a very simple mathematical expression language used in graphics software (RenderMan, Maya, Mudbox, Yeti).

See the [SeExpr Home Page](#) and [SeExpr Language Documentation](#) for more information.

SeExpr is licensed under the Apache License, Version 2.0, and is Copyright Disney Enterprises, Inc.

SeExpr vs. SeExprSimple

The SeExpr plugin comes in two versions:

- *SeExpr* has a single vector expression for the color channels, and a scalar expression for the alpha channel. The source color is accessed through the `Cs` vector, and alpha through the `As` scalar, as specified in the original SeExpr language.
- *SeExprSimple* has one scalar expression per channel, and the source channels may also be accessed through scalars (`r`, `g`, `b`, `a`).

SeExpr extensions

A few pre-defined variables and functions were added to the language for filtering and blending several input images.

The following pre-defined variables can be used in the script:

- `x`: X coordinate (in pixel units) of the pixel to render.
- `y`: Y coordinate (in pixel units) of the pixel to render.
- `u`: X coordinate (normalized in the [0,1] range) of the output pixel to render.
- `v`: Y coordinate (normalized in the [0,1] range) of the output pixel to render.
- `sx`, `sy`: Scale at which the image is being rendered. Depending on the zoom level of the viewer, the image might be rendered at a lower scale than usual. This parameter is useful when producing spatial effects that need to be invariant to the pixel scale, especially when using X and Y coordinates. (0.5,0.5) means that the image is being rendered at half of its original size.
- `par`: The pixel aspect ratio.
- `cx`, `cy`: Shortcuts for $(x + 0.5) / par / sx$ and $(y + 0.5) / sy$, i.e. the canonical coordinates of the current pixel.
- `frame`: Current frame being rendered
- `Cs`, `As`: Color (RGB vector) and alpha (scalar) of the image from input 1.
- `CsN`, `AsN`: Color (RGB vector) and alpha (scalar) of the image from input N, e.g. `Cs2` and `As2` for input 2.
- `output_width`, `output_height`: Dimensions of the output image being rendered.
- `input_width`, `input_height`: Dimensions of image from input 1, in pixels.
- `input_widthN`, `input_heightN`: Dimensions of image from input N, e.g. `input_width2` and `input_height2` for input 2.

The following additional functions are available:

- `color cpixel(int i, int f, float x, float y, int interp = 0)`: interpolates the color from input `i` at the pixel position (x,y) in the image, at frame `f`.
- `float apixel(int i, int f, float x, float y, int interp = 0)`: interpolates the alpha from input `i` at the pixel position (x,y) in the image, at frame `f`.

The pixel position of the center of the bottom-left pixel is (0., 0.).

The first input has index $i=1$.

`interp` controls the interpolation filter, and can take one of the following values:

- 0: impulse - (nearest neighbor / box) Use original values
- 1: bilinear - (tent / triangle) Bilinear interpolation between original values
- 2: cubic - (cubic spline) Some smoothing
- 3: Keys - (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*)
- 4: Simon - Some smoothing, plus medium sharpening (*)
- 5: Rifman - Some smoothing, plus significant sharpening (*)
- 6: Mitchell - Some smoothing, plus blurring to hide pixelation (*+)
- 7: Parzen - (cubic B-spline) Greatest smoothing of all filters (+)
- 8: notch - Flat smoothing (which tends to hide moire' patterns) (+)

Some filters may produce values outside of the initial range (*) or modify the values even at integer positions (+).

Sample scripts

Add green channel to red, keep green, and apply a 50% gain on blue

SeExprSimple:

```
r+g
g
0.5*b
```

SeExpr:

```
[Cs[0]+Cs[1], Cs[1], 0.5*Cs[2]]
```

“Multiply” merge operator on inputs 1 and 2

SeExprSimple:

```
r*r2
g*g2
b*b2
a+a2-a*a2
```

SeExpr:

```
Cs * Cs2
As + As2 - As * As2
```

“Over” merge operator on inputs 1 and 2

SeExprSimple:

```
r+r2*(1-a)
g+g2*(1-a)
b+b2*(1-a)
a+a2-a*a2
```

SeExpr:

```
Cs + Cs2 * (1 - As)
As + As2 - As * As2
```

Generating a time-varying colored Perlin noise with size x1

```
cnoise([cx/x1,cy/x1,frame])
```

Average pixels over the previous, current and next frame

SeExpr:

```
prev = cpixel(1,frame - 1,x,y);
cur = Cs;
next = cpixel(1,frame + 1,x,y);
(prev + cur + next) / 3;
```

“Wave” - displace columns of pixels vertically according to a sine wave function

SeExpr:

```
cpixel(1,frame,x,y+x2*sy*sin(2*3.1416*(x/sx - x3)/x1),2)
```

Set the No. of scalar params to 3.

- x1 is the horizontal wavelength in pixels.
- x2 is the vertical amplitude in pixels.
- x3 is the horizontal shift in pixels.

Custom parameters

To use custom variables that are pre-defined in the plug-in (scalars, positions and colors) you must reference them using their script-name in the expression. For example, the parameter x1 can be referenced using x1 in the script:

```
Cs + x1
```

Multi-instruction expressions

If an expression spans multiple instructions (usually written one per line), each instruction must end with a semi-column (;). The last instruction of the expression is considered as the final value of the pixel (a RGB vector or an Alpha scalar, depending on the script), and must not be terminated by a semicolon. More documentation is available on the [SeExpr website](#).

Accessing pixel values from other frames

The input frame range used to render a given output frame is computed automatically if the following conditions hold:

- The `frame` parameter to `cpixel/apixel` must not depend on the color or alpha of a pixel, nor on the result of another call to `cpixel/apixel`

- A call to `cpixel/apixel` must not depend on the color or alpha of a pixel, as in the following:

```
if (As > 0.1) { src = cpixel(1,frame,x,y); } else { src = [0,0,0]; }
```

If one of these conditions does not hold, all frames from the specified input frame range are asked for.

Inputs

Input	Description	Optional
1		Yes
2		Yes
3		Yes
4		Yes

Controls

Parameter / script name	Type	Default	Function
Region of Definition / <code>rod</code>	Choice	Union	<p>Region of definition (extent) of the output.</p> <p>Union (union): The output region is the union of the regions of definition of all connected inputs.</p> <p>Intersection (intersection): The output region is the intersection the regions of definition of all connected inputs.</p> <p>Size (size): The output region is the size of the rectangle overlay.</p> <p>Format (format): The output region is the specified format.</p> <p>Project (project): The output region is the size of the project.</p> <p>Input1 (input1): The output region is the region of definition of input 1</p> <p>Input2 (input2): The output region is the region of definition of input 2</p> <p>Input3 (input3): The output region is the region of definition of input 3</p> <p>Input4 (input4): The output region is the region of definition of input 4</p> <p>Input5 (input5): The output region is the region of definition of input 5</p> <p>Input6 (input6): The output region is the region of definition of input 6</p> <p>Input7 (input7): The output region is the region of definition of input 7</p> <p>Input8 (input8): The output region is the region of definition of input 8</p> <p>Input9 (input9): The output region is the region of definition of input 9</p> <p>Input10 (input10): The output region is the region of definition of input 10</p>

Continued on next page

Table 152 – continued from previous page

Parameter / script name	Type	Default	Function
Output components / outputComponents	Choice	RGBA	Specify what components to output. In RGB only, the alpha script will not be executed. Similarly, in alpha only, the RGB script will not be executed. RGBA RGB Alpha
Format / format	Choice	PC_Video 640x480	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x1720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super35(full-ap) 1024x778 (1K_Super35(full-ap)) 1K_Cinemascope 914x778 2 (1K_Cinemascope) 2K_Super35(full-ap) 2048x1556 (2K_Super35(full-ap)) 2K_Cinemascope 1828x1556 2 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super35(full-ap) 4096x3112 (4K_Super35(full-ap)) 4K_Cinemascope 3656x3112 2 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
No. of Scalar Params / doubleParamsNb	Integer	0	Use this to control how many scalar parameters should be exposed to the SeExpr expression.
x1 / x1	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x1
x2 / x2	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x2
x3 / x3	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x3
x4 / x4	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x4
x5 / x5	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x5

Continued on next page

Table 152 – continued from previous page

Parameter / script name	Type	Default	Function
x6 / x6	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x6
x7 / x7	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x7
x8 / x8	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x8
x9 / x9	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x9
x10 / x10	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x10
No. of 2D Params / double2DParamsNb	Integer	0	Use this to control how many 2D (position) parameters should be exposed to the SeExpr expression.
pos1 / pos1	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos1
pos2 / pos2	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos2
pos3 / pos3	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos3
pos4 / pos4	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos4
pos5 / pos5	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos5
pos6 / pos6	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos6
pos7 / pos7	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos7
pos8 / pos8	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos8
pos9 / pos9	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos9
pos10 / pos10	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos10
No. of Color Params / colorParamsNb	Integer	0	Use this to control how many color parameters should be exposed to the SeExpr expression.
color1 / color1	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color1
color2 / color2	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color2
color3 / color3	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color3
color4 / color4	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color4
color5 / color5	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color5
color6 / color6	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color6
color7 / color7	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color7
color8 / color8	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color8
color9 / color9	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color9

Continued on next page

Table 152 – continued from previous page

Parameter / script name	Type	Default	Function
color10 / color10	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color10
Input Frame Range / frameRange	Integer	min: 0 max: 0	Default input frame range to fetch images from (may be relative or absolute, depending on the “frameRangeAbsolute” parameter). Only used if the frame range cannot be statically computed from the expression. This parameter can be animated.
Absolute Frame Range / frameRangeAbsolute	Boolean	Off	If checked, the frame range is given as absolute frame numbers, else it is relative to the current frame.
RGB Script / script	String		Contents of the SeExpr expression. This expression should output the RGB components as a SeExpr vector. See the description of the plug-in and http://www.disneyanimation.com/technology/seexpr.html for documentation.
Alpha Script / alphaScript	String		Contents of the SeExpr expression. This expression should output the alpha component only as a scalar. See the description of the plug-in and http://www.disneyanimation.com/technology/seexpr.html for documentation.
Help... / helpButton	Button		Display help about using SeExpr.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.9.19 SeExprSimple node

This documentation is for version 2.0 of SeExprSimple.

Description

Use the SeExpr expression language (by Walt Disney Animation Studios) to process images.

What is SeExpr?

SeExpr is a very simple mathematical expression language used in graphics software (RenderMan, Maya, Mudbox, Yeti).

See the [SeExpr Home Page](#) and [SeExpr Language Documentation](#) for more information.

SeExpr is licensed under the Apache License, Version 2.0, and is Copyright Disney Enterprises, Inc.

SeExpr vs. SeExprSimple

The SeExpr plugin comes in two versions:

- *SeExpr* has a single vector expression for the color channels, and a scalar expression for the alpha channel. The source color is accessed through the `Csvector`, and alpha through the `As` scalar, as specified in the original SeExpr language.
- *SeExprSimple* has one scalar expression per channel, and the source channels may also be accessed through scalars (`r`, `g`, `b`, `a`).

SeExpr extensions

A few pre-defined variables and functions were added to the language for filtering and blending several input images.

The following pre-defined variables can be used in the script:

- `x`: X coordinate (in pixel units) of the pixel to render.
- `y`: Y coordinate (in pixel units) of the pixel to render.
- `u`: X coordinate (normalized in the [0,1] range) of the output pixel to render.
- `v`: Y coordinate (normalized in the [0,1] range) of the output pixel to render.
- `sx`, `sy`: Scale at which the image is being rendered. Depending on the zoom level of the viewer, the image might be rendered at a lower scale than usual. This parameter is useful when producing spatial effects that need to be invariant to the pixel scale, especially when using X and Y coordinates. (0.5,0.5) means that the image is being rendered at half of its original size.
- `par`: The pixel aspect ratio.
- `cx`, `cy`: Shortcuts for $(x + 0.5) / par / sx$ and $(y + 0.5) / sy$, i.e. the canonical coordinates of the current pixel.
- `frame`: Current frame being rendered
- *SeExprSimple only*: `r`, `g`, `b`, `a`: RGBA channels (scalar) of the image from input 1.
- *SeExprSimple only*: `rN`, `gN`, `bN`, `aN`: RGBA channels (scalar) of the image from input N, e.g. `r2` and `a2` are red and alpha channels from input 2.
- `Cs`, `As`: Color (RGB vector) and alpha (scalar) of the image from input 1.
- `CsN`, `AsN`: Color (RGB vector) and alpha (scalar) of the image from input N, e.g. `Cs2` and `As2` for input 2.
- `output_width`, `output_height`: Dimensions of the output image being rendered.
- `input_width`, `input_height`: Dimensions of image from input 1, in pixels.
- `input_widthN`, `input_heightN`: Dimensions of image from input N, e.g. `input_width2` and `input_height2` for input 2.

The following additional functions are available:

- `color cpixel(int i, int f, float x, float y, int interp = 0)`: interpolates the color from input `i` at the pixel position (x,y) in the image, at frame `f`.
- `float apixel(int i, int f, float x, float y, int interp = 0)`: interpolates the alpha from input `i` at the pixel position (x,y) in the image, at frame `f`.

The pixel position of the center of the bottom-left pixel is (0., 0.).

The first input has index `i=1`.

`interp` controls the interpolation filter, and can take one of the following values:

- 0: impulse - (nearest neighbor / box) Use original values
- 1: bilinear - (tent / triangle) Bilinear interpolation between original values
- 2: cubic - (cubic spline) Some smoothing
- 3: Keys - (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*)
- 4: Simon - Some smoothing, plus medium sharpening (*)
- 5: Rifman - Some smoothing, plus significant sharpening (*)
- 6: Mitchell - Some smoothing, plus blurring to hide pixelation (*+)
- 7: Parzen - (cubic B-spline) Greatest smoothing of all filters (+)

- 8: notch - Flat smoothing (which tends to hide moire' patterns) (+)

Some filters may produce values outside of the initial range (*) or modify the values even at integer positions (+).

Sample scripts

Add green channel to red, keep green, and apply a 50% gain on blue

SeExprSimple:

```
r+g
g
0.5*b
```

SeExpr:

```
[Cs[0]+Cs[1], Cs[1], 0.5*Cs[2]]
```

“Multiply” merge operator on inputs 1 and 2

SeExprSimple:

```
r*r2
g*g2
b*b2
a+a2-a*a2
```

SeExpr:

```
Cs * Cs2
As + As2 - As * As2
```

“Over” merge operator on inputs 1 and 2

SeExprSimple:

```
r+r2*(1-a)
g+g2*(1-a)
b+b2*(1-a)
a+a2-a*a2
```

SeExpr:

```
Cs + Cs2 * (1 - As)
As + As2 - As * As2
```

Generating a time-varying colored Perlin noise with size x1

```
cnoise([cx/x1,cy/x1,frame])
```

Average pixels over the previous, current and next frame

SeExpr:

```
prev = cpixel(1, frame - 1, x, y);
cur = Cs;
next = cpixel(1, frame + 1, x, y);
(prev + cur + next) / 3;
```

“Wave” - displace columns of pixels vertically according to a sine wave function

SeExpr:

```
cpixel(1, frame, x, y+x2*sy*sin(2*3.1416*(x/sx - x3)/x1), 2)
```

Set the No. of scalar params to 3.

- x1 is the horizontal wavelength in pixels.
- x2 is the vertical amplitude in pixels.
- x3 is the horizontal shift in pixels.

Custom parameters

To use custom variables that are pre-defined in the plug-in (scalars, positions and colors) you must reference them using their script-name in the expression. For example, the parameter x1 can be referenced using x1 in the script:

```
Cs + x1
```

Multi-instruction expressions

If an expression spans multiple instructions (usually written one per line), each instruction must end with a semi-column (;). The last instruction of the expression is considered as the final value of the pixel (a RGB vector or an Alpha scalar, depending on the script), and must not be terminated by a semicolon. More documentation is available on the [SeExpr website](#).

Accessing pixel values from other frames

The input frame range used to render a given output frame is computed automatically if the following conditions hold:

- The `frame` parameter to `cpixel/apixel` must not depend on the color or alpha of a pixel, nor on the result of another call to `cpixel/apixel`
- A call to `cpixel/apixel` must not depend on the color or alpha of a pixel, as in the following:
if (As > 0.1) { src = cpixel(1, frame, x, y); } else { src = [0,0,0]; }

If one of these conditions does not hold, all frames from the specified input frame range are asked for.

Inputs

Input	Description	Optional
1		Yes
2		Yes
3		Yes
4		Yes

Controls

Parameter / script name	Type	Default	Function
Region of Definition / rod	Choice	Union	<p>Region of definition (extent) of the output.</p> <p>Union (union): The output region is the union of the regions of definition of all connected inputs.</p> <p>Intersection (intersection): The output region is the intersection the regions of definition of all connected inputs.</p> <p>Size (size): The output region is the size of the rectangle overlay.</p> <p>Format (format): The output region is the specified format.</p> <p>Project (project): The output region is the size of the project.</p> <p>Input1 (input1): The output region is the region of definition of input 1</p> <p>Input2 (input2): The output region is the region of definition of input 2</p> <p>Input3 (input3): The output region is the region of definition of input 3</p> <p>Input4 (input4): The output region is the region of definition of input 4</p> <p>Input5 (input5): The output region is the region of definition of input 5</p> <p>Input6 (input6): The output region is the region of definition of input 6</p> <p>Input7 (input7): The output region is the region of definition of input 7</p> <p>Input8 (input8): The output region is the region of definition of input 8</p> <p>Input9 (input9): The output region is the region of definition of input 9</p> <p>Input10 (input10): The output region is the region of definition of input 10</p>
Output components / outputComponents	Choice	RGBA	<p>Specify what components to output. In RGB only, the alpha script will not be executed. Similarly, in alpha only, the RGB script will not be executed.</p> <p>RGBA</p> <p>RGB</p> <p>Alpha</p>

Continued on next page

Table 153 – continued from previous page

Parameter / script name	Type	Default	Function
Format / format	Choice	PC_Video 640x480	<p>The output format</p> <p>PC_Video 640x480 (PC_Video)</p> <p>NTSC 720x486 0.91 (NTSC)</p> <p>PAL 720x576 1.09 (PAL)</p> <p>NTSC_16:9 720x486 1.21 (NTSC_16:9)</p> <p>PAL_16:9 720x576 1.46 (PAL_16:9)</p> <p>HD_720 1280x1720 (HD_720)</p> <p>HD 1920x1080 (HD)</p> <p>UHD_4K 3840x2160 (UHD_4K)</p> <p>1K_Super35(full-ap) 1024x778 (1K_Super35(full-ap))</p> <p>1K_Cinemascope 914x778 2 (1K_Cinemascope)</p> <p>2K_Super35(full-ap) 2048x1556 (2K_Super35(full-ap))</p> <p>2K_Cinemascope 1828x1556 2 (2K_Cinemascope)</p> <p>2K_DCP 2048x1080 (2K_DCP)</p> <p>4K_Super35(full-ap) 4096x3112 (4K_Super35(full-ap))</p> <p>4K_Cinemascope 3656x3112 2 (4K_Cinemascope)</p> <p>4K_DCP 4096x2160 (4K_DCP)</p> <p>square_256 256x256 (square_256)</p> <p>square_512 512x512 (square_512)</p> <p>square_1K 1024x1024 (square_1K)</p> <p>square_2K 2048x2048 (square_2K)</p>
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
No. of Scalar Params / doubleParamsNb	Integer	0	Use this to control how many scalar parameters should be exposed to the SeExpr expression.
x1 / x1	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x1
x2 / x2	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x2
x3 / x3	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x3
x4 / x4	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x4
x5 / x5	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x5
x6 / x6	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x6
x7 / x7	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x7
x8 / x8	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x8
x9 / x9	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x9
x10 / x10	Double	0	A custom 1-dimensional variable that can be referenced in the expression by its script-name, x10

Continued on next page

Table 153 – continued from previous page

Parameter / script name	Type	Default	Function
No. of 2D Params / double2DParamsNb	Integer	0	Use this to control how many 2D (position) parameters should be exposed to the SeExpr expression.
pos1 / pos1	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos1
pos2 / pos2	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos2
pos3 / pos3	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos3
pos4 / pos4	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos4
pos5 / pos5	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos5
pos6 / pos6	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos6
pos7 / pos7	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos7
pos8 / pos8	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos8
pos9 / pos9	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos9
pos10 / pos10	Double	x: 0 y: 0	A custom 2-dimensional variable that can be referenced in the expression by its script-name, pos10
No. of Color Params / colorParamsNb	Integer	0	Use this to control how many color parameters should be exposed to the SeExpr expression.
color1 / color1	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color1
color2 / color2	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color2
color3 / color3	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color3
color4 / color4	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color4
color5 / color5	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color5
color6 / color6	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color6
color7 / color7	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color7
color8 / color8	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color8
color9 / color9	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color9
color10 / color10	Color	r: 0 g: 0 b: 0	A custom RGB variable that can be referenced in the expression by its script-name, color10
Input Frame Range / frameRange	Integer	min: 0 max: 0	Default input frame range to fetch images from (may be relative or absolute, depending on the “frameRangeAbsolute” parameter). Only used if the frame range cannot be statically computed from the expression. This parameter can be animated.
Absolute Frame Range / frameRangeAbsolute	Boolean	Off	If checked, the frame range is given as absolute frame numbers, else it is relative to the current frame.
R= / rExpr	String		Expression to compute the output red channel. If empty, the channel is left unchanged.

Continued on next page

Table 153 – continued from previous page

Parameter / script name	Type	Default	Function
G= / gExpr	String		Expression to compute the output green channel. If empty, the channel is left unchanged.
B= / bExpr	String		Expression to compute the output blue channel. If empty, the channel is left unchanged.
A= / aExpr	String		Expression to compute the output alpha channel. If empty, the channel is left unchanged.
Help... / helpButton	Button		Display help about using SeExpr.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.9.20 Switch node



This documentation is for version 1.0 of Switch.

Description

Lets you switch between any number of inputs.

The selected input number may be manually selected using the “which” parameter, or selected automatically if “automatic” is checked.

Automatic selection works by selecting, at any given time, the first input which is connected and has a non-empty region of definition.

A typical use case is a graph where an edited movie is used as input, then split into shots using one FrameRange plugin per shot (with “before” and “after” set to “Black”), followed by a different processing for each shot (e.g. stabilization, color correction, cropping), and all outputs are gathered into an edited movie using a single “Switch” plug-in in automatic mode. In this graph, no plug-in shifts time, and thus there is no risk of desynchronization, whereas using “AppendClip” instead of “Switch” may shift time if there is an error in one of the FrameRange ranges (a typical error is to use the same frame number as the last frame of shot n and the first frame of shot n+1).

This plugin concatenates transforms.

See also: <http://opticalenquiry.com/nuke/index.php?title=Switch>

Inputs

Input	Description	Optional
0		Yes
1		Yes
2		Yes
3		Yes

Controls

Parameter / script name	Type	Default	Function
Which / which	Integer	0	The input to display. Each input is displayed at the value corresponding to the number of the input. For example, setting which to 4 displays the image from input 4.
Automatic / automatic	Boolean	Off	When checked, automatically switch to the first connected input with a non-empty region of definition. This can be used to recompose a single clip from effects applied to different frame ranges.

2.9.21 TimeDissolve node

This documentation is for version 1.0 of TimeDissolve.

Description

Dissolves between two inputs, starting the dissolve at the in frame and ending at the out frame.

You can specify the dissolve curve over time, if the OFX host supports it (else it is a traditional smoothstep).

See also <http://opticalenquiry.com/nuke/index.php?title=TimeDissolve>

Inputs

Input	Description	Optional
B	The input you intend to dissolve from.	Yes
A	The input you intend to dissolve from.	Yes

Controls

Parameter / script name	Type	Default	Function
In / dissolveIn	Integer	1	Start dissolve at this frame number.
Out / dissolveOut	Integer	10	End dissolve at this frame number.
Curve / dissolveCurve	Parametric		Shape of the dissolve. Horizontal value is from 0 to 1: 0 is the frame before the In frame and should have a value of 0; 1 is the frame after the Out frame and should have a value of 1.

2.9.22 Unpremult node

α

This documentation is for version 2.0 of Unpremult.

Description

Divide the selected channels by alpha (or another channel)

If no channel is selected, or the premultChannel is set to None, the image data is left untouched, but its premultiplication state is set to UnPreMultiplied.

See also: <http://opticalenquiry.com/nuke/index.php?title=Premultiplication>

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Plane / <code>inputPlane</code>	Choice	Color.RGBA	<p>The plane channels to premult</p> <p>Color.RGBA (<code>uk.co.thefoundry.OfxImagePlaneColour</code>)</p> <p>DisparityLeft.Disparity (<code>uk.co.thefoundry.OfxImagePlaneStereoDisparityLeft</code>)</p> <p>DisparityRight.Disparity (<code>uk.co.thefoundry.OfxImagePlaneStereoDisparityRight</code>)</p> <p>Backward.Motion (<code>uk.co.thefoundry.OfxImagePlaneBackMotionVector</code>)</p> <p>Forward.Motion (<code>uk.co.thefoundry.OfxImagePlaneForwardMotionVector</code>)</p>
By / <code>premultChannel</code>	Choice	Color.A	<p>The channel to use for (un)premult.</p> <p>Color.R (<code>uk.co.thefoundry.OfxImagePlaneColour.R</code>): R channel from input Source</p> <p>Color.G (<code>uk.co.thefoundry.OfxImagePlaneColour.G</code>): G channel from input Source</p> <p>Color.B (<code>uk.co.thefoundry.OfxImagePlaneColour.B</code>): B channel from input Source</p> <p>Color.A (<code>uk.co.thefoundry.OfxImagePlaneColour.A</code>): A channel from input Source</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p>
All Planes / <code>processAllPlanes</code>	Boolean	Off	When checked all planes in input will be processed and output to the same plane as in input. It is useful for example to apply a Transform effect on all planes.
Clip Info... / <code>clipInfo</code>	Button		Display information about the inputs

2.10 Transform nodes

The following sections contain documentation about every node in the Transform group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.10.1 AdjustRoD node



This documentation is for version 1.1 of AdjustRoD.

Description

Enlarges the input image by a given amount of black and transparent pixels.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Add Pixels / <code>addPixels</code>	Double	w: 0 h: 0	How many pixels to add on each side for both dimensions (width/height)
Border Conditions / <code>boundary</code>	Choice	Nearest	<p>Specifies how pixel values are computed out of the image domain. This mostly affects values at the boundary of the image. If the image represents intensities, Nearest (Neumann) conditions should be used. If the image represents gradients or derivatives, Black (Dirichlet) boundary conditions should be used.</p> <p>Black (black): Dirichlet boundary condition: pixel values out of the image domain are zero.</p> <p>Nearest (nearest): Neumann boundary condition: pixel values out of the image domain are those of the closest pixel location in the image domain.</p>

2.10.2 Card3D node

This documentation is for version 1.0 of Card3D.

Description

Card3D.

This effect applies a transform that corresponds to projection the source image onto a 3D card in space. The 3D card is positionned with relative to the Axis position, and the Camera position may also be given. The Axis may be used to apply the same global motion to several cards.

This plugin concatenates transforms.

<http://opticalenquiry.com/nuke/index.php?title=Card3D>

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Import Format / <code>axisImportFormat</code>	Choice	chan	<p>The format of the file to import.</p> <p>chan: Chan format, each line is FRAME TX TY TZ RX RY RZ VFOV. Can be created using Natron, Nuke, 3D-Equalizer, Maya and other 3D tracking software. Be careful that the rotation order must be exactly the same when exporting and importing the chan file.</p> <p>Boujou (boujou): Boujou text export. In Boujou, after finishing the track and solving, go to Export > Export Camera Solve (Or press F12) > choose where to save the data and give it a name, click the drop down Export Type and make sure it will save as a .txt, then click Save. Each camera line is R(0,0) R(0,1) R(0,2) R(1,0) R(1,1) R(1,2) R(2,0) R(2,1) R(2,2) Tx Ty Tz F(mm).</p>
Import / <code>axisImportFile</code>	N/A		Import a chan file created using 3D tracking software, or a txt file created using Boujou.
Export / <code>axisExportChan</code>	N/A		Export a .chan file which can be used in Natron, Nuke or 3D tracking software, such as 3D-Equalizer, Maya, or Boujou. Be careful that the rotation order must be exactly the same when exporting and importing the chan file.
Transform Order / <code>axisXformOrder</code>	Choice	SRT	<p>Order in which scale (S), rotation (R) and translation (T) are applied.</p> <p>SRT (srt): Scale, Rotation, Translation.</p> <p>STR (str): Scale, Translation, Rotation.</p> <p>RST (rst): Rotation, Scale, Translation.</p> <p>RTS (rts): Rotation, Translation, Scale.</p> <p>TSR (tsr): Translation, Scale, Rotation.</p> <p>TRS (trs): Translation, Rotation, Scale.</p>
Rotation Order / <code>axisRotOrder</code>	Choice	ZXY	<p>Order in which Euler angles are applied in the rotation.</p> <p>XYZ (xyz): Rotation over X axis, then Y and Z.</p> <p>XZY (xzy): Rotation over X axis, then Z and Y.</p> <p>YXZ (yxz): Rotation over Y axis, then X and Z.</p> <p>YZX (yzx): Rotation over Y axis, then Z and X.</p> <p>ZXY (zxy): Rotation over Z axis, then X and Y.</p> <p>ZYX (zyx): Rotation over Z axis, then Y and X.</p>
Translate / <code>axisTranslate</code>	Double	x: 0 y: 0 z: 0	Translation component.
Rotate / <code>axisRotate</code>	Double	x: 0 y: 0 z: 0	Euler angles (in degrees).
Scale / <code>axisScaling</code>	Double	x: 1 y: 1 z: 1	Scale factor over each axis.
Uniform Scale / <code>axisUniformScale</code>	Double	1	Scale factor over all axis. It is multiplied by the scale factor over each axis.
Skew / <code>axisSkew</code>	Double	x: 0 y: 0 z: 0	Skew over each axis, in degrees.
Pivot / <code>axisPivot</code>	Double	x: 0 y: 0 z: 0	The position of the origin for position, scaling, skewing, and rotation.

Continued on next page

Table 158 – continued from previous page

Parameter / script name	Type	Default	Function
Specify Matrix / axisUseMatrix	Boolean	Off	Check to specify manually all the values for the position matrix.
/axisMatrix11	Double	1	Matrix coefficient.
/axisMatrix12	Double	0	Matrix coefficient.
/axisMatrix13	Double	0	Matrix coefficient.
/axisMatrix14	Double	0	Matrix coefficient.
/axisMatrix21	Double	0	Matrix coefficient.
/axisMatrix22	Double	1	Matrix coefficient.
/axisMatrix23	Double	0	Matrix coefficient.
/axisMatrix24	Double	0	Matrix coefficient.
/axisMatrix31	Double	0	Matrix coefficient.
/axisMatrix32	Double	0	Matrix coefficient.
/axisMatrix33	Double	1	Matrix coefficient.
/axisMatrix34	Double	-1	Matrix coefficient.
/axisMatrix41	Double	0	Matrix coefficient.
/axisMatrix42	Double	0	Matrix coefficient.
/axisMatrix43	Double	0	Matrix coefficient.
/axisMatrix44	Double	1	Matrix coefficient.
Enable Camera / camEnable	Boolean	Off	Enable the camera projection parameters.
Import Format / camImportFormat	Choice	chan	<p>The format of the file to import.</p> <p>chan: Chan format, each line is FRAME TX TY TZ RX RY RZ VFOV. Can be created using Natron, Nuke, 3D-Equalizer, Maya and other 3D tracking software. Be careful that the rotation order must be exactly the same when exporting and importing the chan file.</p> <p>Boujou (boujou): Boujou text export. In Boujou, after finishing the track and solving, go to Export > Export Camera Solve (Or press F12) > choose where to save the data and give it a name, click the drop down Export Type and make sure it will save as a .txt, then click Save. Each camera line is R(0,0) R(0,1) R(0,2) R(1,0) R(1,1) R(1,2) R(2,0) R(2,1) R(2,2) Tx Ty Tz F(mm).</p>
Import / camImportFile	N/A		Import a chan file created using 3D tracking software, or a txt file created using Boujou.
Export / camExportChan	N/A		Export a .chan file which can be used in Natron, Nuke or 3D tracking software, such as 3D-Equalizer, Maya, or Boujou. Be careful that the rotation order must be exactly the same when exporting and importing the chan file.
Transform Order / camXformOrder	Choice	SRT	<p>Order in which scale (S), rotation (R) and translation (T) are applied.</p> <p>SRT (srt): Scale, Rotation, Translation.</p> <p>STR (str): Scale, Translation, Rotation.</p> <p>RST (rst): Rotation, Scale, Translation.</p> <p>RTS (rts): Rotation, Translation, Scale.</p> <p>TSR (tsr): Translation, Scale, Rotation.</p> <p>TRS (trs): Translation, Rotation, Scale.</p>

Continued on next page

Table 158 – continued from previous page

Parameter / script name	Type	Default	Function
Rotation Order / camRotOrder	Choice	ZXY	Order in which Euler angles are applied in the rotation. XYZ (xyz) : Rotation over X axis, then Y and Z. XZY (xzy) : Rotation over X axis, then Z and Y. YXZ (yxz) : Rotation over Y axis, then X and Z. YZX (yzx) : Rotation over Y axis, then Z and X. ZXY (zxy) : Rotation over Z axis, then X and Y. ZYX (zyx) : Rotation over Z axis, then Y and X.
Translate / camTranslate	Double	x: 0 y: 0 z: 0	Translation component.
Rotate / camRotate	Double	x: 0 y: 0 z: 0	Euler angles (in degrees).
Scale / camScaling	Double	x: 1 y: 1 z: 1	Scale factor over each axis.
Uniform Scale / camUniformScale	Double	1	Scale factor over all axis. It is multiplied by the scale factor over each axis.
Skew / camSkew	Double	x: 0 y: 0 z: 0	Skew over each axis, in degrees.
Pivot / camPivot	Double	x: 0 y: 0 z: 0	The position of the origin for position, scaling, skewing, and rotation.
Specify Matrix / camUseMatrix	Boolean	Off	Check to specify manually all the values for the position matrix.
/ camMatrix11	Double	1	Matrix coefficient.
/ camMatrix12	Double	0	Matrix coefficient.
/ camMatrix13	Double	0	Matrix coefficient.
/ camMatrix14	Double	0	Matrix coefficient.
/ camMatrix21	Double	0	Matrix coefficient.
/ camMatrix22	Double	1	Matrix coefficient.
/ camMatrix23	Double	0	Matrix coefficient.
/ camMatrix24	Double	0	Matrix coefficient.
/ camMatrix31	Double	0	Matrix coefficient.
/ camMatrix32	Double	0	Matrix coefficient.
/ camMatrix33	Double	1	Matrix coefficient.
/ camMatrix34	Double	-1	Matrix coefficient.
/ camMatrix41	Double	0	Matrix coefficient.
/ camMatrix42	Double	0	Matrix coefficient.
/ camMatrix43	Double	0	Matrix coefficient.
/ camMatrix44	Double	1	Matrix coefficient.
Projection / camprojection_mode	Choice	Perspective	Perspective (perspective) : Perspective projection. Orthographic (orthographic) : Orthographic projection
Focal Length / camfocal	Double	50	The camera focal length, in arbitrary units (usually either millimeters or 35 mm equivalent focal length). haperture and vaperture must be expressed in the same units.
Horiz. Aperture / camhaperture	Double	24.576	The camera horizontal aperture (or film back width), in the same units as the focal length. In the case of scanned film, this can be obtained as image_width * scanner_pitch.

Continued on next page

Table 158 – continued from previous page

Parameter / script name	Type	Default	Function
Vert. Aperture / <code>camvaperature</code>	Double	18.672	The camera vertical aperture (or film back height), in the same units as the focal length. This does not affect the projection (which is computed from haperture and the image aspect ratio), but it is used to compute the focal length from vertical FOV when importing chan files, using the formula: $\text{focal} = 0.5 * \text{vaperature} / \tan(\text{vfov}/2)$. It is thus best set as: $\text{haperture} = \text{vaperature} * \text{image_width}/\text{image_height}$. In the case of scanned film, this can be obtained as $\text{image_height} * \text{scanner_pitch}$.
Window Translate / <code>camwin_translate</code>	Double	x: 0 y: 0	The camera window (or film back) is translated by this fraction of the horizontal aperture, without changing the position of the camera center. This can be used to model tilt-shift or perspective-control lens.
Window Scale / <code>camwin_scale</code>	Double	x: 1 y: 1	Scale the camera window (or film back).
Window Roll / <code>camwinroll</code>	Double	0	Rotation (in degrees) of the camera window (or film back) around the z axis.
Import Format / <code>cardImportFormat</code>	Choice	chan	<p>The format of the file to import.</p> <p>chan: Chan format, each line is FRAME TX TY TZ RX RY RZ VFOV. Can be created using Natron, Nuke, 3D-Equalizer, Maya and other 3D tracking software. Be careful that the rotation order must be exactly the same when exporting and importing the chan file.</p> <p>Boujou (boujou): Boujou text export. In Boujou, after finishing the track and solving, go to Export > Export Camera Solve (Or press F12) > choose where to save the data and give it a name, click the drop down Export Type and make sure it will save as a .txt, then click Save. Each camera line is R(0,0) R(0,1) R(0,2) R(1,0) R(1,1) R(1,2) R(2,0) R(2,1) R(2,2) Tx Ty Tz F(mm).</p>
Import / <code>cardImportFile</code>	N/A		Import a chan file created using 3D tracking software, or a txt file created using Boujou.
Export / <code>cardExportChan</code>	N/A		Export a .chan file which can be used in Natron, Nuke or 3D tracking software, such as 3D-Equalizer, Maya, or Boujou. Be careful that the rotation order must be exactly the same when exporting and importing the chan file.
Transform Order / <code>cardXformOrder</code>	Choice	SRT	<p>Order in which scale (S), rotation (R) and translation (T) are applied.</p> <p>SRT (srt): Scale, Rotation, Translation.</p> <p>STR (str): Scale, Translation, Rotation.</p> <p>RST (rst): Rotation, Scale, Translation.</p> <p>RTS (rts): Rotation, Translation, Scale.</p> <p>TSR (tsr): Translation, Scale, Rotation.</p> <p>TRS (trs): Translation, Rotation, Scale.</p>
Rotation Order / <code>cardRotOrder</code>	Choice	ZXY	<p>Order in which Euler angles are applied in the rotation.</p> <p>XYZ (xyz): Rotation over X axis, then Y and Z.</p> <p>XZY (xzy): Rotation over X axis, then Z and Y.</p> <p>YXZ (yxz): Rotation over Y axis, then X and Z.</p> <p>YZX (yzx): Rotation over Y axis, then Z and X.</p> <p>ZXY (zxy): Rotation over Z axis, then X and Y.</p> <p>ZYX (zyx): Rotation over Z axis, then Y and X.</p>

Continued on next page

Table 158 – continued from previous page

Parameter / script name	Type	Default	Function
Translate / cardTranslate	Double	x: 0 y: 0 z: -1	Translation component.
Rotate / cardRotate	Double	x: 0 y: 0 z: 0	Euler angles (in degrees).
Scale / cardScaling	Double	x: 1 y: 1 z: 1	Scale factor over each axis.
Uniform Scale / cardUniformScale	Double	1	Scale factor over all axis. It is multiplied by the scale factor over each axis.
Skew / cardSkew	Double	x: 0 y: 0 z: 0	Skew over each axis, in degrees.
Pivot / cardPivot	Double	x: 0 y: 0 z: 0	The position of the origin for position, scaling, skewing, and rotation.
Specify Matrix / cardUseMatrix	Boolean	Off	Check to specify manually all the values for the position matrix.
/ cardMatrix11	Double	1	Matrix coefficient.
/ cardMatrix12	Double	0	Matrix coefficient.
/ cardMatrix13	Double	0	Matrix coefficient.
/ cardMatrix14	Double	0	Matrix coefficient.
/ cardMatrix21	Double	0	Matrix coefficient.
/ cardMatrix22	Double	1	Matrix coefficient.
/ cardMatrix23	Double	0	Matrix coefficient.
/ cardMatrix24	Double	0	Matrix coefficient.
/ cardMatrix31	Double	0	Matrix coefficient.
/ cardMatrix32	Double	0	Matrix coefficient.
/ cardMatrix33	Double	1	Matrix coefficient.
/ cardMatrix34	Double	-1	Matrix coefficient.
/ cardMatrix41	Double	0	Matrix coefficient.
/ cardMatrix42	Double	0	Matrix coefficient.
/ cardMatrix43	Double	0	Matrix coefficient.
/ cardMatrix44	Double	1	Matrix coefficient.
Lens-In Focal / lensInFocal	Double	1	The focal length of the camera that took the picture on the card. The card is scaled so that at distance 1 (which is the default card Z) it occupies the field of view corresponding to lensInFocal and lensInHAperture.
Lens-In H.Aperture / lensInHAperture	Double	1	The horizontal aperture (or sensor/film back width) of the camera that took the picture on the card. The card is scaled so that at distance 1 (which is the default card Z) it occupies the field of view corresponding to lensInFocal and lensInHAperture.
Output Format / format	Choice	Project	Desired format for the output sequence. Format (format): Use a pre-defined image format. Size (size): Use a specific extent (size and offset). Project (project): Use the project extent (size and offset).
Center / recenter	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.

Continued on next page

Table 158 – continued from previous page

Parameter / script name	Type	Default	Function
Format / NatronParamFormatChoice	Choice	HD 1920x1080	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Invert / invert	Boolean	Off	Invert the transform.
Filter / filter	Choice	Cubic	Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+). Impulse (impulse): (nearest neighbor / box) Use original values. Box (box): Integrate the source image over the bounding box of the back-transformed pixel. Bilinear (bilinear): (tent / triangle) Bilinear interpolation between original values. Cubic (cubic): (cubic spline) Some smoothing. Keys (keys): (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*). Simon (simon): Some smoothing, plus medium sharpening (*). Rifman (rifman): Some smoothing, plus significant sharpening (*). Mitchell (mitchell): Some smoothing, plus blurring to hide pixelation (*+). Parzen (parzen): (cubic B-spline) Greatest smoothing of all filters (+). Notch (notch): Flat smoothing (which tends to hide moire' patterns) (+).
Clamp / clamp	Boolean	Off	Clamp filter output within the original range - useful to avoid negative values in mattes

Continued on next page

Table 158 – continued from previous page

Parameter / script name	Type	Default	Function
Black outside / <code>black_outside</code>	Boolean	On	Fill the area outside the source image with black
Motion Blur / <code>motionBlur</code>	Double	0	Quality of motion blur rendering. 0 disables motion blur, 1 is a good value. Increasing this slows down rendering.
Directional Blur Mode / <code>directionalBlur</code>	Boolean	Off	Motion blur is computed from the original image to the transformed image, each parameter being interpolated linearly. The <code>motionBlur</code> parameter must be set to a nonzero value, and the <code>blackOutside</code> parameter may have an important effect on the result.
Shutter / <code>shutter</code>	Double	0.5	Controls how long (in frames) the shutter should remain open.
Shutter Offset / <code>shutterOffset</code>	Choice	Start	Controls when the shutter should be open/closed. Ignored if there is no motion blur (i.e. <code>shutter=0</code> or <code>motionBlur=0</code>). Centered (centered): Centers the shutter around the frame (from $t - \text{shutter}/2$ to $t + \text{shutter}/2$) Start (start): Open the shutter at the frame (from t to $t + \text{shutter}$) End (end): Close the shutter at the frame (from $t - \text{shutter}$ to t) Custom (custom): Open the shutter at $t + \text{shuttercustomoffset}$ (from $t + \text{shuttercustomoffset}$ to $t + \text{shuttercustomoffset} + \text{shutter}$)
Custom Offset / <code>shutterCustomOffset</code>	Double	0	When custom is selected, the shutter is open at current time plus this offset (in frames). Ignored if there is no motion blur (i.e. <code>shutter=0</code> or <code>motionBlur=0</code>).
Interactive Update / <code>interactive</code>	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.

2.10.3 CornerPin node



This documentation is for version 1.0 of CornerPin.

Description

Allows an image to fit another in translation, rotation and scale.

The resulting transform is a translation if 1 point is enabled, a similarity if 2 are enabled, an affine transform if 3 are enabled, and a homography if they are all enabled.

An effect where an image transitions from a full-frame image to an image placed on a billboard or a screen, or a crash zoom effect, can be obtained by combining the Transform and CornerPin effects and using the Amount parameter on both effects.

Apply a CornerPin followed by a Transform effect (the order is important) and visualize the output superimposed on the target image. While leaving the value of the Amount parameter at 1, tune the Transform parameters (including Scale and Skew) so that the transformed image is as close as possible to the desired target location.

Then, adjust the ‘to’ points of the CornerPin effect (which should be affected by the Transform) so that the warped image perfectly matches the desired target location. Link the Amount parameter of the Transform and CornerPin effects.

Finally, by animating the Amount parameter of both effects from 0 to 1, the image goes progressively, and with minimal deformations, from full-frame to the target location, creating the desired effect (motion blur can be added on the Transform node, too).

Note that if only the CornerPin effect is used instead of combining CornerPin and Transform, the position of the CornerPin points is linearly interpolated between their ‘from’ position and their ‘to’ position, which may result in unrealistic image motion, where the image shrinks and expands, especially when the image rotates.

This plugin concatenates transforms.

See also: <http://opticalenquiry.com/nuke/index.php?title=CornerPin>

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
to1 / to1	Double	x: 0 y: 0	
enable1 / enable1	Boolean	On	Enables the point on the left.
to2 / to2	Double	x: 1 y: 0	
enable2 / enable2	Boolean	On	Enables the point on the left.
to3 / to3	Double	x: 1 y: 1	
enable3 / enable3	Boolean	On	Enables the point on the left.
to4 / to4	Double	x: 0 y: 1	
enable4 / enable4	Boolean	On	Enables the point on the left.
Copy “From” / copyFrom	Button		Copy the contents (including animation) of the “from” points to the “to” points.
Copy “From” (Single) / copyFromSingle	Button		Copy the current values of the “from” points to the “to” points.
from1 / from1	Double	x: 0 y: 0	
from2 / from2	Double	x: 1 y: 0	
from3 / from3	Double	x: 1 y: 1	
from4 / from4	Double	x: 0 y: 1	
Set to input rod / setToInputRod	Button		Copy the values from the source region of definition into the “from” points.
Copy “To” / copyTo	Button		Copy the contents (including animation) of the “to” points to the “from” points.
Copy “To” (Single) / copyToSingle	Button		Copy the current values of the “to” points to the “from” points.
Amount / transformAmount	Double	1	Amount of transform to apply (excluding the extra matrix, which is always applied). 0 means the transform is identity, 1 means to apply the full transform. Intermediate transforms are computed by linear interpolation between the ‘from’ and the ‘to’ points. See the plugin description on how to use the amount parameter for a crash zoom effect.

Continued on next page

Table 159 – continued from previous page

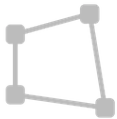
Parameter / script name	Type	Default	Function
Extra Matrix / transform	Double	x: 1 y: 0 z: 0 x: 0 y: 1 z: 0 x: 0 y: 0 z: 1	
Overlay Points / overlayPoints	Choice	To	Whether to display the “from” or the “to” points in the overlay To (to) : Display the “to” points. From (from) : Display the “from” points.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Invert / invert	Boolean	Off	Invert the transform.
Filter / filter	Choice	Cubic	Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+). Impulse (impulse) : (nearest neighbor / box) Use original values. Box (box) : Integrate the source image over the bounding box of the back-transformed pixel. Bilinear (bilinear) : (tent / triangle) Bilinear interpolation between original values. Cubic (cubic) : (cubic spline) Some smoothing. Keys (keys) : (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*). Simon (simon) : Some smoothing, plus medium sharpening (*). Rifman (rifman) : Some smoothing, plus significant sharpening (*). Mitchell (mitchell) : Some smoothing, plus blurring to hide pixelation (*+). Parzen (parzen) : (cubic B-spline) Greatest smoothing of all filters (+). Notch (notch) : Flat smoothing (which tends to hide moire’ patterns) (+).
Clamp / clamp	Boolean	Off	Clamp filter output within the original range - useful to avoid negative values in mattes
Black outside / black_outside	Boolean	On	Fill the area outside the source image with black
Motion Blur / motionBlur	Double	0	Quality of motion blur rendering. 0 disables motion blur, 1 is a good value. Increasing this slows down rendering.
Directional Blur Mode / directionalBlur	Boolean	Off	Motion blur is computed from the original image to the transformed image, each parameter being interpolated linearly. The motionBlur parameter must be set to a nonzero value, and the blackOutside parameter may have an important effect on the result.
Shutter / shutter	Double	0.5	Controls how long (in frames) the shutter should remain open.

Continued on next page

Table 159 – continued from previous page

Parameter / script name	Type	Default	Function
Shutter Offset / shutterOffset	Choice	Start	Controls when the shutter should be open/closed. Ignored if there is no motion blur (i.e. shutter=0 or motionBlur=0). Centered (centered) : Centers the shutter around the frame (from t-shutter/2 to t+shutter/2) Start (start) : Open the shutter at the frame (from t to t+shutter) End (end) : Close the shutter at the frame (from t-shutter to t) Custom (custom) : Open the shutter at t+shuttercustomoffset (from t+shuttercustomoffset to t+shuttercustomoffset+shutter)
Custom Offset / shutterCustomOffset	Double	0	When custom is selected, the shutter is open at current time plus this offset (in frames). Ignored if there is no motion blur (i.e. shutter=0 or motionBlur=0).

2.10.4 CornerPinMasked node



This documentation is for version 1.0 of CornerPinMasked.

Description

Allows an image to fit another in translation, rotation and scale.

The resulting transform is a translation if 1 point is enabled, a similarity if 2 are enabled, an affine transform if 3 are enabled, and a homography if they are all enabled.

An effect where an image transitions from a full-frame image to an image placed on a billboard or a screen, or a crash zoom effect, can be obtained by combining the Transform and CornerPin effects and using the Amount parameter on both effects.

Apply a CornerPin followed by a Transform effect (the order is important) and visualize the output superimposed on the target image. While leaving the value of the Amount parameter at 1, tune the Transform parameters (including Scale and Skew) so that the transformed image is as close as possible to the desired target location.

Then, adjust the ‘to’ points of the CornerPin effect (which should be affected by the Transform) so that the warped image perfectly matches the desired target location. Link the Amount parameter of the Transform and CornerPin effects.

Finally, by animating the Amount parameter of both effects from 0 to 1, the image goes progressively, and with minimal deformations, from full-frame to the target location, creating the desired effect (motion blur can be added on the Transform node, too).

Note that if only the CornerPin effect is used instead of combining CornerPin and Transform, the position of the CornerPin points is linearly interpolated between their ‘from’ position and their ‘to’ position, which may result in unrealistic image motion, where the image shrinks and expands, especially when the image rotates.

This plugin concatenates transforms.

See also: <http://opticalenquiry.com/nuke/index.php?title=CornerPin>

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
to1 / to1	Double	x: 0 y: 0	
enable1 / enable1	Boolean	On	Enables the point on the left.
to2 / to2	Double	x: 1 y: 0	
enable2 / enable2	Boolean	On	Enables the point on the left.
to3 / to3	Double	x: 1 y: 1	
enable3 / enable3	Boolean	On	Enables the point on the left.
to4 / to4	Double	x: 0 y: 1	
enable4 / enable4	Boolean	On	Enables the point on the left.
Copy “From” / copyFrom	Button		Copy the contents (including animation) of the “from” points to the “to” points.
Copy “From” (Single) / copyFromSingle	Button		Copy the current values of the “from” points to the “to” points.
from1 / from1	Double	x: 0 y: 0	
from2 / from2	Double	x: 1 y: 0	
from3 / from3	Double	x: 1 y: 1	
from4 / from4	Double	x: 0 y: 1	
Set to input rod / setToInputRod	Button		Copy the values from the source region of definition into the “from” points.
Copy “To” / copyTo	Button		Copy the contents (including animation) of the “to” points to the “from” points.
Copy “To” (Single) / copyToSingle	Button		Copy the current values of the “to” points to the “from” points.
Amount / transformAmount	Double	1	Amount of transform to apply (excluding the extra matrix, which is always applied). 0 means the transform is identity, 1 means to apply the full transform. Intermediate transforms are computed by linear interpolation between the ‘from’ and the ‘to’ points. See the plugin description on how to use the amount parameter for a crash zoom effect.
Extra Matrix / transform	Double	x: 1 y: 0 z: 0 x: 0 y: 1 z: 0 x: 0 y: 0 z: 1	

Continued on next page

Table 160 – continued from previous page

Parameter / script name	Type	Default	Function
Overlay Points / overlayPoints	Choice	To	Whether to display the “from” or the “to” points in the overlay To (to) : Display the “to” points. From (from) : Display the “from” points.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Invert / invert	Boolean	Off	Invert the transform.
Filter / filter	Choice	Cubic	Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+). Impulse (impulse) : (nearest neighbor / box) Use original values. Box (box) : Integrate the source image over the bounding box of the back-transformed pixel. Bilinear (bilinear) : (tent / triangle) Bilinear interpolation between original values. Cubic (cubic) : (cubic spline) Some smoothing. Keys (keys) : (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*). Simon (simon) : Some smoothing, plus medium sharpening (*). Rifman (rifman) : Some smoothing, plus significant sharpening (*). Mitchell (mitchell) : Some smoothing, plus blurring to hide pixelation (*+). Parzen (parzen) : (cubic B-spline) Greatest smoothing of all filters (+). Notch (notch) : Flat smoothing (which tends to hide moire’ patterns) (+).
Clamp / clamp	Boolean	Off	Clamp filter output within the original range - useful to avoid negative values in mattes
Black outside / black_outside	Boolean	On	Fill the area outside the source image with black
Motion Blur / motionBlur	Double	0	Quality of motion blur rendering. 0 disables motion blur, 1 is a good value. Increasing this slows down rendering.
Directional Blur Mode / directionalBlur	Boolean	Off	Motion blur is computed from the original image to the transformed image, each parameter being interpolated linearly. The motionBlur parameter must be set to a nonzero value, and the blackOutside parameter may have an important effect on the result.
Shutter / shutter	Double	0.5	Controls how long (in frames) the shutter should remain open.
Shutter Offset / shutterOffset	Choice	Start	Controls when the shutter should be open/closed. Ignored if there is no motion blur (i.e. shutter=0 or motionBlur=0). Centered (centered) : Centers the shutter around the frame (from t-shutter/2 to t+shutter/2) Start (start) : Open the shutter at the frame (from t to t+shutter) End (end) : Close the shutter at the frame (from t-shutter to t) Custom (custom) : Open the shutter at t+shuttercustomoffset (from t+shuttercustomoffset to t+shuttercustomoffset+shutter)
Custom Offset / shutterCustomOffset	Double	0	When custom is selected, the shutter is open at current time plus this offset (in frames). Ignored if there is no motion blur (i.e. shutter=0 or motionBlur=0).

Continued on next page

Table 160 – continued from previous page

Parameter / script name	Type	Default	Function
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.10.5 Crop node



This documentation is for version 1.0 of Crop.

Description

Removes everything outside the defined rectangle and optionally adds black edges so everything outside is black.

If the 'Extent' parameter is set to 'Format', and 'Reformat' is checked, the output pixel aspect ratio is also set to this of the format.

This plugin does not concatenate transforms.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Extent / extent	Choice	Size	Extent (size and offset) of the output. Format (format): Use a pre-defined image format. Size (size): Use a specific extent (size and offset). Project (project): Use the project extent (size and offset). Default (default): Use the default extent (e.g. the source clip extent, if connected).
Center / recenter	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.

Continued on next page

Table 161 – continued from previous page

Parameter / script name	Type	Default	Function
Format / NatronParamFormatChoice	Choice	HD 1920x1080	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Frame Range / frameRange	Integer	min: 1 max: 1	Time domain.
Softness / softness	Double	0	Size of the fade to black around edges to apply.
Reformat / reformat	Boolean	Off	Translates the bottom left corner of the crop rectangle to be in (0,0). This sets the output format only if 'Format' or 'Project' is selected as the output Extend. In order to actually change the format of this image stream for other Extent choices, feed the output of this node to a either a NoOp node which sets the proper format, or a Reformat node with the same extent and with 'Resize Type' set to None and 'Center' unchecked. The reason is that the Crop size may be animated, but the output format can not be animated.
Intersect / intersect	Boolean	Off	Intersects the crop rectangle with the input region of definition instead of extending it.
Black Outside / blackOutside	Boolean	Off	Add 1 black and transparent pixel to the region of definition so that all the area outside the crop rectangle is black.

2.10.6 IDistort node



This documentation is for version 2.0 of IDistort.

Description

Distort an image, based on a displacement map.

The U and V channels give the offset in pixels in the destination image to the pixel where the color is taken. For example, if at pixel (45,12) the UV value is (-1.5,3.2), then the color at this pixel is taken from (43.5,15.2) in the source image. This plugin concatenates transforms upstream, so that if the nodes upstream output a 3x3 transform (e.g. Transform, CornerPin, Dot, NoOp, Switch), the original image is sampled only once.

This plugin concatenates transforms upstream.

Inputs

Input	Description	Optional
Source		No
UV		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
U Channel / channelU	Choice	Color.R	<p>Input U channel from UV.</p> <p>Color.R (uk.co.thefoundry.OfxImagePlaneColour.R): R channel from input UV</p> <p>Color.G (uk.co.thefoundry.OfxImagePlaneColour.G): G channel from input UV</p> <p>Color.B (uk.co.thefoundry.OfxImagePlaneColour.B): B channel from input UV</p> <p>Color.A (uk.co.thefoundry.OfxImagePlaneColour.A): A channel from input UV</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p>
V Channel / channelV	Choice	Color.G	<p>Input V channel from UV.</p> <p>Color.R (uk.co.thefoundry.OfxImagePlaneColour.R): R channel from input UV</p> <p>Color.G (uk.co.thefoundry.OfxImagePlaneColour.G): G channel from input UV</p> <p>Color.B (uk.co.thefoundry.OfxImagePlaneColour.B): B channel from input UV</p> <p>Color.A (uk.co.thefoundry.OfxImagePlaneColour.A): A channel from input UV</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p>

Continued on next page

Table 162 – continued from previous page

Parameter / script name	Type	Default	Function
Alpha Channel / <code>channelA</code>	Choice	Color.A	<p>Input Alpha channel from UV. The Output alpha is set to this value. If “Unpremult UV” is checked, the UV values are divided by alpha.</p> <p>Color.R (<code>uk.co.thefoundry.OfxImagePlaneColour.R</code>): R channel from input UV</p> <p>Color.G (<code>uk.co.thefoundry.OfxImagePlaneColour.G</code>): G channel from input UV</p> <p>Color.B (<code>uk.co.thefoundry.OfxImagePlaneColour.B</code>): B channel from input UV</p> <p>Color.A (<code>uk.co.thefoundry.OfxImagePlaneColour.A</code>): A channel from input UV</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p>
Unpremult UV / <code>unpremultUV</code>	Boolean	Off	Unpremult UV by Alpha from UV. Check if UV values look small for small values of Alpha (3D software sometimes write premultiplied UV values).
UV Offset / <code>uvOffset</code>	Double	U: 0 V: 0	Offset to apply to the U and V channel (useful if these were stored in a file that cannot handle negative numbers)
UV Scale / <code>uvScale</code>	Double	U: 1 V: 1	Scale factor to apply to the U and V channel (useful if these were stored in a file that can only store integer values)
Filter / <code>filter</code>	Choice	Cubic	<p>Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+).</p> <p>Impulse (impulse): (nearest neighbor / box) Use original values.</p> <p>Box (box): Integrate the source image over the bounding box of the back-transformed pixel.</p> <p>Bilinear (bilinear): (tent / triangle) Bilinear interpolation between original values.</p> <p>Cubic (cubic): (cubic spline) Some smoothing.</p> <p>Keys (keys): (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*).</p> <p>Simon (simon): Some smoothing, plus medium sharpening (*).</p> <p>Rifman (rifman): Some smoothing, plus significant sharpening (*).</p> <p>Mitchell (mitchell): Some smoothing, plus blurring to hide pixelation (*+).</p> <p>Parzen (parzen): (cubic B-spline) Greatest smoothing of all filters (+).</p> <p>Notch (notch): Flat smoothing (which tends to hide moire’ patterns) (+).</p>
Clamp / <code>clamp</code>	Boolean	Off	Clamp filter output within the original range - useful to avoid negative values in mattes
Black outside / <code>black_outside</code>	Boolean	Off	Fill the area outside the source image with black
(Un)premult / <code>premult</code>	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / <code>maskInvert</code>	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / <code>mix</code>	Double	1	Mix factor between the original and the transformed image.

2.10.7 LensDistortion node



This documentation is for version 4.0 of LensDistortion.

Description

Add or remove lens distortion, or produce an STMap that can be used to apply that transform.

The region of definition of the transformed image is computed from the region of definition of the Source input. If the input is defined outside of the project format, this may result in a very large region. A Crop effect may be inserted before LensDistortion to avoid this. If the input region of definition is inside the format, the Crop To Format parameter may be used to avoid expanding it.

LensDistortion can directly apply distortion/undistortion, but if the distortion parameters are not animated, the most efficient way to use LensDistortion and avoid repeated distortion function calculations is the following:

- If the footage size is not the same as the project size, insert a FrameHold plugin between the footage to distort or undistort and the Source input of LensDistortion. This connection is only used to get the size of the input footage.
- Set Output Mode to “STMap” in LensDistortion.
- feed the LensDistortion output into the UV input of STMap, and feed the footage into the Source input of STMap.

This plugin concatenates transforms upstream.

Inputs

Input	Description	Optional
Source		Yes
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Format / extent	Choice	Default	Reference format for lens distortion. Format (format): Use a pre-defined image format. Size (size): Use a specific extent (size and offset). Project (project): Use the project extent (size and offset). Default (default): Use the default extent (e.g. the source clip extent, if connected).
Center / recenter	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.

Continued on next page

Table 163 – continued from previous page

Parameter / script name	Type	Default	Function
Format / NatronParamFormatChoice	Choice	HD 1920x1080	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.

Continued on next page

Table 163 – continued from previous page

Parameter / script name	Type	Default	Function
Model / model	Choice	Nuke	<p>Choice of the distortion model, i.e. the function that goes from distorted to undistorted image coordinates.</p> <p>Nuke (nuke): The model used in Nuke’s LensDistortion plugin.</p> <p>PFBarel (pfbarel): The PFBarel model used in PFTrack by PixelFarm.</p> <p>3DE Classic (3declassic): Degree-2 anamorphic and degree-4 radial mixed model, used in 3DEqualizer by Science-D-Visions. Works, but it is recommended to use 3DE4 Radial Standard Degree 4 or 3DE4 Anamorphic Standard Degree 4 instead.</p> <p>3DE4 Anamorphic Degree 6 (3deanamorphic6): Degree-6 anamorphic model, used in 3DEqualizer by Science-D-Visions.</p> <p>3DE4 Radial Fisheye Degree 8 (3defisheye8): Radial lens distortion model with equisolid-angle fisheye projection, used in 3DEqualizer by Science-D-Visions.</p> <p>3DE4 Radial Standard Degree 4 (3deradial4): Radial lens distortion model, a.k.a. radial decentered cylindric degree 4, which compensates for decentered lenses (and beam splitter artefacts in stereo rigs), used in 3DEqualizer by Science-D-Visions.</p> <p>3DE4 Anamorphic Standard Degree 4 (3deanamorphic4): Degree-4 anamorphic model with anamorphic lens rotation, which handles ‘human-touched’ mounted anamorphic lenses, used in 3DEqualizer by Science-D-Visions.</p> <p>PanoTools (panotools): The model used in PanoTools, PTGui, PTAssembler, Hugin. See http://wiki.panotools.org/Lens_correction_model</p>
Direction / direction	Choice	Distort	<p>Should the output correspond to applying or to removing distortion.</p> <p>Distort: The output corresponds to applying distortion.</p> <p>Undistort: The output corresponds to removing distortion.</p>
Output Mode / outputMode	Choice	Image	<p>Choice of the output, which may be either a distorted/undistorted image, or a distortion/undistortion STMap.</p> <p>Image: The output is the distorted/undistorted Source.</p> <p>STMap: The output is a distortion/undistortion STMap. It is recommended to insert a FrameHold node at the Source input so that the STMap is computed only once if the parameters are not animated.</p>
K1 / k1	Double	0	Nuke: First radial distortion coefficient (coefficient for r^2).
K2 / k2	Double	0	Nuke: Second radial distortion coefficient (coefficient for r^4).
Center / center	Double	x: 0 y: 0	Nuke: Offset of the distortion center from the image center.
Squeeze / anamorphicSqueeze	Double	1	Nuke: Anamorphic squeeze (only for anamorphic lens).
Asymmetric / asymmetricDistortion	Double	x: 0 y: 0	Nuke: Asymmetric distortion (only for anamorphic lens).
File / pfFile	N/A		The location of the PFBarel .pfb file to use. Keyframes are set if present in the file.
C3 / pfC3	Double	0	PFBarel: Low order radial distortion coefficient.
C5 / pfC5	Double	0	PFBarel: Low order radial distortion coefficient.

Continued on next page

Table 163 – continued from previous page

Parameter / script name	Type	Default	Function
Center / pfP	Double	x: 0.5 y: 0.5	PFBarrel: The distortion center of the lens (specified as a factor rather than a pixel value)
Squeeze / pfSqueeze	Double	1	PFBarrel: Anamorphic squeeze (only for anamorphic lens).
fov left [unit coord] / tde4_field_of_view_xa_unit	Double	0	3DE4: Field of view.
fov bottom [unit coord] / tde4_field_of_view_ya_unit	Double	0	3DE4: Field of view.
fov right [unit coord] / tde4_field_of_view_xb_unit	Double	1	3DE4: Field of view.
fov top [unit coord] / tde4_field_of_view_yb_unit	Double	1	3DE4: Field of view.
tde4 focal length [cm] / tde4_focal_length_cm	Double	1	3DE4: Focal length.
tde4 focus distance [cm] / tde4_custom_focus_distance_cm	Double	100	3DE4: Focus distance.
tde4 filmback width [cm] / tde4_filmback_width_cm	Double	0.8	3DE4: Filmback width.
tde4 filmback height [cm] / tde4_filmback_height_cm	Double	0.6	3DE4: Filmback height.
tde4 lens center offset x [cm] / tde4_lens_center_offset_x_cm	Double	0	3DE4: Lens center horizontal offset.
tde4 lens center offset y [cm] / tde4_lens_center_offset_y_cm	Double	0	3DE4: Lens center vertical offset.
tde4 pixel aspect / tde4_pixel_aspect	Double	1	3DE4: Pixel aspect ratio.
Distortion / tde4_Distortion	Double	0	3DE Classic: Distortion.
Anamorphic Squeeze / tde4_Anamorphic_Squeeze	Double	1	3DE Classic: Anamorphic Squeeze.
Curvature X / tde4_Curvature_X	Double	0	3DE Classic: Curvature X.
Curvature Y / tde4_Curvature_Y	Double	0	3DE Classic: Curvature Y.
Quartic Distortion / tde4_Quartic_Distortion	Double	0	3DE Classic: Quartic Distortion.
Distortion - Degree 2 / tde4_Distortion_Degree_2	Double	0	3DE Standard and Fisheye: Distortion.
U - Degree 2 / tde4_U_Degree_2	Double	0	3DE Standard: U - Degree 2.
V - Degree 2 / tde4_V_Degree_2	Double	0	3DE Standard: V - Degree 2.
Quartic Distortion - Degree 4 / tde4_Quartic_Distortion_Degree_4	Double	0	3DE Standard and Fisheye: Quartic Distortion - Degree 4.

Continued on next page

Table 163 – continued from previous page

Parameter / script name	Type	Default	Function
U - Degree 4 / tde4_U_Degree_4	Double	0	3DE Standard: U - Degree 4.
V - Degree 4 / tde4_V_Degree_4	Double	0	3DE Standard: V - Degree 4.
Phi - Cylindric Direction / tde4_Phi_Cylindric_Direction	Double	0	3DE Standard: Phi - Cylindric Direction.
B - Cylindric Bending / tde4_B_Cylindric_Bending	Double	0	3DE Standard: B - Cylindric Bending.
Cx02 - Degree 2 / tde4_Cx02_Degree_2	Double	0	3DE Anamorphic 4 and 6: Cx02 - Degree 2.
Cy02 - Degree 2 / tde4_Cy02_Degree_2	Double	0	3DE Anamorphic 4 and 6: Cy02 - Degree 2.
Cx22 - Degree 2 / tde4_Cx22_Degree_2	Double	0	3DE Anamorphic 4 and 6: Cx22 - Degree 2.
Cy22 - Degree 2 / tde4_Cy22_Degree_2	Double	0	3DE Anamorphic 4 and 6: Cy22 - Degree 2.
Cx04 - Degree 4 / tde4_Cx04_Degree_4	Double	0	3DE Anamorphic 4 and 6: Cx04 - Degree 4.
Cy04 - Degree 4 / tde4_Cy04_Degree_4	Double	0	3DE Anamorphic 4 and 6: Cy04 - Degree 4.
Cx24 - Degree 4 / tde4_Cx24_Degree_4	Double	0	3DE Anamorphic 4 and 6: Cx24 - Degree 4.
Cy24 - Degree 4 / tde4_Cy24_Degree_4	Double	0	3DE Anamorphic 4 and 6: Cy24 - Degree 4.
Cx44 - Degree 4 / tde4_Cx44_Degree_4	Double	0	3DE Anamorphic 4 and 6: Cx44 - Degree 4.
Cy44 - Degree 4 / tde4_Cy44_Degree_4	Double	0	3DE Anamorphic 4 and 6: Cy44 - Degree 4.
Cx06 - Degree 6 / tde4_Cx06_Degree_6	Double	0	3DE Anamorphic 6: Cx06 - Degree 6.
Cy06 - Degree 6 / tde4_Cy06_Degree_6	Double	0	3DE Anamorphic 6: Cy06 - Degree 6.
Cx26 - Degree 6 / tde4_Cx26_Degree_6	Double	0	3DE Anamorphic 6: Cx26 - Degree 6.
Cy26 - Degree 6 / tde4_Cy26_Degree_6	Double	0	3DE Anamorphic 6: Cy26 - Degree 6.
Cx46 - Degree 6 / tde4_Cx46_Degree_6	Double	0	3DE Anamorphic 6: Cx46 - Degree 6.
Cy46 - Degree 6 / tde4_Cy46_Degree_6	Double	0	3DE Anamorphic 6: Cy46 - Degree 6.
Cx66 - Degree 6 / tde4_Cx66_Degree_6	Double	0	3DE Anamorphic 6: Cx66 - Degree 6.
Cy66 - Degree 6 / tde4_Cy66_Degree_6	Double	0	3DE Anamorphic 6: Cy66 - Degree 6.
Lens Rotation 4 / tde4_Lens_Rotation	Double	0	3DE Anamorphic 4: Lens Rotation 4.
Squeeze-X / tde4_Squeeze_X	Double	1	3DE Anamorphic 4: Squeeze-X.
Squeeze-Y / tde4_Squeeze_Y	Double	1	3DE Anamorphic 4: Squeeze-Y.

Continued on next page

Table 163 – continued from previous page

Parameter / script name	Type	Default	Function
Degree 6 / tde4_Degree_6	Double	0	3DE Fisheye: Degree 6.
Degree 8 / tde4_Degree_8	Double	0	3DE Fisheye: Degree 8.
a / pt_a	Double	0	PanoTools: Radial lens distortion 3rd degree coefficient a.
b / pt_b	Double	0	PanoTools: Radial lens distortion 2nd degree coefficient b.
c / pt_c	Double	0	PanoTools: Radial lens distortion 1st degree coefficient c.
d / pt_d	Double	0	PanoTools: Horizontal lens shift (in pixels).
e / pt_e	Double	0	PanoTools: Vertical lens shift (in pixels).
g / pt_g	Double	0	PanoTools: Vertical lens shear (in pixels). Use to remove slight misalignment of the line scanner relative to the film transport.
t / pt_t	Double	0	PanoTools: Horizontal lens shear (in pixels).
Filter / filter	Choice	Cubic	<p>Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+).</p> <p>Impulse (impulse): (nearest neighbor / box) Use original values.</p> <p>Box (box): Integrate the source image over the bounding box of the back-transformed pixel.</p> <p>Bilinear (bilinear): (tent / triangle) Bilinear interpolation between original values.</p> <p>Cubic (cubic): (cubic spline) Some smoothing.</p> <p>Keys (keys): (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*).</p> <p>Simon (simon): Some smoothing, plus medium sharpening (*).</p> <p>Rifman (rifman): Some smoothing, plus significant sharpening (*).</p> <p>Mitchell (mitchell): Some smoothing, plus blurring to hide pixelation (*+).</p> <p>Parzen (parzen): (cubic B-spline) Greatest smoothing of all filters (+).</p> <p>Notch (notch): Flat smoothing (which tends to hide moire' patterns) (+).</p>
Clamp / clamp	Boolean	Off	Clamp filter output within the original range - useful to avoid negative values in mattes
Black outside / black_outside	Boolean	Off	Fill the area outside the source image with black
Crop To Format / cropToFormat	Boolean	On	If the source is inside the format and the effect extends it outside of the format, crop it to avoid unnecessary calculations. To avoid unwanted crops, only the borders that were inside of the format in the source clip will be cropped.
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.10.8 Mirror node



This documentation is for version 1.0 of Mirror.

Description

Flip (vertical mirror) or flop (horizontal mirror) an image. Interlaced video can not be flipped.

This plugin does not concatenate transforms.

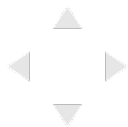
Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Vertical (flip) / <code>flip</code>	Boolean	Off	Upside-down (swap top and bottom). Only possible if input is not interlaced.
Horizontal (flop) / <code>flop</code>	Boolean	Off	Mirror image (swap left and right)

2.10.9 Position node



This documentation is for version 1.0 of Position.

Description

Translate an image by an integer number of pixels.

This plugin does not concatenate transforms.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Translate / <code>translate</code>	Double	x: 0 y: 0	New position of the bottom-left pixel. Rounded to the closest pixel.
Interactive / <code>interactive</code>	Boolean	Off	When checked the image will be rendered whenever moving the overlay interact instead of when releasing the mouse button.

2.10.10 Reformat node



This documentation is for version 1.1 of Reformat.

Description

Convert the image to another format or size.

An image transform is computed that goes from the input format, regardless of the region of definition (RoD), to the selected format. The Resize Type parameter adjust the way the transform is computed.

The output format is set by this effect.

In order to set the output format without transforming the image content, use the NoOp effect.

This plugin concatenates transforms.

See also: <http://opticalenquiry.com/nuke/index.php?title=Reformat>

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Use Source RoD / useRoD	Boolean	Off	Use the region of definition of the source as the source format.
Type / reformatType	Choice	To Format	<p>To Format: Converts between formats, the image is resized to fit in the target format. To Box: Scales to fit into a box of a given width and height. Scale: Scales the image (rounding to integer pixel sizes).</p> <p>To Format (format): Resize to predefined format.</p> <p>To Box (box): Resize to given bounding box.</p> <p>Scale (scale): Apply scale.</p>

Continued on next page

Table 166 – continued from previous page

Parameter / script name	Type	Default	Function
Format / NatronParamFormatChoice	Choice	HD 1920x1080	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
Size / boxSize	Integer	x: 200 y: 200	The output dimensions of the image in pixels.
Force This Shape / boxFixed	Boolean	Off	If checked, the output image is cropped to this size. Else, image is resized according to the resize type but the whole image is kept.
Pixel Aspect Ratio / boxPar	Double	1	Output pixel aspect ratio.
Scale / reformatScale	Double	x: 1 y: 1	The scale factor to apply to the image. The scale factor is rounded slightly, so that the output image is an integer number of pixels in the direction chosen under resize type.
Uniform / reformatScaleUniform	Boolean	Off	Use the X scale for both directions
Resize Type / resize	Choice	Width	<p>Format: Converts between formats, the image is resized to fit in the target format. Size: Scales to fit into a box of a given width and height. Scale: Scales the image.</p> <p>None (none): Do not resize the original.</p> <p>Width (width): Scale the original so that its width fits the output width, while preserving the aspect ratio.</p> <p>Height (height): Scale the original so that its height fits the output height, while preserving the aspect ratio.</p> <p>Fit (fit): Scale the original so that its smallest size fits the output width or height, while preserving the aspect ratio.</p> <p>Fill (fill): Scale the original so that its longest size fits the output width or height, while preserving the aspect ratio.</p> <p>Distort (distort): Scale the original so that both sides fit the output dimensions. This does not preserve the aspect ratio.</p>

Continued on next page

Table 166 – continued from previous page

Parameter / script name	Type	Default	Function
Center / reformatCentered	Boolean	On	Translate the center of the image to the center of the output. Otherwise, the lower left corner is left untouched.
Flip / flip	Boolean	Off	Mirror the image vertically.
Flop / flop	Boolean	Off	Mirror the image horizontally.
Turn / turn	Boolean	Off	Rotate the image by 90 degrees counter-clockwise.
Preserve BBox / preserveBB	Boolean	Off	<p>If checked, preserve the whole image bounding box and concatenate transforms downstream.</p> <p>Normally, all pixels outside of the outside format are clipped off. If this is checked, the whole image RoD is kept.</p> <p>By default, transforms are only concatenated upstream, i.e. the image is rendered by this effect by concatenating upstream transforms (e.g. CornerPin, Transform...), and the original image is resampled only once. If checked, and there are concatenating transform effects downstream, the image is rendered by the last consecutive concatenating effect.</p>
Filter / filter	Choice	Cubic	<p>Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+).</p> <p>Impulse (impulse): (nearest neighbor / box) Use original values.</p> <p>Box (box): Integrate the source image over the bounding box of the back-transformed pixel.</p> <p>Bilinear (bilinear): (tent / triangle) Bilinear interpolation between original values.</p> <p>Cubic (cubic): (cubic spline) Some smoothing.</p> <p>Keys (keys): (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*).</p> <p>Simon (simon): Some smoothing, plus medium sharpening (*).</p> <p>Rifman (rifman): Some smoothing, plus significant sharpening (*).</p> <p>Mitchell (mitchell): Some smoothing, plus blurring to hide pixelation (*+).</p> <p>Parzen (parzen): (cubic B-spline) Greatest smoothing of all filters (+).</p> <p>Notch (notch): Flat smoothing (which tends to hide moire' patterns) (+).</p>
Clamp / clamp	Boolean	Off	Clamp filter output within the original range - useful to avoid negative values in mattes
Black outside / black_outside	Boolean	Off	Fill the area outside the source image with black

2.10.11 Resize node



This documentation is for version 2.0 of Resize.

Description

Resize input stream, using OpenImageIO.

Note that only full images can be rendered, so it may be slower for interactive editing than the Reformat plugin.

However, the rendering algorithms are different between Reformat and Resize: Resize applies 1-dimensional filters in the horizontal and vertical directions, whereas Reformat resamples the image, so in some cases this plugin may give more visually pleasant results than Reformat.

This plugin does not concatenate transforms (as opposed to Reformat).

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Type / <code>type</code>	Choice	Format	<p>Format: Converts between formats, the image is resized to fit in the target format. Size: Scales to fit into a box of a given width and height. Scale: Scales the image.</p> <p>Format (format) Size (size) Scale (scale)</p>
Format / <code>format</code>	Choice	PC_Video 640x480	<p>The output format</p> <p>PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x1720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super35(full-ap) 1024x778 (1K_Super35(full-ap)) 1K_Cinemascope 914x778 2 (1K_Cinemascope) 2K_Super35(full-ap) 2048x1556 (2K_Super35(full-ap)) 2K_Cinemascope 1828x1556 2 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super35(full-ap) 4096x3112 (4K_Super35(full-ap)) 4K_Cinemascope 3656x3112 2 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)</p>

Continued on next page

Table 167 – continued from previous page

Parameter / script name	Type	Default	Function
Size / size	Integer	x: 200 y: 200	The output size
Preserve PAR / preservePAR	Boolean	On	Preserve Pixel Aspect Ratio (PAR). When checked, one direction will be clipped.
Scale / scale	Double	x: 1 y: 1	The scale factor to apply to the image.
Filter / filter	Choice	Default	<p>The filter used to resize. Lanczos3 is great for downscaling and blackman-harris is great for upscaling.</p> <p>Impulse (impulse): No interpolation.</p> <p>box</p> <p>triangle</p> <p>gaussian</p> <p>sharp-gaussian</p> <p>catmull-rom</p> <p>blackman-harris</p> <p>sinc</p> <p>lanczos3</p> <p>radial-lanczos3</p> <p>mitchell</p> <p>bspline</p> <p>disk</p> <p>cubic</p> <p>keys</p> <p>simon</p> <p>rifman</p> <p>Default (default): blackman-harris when increasing resolution, lanczos3 when decreasing resolution.</p>

2.10.12 STMap node



This documentation is for version 2.0 of STMap.

Description

Move pixels around an image, based on a UVmap.

The U and V channels give, for each pixel in the destination image, the normalized position of the pixel where the color is taken. (0,0) is the bottom left corner of the input image, while (1,1) is the top right corner. This plugin concatenates transforms upstream, so that if the nodes upstream output a 3x3 transform (e.g. Transform, CornerPin, Dot, NoOp, Switch), the original image is sampled only once.

This plugin concatenates transforms upstream.

Inputs

Input	Description	Optional
UV		No
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
U Channel / <code>channelU</code>	Choice	Color.R	<p>Input U channel from UV.</p> <p>Color.R (<code>uk.co.thefoundry.OfxImagePlaneColour.R</code>): R channel from input UV</p> <p>Color.G (<code>uk.co.thefoundry.OfxImagePlaneColour.G</code>): G channel from input UV</p> <p>Color.B (<code>uk.co.thefoundry.OfxImagePlaneColour.B</code>): B channel from input UV</p> <p>Color.A (<code>uk.co.thefoundry.OfxImagePlaneColour.A</code>): A channel from input UV</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p>
V Channel / <code>channelV</code>	Choice	Color.G	<p>Input V channel from UV.</p> <p>Color.R (<code>uk.co.thefoundry.OfxImagePlaneColour.R</code>): R channel from input UV</p> <p>Color.G (<code>uk.co.thefoundry.OfxImagePlaneColour.G</code>): G channel from input UV</p> <p>Color.B (<code>uk.co.thefoundry.OfxImagePlaneColour.B</code>): B channel from input UV</p> <p>Color.A (<code>uk.co.thefoundry.OfxImagePlaneColour.A</code>): A channel from input UV</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p>
Alpha Channel / <code>channelA</code>	Choice	Color.A	<p>Input Alpha channel from UV. The Output alpha is set to this value. If “Unpremult UV” is checked, the UV values are divided by alpha.</p> <p>Color.R (<code>uk.co.thefoundry.OfxImagePlaneColour.R</code>): R channel from input UV</p> <p>Color.G (<code>uk.co.thefoundry.OfxImagePlaneColour.G</code>): G channel from input UV</p> <p>Color.B (<code>uk.co.thefoundry.OfxImagePlaneColour.B</code>): B channel from input UV</p> <p>Color.A (<code>uk.co.thefoundry.OfxImagePlaneColour.A</code>): A channel from input UV</p> <p>0: 0 constant channel</p> <p>1: 1 constant channel</p>

Continued on next page

Table 168 – continued from previous page

Parameter / script name	Type	Default	Function
Unpremult UV / unpremultUV	Boolean	Off	Unpremult UV by Alpha from UV. Check if UV values look small for small values of Alpha (3D software sometimes write premultiplied UV values).
UV Offset / uvOffset	Double	U: 0 V: 0	Offset to apply to the U and V channel (useful if these were stored in a file that cannot handle negative numbers)
UV Scale / uvScale	Double	U: 1 V: 1	Scale factor to apply to the U and V channel (useful if these were stored in a file that can only store integer values)
U Wrap Mode / wrapU	Choice	Clamp	Wrap mode for U coordinate. Clamp (clamp) : Texture edges are black (if blackOutside is checked) or stretched indefinitely. Repeat (repeat) : Texture is repeated. Mirror (mirror) : Texture is mirrored alternatively.
V Wrap Mode / wrapV	Choice	Clamp	Wrap mode for V coordinate. Clamp (clamp) : Texture edges are black (if blackOutside is checked) or stretched indefinitely. Repeat (repeat) : Texture is repeated. Mirror (mirror) : Texture is mirrored alternatively.
Filter / filter	Choice	Cubic	Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+). Impulse (impulse) : (nearest neighbor / box) Use original values. Box (box) : Integrate the source image over the bounding box of the back-transformed pixel. Bilinear (bilinear) : (tent / triangle) Bilinear interpolation between original values. Cubic (cubic) : (cubic spline) Some smoothing. Keys (keys) : (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*). Simon (simon) : Some smoothing, plus medium sharpening (*). Rifman (rifman) : Some smoothing, plus significant sharpening (*). Mitchell (mitchell) : Some smoothing, plus blurring to hide pixelation (*+). Parzen (parzen) : (cubic B-spline) Greatest smoothing of all filters (+). Notch (notch) : Flat smoothing (which tends to hide moire' patterns) (+).
Clamp / clamp	Boolean	Off	Clamp filter output within the original range - useful to avoid negative values in mattes
Black outside / black_outside	Boolean	On	Fill the area outside the source image with black
(Un)premult / premult	Boolean	Off	Divide the image by the alpha channel before processing, and re-multiply it afterwards. Use if the input images are premultiplied.
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.10.13 SpriteSheet node

This documentation is for version 1.0 of SpriteSheet.

Description

Read individual frames from a sprite sheet. A sprite sheet is a series of images (usually animation frames) combined into a larger image (or images). For example, an animation consisting of eight 100x100 images could be combined into a single 400x200 sprite sheet (4 frames across by 2 high). The sprite with index 0 is at the top-left of the source image, and sprites are ordered left-to-right and top-to-bottom. The output is an animated sprite that repeats the sprites given in the sprite range. The ContactSheet effect can be used to make a spritesheet from a series of images or a video.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Sprite Size / <code>spriteSize</code>	Integer	x: 64 y: 64	Size in pixels of an individual sprite.
Sprite Range / <code>spriteRange</code>	Integer	first: 0 last: 0	Index of the first and last sprite in the animation. The sprite index starts at zero.
Frame Offset / <code>frameOffset</code>	Integer	1	Output frame number for the first sprite.

2.10.14 Tracker node

This documentation is for version 1.0 of Tracker.

Description

Track one or more 2D point(s) using LibMV from the Blender open-source software.

Goal

Track one or more 2D point and use them to either make another object/image match-move their motion or to stabilize the input image.

Tracking

- Connect a Tracker node to the image containing the item you need to track
- Place tracking markers with CTRL+ALT+Click on the Viewer or by clicking the + button below the track table in the settings panel
- Setup the motion model to match the motion type of the item you need to track. By default the tracker will only assume the item is undergoing a translation. Other motion models can be used for complex tracks but may be slower.

- Select in the settings panel or on the Viewer the markers you want to track and then start tracking with the player buttons on the top of the Viewer.
- If a track is getting lost or fails at some point, you may refine it by moving the marker at its correct position, this will force a new keyframe on the pattern which will be visible in the Viewer and on the timeline.

Using the tracks data

You can either use the Tracker node itself to use the track data or you may export it to another node.

Using the Transform within the Tracker node

Go to the Transform tab in the settings panel, and set the Transform Type to the operation you want to achieve. During tracking, the Transform Type should always been set to None if you want to correctly see the tracks on the Viewer.

You will notice that the transform parameters will be set automatically when the tracking is finished. Depending on the Transform Type, the values will be computed either to match-move the motion of the tracked points or to stabilize the image.

Exporting the tracking data

You may export the tracking data either to a CornerPin node or to a Transform node. The CornerPin node performs a warp that may be more stable than a Transform node when using 4 or more tracks: it retains more information than the Transform node.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Motion Type / <code>motionType</code>	Choice	None	<p>The type of motion in output of this node.</p> <p>None: No transformation applied in output to the image: this node is a pass-through. Set it to this mode when tracking to correctly see the input image on the viewer</p> <p>Stabilize: Transforms the image so that the tracked points do not move</p> <p>Match-Move: Transforms a different image so that it moves to match the tracked points</p> <p>Remove Jitter: Transforms the image so that the tracked points move smoothly with high frequencies removed</p> <p>Add Jitter: Transforms the image by the high frequencies of the animation of the tracks to increase the shake or apply it on another image</p>

Continued on next page

Table 170 – continued from previous page

Parameter / script name	Type	Default	Function
Transform Type / <code>transformType</code>	Choice	CornerPin	<p>The type of transform used to produce the results.</p> <p>Transform: The tracks motion will be used to compute the translation, scale and rotation parameter of a Transform node. At least 1 track is required to compute the translation and 2 for scale and rotation. The more tracks you use, the more stable and precise the resulting transform will be.</p> <p>CornerPin: The tracks motion will be used to compute a CornerPin. A CornerPin is useful if you are tracking an image portion that has a perspective distortion. At least 1 track is required to compute the homography transforming the “From” points to the “To” points, and 4 required to track a perspective transformation. The more points you add, the more stable and precise the resulting CornerPin will be.</p>
Reference Frame / <code>referenceFrame</code>	Integer	1	When exporting tracks to a CornerPin or Transform, this will be the frame number at which the transform will be an identity.
Set to Current Frame / <code>setReferenceButton</code>	Button		Set the reference frame to the timeline’s current frame
Jitter Period / <code>jitterPeriod</code>	Integer	10	Number of frames to average together to remove high frequencies for the add/remove jitter transform type
Smooth / <code>smooth</code>	Integer	t: 0 r: 0 s: 0	Smooth the translation/rotation/scale by averaging this number of frames together
Smooth / <code>smoothCornerPin</code>	Integer	0	Smooth the CornerPin by averaging this number of frames together
Compute Transform Automatically / <code>autoComputeTransform</code>	Boolean	On	When checked, whenever changing a parameter controlling the Transform Generation (such as Motion Type, Transform Type, Reference Frame, etc...) or changing the Enabled parameter of a track, the transform parameters will be re-computed automatically. When unchecked, you must press the Compute button to compute it.
Compute / <code>computeTransform</code>	Button		Click to compute the parameters of the Transform Controls or CornerPin Controls (depending on the Transform Type) from the data acquired on the tracks during the tracking. This should be done after the tracking is finished and when you feel the results are satisfying. For each frame, the resulting parameter is computed from the tracks that are enabled at this frame and that have a keyframe on the center point (e.g: are valid).
Robust Model / <code>robustModel</code>	Boolean	On	When checked, the solver will assume that the model generated (i.e: the Transform or the CornerPin) is possible given the motion of the video and will eliminate points that do not match the model to compute the resulting parameters. When unchecked, the solver assumes that all points that are enabled and have a keyframe are valid and fit the model: this may in some situations work better if you are trying to find a model that is just not correct for the given motion of the video.
Fitting Error (px) / <code>fittingError</code>	Double	0	This parameter indicates the error for each frame of the fitting of the model (i.e: Transform / CornerPin) to the tracks data. This value is in pixels and represents the rooted weighted sum of squared errors for each track. The error is essentially the difference between the point position computed from the original point onto which is applied the fitted model and the original tracked point.
Warn If Error Is Above / <code>fittingErrorWarnAbove</code>	Double	1	A warning will appear if the model fitting error reaches this value (or higher). The warning indicates that the calculated model is probably poorly suited for the stabilization/match-move you want to achieve and you should either refine your tracking data or pick another model

Continued on next page

Table 170 – continued from previous page

Parameter / script name	Type	Default	Function
Disable Transform / disableProcess	Boolean	Off	When checked, the CornerPin/Transform applied by the parameters is disabled temporarily. This is useful if you are using a CornerPin and you need to edit the From or To points. For example, in match-move mode to replace a portion of the image by another one. To achieve such effect, you would need to place the From points of the CornerPin controls to the desired 4 corners in the image. Similarly, you may want to stabilize the image onto a moving vehicle, in which case you would want to set the CornerPin points to enclose the vehicle.
Set to Input Rod / setToInputRod	Button		Set the 4 from points to the image rectangle in input of the tracker node
Export / export	Button		Creates a node referencing the tracked data. The node type depends on the node selected by the Transform Type parameter. The type of transformation applied by the created node depends on the Motion Type parameter. To activate this button you must select set the Motion Type to something other than None
Mag. Window Size / magWindowSize	Integer	200	The size of the selected track magnification winow in pixels

2.10.15 Transform node



This documentation is for version 1.0 of Transform.

Description

Translate / Rotate / Scale a 2D image.

This plugin concatenates transforms.

See also <http://opticalenquiry.com/nuke/index.php?title=Transform>

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Translate / translate	Double	x: 0 y: 0	Translation along the x and y axes in pixels. Can also be adjusted by clicking and dragging the center handle in the Viewer.
Rotate / rotate	Double	0	Rotation angle in degrees around the Center. Can also be adjusted by clicking and dragging the rotation bar in the Viewer.
Scale / scale	Double	x: 1 y: 1	Scale factor along the x and y axes. Can also be adjusted by clicking and dragging the outer circle or the diameter handles in the Viewer.
Uniform / uniform	Boolean	Off	Use the X scale for both directions

Continued on next page

Table 171 – continued from previous page

Parameter / script name	Type	Default	Function
Skew X / skewX	Double	0	Skew along the x axis. Can also be adjusted by clicking and dragging the skew bar in the Viewer.
Skew Y / skewY	Double	0	Skew along the y axis.
Skew Order / skewOrder	Choice	XY	The order in which skew transforms are applied: X then Y, or Y then X. XY YX
Amount / transformAmount	Double	1	Amount of transform to apply. 0 means the transform is identity, 1 means to apply the full transform.
Center / center	Double	x: 0.5 y: 0.5	Center of rotation and scale.
Reset Center / resetCenter	Button		Reset the position of the center to the center of the input region of definition
Interactive Update / interactive	Boolean	On	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Invert / invert	Boolean	Off	Invert the transform.
Filter / filter	Choice	Cubic	Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+). Impulse (impulse) : (nearest neighbor / box) Use original values. Box (box) : Integrate the source image over the bounding box of the back-transformed pixel. Bilinear (bilinear) : (tent / triangle) Bilinear interpolation between original values. Cubic (cubic) : (cubic spline) Some smoothing. Keys (keys) : (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*). Simon (simon) : Some smoothing, plus medium sharpening (*). Rifman (rifman) : Some smoothing, plus significant sharpening (*). Mitchell (mitchell) : Some smoothing, plus blurring to hide pixelation (*+). Parzen (parzen) : (cubic B-spline) Greatest smoothing of all filters (+). Notch (notch) : Flat smoothing (which tends to hide moire' patterns) (+).
Clamp / clamp	Boolean	Off	Clamp filter output within the original range - useful to avoid negative values in mattes
Black outside / black_outside	Boolean	On	Fill the area outside the source image with black
Motion Blur / motionBlur	Double	0	Quality of motion blur rendering. 0 disables motion blur, 1 is a good value. Increasing this slows down rendering.
Directional Blur Mode / directionalBlur	Boolean	Off	Motion blur is computed from the original image to the transformed image, each parameter being interpolated linearly. The motionBlur parameter must be set to a nonzero value, and the blackOutside parameter may have an important effect on the result.
Shutter / shutter	Double	0.5	Controls how long (in frames) the shutter should remain open.

Continued on next page

Table 171 – continued from previous page

Parameter / script name	Type	Default	Function
Shutter Offset / shutterOffset	Choice	Start	Controls when the shutter should be open/closed. Ignored if there is no motion blur (i.e. shutter=0 or motionBlur=0). Centered (centered) : Centers the shutter around the frame (from t-shutter/2 to t+shutter/2) Start (start) : Open the shutter at the frame (from t to t+shutter) End (end) : Close the shutter at the frame (from t-shutter to t) Custom (custom) : Open the shutter at t+shuttercustomoffset (from t+shuttercustomoffset to t+shuttercustomoffset+shutter)
Custom Offset / shutterCustomOffset	Double	0	When custom is selected, the shutter is open at current time plus this offset (in frames). Ignored if there is no motion blur (i.e. shutter=0 or motionBlur=0).

2.10.16 TransformMasked node



This documentation is for version 1.0 of TransformMasked.

Description

Translate / Rotate / Scale a 2D image, with optional masking.

This plugin concatenates transforms upstream.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Translate / translate	Double	x: 0 y: 0	Translation along the x and y axes in pixels. Can also be adjusted by clicking and dragging the center handle in the Viewer.
Rotate / rotate	Double	0	Rotation angle in degrees around the Center. Can also be adjusted by clicking and dragging the rotation bar in the Viewer.
Scale / scale	Double	x: 1 y: 1	Scale factor along the x and y axes. Can also be adjusted by clicking and dragging the outer circle or the diameter handles in the Viewer.
Uniform / uniform	Boolean	Off	Use the X scale for both directions
Skew X / skewX	Double	0	Skew along the x axis. Can also be adjusted by clicking and dragging the skew bar in the Viewer.
Skew Y / skewY	Double	0	Skew along the y axis.

Continued on next page

Table 172 – continued from previous page

Parameter / script name	Type	Default	Function
Skew Order / skewOrder	Choice	XY	The order in which skew transforms are applied: X then Y, or Y then X. XY YX
Amount / transformAmount	Double	1	Amount of transform to apply. 0 means the transform is identity, 1 means to apply the full transform.
Center / center	Double	x: 0.5 y: 0.5	Center of rotation and scale.
Reset Center / resetCenter	Button		Reset the position of the center to the center of the input region of definition
Interactive Update / interactive	Boolean	On	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Invert / invert	Boolean	Off	Invert the transform.
Filter / filter	Choice	Cubic	Filtering algorithm - some filters may produce values outside of the initial range (*) or modify the values even if there is no movement (+). Impulse (impulse) : (nearest neighbor / box) Use original values. Box (box) : Integrate the source image over the bounding box of the back-transformed pixel. Bilinear (bilinear) : (tent / triangle) Bilinear interpolation between original values. Cubic (cubic) : (cubic spline) Some smoothing. Keys (keys) : (Catmull-Rom / Hermite spline) Some smoothing, plus minor sharpening (*). Simon (simon) : Some smoothing, plus medium sharpening (*). Rifman (rifman) : Some smoothing, plus significant sharpening (*). Mitchell (mitchell) : Some smoothing, plus blurring to hide pixelation (*+). Parzen (parzen) : (cubic B-spline) Greatest smoothing of all filters (+). Notch (notch) : Flat smoothing (which tends to hide moire' patterns) (+).
Clamp / clamp	Boolean	Off	Clamp filter output within the original range - useful to avoid negative values in mattes
Black outside / black_outside	Boolean	On	Fill the area outside the source image with black
Motion Blur / motionBlur	Double	0	Quality of motion blur rendering. 0 disables motion blur, 1 is a good value. Increasing this slows down rendering.
Directional Blur Mode / directionalBlur	Boolean	Off	Motion blur is computed from the original image to the transformed image, each parameter being interpolated linearly. The motionBlur parameter must be set to a nonzero value, and the blackOutside parameter may have an important effect on the result.
Shutter / shutter	Double	0.5	Controls how long (in frames) the shutter should remain open.

Continued on next page

Table 172 – continued from previous page

Parameter / script name	Type	Default	Function
Shutter Offset / shutterOffset	Choice	Start	Controls when the shutter should be open/closed. Ignored if there is no motion blur (i.e. shutter=0 or motionBlur=0). Centered (centered) : Centers the shutter around the frame (from t-shutter/2 to t+shutter/2) Start (start) : Open the shutter at the frame (from t to t+shutter) End (end) : Close the shutter at the frame (from t-shutter to t) Custom (custom) : Open the shutter at t+shuttercustomoffset (from t+shuttercustomoffset to t+shuttercustomoffset+shutter)
Custom Offset / shutterCustomOffset	Double	0	When custom is selected, the shutter is open at current time plus this offset (in frames). Ignored if there is no motion blur (i.e. shutter=0 or motionBlur=0).
Invert Mask / maskInvert	Boolean	Off	When checked, the effect is fully applied where the mask is 0.
Mix / mix	Double	1	Mix factor between the original and the transformed image.

2.11 Views nodes

The following sections contain documentation about every node in the Views group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.11.1 Anaglyph node



This documentation is for version 1.0 of Anaglyph.

Description

Make an anaglyph image out of the two views of the input.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Color Amount / amtcolor	Double	0	Amount of colour in the anaglyph: 0 = grayscale anaglyph, 1 = full-color anaglyph. Fusion is more difficult with full-color anaglyphs.
(right=red) / swap	Boolean	Off	Swap left and right views

Continued on next page

Table 173 – continued from previous page

Parameter / script name	Type	Default	Function
Horizontal Offset / offset	Integer	0	Horizontal offset. The red view is shifted to the left by half this amount, and the cyan view is shifted to the right by half this amount (in pixels).

2.11.2 JoinViews node

This documentation is for version 1.0 of JoinViews.

Description

Take in input separate views to make a multiple view stream output. The first view from each input is copied to one of the view of the output.

Inputs

Input	Description	Optional
Main		Yes

Controls

Parameter / script name	Type	Default	Function

2.11.3 MixViews node



This documentation is for version 1.0 of MixViews.

Description

Mix two views together.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Mix / <code>mix</code>	Double	0	Mix factor for the right view

2.11.4 OneView node

This documentation is for version 1.0 of OneView.

Description

Takes one view from the input.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
View / <code>view</code>	Choice		View to take from the input

2.11.5 SideBySide node



This documentation is for version 1.0 of SideBySide.

Description

Put the left and right view of the input next to each other.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Vertical / <code>vertical</code>	Boolean	Off	Stack views vertically instead of horizontally

Continued on next page

Table 177 – continued from previous page

Parameter / script name	Type	Default	Function
View 1 / view1	Choice	Left	First view Left (left) : Left view. Right (right) : Right view.
View 2 / view2	Choice	Right	Second view Left (left) : Left view. Right (right) : Right view.

2.12 Other nodes

The following sections contain documentation about every node in the Other group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.12.1 Backdrop node

This documentation is for version 1.0 of Backdrop.

Description

The Backdrop node is useful to group nodes and identify them in the node graph.

You can also move all the nodes inside the backdrop.

Inputs

Input	Description	Optional

Controls

Parameter / script name	Type	Default	Function
Label / Label	String		Text to display on the backdrop.

2.12.2 DiskCache node

This documentation is for version 1.0 of DiskCache.

Description

This node caches all images of the connected input node onto the disk with full 32bit floating point raw data. When an image is found in the cache, Natron will then not request the input branch to render out that image. The DiskCache node only caches full images and does not split up the images in chunks. The DiskCache node is useful if working with a large and complex node tree: this allows to break the tree into smaller branches and cache any branch that you're no longer working on. The cached images are saved by default in the same directory

that is used for the viewer cache but you can set its location and size in the preferences. A solid state drive disk is recommended for efficiency of this node. By default all images that pass into the node are cached but they depend on the zoom-level of the viewer. For convenience you can cache a specific frame range at scale 100% much like a writer node would do.

WARNING: The DiskCache node must be part of the tree when you want to read cached data from it.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Frame range / frameRange	Choice	Input frame range	Input frame range Project frame range Manual
Pre-cache / preRender	Button		Cache the frame range specified by rendering images at zoom-level 100% only.

2.12.3 Dot node

This documentation is for version 1.0 of Dot.

Description

Does not do anything to the input image, this is used in the node graph to make bends in the links.

Inputs

Input	Description	Optional
		No

Controls

Parameter / script name	Type	Default	Function

2.12.4 Group node

This documentation is for version 1.0 of Group.

Description

Use this to nest multiple nodes into a single node. The original nodes will be replaced by the Group node and its content is available in a separate NodeGraph tab. You can add user parameters to the Group node which can drive parameters of nodes nested within the Group. To specify the outputs and inputs of the Group node, you may add multiple Input node within the group and exactly 1 Output node.

Inputs

Input	Description	Optional

Controls

Parameter / script name	Type	Default	Function
Convert to Group / <code>convertToGroup</code>	Button		Converts this node to a Group: the internal node-graph and the user parameters will become editable

2.12.5 ImageStatistics node



This documentation is for version 1.0 of ImageStatistics.

Description

Compute image statistics over the whole image or over a rectangle. The statistics can be computed either on RGBA components, in the HSVL colorspace (which is the HSV colorspace with an additional L component from HSL), or the position and value of the pixels with the maximum and minimum luminance values can be computed.

The color values of the minimum and maximum luma pixels for an image sequence can be used as black and white point in a Grade node to remove flicker from the same sequence.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Restrict to Rectangle / <code>restrictToRectangle</code>	Boolean	On	Restrict statistics computation to a rectangle.
Bottom Left / <code>bottomLeft</code>	Double	x: 0 y: 0	Coordinates of the bottom left corner of the rectangle

Continued on next page

Table 182 – continued from previous page

Parameter / script name	Type	Default	Function
Size / size	Double	w: 1 h: 1	Width and height of the rectangle
Auto Update / autoUpdate	Boolean	On	Automatically update values when input or rectangle changes if an analysis was performed at current frame. If not checked, values are only updated if the plugin parameters change.
Interactive Update / interactive	Boolean	Off	If checked, update the parameter values during interaction with the image viewer, else update the values when pen is released.
Min. / statMin	Color	r: 0 g: 0 b: 0 a: 0	Minimum value.
Max. / statMax	Color	r: 0 g: 0 b: 0 a: 0	Maximum value.
Mean / statMean	Color	r: 0 g: 0 b: 0 a: 0	The mean is the average. Add up the values, and divide by the number of values.
S.Dev. / statSDev	Color	r: 0 g: 0 b: 0 a: 0	The standard deviation (S.Dev.) quantifies variability or scatter, and it is expressed in the same units as your data.
Skewness / statSkewness	Color	r: 0 g: 0 b: 0 a: 0	<p>Skewness quantifies how symmetrical the distribution is.</p> <ul style="list-style-type: none"> • A symmetrical distribution has a skewness of zero. • An asymmetrical distribution with a long tail to the right (higher values) has a positive skew. • An asymmetrical distribution with a long tail to the left (lower values) has a negative skew. • The skewness is unitless. • Any threshold or rule of thumb is arbitrary, but here is one: If the skewness is greater than 1.0 (or less than -1.0), the skewness is substantial and the distribution is far from symmetrical.
Kurtosis / statKurtosis	Color	r: 0 g: 0 b: 0 a: 0	<p>Kurtosis quantifies whether the shape of the data distribution matches the Gaussian distribution.</p> <ul style="list-style-type: none"> • A Gaussian distribution has a kurtosis of 0. • A flatter distribution has a negative kurtosis, • A distribution more peaked than a Gaussian distribution has a positive kurtosis. • Kurtosis has no units. • The value that this plugin reports is sometimes called the excess kurtosis since the expected kurtosis for a Gaussian distribution is 0.0. • An alternative definition of kurtosis is computed by adding 3 to the value reported by this plugin. With this definition, a Gaussian distribution is expected to have a kurtosis of 3.0.
Analyze Frame / analyzeFrame	Button		Analyze current frame and set values.
Analyze Sequence / analyzeSequence	Button		Analyze all frames from the sequence and set values.
Clear Frame / clearFrame	Button		Clear analysis for current frame.

Continued on next page

Table 182 – continued from previous page

Parameter / script name	Type	Default	Function
Clear Sequence / <code>clearSequence</code>	Button		Clear analysis for all frames from the sequence.
HSV L Min. / <code>statHSV LMin</code>	Color	h: 0 s: 0 v: 0 l: 0	Minimum value.
HSV L Max. / <code>statHSV LMax</code>	Color	h: 0 s: 0 v: 0 l: 0	Maximum value.
HSV L Mean / <code>statHSV LMean</code>	Color	h: 0 s: 0 v: 0 l: 0	The mean is the average. Add up the values, and divide by the number of values.
HSV L S.Dev. / <code>statHSV LSDev</code>	Color	h: 0 s: 0 v: 0 l: 0	The standard deviation (S.Dev.) quantifies variability or scatter, and it is expressed in the same units as your data.
HSV L Skewness / <code>statHSV LSkewness</code>	Color	h: 0 s: 0 v: 0 l: 0	Skewness quantifies how symmetrical the distribution is. <ul style="list-style-type: none"> • A symmetrical distribution has a skewness of zero. • An asymmetrical distribution with a long tail to the right (higher values) has a positive skew. • An asymmetrical distribution with a long tail to the left (lower values) has a negative skew. • The skewness is unitless. • Any threshold or rule of thumb is arbitrary, but here is one: If the skewness is greater than 1.0 (or less than -1.0), the skewness is substantial and the distribution is far from symmetrical.
HSV L Kurtosis / <code>statHSV LKurtosis</code>	Color	h: 0 s: 0 v: 0 l: 0	Kurtosis quantifies whether the shape of the data distribution matches the Gaussian distribution. <ul style="list-style-type: none"> • A Gaussian distribution has a kurtosis of 0. • A flatter distribution has a negative kurtosis, • A distribution more peaked than a Gaussian distribution has a positive kurtosis. • Kurtosis has no units. • The value that this plugin reports is sometimes called the excess kurtosis since the expected kurtosis for a Gaussian distribution is 0.0. • An alternative definition of kurtosis is computed by adding 3 to the value reported by this plugin. With this definition, a Gaussian distribution is expected to have a kurtosis of 3.0.
Analyze Frame / <code>analyzeFrameHSV L</code>	Button		Analyze current frame as HSV L and set values.
Analyze Sequence / <code>analyzeSequenceHSV L</code>	Button		Analyze all frames from the sequence as HSV L and set values.
Clear Frame / <code>clearFrameHSV L</code>	Button		Clear HSV L analysis for current frame.
Clear Sequence / <code>clearSequenceHSV L</code>	Button		Clear HSV L analysis for all frames from the sequence.

Continued on next page

Table 182 – continued from previous page

Parameter / script name	Type	Default	Function
Luminance Math / luminanceMath	Choice	Rec. 709	Formula used to compute luminance from RGB values. Rec. 709 (rec709) : Use Rec. 709 ($0.2126r + 0.7152g + 0.0722b$). Rec. 2020 (rec2020) : Use Rec. 2020 ($0.2627r + 0.6780g + 0.0593b$). ACES AP0 (acesap0) : Use ACES AP0 ($0.3439664498r + 0.7281660966g + -0.0721325464b$). ACES AP1 (acesap1) : Use ACES AP1 ($0.2722287168r + 0.6740817658g + 0.0536895174b$). CCIR 601 (ccir601) : Use CCIR 601 ($0.2989r + 0.5866g + 0.1145b$). Average (average) : Use average of r, g, b. Max (max) : Use max of r, g, b.
Max Luma Pixel / maxLumaPix	Double	x: 0 y: 0	Position of the pixel with the maximum luma value.
Max Luma Pixel Value / maxLumaPixVal	Color	r: 0 g: 0 b: 0 a: 0	RGB value for the pixel with the maximum luma value.
Min Luma Pixel / minLumaPix	Double	x: 0 y: 0	Position of the pixel with the minimum luma value.
Min Luma Pixel Value / minLumaPixVal	Color	r: 0 g: 0 b: 0 a: 0	RGB value for the pixel with the minimum luma value.
Analyze Frame / analyzeFrameLuma	Button		Analyze current frame and set min/max luma values.
Analyze Sequence / analyzeSequenceLuma	Button		Analyze all frames from the sequence and set min/max luma values.
Clear Frame / clearFrameLuma	Button		Clear luma analysis for current frame.
Clear Sequence / clearSequenceLuma	Button		Clear luma analysis for all frames from the sequence.

2.12.6 Input node

This documentation is for version 1.0 of Input.

Description

This node can only be used within a Group. It adds an input arrow to the group.

Inputs

Input	Description	Optional

Controls

Parameter / script name	Type	Default	Function
Optional / optional	Boolean	Off	When checked, this input of the group will be optional, i.e. it will not be required that it is connected for the render to work.
Mask / isMask	Boolean	Off	When checked, this input of the group will be considered as a mask. A mask is always optional.

2.12.7 NoOp node



This documentation is for version 2.0 of NoOp.

Description

Copies the input to the output.

This effect does not modify the actual content of the image, but can be used to modify the metadata associated with the clip (premultiplication, field order, format, pixel aspect ratio, frame rate).

This plugin concatenates transforms.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Force Copy / forceCopy	Boolean	Off	Force copy from input to output
Supports Tiles / supportsTiles	Boolean	On	Does the plugin support image tiling, i.e. rendering only a subset of the full region of definition? Only supported on OpenFX 1.4 hosts.
Set Premultiplication / setPremult	Boolean	Off	Set the premultiplication state of the output clip, without modifying the raw content. Use the Premult or UnPremult plu-gins to affect the content.
Output Premultiplication / outputPremult	Choice	PreMultiplied	Premultiplication state of the output clip. Opaque PreMultiplied UnPreMultiplied
Set Format / setFormat	Boolean	Off	Set the format of the output clip, without modifying the raw content.

Continued on next page

Table 184 – continued from previous page

Parameter / script name	Type	Default	Function
Extent / extent	Choice	Format	Extent (size and offset) of the output. Format (format): Use a pre-defined image format. Size (size): Use a specific extent (size and offset). Project (project): Use the project extent (size and offset).
Center / recenter	Button		Centers the region of definition to the input region of definition. If there is no input, then the region of definition is centered to the project window.
Format / NatronParamFormatChoice	Choice	HD 1920x1080	The output format PC_Video 640x480 (PC_Video) NTSC 720x486 0.91 (NTSC) PAL 720x576 1.09 (PAL) NTSC_16:9 720x486 1.21 (NTSC_16:9) PAL_16:9 720x576 1.46 (PAL_16:9) HD_720 1280x720 (HD_720) HD 1920x1080 (HD) UHD_4K 3840x2160 (UHD_4K) 1K_Super_35(full-ap) 1024x778 (1K_Super_35(full-ap)) 1K_Cinemascope 914x778 2.00 (1K_Cinemascope) 2K_Super_35(full-ap) 2048x1556 (2K_Super_35(full-ap)) 2K_Cinemascope 1828x1556 2.00 (2K_Cinemascope) 2K_DCP 2048x1080 (2K_DCP) 4K_Super_35(full-ap) 4096x3112 (4K_Super_35(full-ap)) 4K_Cinemascope 3656x3112 2.00 (4K_Cinemascope) 4K_DCP 4096x2160 (4K_DCP) square_256 256x256 (square_256) square_512 512x512 (square_512) square_1K 1024x1024 (square_1K) square_2K 2048x2048 (square_2K)
Bottom Left / bottomLeft	Double	x: 0 y: 0	Coordinates of the bottom left corner of the size rectangle.
Size / size	Double	w: 1 h: 1	Width and height of the size rectangle.
Set Pixel Aspect Ratio / setPixelAspectRatio	Boolean	Off	Set the pixel aspect ratio of the output clip, without modifying the raw content.
Output Pixel Aspect Ratio / outputPixelAspectRatio	Double	1	Pixel aspect ratio of the output clip.
Set Frame Rate / setFrameRate	Boolean	Off	Set the frame rate state of the output clip, without modifying the raw content.
Output Frame Rate / outputFrameRate	Double	24	Frame rate of the output clip.
Clip Info... / clipInfo	Button		Display information about the inputs

2.12.8 Output node

This documentation is for version 1.0 of Output.

Description

This node can only be used within a Group. There can only be 1 Output node in the group. It defines the output of the group.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function

2.12.9 Precomp node

This documentation is for version 1.0 of Precomp.

Description

The Precomp node is like a Group node, but references an external Natron project (.ntp) instead.

This allows you to save a subset of the node tree as a separate project. A Precomp node can be useful in at least two ways:

It can be used to reduce portions of the node tree to pre-rendered image inputs. This speeds up render time: Natron only has to process the single image input instead of all the nodes within the project. Since this is a separate project, you also maintain access to the internal tree and can edit it any time.

It enables a collaborative project: while one user works on the main project, others can work on other parts referenced by the Precomp node.

Inputs

Input	Description	Optional

Controls

Parameter / script name	Type	Default	Function
Project Filename (.ntp) / projectFilename	N/A		The absolute file path of the project to use as a pre-comp.

Continued on next page

Table 186 – continued from previous page

Parameter / script name	Type	Default	Function
Edit Project... / <code>editProject</code>	Button		Opens the specified project in a new Natron instance
Pre-Render / <code>preRender</code>	Boolean	On	<p>When checked the output of this node will be the images read directly from what is rendered by the node indicated by “Write Node”. If no Write is selected, or if the rendered images do not exist this node will have the behavior determined by the “On Error” parameter. To pre-render images, select a write node, a frame-range and hit “Render”.</p> <p>When unchecked, this node will output the image rendered by the node indicated in the “Output Node” parameter by rendering the full-tree of the sub-project. In that case no writing on disk will occur and the images will be cached with the same policy as if the nodes were used in the active project in the first place.</p>
Write Node / <code>writeNode</code>	Choice		Choose here the Write node in the pre-comp from which to render images then specify a frame-range and hit the “Render” button.
First-Frame / <code>first</code>	Integer	0	The first-frame to render
Last-Frame / <code>last</code>	Integer	0	The last-frame to render
On Error / <code>onError</code>	Choice	Error	<p>Indicates the behavior when an image is missing from the render of the pre-comp project</p> <p>Load previous: Loads the previous frame in the sequence.</p> <p>Load next: Loads the next frame in the sequence.</p> <p>Load nearest: Loads the nearest frame in the sequence.</p> <p>Error: Fails to render.</p> <p>Black: Black Image.</p>
Render / <code>render</code>	Button		

2.13 GMIC nodes

The following sections contain documentation about every node in the GMIC group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.13.1 About G’MIC node

This documentation is for version 1.0 of About G’MIC.

Description

Wrapper for the G’MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.2 G'MIC 3d blocks node

This documentation is for version 1.0 of G'MIC 3d blocks.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Resolution / Resolution	Integer	32	
Smoothness / Smoothness	Double	0	
Elevation / Elevation	Double	4	
Size / Size	Double	1.5	
Angle / Angle	Double	30	
Tilt / Tilt	Double	60	
FOV / FOV	Double	45	
Centering (%) / Centering_	Double	x: 0.5 y: 0.5	
X-light / Xlight	Double	0	
Y-light / Ylight	Double	-50	
Z-light / Zlight	Double	-100	
Specular lightness / Specular_lightness	Double	0.5	
Specular shininess / Specular_shininess	Double	0.7	
Use light / Use_light	Boolean	On	
Antialiasing / Antialiasing	Boolean	On	
Outline color / Outline_color	Color	r: 0 g: 0 b: 0 a: 0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 188 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.3 G'MIC 3d colored object node

This documentation is for version 1.0 of G'MIC 3d colored object.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Type / Type	Choice	Box	Plane Box Pyramid Ellipsoid Torus Gyroid Weird Cup
Color / Color	Color	r: 0.501961 g: 0.501961 b: 0.501961 a: 0.501961	
Size-1 / Size1	Double	0.5	

Continued on next page

Table 189 – continued from previous page

Parameter / script name	Type	Default	Function
Size-2 / Size2	Double	0.5	
Size-3 / Size3	Double	0.5	
X-angle / Xangle	Double	57	
Y-angle / Yangle	Double	41	
Z-angle / Zangle	Double	21	
FOV / FOV	Double	45	
X-light / Xlight	Double	0	
Y-light / Ylight	Double	0	
Z-light / Zlight	Double	-100	
Specular lightness / Specular_lightness	Double	0.5	
Specular shininess / Specular_shininess	Double	0.7	
Rendering / Rendering	Choice	Gouraud	Dots Wireframe Flat Flat shaded Gouraud Phong
Antialiasing / Antialiasing	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 189 – continued from previous page

Parameter / script name	Type	Default	Function
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.4 G'MIC 3d elevation node

This documentation is for version 1.0 of G'MIC 3d elevation.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Factor / Factor	Double	100	
Smoothness / Smoothness	Double	1	
Width / Width	Integer	1024	
Height / Height	Integer	1024	
Size / Size	Double	0.8	
X-angle / Xangle	Double	25	
Y-angle / Yangle	Double	0	
Z-angle / Zangle	Double	21	
FOV / FOV	Double	45	
X-light / Xlight	Double	0	
Y-light / Ylight	Double	0	
Z-light / Zlight	Double	-100	
Specular lightness / Specular_lightness	Double	0.5	
Specular shininess / Specular_shininess	Double	0.7	

Continued on next page

Table 190 – continued from previous page

Parameter / script name	Type	Default	Function
Rendering / Rendering	Choice	Flat	Dots Wireframe Flat Flat shaded Gouraud Phong
Antialiasing / Antialiasing	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.5 G'MIC 3d extrusion node

This documentation is for version 1.0 of G'MIC 3d extrusion.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Depth / Depth	Double	10	
Resolution / Resolution	Integer	512	
Smoothness / Smoothness	Double	0.6	
Width / Width	Integer	1024	
Height / Height	Integer	1024	
Size / Size	Double	0.5	
X-angle / Xangle	Double	57	
Y-angle / Yangle	Double	41	
Z-angle / Zangle	Double	21	
FOV / FOV	Double	45	
X-light / Xlight	Double	0	
Y-light / Ylight	Double	0	
Z-light / Zlight	Double	-100	
Specular lightness / Specular_lightness	Double	0.5	
Specular shininess / Specular_shininess	Double	0.7	
Rendering / Rendering	Choice	Gouraud	Dots Wireframe Flat Flat shaded Gouraud Phong
Antialiasing / Antialiasing	Boolean	On	

Continued on next page

Table 191 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.6 G'MIC 3d image object node

This documentation is for version 1.0 of G'MIC 3d image object.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Type / Type	Choice	Cube	Plane Cube Pyramid Sphere Torus Gyroid Weird Cup Rubik
Width / Width	Integer	1024	
Height / Height	Integer	1024	
Size / Size	Double	0.5	
X-angle / Xangle	Double	57	
Y-angle / Yangle	Double	41	
Z-angle / Zangle	Double	21	
FOV / FOV	Double	45	
X-light / Xlight	Double	0	
Y-light / Ylight	Double	0	
Z-light / Zlight	Double	-100	
Specular lightness / Specular_lightness	Double	0.5	
Specular shininess / Specular_shininess	Double	0.7	
Rendering / Rendering	Choice	Gouraud	Dots Wireframe Flat Flat shaded Gouraud Phong
Antialiasing / Antialiasing	Boolean	On	

Continued on next page

Table 192 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.7 G'MIC 3d lathing node

This documentation is for version 1.0 of G'MIC 3d lathing.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Resolution / Resolution	Integer	76	
Smoothness / Smoothness	Double	2	
Max angle / Max_angle	Double	361	
Width / Width	Integer	1024	
Height / Height	Integer	1024	
Size / Size	Double	0.5	
X-angle / Xangle	Double	0	
Y-angle / Yangle	Double	0	
Z-angle / Zangle	Double	0	
FOV / FOV	Double	45	
X-light / Xlight	Double	0	
Y-light / Ylight	Double	0	
Z-light / Zlight	Double	-100	
Specular lightness / Specular_lightness	Double	0.5	
Specular shininess / Specular_shininess	Double	0.7	
Rendering / Rendering	Choice	Gouraud	Dots Wireframe Flat Flat shaded Gouraud Phong
Antialiasing / Antialiasing	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 193 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.8 G'MIC 3d random objects node

This documentation is for version 1.0 of G'MIC 3d random objects.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Type / Type	Choice	Cube	Cube Cone Cylinder Sphere Torus

Continued on next page

Table 194 – continued from previous page

Parameter / script name	Type	Default	Function
Density / Density	Integer	50	
Size / Size	Double	3	
Z-range / Zrange	Double	100	
FOV / FOV	Double	45	
X-light / Xlight	Double	0	
Y-light / Ylight	Double	0	
Z-light / Zlight	Double	-100	
Specular lightness / Specular_lightness	Double	0.5	
Specular shininess / Specular_shininess	Double	0.7	
Rendering / Rendering	Choice	Flat shaded	Dots Wireframe Flat Flat shaded Gouraud Phong
Opacity / Opacity	Double	1	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 194 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.9 G'MIC 3d text pointcloud node

This documentation is for version 1.0 of G'MIC 3d text pointcloud.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Frames / Frames	Integer	64	
1st text / p1st_text	String	G'MIC	
2nd text / p2nd_text	String	Rocks!	
Smoothness / Smoothness	Double	1	
Color / Color	Color	r: 0.784314 g: 0.862745 b: 1 a: 1	
Background / Background	Color	r: 1 g: 1 b: 1 a: 1	
X-shado / Xshado	Double	2	
Y-shado / Yshado	Double	2	
Shadow smoothness / Shadow_smoothness	Double	1	
Stationary frames / Stationary_frames	Integer	19	

Continued on next page

Table 195 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.10 G'MIC 3d tiles node

This documentation is for version 1.0 of G'MIC 3d tiles.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Inter-frames / Interframes	Integer	10	
X-tiles / Xtiles	Integer	8	
Y-tiles / Ytiles	Integer	8	
X-rotation / Xrotation	String	1	
Y-rotation / Yrotation	String	1	
Z-rotation / Zrotation	String	0	
Focale / Focale	Double	800	
Enable antialiasing / Enable_antialiasing	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 196 – continued from previous page

Parameter / script name	Type	Default	Function
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.11 G'MIC Abstraction node

This documentation is for version 1.0 of G'MIC Abstraction.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	1	
Levels / Levels	Integer	10	
Contrast / Contrast	Double	0.2	

Continued on next page

Table 197 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 197 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.12 G'MIC Add grain node

This documentation is for version 1.0 of G'MIC Add grain.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Preset / Preset	Choice	Orwo NP20-GDR	Orwo NP20-GDR Kodak TMAX 400 Kodak TMAX 3200 Kodak TRI-X 1600 Unknown
Blend mode / Blend_mode	Choice	Grain merge	Alpha Grain merge Hard light Overlay Soft light Grain only
Opacity / Opacity	Double	0.2	
Scale / Scale	Double	100	
Colored grain / Colored_grain	Boolean	Off	

Continued on next page

Table 198 – continued from previous page

Parameter / script name	Type	Default	Function
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview grain alone / Preview_grain_alone	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 198 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.13 G'MIC Align layers node

This documentation is for version 1.0 of G'MIC Align layers.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Alignment type / Alignment_type	Choice	Rigid	Rigid Non-rigid

Continued on next page

Table 199 – continued from previous page

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	0.7	
Scales / Scales	Choice	Auto	Auto 1 2 3 4 5 6 7 8
Revert layers / Revert_layers	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 199 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.14 G'MIC Array faded node

This documentation is for version 1.0 of G'MIC Array faded.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-tiles / Xtiles	Integer	2	
Y-tiles / Ytiles	Integer	2	
X-offset (%) / Xoffset_	Double	0	
Y-offset (%) / Yoffset_	Double	0	
Fade start (%) / Fade_start_	Double	80	
Fade end (%) / Fade_end_	Double	90	
Mirror / Mirror	Choice	None	None x-axis y-axis xy-axes

Continued on next page

Table 200 – continued from previous page

Parameter / script name	Type	Default	Function
Size / Size	Choice	Shrink	Shrink Expand Repeat [Memory consuming!]
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.15 G'MIC Array mirrored node

This documentation is for version 1.0 of G'MIC Array mirrored.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Iterations / Iterations	Integer	1	
X-offset (%) / Xoffset_	Double	0	
Y-offset (%) / Yoffset_	Double	0	
Array mode / Array_mode	Choice	xy-axes	x-axis y-axis xy-axes 2xy-axes
Initialization / Initialization	Choice	Original	Original Mirror X Mirror Y Rotate 90 deg. Rotate 180 deg. Rotate 270 deg.
Expand size / Expand_size	Boolean	Off	
Crop (%) / Crop_	Integer	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 201 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.16 G'MIC Array random colors node

This documentation is for version 1.0 of G'MIC Array random colors.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Source X-tiles / Source_Xtiles	Integer	5	
Source Y-tiles / Source_Ytiles	Integer	5	
Destination X-tiles / Destination_Xtiles	Integer	7	
Destination Y-tiles / Destination_Ytiles	Integer	7	
X-tiles / Xtiles	Integer	5	

Continued on next page

Table 202 – continued from previous page

Parameter / script name	Type	Default	Function
Y-tiles / Ytiles	Integer	5	
Opacity / Opacity	Double	0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.17 G'MIC Array regular node

This documentation is for version 1.0 of G'MIC Array regular.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-tiles / Xtiles	Integer	2	
Y-tiles / Ytiles	Integer	2	
X-offset (%) / Xoffset_	Double	0	
Y-offset (%) / Yoffset_	Double	0	
Mirror / Mirror	Choice	None	None x-axis y-axis xy-axes
Size / Size	Choice	Shrink	Shrink Expand Repeat [Memory consuming!]
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 203 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.18 G'MIC Ascii art node

This documentation is for version 1.0 of G'MIC Ascii art.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Charset / Charset	Choice	Ascii	Custom Binary digits Digits Lowercase letters Uppercase letters Ascii Card suits Math symbols
Custom dictionary / Custom_dictionary	String	.oO0	
Analysis scale / Analysis_scale	Integer	16	
Analysis smoothness / Analysis_smoothness	Double	15	
Synthesis scale / Synthesis_scale	Integer	16	

Continued on next page

Table 204 – continued from previous page

Parameter / script name	Type	Default	Function
Result type / Result_type	Choice	Colored on black	White on black Black on white Colored on black Colored on transparent
Gamma / Gamma	Double	0	
Smoothness / Smoothness	Double	0.2	
Colors / Colors	Choice	Full colors	Full colors 2 colors 3 colors 4 colors 8 colors 12 colors 16 colors Grayscale 2 grays 3 grays 4 grays 8 grays 12 grays 16 grays
Output ascii file / Output_ascii_file	Boolean	Off	
Output folder / Output_folder	N/A		
Output filename / Output_filename	String	gmic_asciiart.txt	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 204 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.19 G'MIC B&W films node

This documentation is for version 1.0 of G'MIC B&W films.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Preset / Preset	Choice	None	<p>None</p> <p>Agfa APX 100</p> <p>Agfa APX 25</p> <p>Fuji Neopan 1600</p> <p>Fuji Neopan Acros 100</p> <p>Ilford Delta 100</p> <p>Ilford Delta 3200</p> <p>Ilford Delta 400</p> <p>Ilford FP4 Plus 125</p> <p>Ilford HP5 Plus 400</p> <p>Ilford HPS 800</p> <p>Ilford Pan F Plus 50</p> <p>Ilford XP2</p> <p>Kodak BW 400 CN</p> <p>Kodak HIE (HS Infra)</p> <p>Kodak T-Max 100</p> <p>Kodak T-Max 3200</p> <p>Kodak T-Max 400</p> <p>Kodak Tri-X 400</p> <p>Polaroid 664</p> <p>Polaroid 667</p> <p>Polaroid 672</p> <p>Rollei IR 400</p> <p>Rollei Ortho 25</p> <p>Rollei Retro 100 Tonal</p> <p>Rollei Retro 80s</p>
Strength (%) / Strength_	Double	100	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Normalize colors / Normalize_colors	Choice	None	<p>None</p> <p>Pre-process</p> <p>Post-process</p> <p>Both</p>
Pseudo-gray dithering / Pseudogray_dithering	Integer	0	

Continued on next page

Table 205 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 205 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.20 G'MIC B&W films collage node

This documentation is for version 1.0 of G'MIC B&W films collage.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Image size / Image_size	Double	512	
Columns for collage / Columns_for_collage	Integer	4	
Label size / Label_size	Integer	16	
Output as / Output_as	Choice	Table	Table Multiple layers

Continued on next page

Table 206 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.21 G'MIC B&W stencil node

This documentation is for version 1.0 of G'MIC B&W stencil.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Threshold / Threshold	Double	10	
Smoothness / Smoothness	Double	10	
Hue / Hue	Double	0	
Saturation / Saturation	Double	0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 207 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.22 G'MIC Ball node

This documentation is for version 1.0 of G'MIC Ball.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Radius / Radius	Integer	128	
Specular light / Specular_light	Double	0.8	
Specular size / Specular_size	Double	1	
Shadow / Shadow	Double	1.5	
Color / Color	Color	r: 1 g: 0 b: 1 a: 1	

Continued on next page

Table 208 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.23 G'MIC Bandpass node

This documentation is for version 1.0 of G'MIC Bandpass.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Low frequency / Low_frequency	Double	0	
High frequency / High_frequency	Double	100	
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Value action / Value_action	Choice	Normalize	<p> None Cut Normalize </p>

Continued on next page

Table 209 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 209 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.24 G'MIC Barnsley fern node

This documentation is for version 1.0 of G'MIC Barnsley fern.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Type / Type	Choice	Asplenium adiantum-nigrum	Asplenium adiantum-nigrum Thelypteridaceae
Density (%) / Density_	Double	100	
Angle / Angle	Double	30	
Opacity (%) / Opacity_	Double	40	
Color / Color	Color	r: 0.0392157 g: 0.698039 b: 0 a: 0	
Add as a new layer / Add_as_a_new_layer	Boolean	On	

Continued on next page

Table 210 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.25 G'MIC Basic adjustments node

This documentation is for version 1.0 of G'MIC Basic adjustments.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 211 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.26 G'MIC Black & white node

This documentation is for version 1.0 of G'MIC Black & white.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Red level / Red_level	Double	0.299	
Red smoothness / Red_smoothness	Double	0	
Green level / Green_level	Double	0.587	
Green smoothness / Green_smoothness	Double	0	

Continued on next page

Table 212 – continued from previous page

Parameter / script name	Type	Default	Function
Blue level / Blue_level	Double	0.114	
Blue smoothness / Blue_smoothness	Double	0	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Grain (shadows) / Grain_shadows	Double	0	
Grain (midtones) / Grain_midtones	Double	0	
Grain (highlights) / Grain_highlights	Double	0	
Grain tone fading / Grain_tone_fading	Double	2	
Grain scale / Grain_scale	Double	0	
Grain type / Grain_type	Choice	Gaussian	Gaussian Uniform Salt and pepper Poisson
Local contrast / Local_contrast	Double	0	
Radius / Radius	Integer	16	
Contrast smoothness / Contrast_smoothness	Double	4	
Pseudo-gray dithering / Pseudogray_dithering	Integer	0	
Use maximum tones / Use_maximum_tones	Boolean	Off	

Continued on next page

Table 212 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 212 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.27 G'MIC Blend average all node

This documentation is for version 1.0 of G'MIC Blend average all.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Colorspace / Colorspace	Choice	sRGB	sRGB Linear RGB Lab

Continued on next page

Table 213 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.28 G'MIC Blend edges node

This documentation is for version 1.0 of G'MIC Blend edges.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Opacity / Opacity	Double	1	
Smoothness / Smoothness	Double	0.8	
Revert layers / Revert_layers	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.29 G'MIC Blend fade node

This documentation is for version 1.0 of G'MIC Blend fade.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Preset / Preset	Choice	Linear	Custom Linear Circular Wave Keftales
Offset / Offset	Double	0	
Thinness / Thinness	Double	0	
Sharpness / Sharpness	Double	5	
Sharpest / Sharpest	Boolean	Off	
Revert layers / Revert_layers	Boolean	Off	
Colorspace / Colorspace	Choice	sRGB	sRGB Linear RGB Lab
1st parameter / p1st_parameter	Double	0	
2nd parameter / p2nd_parameter	Double	0	
3rd parameter / p3rd_parameter	Double	0	
Formula / Formula	String	$\cos(4*\pi*x/w)$ $*$ $\sin(4*\pi*y/h)$	

Continued on next page

Table 215 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.30 G'MIC Blend median node

This documentation is for version 1.0 of G'MIC Blend median.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Colorspace / Colorspace	Choice	sRGB	sRGB Linear RGB Lab
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.31 G'MIC Blend seamless node

This documentation is for version 1.0 of G'MIC Blend seamless.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Mixed mode / Mixed_mode	Boolean	Off	
Inner fading / Inner_fading	Double	0	
Outer fading / Outer_fading	Double	25	
Colorspace / Colorspace	Choice	sRGB	sRGB Linear RGB Lab
Output as separate layers / Output_as_separate_layers	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 217 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.32 G'MIC Blend standard node

This documentation is for version 1.0 of G'MIC Blend standard.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Continued on next page

Table 218 – continued from previous page

Parameter / script name	Type	Default	Function
Parameter / script name	Type	Default	Function
Mode / Mode	Choice	Custom formula	<p> Add Alpha And Average Blue Burn Custom formula Darken Difference Divide Dodge Edges Exclusion Freeze Grain extract Grain merge Green Hard light Hard mix Hue Interpolation Lighten Lightness Linear burn Linear light Luminance Multiply Negation Or Overlay Pin light Red Reflect Saturation Shape area max Shape area max0 Shape area min Shape area min0 Shape average Shape average0 Shape median Shape median0 Shape min Shape min0 Shape max Shape max0 </p>
2.13. GMIC nodes			<p> Soft burn Soft dodge Soft light Screen </p>

Table 218 – continued from previous page

Parameter / script name	Type	Default	Function
Process as / Process_as	Choice	Two-by-two	Two-by-two Upper layer is the top layer for all blends Lower layer is the bottom layer for all blends
Opacity (%) / Opacity_	Double	100	
Preview all outputs / Preview_all_outputs	Boolean	On	
Custom formula / Custom_formula	String	1/2 - 1/4*cos(pi*a) - 1/4*cos(pi*b)	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.33 G'MIC Blur angular node

This documentation is for version 1.0 of G'MIC Blur angular.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	2	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Sharpness / Sharpness	Double	0	
Preview guides / Preview_guides	Boolean	On	

Continued on next page

Table 219 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	Linear RGB [all]	All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas]
Value action / Value_action	Choice	None	None Cut Normalize

Continued on next page

Table 219 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.34 G'MIC Blur bloom node

This documentation is for version 1.0 of G'MIC Blur bloom.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	1	
Ratio / Ratio	Double	2	
Iterations / Iterations	Integer	5	
Operator / Operator	Choice	Add	Add Max Min
Kernel / Kernel	Choice	Quasi-gaussian	Quasi-gaussian Gaussian Box Triangle Quadratic
Normalize scales / Normalize_scales	Boolean	Off	
Anisotropy / Anisotropy	Double	0	
Angle / Angle	Double	0	

Continued on next page

Table 220 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	Linear RGB [all]	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 220 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 220 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.35 G'MIC Blur depth-of-field node

This documentation is for version 1.0 of G'MIC Blur depth-of-field.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Blur amplitude / Blur_amplitude	Double	3	
Blur precision / Blur_precision	Integer	16	
Depth-of-field type / Depthoffield_type	Choice	Gaussian	Gaussian User-defined (bottom layer)
Invert blur / Invert_blur	Boolean	Off	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
First radius / First_radius	Double	30	
Second radius / Second_radius	Double	30	
Angle / Angle	Double	0	
Sharpness / Sharpness	Double	1	

Continued on next page

Table 221 – continued from previous page

Parameter / script name	Type	Default	Function
Preview guides / Preview_guides	Boolean	On	
Gamma / Gamma	Double	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.36 G'MIC Blur gaussian node

This documentation is for version 1.0 of G'MIC Blur gaussian.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
XY-amplitude / XYamplitude	Double	3	
X-amplitude / Xamplitude	Double	0	
Y-amplitude / Yamplitude	Double	0	
Boundary / Boundary	Choice	Nearest	Black Nearest

Continued on next page

Table 222 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Value action / Value_action	Choice	None	<p> None Cut Normalize </p>

Continued on next page

Table 222 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 222 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.37 G'MIC Blur glow node

This documentation is for version 1.0 of G'MIC Blur glow.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	6	

Continued on next page

Table 223 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	Linear RGB [all]	All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas]
Value action / Value_action	Choice	None	None Cut Normalize

Continued on next page

Table 223 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 223 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.38 G'MIC Blur linear node

This documentation is for version 1.0 of G'MIC Blur linear.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Tangent radius / Tangent_radius	Double	10	
Orthogonal radius / Orthogonal_radius	Double	0.5	
Angle / Angle	Double	0	
Sharpness / Sharpness	Double	0	
Boundary / Boundary	Choice	Nearest	Black Nearest

Continued on next page

Table 224 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	Linear RGB [all]	All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas]
Value action / Value_action	Choice	None	None Cut Normalize

Continued on next page

Table 224 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 224 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.39 G'MIC Blur radial node

This documentation is for version 1.0 of G'MIC Blur radial.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	3	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Sharpness / Sharpness	Double	0	
Preview guides / Preview_guides	Boolean	On	

Continued on next page

Table 225 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	Linear RGB [all]	All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas]
Value action / Value_action	Choice	None	None Cut Normalize

Continued on next page

Table 225 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.40 G'MIC Bokeh node

This documentation is for version 1.0 of G'MIC Bokeh.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of scales / Number_of_scales	Integer	3	
Shape / Shape	Choice	Circular	Triangle Square Diamond Pentagon Hexagon Octagon Decagon Star Circular
Random seed / Random_seed	Integer	0	
Density / Density	Integer	30	
Radius (%) / Radius_	Double	8	
Outline (%) / Outline_	Double	4	
Inner shade / Inner_shade	Double	0.3	
Smoothness / Smoothness	Double	0.2	
Color / Color	Color	r: 0.823529 g: 0.823529 b: 0.313726 a: 0.313726	
Color dispersion / Color_dispersion	Double	0.7	
Density_2 / Density_2	Integer	30	
Radius (%)_2 / Radius__2	Double	20	
Outline (%)_2 / Outline__2	Double	20	
Inner shade_2 / Inner_shade_2	Double	1	
Smoothness_2 / Smoothness_2	Double	2	

Continued on next page

Table 226 – continued from previous page

Parameter / script name	Type	Default	Function
Color_2 / Color_2	Color	r: 0.666667 g: 0.509804 b: 0.0784314 a: 0.0784314	
Color dispersion_2 / Color_dispersion_2	Double	0.15	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 226 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.41 G'MIC Boost chromaticity node

This documentation is for version 1.0 of G'MIC Boost chromaticity.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude (%) / Amplitude_	Double	50	
Color space / Color_space	Choice	YCbCr (dis- tinct)	YCbCr (distinct) YCbCr (mixed) Lab (distinct) Lab (mixed)

Continued on next page

Table 227 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 227 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.42 G'MIC Boost-fade node

This documentation is for version 1.0 of G'MIC Boost-fade.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	5	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 228 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.43 G'MIC Box fitting node

This documentation is for version 1.0 of G'MIC Box fitting.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Starting pattern / Starting_pattern	Choice	Red-Green	Red-Green Blue-Green Green-Red Green-Blue
Keep colors / Keep_colors	Boolean	On	
Minimal size / Minimal_size	Integer	3	
Maximal size / Maximal_size	Integer	0	
Initial density / Initial_density	Double	0.1	
Transparency / Transparency	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 229 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.44 G'MIC Brushify node

This documentation is for version 1.0 of G'MIC Brushify.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Shape / Shape	Choice	Ellipse	Bottom layer Top layer Rectangle Diamond Pentagon Hexagon Octagon Ellipse Gaussian Star Heart
Ratio / Ratio	Double	0.25	
Number of sizes / Number_of_sizes	Integer	4	
Maximal size / Maximal_size	Integer	64	
Minimal size (%) / Minimal_size_	Double	25	

Continued on next page

Table 230 – continued from previous page

Parameter / script name	Type	Default	Function
Number of orientations / Number_of_orientations	Integer	12	
Fuzzyness / Fuzzyness	Double	0	
Smoothness / Smoothness	Double	2	
Light type / Light_type	Choice	Full	None Flat Darken Lighten Full
Light strength / Light_strength	Double	0.2	
Opacity / Opacity	Double	0.5	
Density (%) / Density_	Double	30	
Contour coherence / Contour_coherence	Double	1	
Orientation coherence / Orientation_coherence	Double	1	
Gradient smoothness / Gradient_smoothness	Double	1	
Structure smoothness / Structure_smoothness	Double	5	
Primary angle / Primary_angle	Double	0	
Angle dispersion / Angle_dispersion	Double	0.2	
Preview brush / Preview_brush	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 230 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.45 G'MIC Burn node

This documentation is for version 1.0 of G'MIC Burn.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	0.5	
Scale / Scale	Double	30	
Smoothness / Smoothness	Double	1	

Continued on next page

Table 231 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Value action / Value_action	Choice	None	<p> None Cut Normalize </p>

Continued on next page

Table 231 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 231 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.46 G'MIC Camouflage node

This documentation is for version 1.0 of G'MIC Camouflage.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Scale / Scale	Integer	9	
Levels / Levels	Integer	12	
Coherence / Coherence	Double	100	
Color 1 / Color_1	Color	r: 0.117647 g: 0.180392 b: 0.129412 a: 0.129412	
Color 2 / Color_2	Color	r: 0.294118 g: 0.352941 b: 0.254902 a: 0.254902	

Continued on next page

Table 232 – continued from previous page

Parameter / script name	Type	Default	Function
Color 3 / <code>Color_3</code>	Color	r: 0.701961 g: 0.741176 b: 0.458824 a: 0.458824	
Color 4 / <code>Color_4</code>	Color	r: 1 g: 0.964706 b: 0.619608 a: 0.619608	
Output Layer / <code>Output_Layer</code>	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / <code>Resize_Mode</code>	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / <code>Ignore_Alpha</code>	Boolean	Off	
Log Verbosity / <code>Log_Verbosity</code>	Choice	Off	Off Level 1 Level 2 Level 3

2.13.47 G'MIC Canvas node

This documentation is for version 1.0 of G'MIC Canvas.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	70	
Angle / Angle	Double	45	
Sharpness / Sharpness	Double	400	
Activate second direction / Activate_second_direction	Boolean	On	
Amplitude_2 / Amplitude_2	Double	70	
Angle_2 / Angle_2	Double	135	
Sharpness_2 / Sharpness_2	Double	400	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 233 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.48 G'MIC Cartesian transform node

This documentation is for version 1.0 of G'MIC Cartesian transform.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-warping / Xwarping	String	$(w+h)/20 * \cos(y*20/h)$	
Y-warping / Ywarping	String	$(w+h)/20 * \sin(x*20/w)$	
Relative warping / Relative_warping	Boolean	On	
Interpolation / Interpolation	Choice	Linear	Nearest neighbor Linear
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 234 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.49 G'MIC Cartoon node

This documentation is for version 1.0 of G'MIC Cartoon.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	3	
Sharpening / Sharpening	Double	200	
Edge threshold / Edge_threshold	Double	20	
Edge thickness / Edge_thickness	Double	0.25	
Color strength / Color_strength	Double	1.5	
Color quantization / Color_quantization	Integer	8	

Continued on next page

Table 235 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 235 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.50 G'MIC Channel processing node

This documentation is for version 1.0 of G'MIC Channel processing.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Smoothness / Smoothness	Double	0	
Value action / Value_action	Choice	None	None Cut Cut & Normalize Normalize Threshold
Low value / Low_value	Double	0	
High value / High_value	Double	100	
Quantization / Quantization	Integer	256	

Continued on next page

Table 236 – continued from previous page

Parameter / script name	Type	Default	Function
Equalization / Equalization	Boolean	Off	
Negation / Negation	Boolean	Off	
Tones range / Tones_range	Choice	All tones	All tones Shadows Mid-tones Highlights
Tones smoothness / Tones_smoothness	Double	2	

Continued on next page

Table 236 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 236 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 236 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.51 G'MIC Channels to layers node

This documentation is for version 1.0 of G'MIC Channels to layers.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Colorspace / Colorspace	Choice	RGB	RGB CMY HSV

Continued on next page

Table 237 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.52 G'MIC Charcoal node

This documentation is for version 1.0 of G'MIC Charcoal.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Granularity / Granularity	Integer	65	
Lowlights crossover point / Lowlights_crossover_point	Integer	70	
Highlights crossover point / Highlights_crossover_point	Integer	170	
Boost contrast / Boost_contrast	Boolean	Off	
Resize image for optimum effect / Resize_image_for_optimum_effect	Boolean	On	
Add chalk highlights / Add_chalk_highlights	Boolean	Off	
Minimal highlights / Minimal_highlights	Integer	50	
Maximal highlights / Maximal_highlights	Integer	70	
Background color / Background_color	Color	r: 1 g: 1 b: 1 a: 1	
Foreground color / Foreground_color	Color	r: 0 g: 0 b: 0 a: 0	
Invert background/foreground / Invert_backgroundforeground	Boolean	Off	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 238 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.53 G'MIC Chessboard node

This documentation is for version 1.0 of G'MIC Chessboard.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
First size / First_size	Integer	64	
Second size / Second_size	Integer	64	
First offset / First_offset	Integer	0	
Second offset / Second_offset	Integer	0	
Angle / Angle	Double	0	
Opacity / Opacity	Double	0.5	
First color / First_color	Color	r: 0 g: 0 b: 0 a: 0	
Second color / Second_color	Color	r: 1 g: 1 b: 1 a: 1	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 239 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.54 G'MIC Chromatic aberrations node

This documentation is for version 1.0 of G'MIC Chromatic aberrations.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Primary color / Primary_color	Color	r: 1 g: 0 b: 0 a: 0	
X-shift / Xshift	Double	2	
Y-shift / Yshift	Double	2	
Secondary color / Secondary_color	Color	r: 0 g: 1 b: 0 a: 0	

Continued on next page

Table 240 – continued from previous page

Parameter / script name	Type	Default	Function
X-shift (px) / Xshift_px	Double	0	
Y-shift (px) / Yshift_px	Double	0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 240 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.55 G'MIC Circle abstraction node

This documentation is for version 1.0 of G'MIC Circle abstraction.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of colors / Number_of_colors	Integer	8	
Density / Density	Integer	5	
Opacity / Opacity	Double	0.8	
Smoothness / Smoothness	Double	0	
Filled circles / Filled_circles	Boolean	On	
Fill transparent holes / Fill_transparent_holes	Boolean	On	
Normalize colors / Normalize_colors	Boolean	On	

Continued on next page

Table 241 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 241 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.56 G'MIC Circle art node

This documentation is for version 1.0 of G'MIC Circle art.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Type / Type	Choice	Lissajous spiral	Random Lissajous spiral
Density / Density	Double	15	
Radius / Radius	Double	0.5	
Modulo / Modulo	Integer	8	
Anti-aliasing / Antialiasing	Boolean	On	
Random colors / Random_colors	Boolean	On	
Curve length / Curve_length	Double	15	
Curve angle / Curve_angle	Double	0	
Minimal radius / Minimal_radius	Double	0	
Maximal radius / Maximal_radius	Double	0.5	
X-dispersion / Xdispersion	Double	1	

Continued on next page

Table 242 – continued from previous page

Parameter / script name	Type	Default	Function
Y-dispersion / Ydispersion	Double	1	
X-factor / Xfactor	Integer	1	
Y-factor / Yfactor	Integer	1	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.57 G'MIC Circle transform node

This documentation is for version 1.0 of G'MIC Circle transform.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Radius / Radius	Double	x: 0.75 y: 0.5	
X-scale / Xscale	Double	-2	
Y-scale / Yscale	Double	-2	
Symmetry / Symmetry	Choice	None	None Inside Outside
Interpolation / Interpolation	Choice	Linear	Nearest neighbor Linear
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror
Preview reference circle / Preview_reference_circle	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 243 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.58 G'MIC Color balance node

This documentation is for version 1.0 of G'MIC Color balance.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Neutral color / Neutral_color	Color	r: 0.501961 g: 0.501961 b: 0.501961 a: 0.501961	

Continued on next page

Table 244 – continued from previous page

Parameter / script name	Type	Default	Function
Stretch colors / Stretch_colors	Boolean	On	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 244 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.59 G'MIC Color blindness node

This documentation is for version 1.0 of G'MIC Color blindness.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Blindness type / Blindness_type	Choice	Protanopia	Protanopia Protanomaly Deutanopia Deuteranomaly Tritanopia Tritanomaly Achromatopsia Achromatomaly

Continued on next page

Table 245 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 245 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.60 G'MIC Colorize lineart auto-fill node

This documentation is for version 1.0 of G'MIC Colorize lineart auto-fill.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Contour threshold (%) / Contour_threshold_	Double	90	
Contour normalization / Contour_normalization	Boolean	On	
Minimal region area / Minimal_region_area	Integer	8	
Tolerance to gaps / Tolerance_to_gaps	Integer	0	
Preview type / Preview_type	Choice	Lineart + colors	Lineart + colors Colors only

Continued on next page

Table 246 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.61 G'MIC Colorize lineart propagation node

This documentation is for version 1.0 of G'MIC Colorize lineart propagation.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Input layers / Input_layers	Choice	Color spots + lineart	Color spots + lineart Lineart + color spots Color spots + extrapolated colors + lineart Lineart + color spots + extrapolated colors
Output layers / Output_layers	Choice	Extrapolated colors + lineart	Single (merged) Extrapolated colors + lineart Lineart + extrapolated colors Color spots + extrapolated colors + lineart Lineart + color spots + extrapolated colors
Extrapolate colors as / Extrapolate_colors_as	Choice	One layer	One layer Two layers Three layers Four layers Five layers Six layers Seven layers Eight layers Nine layers Ten layers One layer per single color One layer per single region
Smoothness / Smoothness	Double	0.05	

Continued on next page

Table 247 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.62 G'MIC Colorize lineart smart coloring node

This documentation is for version 1.0 of G'MIC Colorize lineart smart coloring.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Colorize mode / Colorize_mode	Choice	Generate random-colors layer	Generate random-colors layer Extrapolate color spots on transparent top layer Auto-clean bottom color layer
Contour detection (%) / Contour_detection_	Double	95	
Contour erosion / Contour_erosion	Integer	0	
Discard contour guides / Discard_contour_guides	Boolean	Off	
Output region delimiters / Output_region_delimiters	Boolean	Off	
Make hue depends on region size / Make_hue_depends_on_region_size	Double	1	
Maximal color saturation / Maximal_color_saturation	Integer	24	
Minimal color intensity / Minimal_color_intensity	Integer	200	
Color shading (%) / Color_shading_	Integer	0	
End point rate (%) / End_point_rate_	Double	85	
End point connectivity / End_point_connectivity	Integer	2	
Spline max length (px) / Spline_max_length_px	Double	60	
Segment max length (px) / Segment_max_length_px	Double	20	
Spline max angle (deg) / Spline_max_angle_deg	Double	90	
Spline roundness / Spline_roundness	Double	1	
Minimal region area / Minimal_region_area	Double	10	
Allow self intersections / Allow_self_intersections	Boolean	On	
Detection rate / Detection_rate	Double	0	

Continued on next page

Table 248 – continued from previous page

Parameter / script name	Type	Default	Function
Max segment curvature (deg) / Max_segment_curvature_deg	Double	20	
Max grouping distance (px) / Max_grouping_distance_px	Double	64	
Max grouping angle (deg) / Max_grouping_angle_deg	Double	7.5	
Max grouping ratio / Max_grouping_ratio	Double	0.5	
Preview type / Preview_type	Choice	Colored geometry	Colored geometry Colored regions Colored lineart
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 248 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.63 G'MIC Colorize photographs node

This documentation is for version 1.0 of G'MIC Colorize photographs.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Integer	2	
Anisotropy / Anisotropy	Double	0.2	
Output mode / Output_mode	Choice	Merge bright- ness/colors	Merge brightness/colors Split brightness/colors

Continued on next page

Table 249 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.64 G'MIC Colorize with colormap node

This documentation is for version 1.0 of G'MIC Colorize with colormap.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Normalize input / Normalize_input	Boolean	Off	
Gradient preset / Gradient_preset	Choice	User-defined	User-defined Black to white White to black Sepia Solarize
Interpolation type / Interpolation_type	Choice	Linear	Nearest Linear Cubic Lanczos
Preserve initial brightness / Preserve_initial_brightness	Boolean	Off	
Number of tones / Number_of_tones	Integer	5	
1st tone / p1st_tone	Color	r: 0 g: 0 b: 0 a: 0	
2nd tone / p2nd_tone	Color	r: 0.168627 g: 0.0980392 b: 0.215686 a: 0.215686	
3rd tone / p3rd_tone	Color	r: 0.619608 g: 0.537255 b: 0.741176 a: 0.741176	

Continued on next page

Table 250 – continued from previous page

Parameter / script name	Type	Default	Function
4th tone / p4th_tone	Color	r: 0.878431 g: 0.74902 b: 0.894118 a: 0.894118	
5th tone / p5th_tone	Color	r: 1 g: 1 b: 1 a: 1	
6th tone / p6th_tone	Color	r: 1 g: 1 b: 1 a: 1	
7th tone / p7th_tone	Color	r: 1 g: 1 b: 1 a: 1	
8th tone / p8th_tone	Color	r: 1 g: 1 b: 1 a: 1	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerred Checkerred inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 250 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.65 G'MIC Colormap node

This documentation is for version 1.0 of G'MIC Colormap.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Colormap / Colormap	Choice	Custom	Adaptive Custom Standard (256) HSV (256) Lines (256) Hot (256) Cool (256) Jet (256) Flag (256) Cube (256)
Dithering / Dithering	Double	1	
Number of tones / Number_of_tones	Integer	32	
Number of colors / Number_of_colors	Integer	8	
1st color / p1st_color	Color	r: 0 g: 0 b: 0 a: 0	
2nd color / p2nd_color	Color	r: 1 g: 1 b: 1 a: 1	
3rd color / p3rd_color	Color	r: 1 g: 0 b: 0 a: 0	
4th color / p4th_color	Color	r: 0 g: 1 b: 0 a: 0	
5th color / p5th_color	Color	r: 0 g: 0 b: 1 a: 1	
6th color / p6th_color	Color	r: 1 g: 1 b: 0 a: 0	
7th color / p7th_color	Color	r: 1 g: 0 b: 1 a: 1	
8th color / p8th_color	Color	r: 0 g: 1 b: 1 a: 1	

Continued on next page

Table 251 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 251 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.66 G'MIC Colors to layers node

This documentation is for version 1.0 of G'MIC Colors to layers.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Color tolerance / Color_tolerance	Double	50	
Maximum number of output layers / Maximum_number_of_output_layers	Integer	16	
Minimal area (%) / Minimal_area_	Double	1	
Autocrop output layers / Autocrop_output_layers	Boolean	Off	

Continued on next page

Table 252 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.67 G'MIC Conformal maps node

This documentation is for version 1.0 of G'MIC Conformal maps.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Mapping / Mapping	Choice	Dipole: $1/(4*z^2-1)$	Custom formula z (z-1)/(z+1) cos(z) sin(z) tan(z) exp(z) log(z) Dipole: $1/(4*z^2-1)$ Star: $-5*(z^3/3-z/4)/2$
Exponent (real) / Exponent_real	Double	1	
Exponent (imaginary) / Exponent_imaginary	Double	0	
Custom formula / Custom_formula	String	$((1.1 + i*z/6)/(1.04 - i*z/6))^6$	
Zoom / Zoom	Double	0	
Angle / Angle	Double	0	
Aspect ratio / Aspect_ratio	Double	0	
X-shift / Xshift	Double	0	
Y-shift / Yshift	Double	0	
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror
Anti-aliasing / Antialiasing	Integer	0	
Specify different output size / Specify_different_output_size	Boolean	Off	
Output width / Output_width	String	1024	
Output height / Output_height	String	1024	

Continued on next page

Table 253 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.68 G'MIC Contrast swiss mask node

This documentation is for version 1.0 of G'MIC Contrast swiss mask.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Blur the mask / Blur_the_mask	Double	2	
Skip to use the mask to boost / Skip_to_use_the_mask_to_boost	Boolean	Off	
Intensity / Intensity	Double	1	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.69 G'MIC Convolve node

This documentation is for version 1.0 of G'MIC Convolve.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Kernel / Kernel	Choice	Custom	Custom Average 3x3 Average 5x5 Average 7x7 Average 9x9 Prewitt-X Prewitt-Y Sobel-X Sobel-Y Rotinv-X Rotinv-Y Laplacian Robert Cross 1 Robert Cross 2 Impulses 5x5 Impulses 7x7 Impulses 9x9
Boundary / Boundary	Choice	Neumann	Dirichlet Neumann
Custom kernel / Custom_kernel	String	0,1,0;1,-4,1;0,1,0	
Value range / Value_range	Choice	Normalize	Cut Normalize
Kernel multiplier / Kernel_multiplier	Double	1	

Continued on next page

Table 255 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 255 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 255 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.70 G'MIC Cracks node

This documentation is for version 1.0 of G'MIC Cracks.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	20	
Fibrousness / Fibrousness	Double	3	
Emboss / Emboss	Double	0.6	
Density (%) / Density_	Double	30	
Relief / Relief	Boolean	On	
Color / Color	Color	r: 1 g: 1 b: 1 a: 1	

Continued on next page

Table 256 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 256 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 256 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.71 G'MIC Crease node

This documentation is for version 1.0 of G'MIC Crease.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	30	
Frequency (%) / Frequency_	Double	10	
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror

Continued on next page

Table 257 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.72 G'MIC Crystal node

This documentation is for version 1.0 of G'MIC Crystal.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Double	50	
Smoothness / Smoothness	Double	0.2	
Edges / Edges	Double	20	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 258 – continued from previous page

Parameter / script name	Type	Default	Function
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.73 G'MIC Crystal background node

This documentation is for version 1.0 of G'MIC Crystal background.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Iterations / Iterations	Integer	10	
Density (%) / Density_	Double	25	
Random seed / Random_seed	Integer	0	
Opacity (%) / Opacity_	Double	100	
Color / Color	Boolean	On	

Continued on next page

Table 259 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.74 G'MIC Cubism node

This documentation is for version 1.0 of G'MIC Cubism.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Iterations / Iterations	Integer	2	
Density / Density	Double	50	
Thickness / Thickness	Double	10	
Angle / Angle	Double	90	
Opacity / Opacity	Double	0.7	
Smoothness / Smoothness	Double	0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 260 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.75 G'MIC Cupid node

This documentation is for version 1.0 of G'MIC Cupid.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Size (%) / Size_	Double	75	
Smoothness / Smoothness	Double	0	
Color / Color	Color	r: 1 g: 0 b: 0 a: 0	
Antialiasing / Antialiasing	Boolean	On	

Continued on next page

Table 261 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.76 G'MIC Curvature node

This documentation is for version 1.0 of G'MIC Curvature.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	2	
Min threshold / Min_threshold	Double	0	
Max threshold / Max_threshold	Double	100	
Absolute value / Absolute_value	Boolean	Off	
Negative colors / Negative_colors	Boolean	Off	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 262 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.77 G'MIC Customize CLUT node

This documentation is for version 1.0 of G'MIC Customize CLUT.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Keypoint influence (%) / Keypoint_influence_	Double	100	

Continued on next page

Table 263 – continued from previous page

Parameter / script name	Type	Default	Function
Lock uniform sampling / Lock_uniform_sampling	Choice	None	None 8 keypoints (RGB corners) 27 keypoints 64 keypoints 125 keypoints 216 keypoints 343 keypoints
Spatial regularization / Spatial_regularization	Integer	10	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Post-normalize / Postnormalize	Boolean	Off	
Output corresponding CLUT / Output_corresponding_CLUT	Choice	Disable	Disable 512x512 layer 4096x4096 layer
Preview type / Preview_type	Choice	3D CLUT (fast)	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate horizontal Duplicate vertical HaldCLUT 3D CLUT (fast) 3D CLUT (precise)
CLUT opacity / CLUT_opacity	Double	0.5	
Action #1 / Action_1	Choice	Lock source	Ignore Lock source Replace source by target

Continued on next page

Table 263 – continued from previous page

Parameter / script name	Type	Default	Function
Source color #1 / Source_color_1	Color	r: 0 g: 0 b: 0 a: 0	
Target color #1 / Target_color_1	Color	r: 0 g: 0 b: 0 a: 0	
Action #2 / Action_2	Choice	Lock source	Ignore Lock source Replace source by target
Source color #2 / Source_color_2	Color	r: 1 g: 1 b: 1 a: 1	
Target color #2 / Target_color_2	Color	r: 1 g: 0.768628 b: 0.501961 a: 0.501961	
Action #3 / Action_3	Choice	Ignore	Ignore Lock source Replace source by target
Source color #3 / Source_color_3	Color	r: 0 g: 0 b: 0 a: 0	
Target color #3 / Target_color_3	Color	r: 0 g: 0 b: 0 a: 0	
Action #4 / Action_4	Choice	Ignore	Ignore Lock source Replace source by target
Source color #4 / Source_color_4	Color	r: 0 g: 0 b: 0 a: 0	
Target color #4 / Target_color_4	Color	r: 0 g: 0 b: 0 a: 0	
Action #5 / Action_5	Choice	Ignore	Ignore Lock source Replace source by target

Continued on next page

Table 263 – continued from previous page

Parameter / script name	Type	Default	Function
Source color #5 / Source_color_5	Color	r: 0 g: 0 b: 0 a: 0	
Target color #5 / Target_color_5	Color	r: 0 g: 0 b: 0 a: 0	
Action #6 / Action_6	Choice	Ignore	Ignore Lock source Replace source by target
Source color #6 / Source_color_6	Color	r: 0 g: 0 b: 0 a: 0	
Target color #6 / Target_color_6	Color	r: 0 g: 0 b: 0 a: 0	
Action #7 / Action_7	Choice	Ignore	Ignore Lock source Replace source by target
Source color #7 / Source_color_7	Color	r: 0 g: 0 b: 0 a: 0	
Target color #7 / Target_color_7	Color	r: 0 g: 0 b: 0 a: 0	
Action #8 / Action_8	Choice	Ignore	Ignore Lock source Replace source by target
Source color #8 / Source_color_8	Color	r: 0 g: 0 b: 0 a: 0	
Target color #8 / Target_color_8	Color	r: 0 g: 0 b: 0 a: 0	
Action #9 / Action_9	Choice	Ignore	Ignore Lock source Replace source by target
Source color #9 / Source_color_9	Color	r: 0 g: 0 b: 0 a: 0	

Continued on next page

Table 263 – continued from previous page

Parameter / script name	Type	Default	Function
Target color #9 / Target_color_9	Color	r: 0 g: 0 b: 0 a: 0	
Action #10 / Action_10	Choice	Ignore	Ignore Lock source Replace source by target
Source color #10 / Source_color_10	Color	r: 0 g: 0 b: 0 a: 0	
Target color #10 / Target_color_10	Color	r: 0 g: 0 b: 0 a: 0	
Action #11 / Action_11	Choice	Ignore	Ignore Lock source Replace source by target
Source color #11 / Source_color_11	Color	r: 0 g: 0 b: 0 a: 0	
Target color #11 / Target_color_11	Color	r: 0 g: 0 b: 0 a: 0	
Action #12 / Action_12	Choice	Ignore	Ignore Lock source Replace source by target
Source color #12 / Source_color_12	Color	r: 0 g: 0 b: 0 a: 0	
Target color #12 / Target_color_12	Color	r: 0 g: 0 b: 0 a: 0	
Action #13 / Action_13	Choice	Ignore	Ignore Lock source Replace source by target
Source color #13 / Source_color_13	Color	r: 0 g: 0 b: 0 a: 0	
Target color #13 / Target_color_13	Color	r: 0 g: 0 b: 0 a: 0	

Continued on next page

Table 263 – continued from previous page

Parameter / script name	Type	Default	Function
Action #14 / Action_14	Choice	Ignore	Ignore Lock source Replace source by target
Source color #14 / Source_color_14	Color	r: 0 g: 0 b: 0 a: 0	
Target color #14 / Target_color_14	Color	r: 0 g: 0 b: 0 a: 0	
Action #15 / Action_15	Choice	Ignore	Ignore Lock source Replace source by target
Source color #15 / Source_color_15	Color	r: 0 g: 0 b: 0 a: 0	
Target color #15 / Target_color_15	Color	r: 0 g: 0 b: 0 a: 0	
Action #16 / Action_16	Choice	Ignore	Ignore Lock source Replace source by target
Source color #16 / Source_color_16	Color	r: 0 g: 0 b: 0 a: 0	
Target color #16 / Target_color_16	Color	r: 0 g: 0 b: 0 a: 0	
Action #17 / Action_17	Choice	Ignore	Ignore Lock source Replace source by target
Source color #17 / Source_color_17	Color	r: 0 g: 0 b: 0 a: 0	
Target color #17 / Target_color_17	Color	r: 0 g: 0 b: 0 a: 0	

Continued on next page

Table 263 – continued from previous page

Parameter / script name	Type	Default	Function
Action #18 / Action_18	Choice	Ignore	Ignore Lock source Replace source by target
Source color #18 / Source_color_18	Color	r: 0 g: 0 b: 0 a: 0	
Target color #18 / Target_color_18	Color	r: 0 g: 0 b: 0 a: 0	
Action #19 / Action_19	Choice	Ignore	Ignore Lock source Replace source by target
Source color #19 / Source_color_19	Color	r: 0 g: 0 b: 0 a: 0	
Target color #19 / Target_color_19	Color	r: 0 g: 0 b: 0 a: 0	
Action #20 / Action_20	Choice	Ignore	Ignore Lock source Replace source by target
Source color #20 / Source_color_20	Color	r: 0 g: 0 b: 0 a: 0	
Target color #20 / Target_color_20	Color	r: 0 g: 0 b: 0 a: 0	
Action #21 / Action_21	Choice	Ignore	Ignore Lock source Replace source by target
Source color #21 / Source_color_21	Color	r: 0 g: 0 b: 0 a: 0	
Target color #21 / Target_color_21	Color	r: 0 g: 0 b: 0 a: 0	

Continued on next page

Table 263 – continued from previous page

Parameter / script name	Type	Default	Function
Action #22 / Action_22	Choice	Ignore	Ignore Lock source Replace source by target
Source color #22 / Source_color_22	Color	r: 0 g: 0 b: 0 a: 0	
Target color #22 / Target_color_22	Color	r: 0 g: 0 b: 0 a: 0	
Action #23 / Action_23	Choice	Ignore	Ignore Lock source Replace source by target
Source color #23 / Source_color_23	Color	r: 0 g: 0 b: 0 a: 0	
Target color #23 / Target_color_23	Color	r: 0 g: 0 b: 0 a: 0	
Action #24 / Action_24	Choice	Ignore	Ignore Lock source Replace source by target
Source color #24 / Source_color_24	Color	r: 0 g: 0 b: 0 a: 0	
Target color #24 / Target_color_24	Color	r: 0 g: 0 b: 0 a: 0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 263 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.78 G'MIC Cutout node

This documentation is for version 1.0 of G'MIC Cutout.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of levels / Number_of_levels	Integer	4	
Edge simplicity / Edge_simplicity	Double	0.5	
Edge fidelity / Edge_fidelity	Integer	4	
Normalize / Normalize	Boolean	On	

Continued on next page

Table 264 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 264 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.79 G'MIC Decompose channels node

This documentation is for version 1.0 of G'MIC Decompose channels.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Color basis / Color_basis	Choice	Lab	RGB HSV HSL HSI YUV YCbCr XYZ Lab Lch CMY CMYK YIQ

Continued on next page

Table 265 – continued from previous page

Parameter / script name	Type	Default	Function
Action / Action	Choice	Decompose	Decompose Recompose
Output multiple layers / Output_multiple_layers	Boolean	Off	
Include opacity layer / Include_opacity_layer	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.80 G'MIC Deinterlace node

This documentation is for version 1.0 of G'MIC Deinterlace.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
G/M smoothness / GM_smoothness	Double	6	
R/B smoothness (principal) / RB_smoothness_principal	Double	6	
R/B smoothness (secondary) / RB_smoothness_secondary	Double	4	
Algorithm / Algorithm	Choice	Standard	Standard Motion-compensated
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 266 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.81 G'MIC Details equalizer node

This documentation is for version 1.0 of G'MIC Details equalizer.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Base scale / Base_scale	Double	5	
Detail scale / Detail_scale	Double	0.5	
Threshold / Threshold	Double	0	
Smoothness / Smoothness	Double	0	
Smoothness type / Smoothness_type	Choice	Diffusion	Gaussian Bilateral Diffusion
Gain / Gain	Double	0	
Threshold_2 / Threshold_2	Double	0	
Smoothness_2 / Smoothness_2	Double	0	
Smoothness type_2 / Smoothness_type_2	Choice	Diffusion	Gaussian Bilateral Diffusion
Gain_2 / Gain_2	Double	0	
Threshold_3 / Threshold_3	Double	0	
Smoothness_3 / Smoothness_3	Double	0	
Smoothness type_3 / Smoothness_type_3	Choice	Diffusion	Gaussian Bilateral Diffusion
Gain_3 / Gain_3	Double	0	
Threshold_4 / Threshold_4	Double	0	
Smoothness_4 / Smoothness_4	Double	0	
Smoothness type_4 / Smoothness_type_4	Choice	Diffusion	Gaussian Bilateral Diffusion
Gain_4 / Gain_4	Double	0	

Continued on next page

Table 267 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	YCbCr [luminance]	All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas]
Value action / Value_action	Choice	None	None Cut Normalize

Continued on next page

Table 267 – continued from previous page

Parameter / script name	Type	Default	Function
Parallel processing / Parallel_processing	Choice	Auto	Auto One thread Two threads Four threads Eight threads Sixteen threads
Spatial overlap / Spatial_overlap	Integer	32	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 267 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.82 G'MIC Detect skin node

This documentation is for version 1.0 of G'MIC Detect skin.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Skin estimation / Skin_estimation	Choice	Automatic	Manual Automatic
Tolerance / Tolerance	Double	0.5	
Smoothness / Smoothness	Double	0.5	

Continued on next page

Table 268 – continued from previous page

Parameter / script name	Type	Default	Function
Threshold / Threshold	Double	1	
Pre-normalize image / Prenormalize_image	Boolean	On	
X-coordinate / Xcoordinate	Double	50	
Y-coordinate / Ycoordinate	Double	50	
Radius / Radius	Double	5	
Output mode / Output_mode	Choice	Opaque skin	Probability map Opaque skin Transparent skin
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 268 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.83 G'MIC Dices node

This documentation is for version 1.0 of G'MIC Dices.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Resolution / Resolution	Double	2	
Size / Size	Integer	24	

Continued on next page

Table 269 – continued from previous page

Parameter / script name	Type	Default	Function
Color model / Color_model	Choice	White dices	Black dices White dices Dices with colored numbers Dices with colored sides
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.84 G'MIC Difference of gaussians node

This documentation is for version 1.0 of G'MIC Difference of gaussians.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
1st variance / p1st_variance	Double	1.4	
2nd variance / p2nd_variance	Double	1.5	
Threshold / Threshold	Double	0	
Negative colors / Negative_colors	Boolean	Off	
Monochrome / Monochrome	Boolean	On	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 270 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.85 G'MIC Diffusion tensors node

This documentation is for version 1.0 of G'MIC Diffusion tensors.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Resolution (%) / Resolution_	Double	10	
Size / Size	Double	5	

Continued on next page

Table 271 – continued from previous page

Parameter / script name	Type	Default	Function
Color mode / Color_mode	Choice	Color	Monochrome Grayscale Orientation Color
Outline / Outline	Integer	1	
Sharpness / Sharpness	Double	0.15	
Anisotropy / Anisotropy	Double	1	
Gradient smoothness / Gradient_smoothness	Double	0	
Tensor smoothness / Tensor_smoothness	Double	3	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 271 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.86 G'MIC Dirty node

This documentation is for version 1.0 of G'MIC Dirty.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	30	
Monochrome / Monochrome	Boolean	On	

Continued on next page

Table 272 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Value action / Value_action	Choice	None	<p> None Cut Normalize </p>

Continued on next page

Table 272 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 272 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.87 G'MIC Distance transform node

This documentation is for version 1.0 of G'MIC Distance transform.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Value / Value	Integer	128	
Metric / Metric	Choice	Euclidean	Chebyshev Manhattan Euclidean Squared-Euclidean
Normalization / Normalization	Choice	Modulo	Cut Normalize Modulo
Modulo value / Modulo_value	Integer	32	

Continued on next page

Table 273 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 273 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.88 G'MIC Distort lens node

This documentation is for version 1.0 of G'MIC Distort lens.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	0.1	
Aspect ratio / Aspect_ratio	Double	0	
Zoom / Zoom	Double	0	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Boundary / Boundary	Choice	Transparent	Transparent Nearest Periodic Mirror

Continued on next page

Table 274 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.89 G'MIC Dithering node

This documentation is for version 1.0 of G'MIC Dithering.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue / Hue	Double	0	
Saturation (%) / Saturation_	Double	0	
Smoothness / Smoothness	Double	0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 275 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.90 G'MIC Drawn montage node

This documentation is for version 1.0 of G'MIC Drawn montage.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Deverny.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Layer / Layer	Choice	1st	1st 2nd 3rd 4th 5th 6th 7th 8th 9th 10th 11th 12th 13th 14th 15th 16th
Associated color / Associated_color	Color	r: 0 g: 0 b: 0 a: 0	
Zoom / Zoom	Double	-10	
X-centering (%) / Xcentering_	Double	50	
Y-centering (%) / Ycentering_	Double	50	
Angle / Angle	Choice	0 deg.	0 deg. 90 deg. 180 deg. 270 deg.
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 276 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.91 G'MIC Drop shadow node

This documentation is for version 1.0 of G'MIC Drop shadow.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-shadow / Xshadow	Double	3	
Y-shadow / Yshadow	Double	3	
Smoothness / Smoothness	Double	1.8	
Curvature / Curvature	Double	0	
Corner brightness / Corner_brightness	Double	0	
Angle / Angle	Double	0	

Continued on next page

Table 277 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.92 G'MIC Drop shadow 3d node

This documentation is for version 1.0 of G'MIC Drop shadow 3d.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-angle / Xangle	Double	0	
Y-angle / Yangle	Double	0	
Z-angle / Zangle	Double	0	
Zoom / Zoom	Double	0	
X-offset / Xoffset	Double	1	
Y-offset / Yoffset	Double	1	
Perspective / Perspective	Double	2	
Smoothness / Smoothness	Double	0.5	
Color / Color	Color	r: 0 g: 0 b: 0 a: 0	
Preview only shadow / Preview_only_shadow	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.93 G'MIC Drop water node

This documentation is for version 1.0 of G'MIC Drop water.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Shapes / Shapes	Choice	Procedural	Procedural Opaque regions on top layer
Density / Density	Double	20	
Radius / Radius	Double	2	
Variability / Variability	Double	80	
Random seed / Random_seed	Integer	0	
Refraction / Refraction	Double	3	
Light angle / Light_angle	Double	35	
Specular size / Specular_size	Double	10	
Specular intensity / Specular_intensity	Double	1	
Specular centering / Specular_centering	Double	0.5	
Shadow size / Shadow_size	Double	0.25	
Shadow intensity / Shadow_intensity	Double	0.5	
Shadow smoothness / Shadow_smoothness	Double	0.75	
Diffuse shadow / Diffuse_shadow	Double	0.05	
Smoothness / Smoothness	Double	0.15	

Continued on next page

Table 279 – continued from previous page

Parameter / script name	Type	Default	Function
Output as separate layers / Output_as_separate_layers	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.94 G'MIC Droste node

This documentation is for version 1.0 of G'MIC Droste.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X0 / X0	Double	20	
Y0 / Y0	Double	20	
X1 / X1	Double	80	
Y1 / Y1	Double	20	
X2 / X2	Double	80	
Y2 / Y2	Double	80	
X3 / X3	Double	20	
Y3 / Y3	Double	80	
Iterations / Iterations	Integer	1	
X-shift / Xshift	Double	0	
Y-shift / Yshift	Double	0	
Angle / Angle	Double	0	
Zoom / Zoom	Double	1	
Mirror / Mirror	Choice	None	None x-axis y-axis xy-axes
Boundary / Boundary	Choice	Nearest	Transparent Nearest Periodic Mirror
Drawing mode / Drawing_mode	Choice	Replace	Replace Replace (sharpest) Behind Below
View outlines only / View_outlines_only	Boolean	Off	

Continued on next page

Table 280 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.95 G'MIC Edges node

This documentation is for version 1.0 of G'MIC Edges.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	0	
Threshold / Threshold	Double	15	
Negative colors / Negative_colors	Boolean	Off	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 281 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.96 G'MIC Edges offsets node

This documentation is for version 1.0 of G'MIC Edges offsets.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	0	
Threshold / Threshold	Double	15	
Scale / Scale	Integer	4	
Thickness / Thickness	Integer	1	
Negative colors / Negative_colors	Boolean	Off	

Continued on next page

Table 282 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 282 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.97 G'MIC Edges on fire node

This documentation is for version 1.0 of G'MIC Edges on fire.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Edges / Edges	Double	0.7	
Attenuation / Attenuation	Double	0.25	
Smoothness / Smoothness	Double	0.5	
Threshold / Threshold	Double	25	
Number of frames / Number_of_frames	Integer	20	
Starting frame / Starting_frame	Integer	20	
Frame skip / Frame_skip	Integer	0	

Continued on next page

Table 283 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 283 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.98 G'MIC Ellipsionism node

This documentation is for version 1.0 of G'MIC Ellipsionism.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Primary radius / Primary_radius	Double	20	
Secondary radius / Secondary_radius	Double	10	
Smoothness / Smoothness	Double	0.5	
Opacity / Opacity	Double	0.7	
Outline / Outline	Double	3	
Density / Density	Double	0.5	

Continued on next page

Table 284 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 284 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.99 G'MIC Engrave node

This documentation is for version 1.0 of G'MIC Engrave.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Radius / Radius	Double	0.5	
Density / Density	Double	50	
Edges / Edges	Double	0	
Coherence / Coherence	Double	8	
Threshold (%) / Threshold_	Double	40	
Minimal area / Minimal_area	Integer	0	
Flat regions removal / Flat_regions_removal	Double	0	
Add color background / Add_color_background	Boolean	Off	
Quantization / Quantization	Double	10	
Shading / Shading	Integer	1	
Hue / Hue	Double	0	
Saturation (%) / Saturation_	Double	0	
Lightness (%) / Lightness_	Double	0	

Continued on next page

Table 285 – continued from previous page

Parameter / script name	Type	Default	Function
Anti-aliasing / Antialiasing	Choice	x1.5	Disabled x1.5 x2 x3
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 285 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.100 G'MIC Equalize HSI-HSL-HSV node

This documentation is for version 1.0 of G'MIC Equalize HSI-HSL-HSV.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Colorspace / Colorspace	Choice	HSL	HSI HSL HSV
Opacity (%) / Opacity_	Double	100	
Value blending / Value_blending	Double	0	
Color blending / Color_blending	Double	0	
Preview mapping / Preview_mapping	Choice	None	None Grey Color

Continued on next page

Table 286 – continued from previous page

Parameter / script name	Type	Default	Function
Hue offset / Hue_offset	Double	0	
Saturation offset / Saturation_offset	Double	0	
Value offset / Value_offset	Double	0	
Hue offset_2 / Hue_offset_2	Double	0	
Saturation offset_2 / Saturation_offset_2	Double	0	
Value offset_2 / Value_offset_2	Double	0	
Hue offset_3 / Hue_offset_3	Double	0	
Saturation offset_3 / Saturation_offset_3	Double	0	
Value offset_3 / Value_offset_3	Double	0	
Hue offset_4 / Hue_offset_4	Double	0	
Saturation offset_4 / Saturation_offset_4	Double	0	
Value offset_4 / Value_offset_4	Double	0	
Hue offset_5 / Hue_offset_5	Double	0	
Saturation offset_5 / Saturation_offset_5	Double	0	
Value offset_5 / Value_offset_5	Double	0	
Hue offset_6 / Hue_offset_6	Double	0	
Saturation offset_6 / Saturation_offset_6	Double	0	
Value offset_6 / Value_offset_6	Double	0	
Hue offset_7 / Hue_offset_7	Double	0	
Saturation offset_7 / Saturation_offset_7	Double	0	
Value offset_7 / Value_offset_7	Double	0	
Hue offset_8 / Hue_offset_8	Double	0	
Saturation offset_8 / Saturation_offset_8	Double	0	
Value offset_8 / Value_offset_8	Double	0	
Hue offset_9 / Hue_offset_9	Double	0	
Saturation offset_9 / Saturation_offset_9	Double	0	

Continued on next page

Table 286 – continued from previous page

Parameter / script name	Type	Default	Function
Value offset_9 / Value_offset_9	Double	0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 286 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.101 G'MIC Equalize HSV node

This documentation is for version 1.0 of G'MIC Equalize HSV.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Preview bands / Preview_bands	Boolean	Off	
Hue band / Hue_band	Double	180	
Band width / Band_width	Double	40	
Hue shift / Hue_shift	Double	0	
Saturation correction / Saturation_correction	Double	0	
Value correction / Value_correction	Double	0	
Hue band_2 / Hue_band_2	Double	180	
Band width_2 / Band_width_2	Double	40	
Hue shift_2 / Hue_shift_2	Double	0	
Saturation correction_2 / Saturation_correction_2	Double	0	
Value correction_2 / Value_correction_2	Double	0	

Continued on next page

Table 287 – continued from previous page

Parameter / script name	Type	Default	Function
Hue band_3 / Hue_band_3	Double	180	
Band width_3 / Band_width_3	Double	40	
Hue shift_3 / Hue_shift_3	Double	0	
Saturation correction_3 / Saturation_correction_3	Double	0	
Value correction_3 / Value_correction_3	Double	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.102 G'MIC Equalize local histograms node

This documentation is for version 1.0 of G'MIC Equalize local histograms.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Strength (%) / Strength_	Double	75	
Mode / Mode	Choice	Soft	Raw Hard Soft
Radius / Radius	Integer	4	
Sigma / Sigma	Double	100	
Regularization / Regularization	Double	8	
Reduce halos / Reduce_halos	Boolean	On	

Continued on next page

Table 288 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	Lab [lightness]	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 288 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 288 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.103 G'MIC Equalize shadow node

This documentation is for version 1.0 of G'MIC Equalize shadow.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	1	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 289 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.104 G'MIC Equation plot parametric node

This documentation is for version 1.0 of G'MIC Equation plot parametric.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X(t) / Xt	String	$\sin(t) * (\exp(\cos(t)) - 2 * \cos(4 * t) - \sin(t/12)^5)$	
Y(t) / Yt	String	$\cos(t) * (\exp(\cos(t)) - 2 * \cos(4 * t) - \sin(t/12)^5)$	
Min-t / Mint	Double	0	
Max-t / Maxt	Double	100	
Resolution / Resolution	Integer	4096	
Outline opacity / Outline_opacity	Double	1	
Dot size / Dot_size	Integer	0	
Start color / Start_color	Color	r: 0.25098 g: 0 b: 0 a: 0	
End color / End_color	Color	r: 0.501961 g: 0 b: 0 a: 0	
Colored outline / Colored_outline	Boolean	On	
Antialiasing / Antialiasing	Boolean	On	
Decoration / Decoration	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 290 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.105 G'MIC Equation plot $y=f(x)$ node

This documentation is for version 1.0 of G'MIC Equation plot $y=f(x)$.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
F(x) / Fx	String	$X*c+10*\cos(X+c+u)$	
X-min / Xmin	Double	-10	
X-max / Xmax	Double	10	
Resolution / Resolution	Integer	100	
Channels / Channels	Integer	3	

Continued on next page

Table 291 – continued from previous page

Parameter / script name	Type	Default	Function
Plot type / Plot_type	Choice	Splines	None Lines Splines Bars
Vertex type / Vertex_type	Choice	None	None Points Crosses 1 Crosses 2 Circles 1 Circles 2 Square 1 Square 2
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 291 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.106 G'MIC Equirectangular to nadir-zenith node

This documentation is for version 1.0 of G'MIC Equirectangular to nadir-zenith.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Mode / Mode	Choice	to nadir/zenith	to nadir/zenith to equirectangular
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 292 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.107 G'MIC Euclidean polar node

This documentation is for version 1.0 of G'MIC Euclidean polar.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Stretch factor / Stretch_factor	Double	1	
Boundary / Boundary	Choice	Nearest	Transparent Nearest Periodic Mirror

Continued on next page

Table 293 – continued from previous page

Parameter / script name	Type	Default	Function
Inverse transform / Inverse_transform	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.108 G'MIC Extract objects node

This documentation is for version 1.0 of G'MIC Extract objects.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Background point (%) / Background_point_	Double	x: 0 y: 0	
Color tolerance / Color_tolerance	Integer	20	
Opacity threshold (%) / Opacity_threshold_	Integer	50	
Minimal area / Minimal_area	Double	0.3	
Connectivity / Connectivity	Choice	Low	Low High
Output as / Output_as	Choice	Crop	Crop Segmentation
Preview guides / Preview_guides	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 294 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.109 G'MIC Fade layers node

This documentation is for version 1.0 of G'MIC Fade layers.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Inter-frames / Interframes	Integer	10	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 295 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.110 G'MIC Felt pen node

This documentation is for version 1.0 of G'MIC Felt pen.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	300	
Density / Density	Double	50	
Smoothness / Smoothness	Double	1	
Opacity / Opacity	Double	0.1	
Edge / Edge	Double	20	
Thickness / Thickness	Integer	5	

Continued on next page

Table 296 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 296 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.111 G'MIC Flip & rotate blocs node

This documentation is for version 1.0 of G'MIC Flip & rotate blocs.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-size (px) / Xsize_px	Integer	4	
Y-size (px) / Ysize_px	Integer	4	
Flip / Flip	Choice	XY-axes	None X-axis Y-axis XY-axes
Rotate / Rotate	Choice	0 deg.	-90 deg. 0 deg. 90 deg.

Continued on next page

Table 297 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 297 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 297 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.112 G'MIC Flower node

This documentation is for version 1.0 of G'MIC Flower.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Radius / Radius	Double	70	
Amplitude / Amplitude	Double	1	
Center (%)_2 / Center__2	Double	x: 0.5 y: 0.5	
Amplitude / Angle / Amplitude__Angle	Double	x: 0.75 y: 0.5	
Petals / Petals	Integer	6	
Offset (%) / Offset_	Double	0	
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror

Continued on next page

Table 298 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.113 G'MIC Fourier analysis node

This documentation is for version 1.0 of G'MIC Fourier analysis.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.114 G'MIC Fourier transform node

This documentation is for version 1.0 of G'MIC Fourier transform.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Magnitude/Phase / MagnitudePhase	Choice	One layer (vertical)	One layer (horizontal) One layer (vertical) Two layers
Discard transparency / Discard_transparency	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 300 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.115 G'MIC Fourier watermark node

This documentation is for version 1.0 of G'MIC Fourier watermark.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Text / Text	String	(c) G'MIC	
Size / Size	Integer	53	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 301 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.116 G'MIC Frame blur node

This documentation is for version 1.0 of G'MIC Frame blur.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Horizontal size (%) / Horizontal_size_	Double	30	
Vertical size (%) / Vertical_size_	Double	30	
Crop / Crop	Double	0	
Blur / Blur	Double	5	
Roundness / Roundness	Double	0	
Apply color balance / Apply_color_balance	Boolean	Off	

Continued on next page

Table 302 – continued from previous page

Parameter / script name	Type	Default	Function
Balance color / Balance_color	Color	r: 0.501961 g: 0.501961 b: 0.501961 a: 0.501961	
Normalization / Normalization	Choice	None	None Stretch Equalize
Outline size / Outline_size	Double	5	
Outline color / Outline_color	Color	r: 1 g: 1 b: 1 a: 1	
X-shadow / Xshadow	Double	2	
Y-shadow / Yshadow	Double	2	
Shadow smoothness / Shadow_smoothness	Double	1	
Shadow contrast / Shadow_contrast	Double	0	
X-centering / Xcentering	Double	0.5	
Y-centering / Ycentering	Double	0.5	
Angle / Angle	Double	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 302 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.117 G'MIC Frame fuzzy node

This documentation is for version 1.0 of G'MIC Frame fuzzy.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Depth / Depth	Double	3	
X-center / Xcenter	Double	0	
Y-center / Ycenter	Double	0	
Left side orientation / Left_side_orientation	Choice	Normal	Normal Mirror-X Mirror-Y Mirror-XY

Continued on next page

Table 303 – continued from previous page

Parameter / script name	Type	Default	Function
Right side orientation / Right_side_orientation	Choice	Normal	Normal Mirror-X Mirror-Y Mirror-XY
Upper side orientation / Upper_side_orientation	Choice	Normal	Normal Mirror-X Mirror-Y Mirror-XY
Lower sider orientation / Lower_sider_orientation	Choice	Normal	Normal Mirror-X Mirror-Y Mirror-XY
Horizontal size (%) / Horizontal_size_	Double	5	
Vertical size (%) / Vertical_size_	Double	5	
Fuzzyness / Fuzzyness	Double	10	
Smoothness / Smoothness	Double	1	
Color / Color	Color	r: 1 g: 1 b: 1 a: 1	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 303 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.118 G'MIC Frame painting node

This documentation is for version 1.0 of G'MIC Frame painting.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Size (%) / Size_	Double	10	
Contrast / Contrast	Double	0.4	
Smoothness / Smoothness	Double	6	

Continued on next page

Table 304 – continued from previous page

Parameter / script name	Type	Default	Function
Color / Color	Color	r: 0.882353 g: 0.784314 b: 0.470588 a: 0.470588	
Vignette size / Vignette_size	Double	2	
Vignette contrast / Vignette_contrast	Double	400	
Defects contrast / Defects_contrast	Double	50	
Defects density / Defects_density	Double	10	
Defects size / Defects_size	Double	1	
Defects smoothness / Defects_smoothness	Double	0.5	
Serial number / Serial_number	Integer	123456	
Frame as a new layer / Frame_as_a_new_layer	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 304 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.119 G'MIC Frame pattern node

This documentation is for version 1.0 of G'MIC Frame pattern.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Tiles / Tiles	Integer	10	
Pattern / Pattern	Choice	Self image	Top layer Self image
Iterations / Iterations	Integer	1	
Constrain image size / Constrain_image_size	Boolean	On	

Continued on next page

Table 305 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.120 G'MIC Frame regular node

This documentation is for version 1.0 of G'MIC Frame regular.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-start (%) / Xstart_	Integer	0	
X-end (%) / Xend_	Integer	100	
Y-start (%) / Ystart_	Integer	0	
Y-end (%) / Yend_	Integer	100	
Width (%) / Width_	Integer	10	
Height (%) / Height_	Integer	10	
Color / Color	Color	r: 0 g: 0 b: 0 a: 0	
Outline size / Outline_size	Integer	1	
Outline color / Outline_color	Color	r: 1 g: 1 b: 1 a: 1	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.121 G'MIC Frame round node

This documentation is for version 1.0 of G'MIC Frame round.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Sharpness / Sharpness	Double	6	
Size (%) / Size_	Double	20	
Smoothness / Smoothness	Double	0.1	
Shade / Shade	Double	0	
Color / Color	Color	r: 1 g: 1 b: 1 a: 1	
Blur frame / Blur_frame	Double	0	
Blur shade / Blur_shade	Double	0.1	
Blur amplitude / Blur_amplitude	Double	3	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 307 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.122 G'MIC Frame smooth node

This documentation is for version 1.0 of G'MIC Frame smooth.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Deverny.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Width (%) / Width_	Integer	10	
Height (%) / Height_	Integer	10	
Roundness / Roundness	Double	0.25	

Continued on next page

Table 308 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.123 G'MIC Freaky B&W node

This documentation is for version 1.0 of G'MIC Freaky B&W.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Strength (%) / Strength_	Double	90	
Oddness (%) / Oddness_	Double	20	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 309 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.124 G'MIC Freaky details node

This documentation is for version 1.0 of G'MIC Freaky details.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Integer	2	
Scale / Scale	Double	10	
Iterations / Iterations	Integer	1	

Continued on next page

Table 310 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	YCbCr [lumi- nance]	All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas]
Parallel processing / Parallel_processing	Choice	Auto	Auto One thread Two threads Four threads Eight threads Sixteen threads
Spatial overlap / Spatial_overlap	Integer	32	

Continued on next page

Table 310 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 310 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.125 G'MIC Fuji xtrans node

This documentation is for version 1.0 of G'MIC Fuji xtrans.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Preset / Preset	Choice	None	None Astia Classic Chrome Pro Neg Hi Pro Neg Std Provia Velvia
Strength (%) / Strength_	Double	100	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	

Continued on next page

Table 311 – continued from previous page

Parameter / script name	Type	Default	Function
Normalize colors / Normalize_colors	Choice	None	None Pre-process Post-process Both
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 311 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.126 G'MIC Games & demos node

This documentation is for version 1.0 of G'MIC Games & demos.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Selection / Selection	Choice	2048	<p> 2048 Blobs editor Bouncing balls Connect-4 Fire effect Fireworks Fish-eye effect Fourier filtering Hanoi tower Histogram Hough transform Jawbreaker Virtual landscape The game of life Light effect Mandelbrot explorer 3d metaballs Minesweeper Minimal path Pacman Paint Plasma effect RGB quantization 3d reflection 3d rubber object Shadebobs Spline editor 3d starfield Tetris Tic-tac-toe 3d waves Fractal whirl </p>
Output Layer / Output_Layer	Choice	Layer 0	<p> Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9 </p>

Continued on next page

Table 312 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.127 G'MIC Gear node

This documentation is for version 1.0 of G'MIC Gear.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Size (%) / Size_	Double	75	
Number of teeth / Number_of_teeth	Integer	12	
Elevation (%) / Elevation_	Double	15	
Angle (%) / Angle_	Double	0	
Inner radius (%) / Inner_radius_	Double	40	

Continued on next page

Table 313 – continued from previous page

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	0	
Color / Color	Color	r: 1 g: 0 b: 0 a: 0	
Antialiasing / Antialiasing	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.128 G'MIC Gradient RGB node

This documentation is for version 1.0 of G'MIC Gradient RGB.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	0	
Min threshold / Min_threshold	Double	0	
Max threshold / Max_threshold	Double	100	
Orientation only / Orientation_only	Boolean	Off	
Negative colors / Negative_colors	Boolean	Off	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 314 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.129 G'MIC Gradient corners node

This documentation is for version 1.0 of G'MIC Gradient corners.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Color 1 (up/left corner) / Color_1_upleft_corner	Color	r: 1 g: 1 b: 1 a: 1	
Color 2 (up/right corner) / Color_2_upright_corner	Color	r: 1 g: 0 b: 0 a: 0	
Color 3 (bottom/left corner) / Color_3_bottomleft_corner	Color	r: 0 g: 1 b: 0 a: 0	

Continued on next page

Table 315 – continued from previous page

Parameter / script name	Type	Default	Function
Color 4 (bottom/right corner) / Color_4_bottomright_corner	Color	r: 0 g: 0 b: 1	
Colorspace / Colorspace	Choice	Linear RGB	sRGB Linear RGB Lab
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.130 G'MIC Gradient custom shape node

This documentation is for version 1.0 of G'MIC Gradient custom shape.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Select by / Select_by	Choice	Auto	Auto Dark pixels Bright pixels Opaque pixels
Smoothness / Smoothness	Double	0	
Threshold / Threshold	Double	0	
Preview shape / Preview_shape	Boolean	On	
Number of colors / Number_of_colors	Integer	4	
Cycles / Cycles	Double	1	
Offset / Offset	Double	0	
Shading / Shading	Double	128	
Inner length / Inner_length	Double	100	
Outer length / Outer_length	Double	100	
Spatial metric / Spatial_metric	Choice	Euclidean	Chebyshev Manhattan Euclidean
Color metric / Color_metric	Choice	RGB	RGB HSV Lab
Shade back to first color / Shade_back_to_first_color	Boolean	On	
Preview gradient / Preview_gradient	Boolean	Off	
Save gradient as / Save_gradient_as	String		

Continued on next page

Table 316 – continued from previous page

Parameter / script name	Type	Default	Function
Colormap type / Colormap_type	Choice	User-defined	Pre-defined User-defined
Pre-defined colormap / Predefined_colormap	Integer	0	
1st color / p1st_color	Color	r: 0 g: 0 b: 0 a: 0	
2nd color / p2nd_color	Color	r: 1 g: 0 b: 0 a: 0	
3rd color / p3rd_color	Color	r: 1 g: 1 b: 0 a: 0	
4th color / p4th_color	Color	r: 1 g: 1 b: 1 a: 1	
5th color / p5th_color	Color	r: 0 g: 1 b: 1 a: 1	
6th color / p6th_color	Color	r: 0 g: 1 b: 0 a: 0	
7th color / p7th_color	Color	r: 0 g: 0 b: 1 a: 1	
8th color / p8th_color	Color	r: 0.501961 g: 0.501961 b: 0.501961 a: 0.501961	
9th color / p9th_color	Color	r: 1 g: 0 b: 1 a: 1	
10th color / p10th_color	Color	r: 0 g: 0 b: 0 a: 0	

Continued on next page

Table 316 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.131 G'MIC Gradient from line node

This documentation is for version 1.0 of G'MIC Gradient from line.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Starting point (%) / Starting_point_	Double	x: 0 y: 0	
Ending point (%) / Ending_point_	Double	x: 1 y: 1	
Sampling / Sampling	Double	100	
Length / Length	Integer	0	
Sort colors / Sort_colors	Choice	Don't sort	Don't sort By red component By green component By blue component By luminance By blue chrominance By red chrominance By lightness
Reverse gradient / Reverse_gradient	Boolean	Off	
Preview gradient / Preview_gradient	Boolean	On	
Save gradient as / Save_gradient_as	String		
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 317 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.132 G'MIC Gradient linear node

This documentation is for version 1.0 of G'MIC Gradient linear.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Starting color / Starting_color	Color	r: 0 g: 0 b: 0 a: 0	
Ending color / Ending_color	Color	r: 1 g: 1 b: 1 a: 1	
Swap colors / Swap_colors	Boolean	Off	
Angle / Angle	Double	45	
Fade start / Fade_start	Double	0	
Fade end / Fade_end	Double	100	
Colorspace / Colorspace	Choice	sRGB	sRGB Linear RGB Lab

Continued on next page

Table 318 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.133 G'MIC Gradient norm node

This documentation is for version 1.0 of G'MIC Gradient norm.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	0	
Linearity / Linearity	Double	0.5	
Min threshold / Min_threshold	Double	0	
Max threshold / Max_threshold	Double	100	
Negative colors / Negative_colors	Boolean	Off	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 319 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.134 G'MIC Gradient radial node

This documentation is for version 1.0 of G'MIC Gradient radial.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Starting color / Starting_color	Color	r: 0 g: 0 b: 0 a: 0	
Ending color / Ending_color	Color	r: 1 g: 1 b: 1 a: 1	
Swap colors / Swap_colors	Boolean	Off	

Continued on next page

Table 320 – continued from previous page

Parameter / script name	Type	Default	Function
Fade start / Fade_start	Double	0	
Fade end / Fade_end	Double	100	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Colorspace / Colorspace	Choice	sRGB	sRGB Linear RGB Lab
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.135 G'MIC Gradient random node

This documentation is for version 1.0 of G'MIC Gradient random.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Integer	32	
Seed / Seed	Integer	0	
Smoothness / Smoothness	Double	0	
Color balance / Color_balance	Color	r: 0.501961 g: 0.501961 b: 0.501961 a: 0.501961	
Opacity / Opacity	Double	1	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 321 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.136 G'MIC Grid cartesian node

This documentation is for version 1.0 of G'MIC Grid cartesian.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-size / Xsize	Integer	10	
Y-size / Ysize	Integer	10	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 322 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.137 G'MIC Grid hexagonal node

This documentation is for version 1.0 of G'MIC Grid hexagonal.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Deverny.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Resolution / Resolution	Integer	32	
Outline / Outline	Double	0.1	
Anti-aliasing / Antialiasing	Boolean	On	

Continued on next page

Table 323 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.138 G'MIC Grid triangular node

This documentation is for version 1.0 of G'MIC Grid triangular.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Pattern width / Pattern_width	Integer	10	
Pattern height / Pattern_height	Integer	18	
Pattern type / Pattern_type	Choice	Horizontal	Horizontal Vertical Crossed Cube Decreasing Increasing
Outline color / Outline_color	Color	r: 1 g: 1 b: 1 a: 1	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.139 G'MIC Halftone node

This documentation is for version 1.0 of G'MIC Halftone.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Smoothness / Smoothness	Double	0	
Number of tones / Number_of_tones	Integer	5	
Size for dark tones / Size_for_dark_tones	Integer	8	
Size for bright tones / Size_for_bright_tones	Integer	8	
Shape / Shape	Choice	Circle (inv.)	Square Diamond Circle Square (inv.) Diamond (inv.) Circle (inv.)
Smoothness_2 / Smoothness_2	Double	0.1	

Continued on next page

Table 325 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 325 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.140 G'MIC Hard sketch node

This documentation is for version 1.0 of G'MIC Hard sketch.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Detail level / Detail_level	Double	0.8	
Amplitude / Amplitude	Double	300	
Density / Density	Double	50	
Smoothness / Smoothness	Double	1	
Opacity / Opacity	Double	0.1	
Edge / Edge	Double	20	
Fast approximation / Fast_approximation	Boolean	Off	
Color model / Color_model	Choice	Color on white	Black on white White on black Black on transparent white White on transparent black Color on white

Continued on next page

Table 326 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 326 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.141 G'MIC Heart node

This documentation is for version 1.0 of G'MIC Heart.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Size (%) / Size_	Double	75	
Smoothness / Smoothness	Double	0	
Color / Color	Color	r: 1 g: 0 b: 0 a: 0	
Antialiasing / Antialiasing	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 327 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.142 G'MIC Hearts node

This documentation is for version 1.0 of G'MIC Hearts.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Double	2	

Continued on next page

Table 328 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 328 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 328 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.143 G'MIC Highlight bloom node

This documentation is for version 1.0 of G'MIC Highlight bloom.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Details strength (%) / Details_strength_	Double	90	
Details scale / Details_scale	Double	60	
Smoothness / Smoothness	Double	60	
Highlight (%) / Highlight_	Integer	30	
Contrast (%) / Contrast_	Double	20	

Continued on next page

Table 329 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 329 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.144 G'MIC Histogram analysis node

This documentation is for version 1.0 of G'MIC Histogram analysis.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of clusters / Number_of_clusters	Integer	256	

Continued on next page

Table 330 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 330 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.145 G'MIC Hope poster node

This documentation is for version 1.0 of G'MIC Hope poster.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Gamma / Gamma	Double	0	
Smoothness / Smoothness	Double	3	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 331 – continued from previous page

Parameter / script name	Type	Default	Function
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.146 G'MIC Hough sketch node

This documentation is for version 1.0 of G'MIC Hough sketch.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	1.25	
Density / Density	Double	10	
Radius / Radius	Integer	5	
Threshold / Threshold	Double	80	
Opacity / Opacity	Double	0.1	
Color model / Color_model	Choice	Color on white	Black on white White on black Black on transparent white White on transparent black Color on white

Continued on next page

Table 332 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 332 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.147 G'MIC Illuminate 2d shape node

This documentation is for version 1.0 of G'MIC Illuminate 2d shape.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Input type / Input_type	Choice	Single opaque shapes over transp. bg	Single opaque shapes over transp. bg Multiple colored shapes over transp. bg Bump map Normal map
Output type / Output_type	Choice	Illumination	Illumination Bump map Normal map
Input guide color / Input_guide_color	Color	r: 1 g: 0 b: 0 a: 0	
Keep base layer as input background / Keep_base_layer_as_input_background	Boolean	On	
Keep transparency in output / Keep_transparency_in_output	Boolean	On	

Continued on next page

Table 333 – continued from previous page

Parameter / script name	Type	Default	Function
Minimal shape area / Minimal_shape_area	Integer	4	
Preview detected shapes / Preview_detected_shapes	Boolean	Off	
Erosion/Dilation / ErosionDilation	Double	0	
Smoothness / Smoothness	Double	2	
Bump factor / Bump_factor	Double	1	
Avg / max weight / Avg__max_weight	Double	1	
Resolution / Resolution	Choice	256	Full (slower) 2048 1024 512 256 128 64 (faster)
Blending mode / Blending_mode	Choice	Hard light	Normal Lighten Screen Dodge Add Darken Multiply Burn Overlay Soft light Hard light Grain merge
Opacity (%) / Opacity_	Double	75	
Ambient (%) / Ambient_	Double	30	
Diffuse (%) / Diffuse_	Double	40	
Specular (%) / Specular_	Double	40	
Shininess / Shininess	Double	80	
Smoothness_2 / Smoothness_2	Double	0.2	
Flatness / Flatness	Double	1	

Continued on next page

Table 333 – continued from previous page

Parameter / script name	Type	Default	Function
Linearity / Linearity	Double	0	
Levels / Levels	Integer	0	
Light-X / LightX	Double	2	
Light-Y / LightY	Double	-2	
Light-Z / LightZ	Double	2	
Normalize illumination / Normalize_illumination	Boolean	Off	
Open interactive preview / Open_interactive_preview	Boolean	Off	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 333 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.148 G'MIC Import data node

This documentation is for version 1.0 of G'MIC Import data.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Filename / Filename	N/A		
Normalize / Normalize	Boolean	On	

Continued on next page

Table 334 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.149 G'MIC Ink wash node

This documentation is for version 1.0 of G'MIC Ink wash.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Size / Size	Double	0.14	
Amplitude / Amplitude	Double	23	
Skip all other steps / Skip_all_other_steps	Boolean	Off	
Smoother sharpness / Smoother_sharpness	Double	0.5	
Smoother edge protection / Smoother_edge_protection	Double	0.54	
Smoother softness / Smoother_softness	Double	2.25	
Stretch contrast / Stretch_contrast	Choice	None	None Automatic Automatic & Contrast Mask Manual Controls
LN amplitude / LN_amplitude	Double	2	
LN size / LN_size	Double	6	
LN neighborhood-smoothness / LN_neighborhoodsmoothness	Double	5	
LN average-smoothness / LN_averagesmoothness	Double	20	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 335 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.150 G'MIC Inpaint holes node

This documentation is for version 1.0 of G'MIC Inpaint holes.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Maximal area / Maximal_area	Double	4	
Tolerance / Tolerance	Double	20	
Connectivity / Connectivity	Choice	High	Low High

Continued on next page

Table 336 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.151 G'MIC Inpaint morphological node

This documentation is for version 1.0 of G'MIC Inpaint morphological.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Mask color / Mask_color	Color	r: 1 g: 0 b: 0 a: 0	
Mask dilation / Mask_dilation	Integer	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.152 G'MIC Inpaint multi-scale node

This documentation is for version 1.0 of G'MIC Inpaint multi-scale.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of scales / Number_of_scales	Integer	0	
Patch size / Patch_size	Integer	9	
Number of iterations per scale / Number_of_iterations_per_scale	Integer	10	
Blend size / Blend_size	Integer	5	
Allow outer blending / Allow_outer_blending	Boolean	On	
Mask color / Mask_color	Color	r: 1 g: 0 b: 0 a: 0	
Mask dilation / Mask_dilation	Integer	0	
Preview progression while running / Preview_progression_while_running	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 338 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.153 G'MIC Inpaint patch-based node

This documentation is for version 1.0 of G'MIC Inpaint patch-based.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Patch size / Patch_size	Integer	7	
Lookup size / Lookup_size	Double	16	
Lookup factor / Lookup_factor	Double	0.1	
Blend size / Blend_size	Double	1.2	
Blend threshold / Blend_threshold	Double	0	
Blend decay / Blend_decay	Double	0.05	
Blend scales / Blend_scales	Integer	10	
Allow outer blending / Allow_outer_blending	Boolean	On	
Mask color / Mask_color	Color	r: 1 g: 0 b: 0 a: 0	

Continued on next page

Table 339 – continued from previous page

Parameter / script name	Type	Default	Function
Mask dilation / Mask_dilation	Integer	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.154 G'MIC Inpaint transport-diffusion node

This documentation is for version 1.0 of G'MIC Inpaint transport-diffusion.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness (%) / Smoothness_	Double	75	
Regularization / Regularization	Choice	Delaunay-oriented	Isotropic Delaunay-oriented Edge-oriented
Regularization iterations / Regularization_iterations	Integer	20	
Mask color / Mask_color	Color	r: 1 g: 0 b: 0 a: 0	
Mask dilation / Mask_dilation	Integer	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 340 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.155 G'MIC Instant collage consumer node

This documentation is for version 1.0 of G'MIC Instant collage consumer.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Image size / Image_size	Double	512	
Columns for collage / Columns_for_collage	Integer	4	
Label size / Label_size	Integer	16	
Output as / Output_as	Choice	Table	Table Multiple layers

Continued on next page

Table 341 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.156 G'MIC Instant collage pro node

This documentation is for version 1.0 of G'MIC Instant collage pro.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Image size / Image_size	Double	512	
Columns for collage / Columns_for_collage	Integer	4	
Label size / Label_size	Integer	16	
Output as / Output_as	Choice	Table	Table Multiple layers
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 342 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.157 G'MIC Instant consumer node

This documentation is for version 1.0 of G'MIC Instant consumer.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Continued on next page

Table 343 – continued from previous page

Parameter / script name	Type	Default	Function
Parameter / script name	Type	Default	Function
Preset / Preset	Choice	None	<p>None</p> <p>Polaroid PX-100UV+ Cold –</p> <p>Polaroid PX-100UV+ Cold -</p> <p>Polaroid PX-100UV+ Cold</p> <p>Polaroid PX-100UV+ Cold +</p> <p>Polaroid PX-100UV+ Cold ++</p> <p>Polaroid PX-100UV+ Cold +++</p> <p>Polaroid PX-100UV+ Warm –</p> <p>Polaroid PX-100UV+ Warm -</p> <p>Polaroid PX-100UV+ Warm</p> <p>Polaroid PX-100UV+ Warm +</p> <p>Polaroid PX-100UV+ Warm ++</p> <p>Polaroid PX-100UV+ Warm +++</p> <p>Polaroid PX-680 –</p> <p>Polaroid PX-680 -</p> <p>Polaroid PX-680</p> <p>Polaroid PX-680 +</p> <p>Polaroid PX-680 ++</p> <p>Polaroid PX-680 Cold –</p> <p>Polaroid PX-680 Cold -</p> <p>Polaroid PX-680 Cold</p> <p>Polaroid PX-680 Cold +</p> <p>Polaroid PX-680 Cold ++</p> <p>Polaroid PX-680 Cold ++a</p> <p>Polaroid PX-680 Warm –</p> <p>Polaroid PX-680 Warm -</p> <p>Polaroid PX-680 Warm</p> <p>Polaroid PX-680 Warm +</p> <p>Polaroid PX-680 Warm ++</p> <p>Polaroid PX-70 –</p> <p>Polaroid PX-70 -</p> <p>Polaroid PX-70</p> <p>Polaroid PX-70 +</p> <p>Polaroid PX-70 ++</p> <p>Polaroid PX-70 +++</p> <p>Polaroid PX-70 Cold –</p> <p>Polaroid PX-70 Cold -</p> <p>Polaroid PX-70 Cold</p> <p>Polaroid PX-70 Cold +</p> <p>Polaroid PX-70 Cold ++</p> <p>Polaroid PX-70 Warm –</p> <p>Polaroid PX-70 Warm -</p> <p>Polaroid PX-70 Warm</p> <p>Polaroid PX-70 Warm +</p> <p>Polaroid PX-70 Warm ++</p> <p>Polaroid Time Zero (Expired) —</p>
744			<p>Polaroid Time Zero (Expired)</p> <p>Polaroid Time Zero (Expired) -</p> <p>Polaroid Time Zero (Expired)</p> <p>Polaroid Time Zero (Expired) +</p>

Table 343 – continued from previous page

Parameter / script name	Type	Default	Function
Strength (%) / Strength_	Double	100	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Normalize colors / Normalize_colors	Choice	None	<p>None</p> <p>Pre-process</p> <p>Post-process</p> <p>Both</p>
Preview type / Preview_type	Choice	Full	<p>Full</p> <p>Forward horizontal</p> <p>Forward vertical</p> <p>Backward horizontal</p> <p>Backward vertical</p> <p>Duplicate top</p> <p>Duplicate left</p> <p>Duplicate bottom</p> <p>Duplicate right</p> <p>Duplicate horizontal</p> <p>Duplicate vertical</p> <p>Checkerboard</p> <p>Checkerboard inverse</p>
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	<p>Merged</p> <p>Layer 0</p> <p>Layer -1</p> <p>Layer -2</p> <p>Layer -3</p> <p>Layer -4</p> <p>Layer -5</p> <p>Layer -6</p> <p>Layer -7</p> <p>Layer -8</p> <p>Layer -9</p>

Continued on next page

Table 343 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.158 G'MIC Instant pro node

This documentation is for version 1.0 of G'MIC Instant pro.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Continued on next page

Table 344 – continued from previous page

Parameter / script name	Type	Default	Function
Parameter / script name	Type	Default	Function
Preset / Preset	Choice	None	<p>None</p> <p>Fuji FP-100c –</p> <p>Fuji FP-100c -</p> <p>Fuji FP-100c</p> <p>Fuji FP-100c +</p> <p>Fuji FP-100c ++</p> <p>Fuji FP-100c ++a</p> <p>Fuji FP-100c +++</p> <p>Fuji FP-100c Cool –</p> <p>Fuji FP-100c Cool -</p> <p>Fuji FP-100c Cool</p> <p>Fuji FP-100c Cool +</p> <p>Fuji FP-100c Cool ++</p> <p>Fuji FP-100c Negative –</p> <p>Fuji FP-100c Negative -</p> <p>Fuji FP-100c Negative</p> <p>Fuji FP-100c Negative +</p> <p>Fuji FP-100c Negative ++</p> <p>Fuji FP-100c Negative ++a</p> <p>Fuji FP-100c Negative +++</p> <p>Fuji FP-3000b –</p> <p>Fuji FP-3000b -</p> <p>Fuji FP-3000b</p> <p>Fuji FP-3000b +</p> <p>Fuji FP-3000b ++</p> <p>Fuji FP-3000b +++</p> <p>Fuji FP-3000b HC</p> <p>Fuji FP-3000b Negative –</p> <p>Fuji FP-3000b Negative -</p> <p>Fuji FP-3000b Negative</p> <p>Fuji FP-3000b Negative +</p> <p>Fuji FP-3000b Negative ++</p> <p>Fuji FP-3000b Negative +++</p> <p>Fuji FP-3000b Negative Early</p> <p>Polaroid 665 –</p> <p>Polaroid 665 -</p> <p>Polaroid 665</p> <p>Polaroid 665 +</p> <p>Polaroid 665 ++</p> <p>Polaroid 665 Negative -</p> <p>Polaroid 665 Negative</p> <p>Polaroid 665 Negative +</p> <p>Polaroid 665 Negative HC</p> <p>Polaroid 669 –</p> <p>Polaroid 669 -</p> <p>Polaroid 669</p>
748			<p>Polaroid 669 +</p> <p>Polaroid 669 ++</p> <p>Polaroid 669 +++</p> <p>Polaroid 669 Cold –</p>

Table 344 – continued from previous page

Parameter / script name	Type	Default	Function
Strength (%) / Strength_	Double	100	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Normalize colors / Normalize_colors	Choice	None	<p>None</p> <p>Pre-process</p> <p>Post-process</p> <p>Both</p>
Preview type / Preview_type	Choice	Full	<p>Full</p> <p>Forward horizontal</p> <p>Forward vertical</p> <p>Backward horizontal</p> <p>Backward vertical</p> <p>Duplicate top</p> <p>Duplicate left</p> <p>Duplicate bottom</p> <p>Duplicate right</p> <p>Duplicate horizontal</p> <p>Duplicate vertical</p> <p>Checkerboard</p> <p>Checkerboard inverse</p>
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	<p>Merged</p> <p>Layer 0</p> <p>Layer -1</p> <p>Layer -2</p> <p>Layer -3</p> <p>Layer -4</p> <p>Layer -5</p> <p>Layer -6</p> <p>Layer -7</p> <p>Layer -8</p> <p>Layer -9</p>

Continued on next page

Table 344 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.159 G'MIC Intarsia node

This documentation is for version 1.0 of G'MIC Intarsia.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Output directory / Output_directory	N/A		
Output HTML file / Output_HTML_file	String	intarsia.html	
Maximum image size / Maximum_image_size	Integer	512	
Maximum number of image colors / Maximum_number_of_image_colors	Integer	12	

Continued on next page

Table 345 – continued from previous page

Parameter / script name	Type	Default	Function
Starting point / Starting_point	Choice	Top right	Top left Top right Bottom left Bottom right
Loop method / Loop_method	Choice	Row by row	Row by row Column by column
Add comment area in HTML page / Add_comment_area_in_HTML_page	Boolean	On	
Preview progress (%) / Preview_progress_	Double	100	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 345 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.160 G'MIC Isophotes node

This documentation is for version 1.0 of G'MIC Isophotes.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Levels / Levels	Integer	8	
Smoothness / Smoothness	Double	0	
Filling / Filling	Choice	Colors	Transparent Colors

Continued on next page

Table 346 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 346 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.161 G'MIC JPEG artefacts node

This documentation is for version 1.0 of G'MIC JPEG artefacts.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Quality (%) / Quality_	Integer	50	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 347 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.162 G'MIC Kaleidoscope blended node

This documentation is for version 1.0 of G'MIC Kaleidoscope blended.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Angular tiles / Angular_tiles	Integer	10	
Smoothness / Smoothness	Double	0.5	
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.163 G'MIC Kaleidoscope polar node

This documentation is for version 1.0 of G'MIC Kaleidoscope polar.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Center (%) / Center_	Double	x: 0.5 y: 0.5	
X-offset (%) / Xoffset_	Double	0	
Y-offset (%) / Yoffset_	Double	0	
Radius cut / Radius_cut	Double	100	
Angle cut / Angle_cut	Double	10	
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 349 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.164 G'MIC Kaleidoscope symmetry node

This documentation is for version 1.0 of G'MIC Kaleidoscope symmetry.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Iterations / Iterations	Integer	4	
Angle / Angle	Double	0	
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror

Continued on next page

Table 350 – continued from previous page

Parameter / script name	Type	Default	Function
Symmetry sides / Symmetry_sides	Choice	Backward	Backward Forward Swap
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.165 G'MIC Kuwahara node

This documentation is for version 1.0 of G'MIC Kuwahara.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Iterations / Iterations	Integer	2	
Radius / Radius	Integer	5	
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 351 – continued from previous page

Parameter / script name	Type	Default	Function
Value action / Value_action	Choice	None	<p>None</p> <p>Cut</p> <p>Normalize</p>
Preview type / Preview_type	Choice	Full	<p>Full</p> <p>Forward horizontal</p> <p>Forward vertical</p> <p>Backward horizontal</p> <p>Backward vertical</p> <p>Duplicate top</p> <p>Duplicate left</p> <p>Duplicate bottom</p> <p>Duplicate right</p> <p>Duplicate horizontal</p> <p>Duplicate vertical</p> <p>Checkered</p> <p>Checkered inverse</p>
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	<p>Merged</p> <p>Layer 0</p> <p>Layer -1</p> <p>Layer -2</p> <p>Layer -3</p> <p>Layer -4</p> <p>Layer -5</p> <p>Layer -6</p> <p>Layer -7</p> <p>Layer -8</p> <p>Layer -9</p>
Resize Mode / Resize_Mode	Choice	Dynamic	<p>Fixed (Inplace)</p> <p>Dynamic</p> <p>Downsample 1/2</p> <p>Downsample 1/4</p> <p>Downsample 1/8</p> <p>Downsample 1/16</p>
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 351 – continued from previous page

Parameter / script name	Type	Default	Function
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.166 G'MIC Laplacian node

This documentation is for version 1.0 of G'MIC Laplacian.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	0	
Min threshold / Min_threshold	Double	0	
Max threshold / Max_threshold	Double	100	
Absolute value / Absolute_value	Boolean	Off	
Negative colors / Negative_colors	Boolean	Off	

Continued on next page

Table 352 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 352 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.167 G'MIC Lava node

This documentation is for version 1.0 of G'MIC Lava.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Perturbation / Perturbation	Integer	8	
Smoothness / Smoothness	Double	5	
Scale / Scale	Double	3	
Sharpness / Sharpness	Double	0	

Continued on next page

Table 353 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 353 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.168 G'MIC Lava lamp node

This documentation is for version 1.0 of G'MIC Lava lamp.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of key-frames / Number_of_keyframes	Integer	3	
Number of inter-frames / Number_of_interframes	Integer	30	
Smooth looping / Smooth_looping	Boolean	On	
Resolution / Resolution	Double	20	
Size / Size	Double	2	
Smoothness / Smoothness	Double	0.01	
Transparent background / Transparent_background	Boolean	Off	

Continued on next page

Table 354 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.169 G'MIC Layers to tiles node

This documentation is for version 1.0 of G'MIC Layers to tiles.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-tiles / Xtiles	Integer	0	
Y-tiles / Ytiles	Integer	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.170 G'MIC Light glow node

This documentation is for version 1.0 of G'MIC Light glow.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Double	30	
Amplitude / Amplitude	Double	0.5	
Mode / Mode	Choice	Overlay	Burn Dodge Freeze Grain merge Hard light Interpolation Lighten Multiply Overlay Reflect Soft light Stamp Value
Opacity / Opacity	Double	0.8	

Continued on next page

Table 356 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 356 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 356 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.171 G'MIC Light leaks node

This documentation is for version 1.0 of G'MIC Light leaks.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Leak type / Leak_type	Integer	0	
Angle / Angle	Double	0	
X-scale / Xscale	Double	1	
Y-scale / Yscale	Double	1	
Hue / Hue	Double	0	
Opacity / Opacity	Double	0.85	

Continued on next page

Table 357 – continued from previous page

Parameter / script name	Type	Default	Function
Blend mode / Blend_mode	Choice	Screen	<p> Normal Lighten Screen Dodge Add Darken Multiply Burn Overlay Soft light Hard light Difference Subtract Grain extract Grain merge Divide Hue Saturation Value </p>
Output as separate layers / Output_as_separate_layers	Boolean	On	
Preview type / Preview_type	Choice	Full	<p> Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse </p>
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 357 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.172 G'MIC Light patch node

This documentation is for version 1.0 of G'MIC Light patch.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Integer	5	
Darkness / Darkness	Double	0.7	
Lightness / Lightness	Double	2.5	
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 358 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.173 G'MIC Light rays node

This documentation is for version 1.0 of G'MIC Light rays.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Double	80	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Length / Length	Double	1	
Attenuation / Attenuation	Double	0.5	
Transparency / Transparency	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.174 G'MIC Lightning node

This documentation is for version 1.0 of G'MIC Lightning.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of streaks / Number_of_streaks	Integer	20	
Size (%) / Size_	Double	90	
Resolution / Resolution	Integer	256	
Randomness / Randomness	Double	3	
Smoothness / Smoothness	Double	1.5	
Balance / Balance	Double	0.75	
Color / Color	Color	r: 1 g: 1 b: 1 a: 1	
Seed / Seed	Integer	0	
XY-coordinates (%) / XYcoordinates_	Double	x: 0.5 y: 0.05	
Angle (deg) / Angle_deg	Double	0	
Thickness (px.) / Thickness_px	Integer	6	
Blur / Blur	Double	0.2	
Min offset (%) / Min_offset_	Double	25	
Max offset (%) / Max_offset_	Double	60	
Min length (%) / Min_length_	Double	95	
Max length (%) / Max_length_	Double	100	
Min angle deviation (deg) / Min_angle_deviation_deg	Double	30	
Max angle deviation (deg) / Max_angle_deviation_deg	Double	40	
Thickness factor / Thickness_factor	Double	-0.25	
Blur factor / Blur_factor	Double	-0.1	
Opacity factor / Opacity_factor	Double	-0.2	

Continued on next page

Table 360 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.175 G'MIC Linify node

This documentation is for version 1.0 of G'MIC Linify.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Double	40	
Spreading / Spreading	Double	2	
Resolution (%) / Resolution_	Double	40	
Line opacity / Line_opacity	Double	10	
Line precision / Line_precision	Integer	24	
Color mode / Color_mode	Choice	Subtractive	Subtractive Additive
Preview progression while running / Preview_progression_while_running	Boolean	On	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 361 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.176 G'MIC Lissajous node

This documentation is for version 1.0 of G'MIC Lissajous.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Resolution / Resolution	Integer	4096	
X-size / Xsize	Double	0.9	
Y-size / Ysize	Double	0.9	
Z-size / Zsize	Double	3	
X-multiplier / Xmultiplier	Double	8	
Y-multiplier / Ymultiplier	Double	7	
Z-multiplier / Zmultiplier	Double	0	
X-offset / Xoffset	Double	0	
Y-offset / Yoffset	Double	0	
Z-offset / Zoffset	Double	0	
X-angle / Xangle	Double	0	
Y-angle / Yangle	Double	0	
Z-angle / Zangle	Double	0	
Thickness / Thickness	Double	0	
Color / Color	Color	r: 1 g: 1 b: 1 a: 1	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 362 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.177 G'MIC Local normalization node

This documentation is for version 1.0 of G'MIC Local normalization.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	2	
Radius / Radius	Integer	6	
Neighborhood smoothness / Neighborhood_smoothness	Double	5	
Average smoothness / Average_smoothness	Double	20	
Constrain values / Constrain_values	Boolean	On	

Continued on next page

Table 363 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	YCbCr [luminance]	<p>All</p> <p>RGBA [all]</p> <p>RGB [all]</p> <p>RGB [red]</p> <p>RGB [green]</p> <p>RGB [blue]</p> <p>RGBA [alpha]</p> <p>Linear RGB [all]</p> <p>Linear RGB [red]</p> <p>Linear RGB [green]</p> <p>Linear RGB [blue]</p> <p>YCbCr [luminance]</p> <p>YCbCr [blue-red chrominances]</p> <p>YCbCr [blue chrominance]</p> <p>YCbCr [red chrominance]</p> <p>YCbCr [green chrominance]</p> <p>Lab [lightness]</p> <p>Lab [ab-chrominances]</p> <p>Lab [a-chrominance]</p> <p>Lab [b-chrominance]</p> <p>Lch [ch-chrominances]</p> <p>Lch [c-chrominance]</p> <p>Lch [h-chrominance]</p> <p>HSV [hue]</p> <p>HSV [saturation]</p> <p>HSV [value]</p> <p>HSI [intensity]</p> <p>HSL [lightness]</p> <p>CMYK [cyan]</p> <p>CMYK [magenta]</p> <p>CMYK [yellow]</p> <p>CMYK [key]</p> <p>YIQ [luma]</p> <p>YIQ [chromas]</p>

Continued on next page

Table 363 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 363 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.178 G'MIC Local orientation node

This documentation is for version 1.0 of G'MIC Local orientation.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	0	
Min threshold / Min_threshold	Double	0	
Max threshold / Max_threshold	Double	100	
Negative colors / Negative_colors	Boolean	Off	

Continued on next page

Table 364 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 364 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 364 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.179 G'MIC Local processing node

This documentation is for version 1.0 of G'MIC Local processing.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Action / Action	Choice	Normalize	Normalize Equalize
Strength (%) / Strength_	Double	75	
Neighborhood size (%) / Neighborhood_size_	Double	10	
Overlap (%) / Overlap_	Double	50	
Regularization (%) / Regularization_	Double	20	
Process channels individually / Process_channels_individually	Boolean	Off	

Continued on next page

Table 365 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	Linear RGB [all]	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 365 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 365 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.180 G'MIC Lomo node

This documentation is for version 1.0 of G'MIC Lomo.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Vignette size / Vignette_size	Double	20	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 366 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.181 G'MIC Lylejk's painting node

This documentation is for version 1.0 of G'MIC Lylejk's painting.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Iterations / Iterations	Integer	10	
Abstraction / Abstraction	Integer	2	
Radius / Radius	Integer	4	
Canvas / Canvas	Double	10	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 367 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.182 G'MIC Magic details node

This documentation is for version 1.0 of G'MIC Magic details.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	6	
Spatial scale / Spatial_scale	Double	3	
Value scale / Value_scale	Double	15	
Edges / Edges	Double	-0.5	
Smoothness / Smoothness	Double	2	

Continued on next page

Table 368 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	HSL [lightness]	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 368 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 368 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.183 G'MIC Make seamless diffusion node

This documentation is for version 1.0 of G'MIC Make seamless diffusion.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Equalize light / Equalize_light	Double	0	
Preview original / Preview_original	Boolean	Off	
Tiled preview / Tiled_preview	Choice	2x2	None 2x1 1x2 2x2 3x3 4x4

Continued on next page

Table 369 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 369 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.184 G'MIC Make seamless patch-based node

This documentation is for version 1.0 of G'MIC Make seamless patch-based.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Frame size / Frame_size	Integer	32	
Patch size / Patch_size	Integer	9	
Blend size / Blend_size	Integer	0	
Frame type / Frame_type	Choice	Outer	Inner Outer
Equalize light / Equalize_light	Double	100	
Preview original / Preview_original	Boolean	Off	

Continued on next page

Table 370 – continued from previous page

Parameter / script name	Type	Default	Function
Tiled preview / Tiled_preview	Choice	2x2	None 2x1 1x2 2x2 3x3 4x4
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 370 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.185 G'MIC Mandelbrot Julia sets node

This documentation is for version 1.0 of G'MIC Mandelbrot Julia sets.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Fractal set / Fractal_set	Choice	Mandelbrot	Mandelbrot Julia
Iterations / Iterations	Integer	1024	
X-seed (julia) / Xseed_julia	Double	0.317	

Continued on next page

Table 371 – continued from previous page

Parameter / script name	Type	Default	Function
Y-seed (julia) / Yseed_julia	Double	0.03	
Number of colors / Number_of_colors	Integer	16	
Smoothness / Smoothness	Integer	8	
Seed / Seed	Integer	255	
Zoom center / Zoom_center	Double	x: 0.5 y: 0.5	
Zoom factor / Zoom_factor	Double	0.25	
Zoom in / Zoom_in	Boolean	Off	
Re-center / Recenter	Boolean	Off	
Zoom out / Zoom_out	Boolean	Off	
Display coordinates / Display_coordinates	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 371 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.186 G'MIC Marble node

This documentation is for version 1.0 of G'MIC Marble.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Image weight / Image_weight	Double	0.5	
Pattern weight / Pattern_weight	Double	1	
Pattern angle / Pattern_angle	Double	0	
Amplitude / Amplitude	Double	0	
Sharpness / Sharpness	Double	0.4	
Anisotropy / Anisotropy	Double	0.6	
Alpha / Alpha	Double	0.6	
Sigma / Sigma	Double	1.1	
Cut low / Cut_low	Double	0	
Cut high / Cut_high	Double	100	

Continued on next page

Table 372 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.187 G'MIC Maze node

This documentation is for version 1.0 of G'MIC Maze.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Cell size / Cell_size	Integer	24	
Thickness / Thickness	Integer	1	
Masking / Masking	Choice	None	None Render on dark areas Render on white areas
Preserve image dimension / Preserve_image_dimension	Boolean	On	
Maze type / Maze_type	Choice	Dark walls	Dark walls White walls
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 373 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.188 G'MIC Mighty details node

This documentation is for version 1.0 of G'MIC Mighty details.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	25	
Details amount / Details_amount	Double	1	
Details scale / Details_scale	Double	25	
Details smoothness / Details_smoothness	Integer	1	

Continued on next page

Table 374 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	YCbCr [luminance]	<p>All</p> <p>RGBA [all]</p> <p>RGB [all]</p> <p>RGB [red]</p> <p>RGB [green]</p> <p>RGB [blue]</p> <p>RGBA [alpha]</p> <p>Linear RGB [all]</p> <p>Linear RGB [red]</p> <p>Linear RGB [green]</p> <p>Linear RGB [blue]</p> <p>YCbCr [luminance]</p> <p>YCbCr [blue-red chrominances]</p> <p>YCbCr [blue chrominance]</p> <p>YCbCr [red chrominance]</p> <p>YCbCr [green chrominance]</p> <p>Lab [lightness]</p> <p>Lab [ab-chrominances]</p> <p>Lab [a-chrominance]</p> <p>Lab [b-chrominance]</p> <p>Lch [ch-chrominances]</p> <p>Lch [c-chrominance]</p> <p>Lch [h-chrominance]</p> <p>HSV [hue]</p> <p>HSV [saturation]</p> <p>HSV [value]</p> <p>HSI [intensity]</p> <p>HSL [lightness]</p> <p>CMYK [cyan]</p> <p>CMYK [magenta]</p> <p>CMYK [yellow]</p> <p>CMYK [key]</p> <p>YIQ [luma]</p> <p>YIQ [chromas]</p>

Continued on next page

Table 374 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 374 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.189 G'MIC Mineral mosaic node

This documentation is for version 1.0 of G'MIC Mineral mosaic.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Double	1	
Area / Area	Double	2	
Smoothness / Smoothness	Double	1	
Shade strength / Shade_strength	Double	100	
Shade angle / Shade_angle	Double	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 375 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.190 G'MIC Ministeck node

This documentation is for version 1.0 of G'MIC Ministeck.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Deverny.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of colors / Number_of_colors	Integer	8	
Resolution (px) / Resolution_px	Integer	64	
Piece size (px) / Piece_size_px	Integer	8	
Piece complexity / Piece_complexity	Integer	2	
Relief amplitude / Relief_amplitude	Double	100	

Continued on next page

Table 376 – continued from previous page

Parameter / script name	Type	Default	Function
Relief size / Relief_size	Double	0.3	
Add 1px outline / Add_1px_outline	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.191 G'MIC Mixer CMYK node

This documentation is for version 1.0 of G'MIC Mixer CMYK.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Cyan factor / Cyan_factor	Double	1	
Cyan shift / Cyan_shift	Double	0	
Cyan smoothness / Cyan_smoothness	Double	0	
Magenta factor / Magenta_factor	Double	1	
Magenta shift / Magenta_shift	Double	0	
Magenta smoothness / Magenta_smoothness	Double	0	
Yellow factor / Yellow_factor	Double	1	
Yellow shift / Yellow_shift	Double	0	
Yellow smoothness / Yellow_smoothness	Double	0	
Key factor / Key_factor	Double	1	
Key shift / Key_shift	Double	0	
Key smoothness / Key_smoothness	Double	0	
Tones range / Tones_range	Choice	All tones	All tones Shadows Mid-tones Highlights
Tones smoothness / Tones_smoothness	Double	2	

Continued on next page

Table 377 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 377 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.192 G'MIC Mixer HSV node

This documentation is for version 1.0 of G'MIC Mixer HSV.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Hue factor / Hue_factor	Double	1	
Hue shift / Hue_shift	Double	0	
Hue smoothness / Hue_smoothness	Double	0	
Saturation factor / Saturation_factor	Double	1	
Saturation shift / Saturation_shift	Double	0	
Saturation smoothness / Saturation_smoothness	Double	0	
Value factor / Value_factor	Double	1	
Value shift / Value_shift	Double	0	
Value smoothness / Value_smoothness	Double	0	

Continued on next page

Table 378 – continued from previous page

Parameter / script name	Type	Default	Function
Tones range / Tones_range	Choice	All tones	All tones Shadows Mid-tones Highlights
Tones smoothness / Tones_smoothness	Double	2	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 378 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.193 G'MIC Mixer Lab node

This documentation is for version 1.0 of G'MIC Mixer Lab.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Lightness factor / Lightness_factor	Double	1	
Lightness shift / Lightness_shift	Double	0	
Lightness smoothness / Lightness_smoothness	Double	0	
A-color factor / Acolor_factor	Double	1	

Continued on next page

Table 379 – continued from previous page

Parameter / script name	Type	Default	Function
A-color shift / Acolor_shift	Double	0	
A-color smoothness / Acolor_smoothness	Double	0	
B-color factor / Bcolor_factor	Double	1	
B-color shift / Bcolor_shift	Double	0	
B-color smoothness / Bcolor_smoothness	Double	0	
Tones range / Tones_range	Choice	All tones	All tones Shadows Mid-tones Highlights
Tones smoothness / Tones_smoothness	Double	2	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 379 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.194 G'MIC Mixer RGB node

This documentation is for version 1.0 of G'MIC Mixer RGB.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Red factor / Red_factor	Double	1	
Red shift / Red_shift	Double	0	
Red smoothness / Red_smoothness	Double	0	
Green factor / Green_factor	Double	1	
Green shift / Green_shift	Double	0	
Green smoothness / Green_smoothness	Double	0	
Blue factor / Blue_factor	Double	1	
Blue shift / Blue_shift	Double	0	
Blue smoothness / Blue_smoothness	Double	0	
Tones range / Tones_range	Choice	All tones	All tones Shadows Mid-tones Highlights
Tones smoothness / Tones_smoothness	Double	2	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 380 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.195 G'MIC Mixer YCbCr node

This documentation is for version 1.0 of G'MIC Mixer YCbCr.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Luminance factor / Luminance_factor	Double	1	
Luminance shift / Luminance_shift	Double	0	
Luminance smoothness / Luminance_smoothness	Double	0	
Blue chroma factor / Blue_chroma_factor	Double	1	
Blue chroma shift / Blue_chroma_shift	Double	0	
Blue chroma smoothness / Blue_chroma_smoothness	Double	0	
Red chroma factor / Red_chroma_factor	Double	1	
Red chroma shift / Red_chroma_shift	Double	0	
Red chroma smoothness / Red_chroma_smoothness	Double	0	
Tones range / Tones_range	Choice	All tones	All tones Shadows Mid-tones Highlights
Tones smoothness / Tones_smoothness	Double	2	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 381 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.196 G'MIC Montage node

This documentation is for version 1.0 of G'MIC Montage.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Montage type / Montage_type	Choice	Auto	Auto Custom layout Horizontal Vertical Horizontal array Vertical array
Custom layout / Custom_layout	String	V(H(0,1),H(2,V(3,4)))	
Merging mode / Merging_mode	Choice	Scaled	Aligned Scaled
Centering / scale / Centering__scale	Double	0.5	
Padding (px) / Padding_px	Integer	0	
Frame (px) / Frame_px	Integer	0	
Frame color / Frame_color	Color	r: 0 g: 0 b: 0 a: 0	
Angle / Angle	Double	0	
Angle variations / Angle_variations	Double	0	
Cycle layers / Cycle_layers	Integer	0	
Revert layer order / Revert_layer_order	Boolean	Off	
Output as / Output_as	Choice	Single layer	Single layer Multiple layers

Continued on next page

Table 382 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.197 G'MIC Morph layers node

This documentation is for version 1.0 of G'MIC Morph layers.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Inter-frames / Interframes	Integer	10	
Smoothness / Smoothness	Double	0.2	
Precision / Precision	Double	0.1	
Revert layers / Revert_layers	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 383 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.198 G'MIC Morphological filter node

This documentation is for version 1.0 of G'MIC Morphological filter.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Action / Action	Choice	Erosion	Erosion Dilation Opening Closing Original - Erosion Dilation - Original Original - Opening Closing - Original Original - (Opening + Closing)/2 Closing - Opening
Kernel / Kernel	Choice	Square	Square Octagonal Circular Custom
Size / Size	Integer	5	

Continued on next page

Table 384 – continued from previous page

Parameter / script name	Type	Default	Function
Custom kernel / Custom_kernel	String	1,0,1; 0,1,0; 1,0,1	
Negative / Negative	Boolean	Off	
Process transparency / Process_transparency	Boolean	Off	
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 384 – continued from previous page

Parameter / script name	Type	Default	Function
Value action / Value_action	Choice	None	<p>None Cut Stretch</p>
Preview type / Preview_type	Choice	Full	<p>Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse</p>
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	<p>Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9</p>
Resize Mode / Resize_Mode	Choice	Dynamic	<p>Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16</p>
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 384 – continued from previous page

Parameter / script name	Type	Default	Function
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.199 G'MIC Mosaic node

This documentation is for version 1.0 of G'MIC Mosaic.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density (%) / Density_	Double	50	

Continued on next page

Table 385 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 385 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 385 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.200 G'MIC Multiscale operator node

This documentation is for version 1.0 of G'MIC Multiscale operator.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of scales / Number_of_scales	Integer	4	
Starting scale (%) / Starting_scale_	Double	25	
Ending scale (%) / Ending_scale_	Double	100	
Non-linearity / Nonlinearity	Double	0	
Rescaling / Rescaling	Choice	Lanczsos	Bloc Linear Cubic Lanczsos
X-centering / Xcentering	Double	0.5	
Y-centering / Ycentering	Double	0.5	
Angle / Angle	Double	0	

Continued on next page

Table 386 – continued from previous page

Parameter / script name	Type	Default	Function
Enable interpolated motion / Enable_interpolated_motion	Boolean	Off	
Ending X-centering / Ending_Xcentering	Double	0.5	
Ending Y-centering / Ending_Ycentering	Double	0.5	
Ending angle / Ending_angle	Double	0	
G'MIC operator / GMIC_operator	String		
Return scaling / Return_scaling	Choice	None	None Bloc Linear Cubic Lanczos
Lock return scaling to source layer / Lock_return_scaling_to_source_layer	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 386 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.201 G'MIC Negative collage color node

This documentation is for version 1.0 of G'MIC Negative collage color.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Image size / Image_size	Double	512	
Columns for collage / Columns_for_collage	Integer	4	
Label size / Label_size	Integer	16	
Output as / Output_as	Choice	Table	Table Multiple layers

Continued on next page

Table 387 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.202 G'MIC Negative collage new node

This documentation is for version 1.0 of G'MIC Negative collage new.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Image size / Image_size	Double	512	
Columns for collage / Columns_for_collage	Integer	4	
Label size / Label_size	Integer	16	
Output as / Output_as	Choice	Table	Table Multiple layers
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 388 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.203 G'MIC Negative collage old node

This documentation is for version 1.0 of G'MIC Negative collage old.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Image size / Image_size	Double	512	
Columns for collage / Columns_for_collage	Integer	4	
Label size / Label_size	Integer	16	
Output as / Output_as	Choice	Table	Table Multiple layers

Continued on next page

Table 389 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.204 G'MIC Negative color node

This documentation is for version 1.0 of G'MIC Negative color.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Preset / Preset	Choice	None	<p>None</p> <p>Agfa Ultra Color 100</p> <p>Agfa Vista 200</p> <p>Fuji Superia 200</p> <p>Fuji Superia HG 1600</p> <p>Fuji Superia Reala 100</p> <p>Fuji Superia X-Tra 800</p> <p>Kodak Elite 100 XPRO</p> <p>Kodak Elite Color 200</p> <p>Kodak Elite Color 400</p> <p>Kodak Portra 160 NC</p> <p>Kodak Portra 160 VC</p> <p>Lomography Redscale 100</p>
Strength (%) / Strength_	Double	100	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Normalize colors / Normalize_colors	Choice	None	<p>None</p> <p>Pre-process</p> <p>Post-process</p> <p>Both</p>

Continued on next page

Table 390 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 390 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.205 G'MIC Negative new node

This documentation is for version 1.0 of G'MIC Negative new.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Preset / Preset	Choice	None	None Fuji 160C Fuji 400H Fuji 800Z Fuji Ilford HP5 Kodak Portra 160 Kodak Portra 400 Kodak Portra 800 Kodak TMAX 3200 Kodak TRI-X 400
Effect / Effect	Choice	Standard	Low Standard High Higher

Continued on next page

Table 391 – continued from previous page

Parameter / script name	Type	Default	Function
Strength (%) / Strength_	Double	100	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Normalize colors / Normalize_colors	Choice	None	<p>None</p> <p>Pre-process</p> <p>Post-process</p> <p>Both</p>
Preview type / Preview_type	Choice	Full	<p>Full</p> <p>Forward horizontal</p> <p>Forward vertical</p> <p>Backward horizontal</p> <p>Backward vertical</p> <p>Duplicate top</p> <p>Duplicate left</p> <p>Duplicate bottom</p> <p>Duplicate right</p> <p>Duplicate horizontal</p> <p>Duplicate vertical</p> <p>Checkerboard</p> <p>Checkerboard inverse</p>
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	<p>Merged</p> <p>Layer 0</p> <p>Layer -1</p> <p>Layer -2</p> <p>Layer -3</p> <p>Layer -4</p> <p>Layer -5</p> <p>Layer -6</p> <p>Layer -7</p> <p>Layer -8</p> <p>Layer -9</p>

Continued on next page

Table 391 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.206 G'MIC Negative old node

This documentation is for version 1.0 of G'MIC Negative old.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Preset / Preset	Choice	None	<p>None</p> <p>Fuji Ilford Delta 3200</p> <p>Fuji Neopan 1600</p> <p>Fuji Superia 100</p> <p>Fuji Superia 400</p> <p>Fuji Superia 800</p> <p>Fuji Superia 1600</p> <p>Kodak Portra 160 NC</p> <p>Kodak Portra 160 VC</p> <p>Kodak Portra 400 NC</p> <p>Kodak Portra 400 UC</p> <p>Kodak Portra 400 VC</p>
Effect / Effect	Choice	Standard	<p>Low</p> <p>Standard</p> <p>High</p> <p>Higher</p>
Strength (%) / Strength_	Double	100	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Normalize colors / Normalize_colors	Choice	None	<p>None</p> <p>Pre-process</p> <p>Post-process</p> <p>Both</p>

Continued on next page

Table 392 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 392 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.207 G'MIC Neon lightning node

This documentation is for version 1.0 of G'MIC Neon lightning.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Source (%) / Source_	Double	x: 0.5 y: 0.5	
R0 / R0	Double	0	
Destination (%) / Destination_	Double	x: 0.5 y: 0.5	
R1 / R1	Double	100	
Density / Density	Integer	50	
Glow / Glow	Double	0.7	
Thickness / Thickness	Double	3	
Color / Color	Color	r: 0.509804 g: 0.313726 b: 0.196078 a: 0.196078	
Color dispersion / Color_dispersiion	Double	0.25	
Transparency / Transparency	Double	0	

Continued on next page

Table 393 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.208 G'MIC Noise additive node

This documentation is for version 1.0 of G'MIC Noise additive.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	10	
Noise type / Noise_type	Choice	Gaussian	Gaussian Uniform Salt and pepper Poisson
Channel(s) / Channels	Choice	All	All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas]

Continued on next page

Table 394 – continued from previous page

Parameter / script name	Type	Default	Function
Value action / Value_action	Choice	Cut	<p>None</p> <p>Cut</p> <p>Normalize</p>
Preview type / Preview_type	Choice	Full	<p>Full</p> <p>Forward horizontal</p> <p>Forward vertical</p> <p>Backward horizontal</p> <p>Backward vertical</p> <p>Duplicate top</p> <p>Duplicate left</p> <p>Duplicate bottom</p> <p>Duplicate right</p> <p>Duplicate horizontal</p> <p>Duplicate vertical</p> <p>Checkered</p> <p>Checkered inverse</p>
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	<p>Merged</p> <p>Layer 0</p> <p>Layer -1</p> <p>Layer -2</p> <p>Layer -3</p> <p>Layer -4</p> <p>Layer -5</p> <p>Layer -6</p> <p>Layer -7</p> <p>Layer -8</p> <p>Layer -9</p>
Resize Mode / Resize_Mode	Choice	Dynamic	<p>Fixed (Inplace)</p> <p>Dynamic</p> <p>Downsample 1/2</p> <p>Downsample 1/4</p> <p>Downsample 1/8</p> <p>Downsample 1/16</p>
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 394 – continued from previous page

Parameter / script name	Type	Default	Function
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.209 G'MIC Noise spread node

This documentation is for version 1.0 of G'MIC Noise spread.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-variations / Xvariations	Double	4	
Y-variations / Yvariations	Double	4	

Continued on next page

Table 395 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Value action / Value_action	Choice	None	<p> None Cut Normalize </p>

Continued on next page

Table 395 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 395 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.210 G'MIC Old photograph node

This documentation is for version 1.0 of G'MIC Old photograph.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Vignette strength / Vignette_strength	Double	200	
Vignette min radius / Vignette_min_radius	Double	50	
Vignette max radius / Vignette_max_radius	Double	85	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 396 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.211 G'MIC Old-movie stripes node

This documentation is for version 1.0 of G'MIC Old-movie stripes.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Frequency / Frequency	Double	10	

Continued on next page

Table 397 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Value action / Value_action	Choice	None	<p> None Cut Normalize </p>

Continued on next page

Table 397 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 397 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.212 G'MIC Oldschool 8bits node

This documentation is for version 1.0 of G'MIC Oldschool 8bits.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Scale / Scale	Double	25	
Dithering / Dithering	Double	800	
Levels / Levels	Integer	16	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse

Continued on next page

Table 398 – continued from previous page

Parameter / script name	Type	Default	Function
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.213 G'MIC Op art node

This documentation is for version 1.0 of G'MIC Op art.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Shape / Shape	Choice	Circles	Custom layers Circles Squares Diamonds Triangles Horizontal stripes Vertical stripes Balls Hearts Stars Arrows Truchet Circles (outline) Squares (outline) Diamonds (outline) Triangles (outline) Hearts (outline) Stars (outline) Arrows (outline)
Number of scales / Number_of_scales	Integer	16	
Resolution / Resolution	Double	10	
Zoom factor / Zoom_factor	Integer	2	
Minimal size / Minimal_size	Double	5	
Maximal size / Maximal_size	Double	90	
Stencil type / Stencil_type	Choice	Black & white	Black & white RGB Color

Continued on next page

Table 399 – continued from previous page

Parameter / script name	Type	Default	Function
Allow angle / Allow_angle	Choice	0 deg.	0 deg. 90 deg. 180 deg.
Negative / Negative	Boolean	On	
Antialiasing / Antialiasing	Boolean	On	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 399 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.214 G'MIC Pack node

This documentation is for version 1.0 of G'MIC Pack.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Order by / Order_by	Choice	Maximum dimension	Width Height Maximum dimension Area
Tends to be square / Tends_to_be_square	Boolean	Off	
Force transparency / Force_transparency	Boolean	On	
Output coordinates file / Output_coordinates_file	Boolean	Off	

Continued on next page

Table 400 – continued from previous page

Parameter / script name	Type	Default	Function
Output folder / Output_folder	N/A		
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.215 G'MIC Pack sprites node

This documentation is for version 1.0 of G'MIC Pack sprites.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Number of scales / Number_of_scales	Integer	5	
Minimal scale (%) / Minimal_scale_	Double	25	
Allow angle / Allow_angle	Choice	Any	0 deg. 180 deg. 90 deg. Any
Spacing / Spacing	Integer	1	
Precision / Precision	Integer	7	
Masking / Masking	Choice	No mask- ing	No masking Mask as bottom layer
Width / Width	Integer	512	
Height / Height	Integer	512	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 401 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.216 G'MIC Painting node

This documentation is for version 1.0 of G'MIC Painting.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Abstraction / Abstraction	Integer	5	
Details scale / Details_scale	Double	2.5	
Color / Color	Double	1.5	
Smoothness / Smoothness	Double	50	
Sharpen shades / Sharpen_shades	Boolean	On	

Continued on next page

Table 402 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 402 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.217 G'MIC Paper texture node

This documentation is for version 1.0 of G'MIC Paper texture.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 403 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 403 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.218 G'MIC Pen drawing node

This documentation is for version 1.0 of G'MIC Pen drawing.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	10	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 404 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.219 G'MIC Pencil node

This documentation is for version 1.0 of G'MIC Pencil.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Size / Size	Double	0.3	
Amplitude / Amplitude	Double	60	
Hue / Hue	Double	0	
Saturation / Saturation	Double	0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 405 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.220 G'MIC Pencil portrait node

This documentation is for version 1.0 of G'MIC Pencil portrait.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Stroke length / Stroke_length	Double	30	
Stroke angle / Stroke_angle	Double	120	
Contour threshold / Contour_threshold	Double	1	
Opacity / Opacity	Double	0.5	
Color / Color	Color	r: 0.564706 g: 0.309804 b: 0.0823529 a: 0.0823529	

Continued on next page

Table 406 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 406 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.221 G'MIC Perspective node

This documentation is for version 1.0 of G'MIC Perspective.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-angle / Xangle	Double	1.73	
Y-angle / Yangle	Double	0	
Zoom / Zoom	Double	1	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
X-offset / Xoffset	Double	0	
Y-offset / Yoffset	Double	0	
Boundary / Boundary	Choice	Periodic	Transparent Nearest Periodic Mirror

Continued on next page

Table 407 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.222 G'MIC PictureFX node

This documentation is for version 1.0 of G'MIC PictureFX.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Preset / Preset	Choice	None	<p>None</p> <p>AnalogFX - Anno 1870 Color</p> <p>AnalogFX - Old Style I</p> <p>AnalogFX - Old Style II</p> <p>AnalogFX - Old Style III</p> <p>AnalogFX - Sepia Color</p> <p>AnalogFX - Soft Sepia I</p> <p>AnalogFX - Soft Sepia II</p> <p>GoldFX - Perfect Sunset 01min</p> <p>GoldFX - Perfect Sunset 05min</p> <p>GoldFX - Perfect Sunset 10min</p> <p>GoldFX - Spring breeze</p> <p>GoldFX - Bright spring breeze</p> <p>GoldFX - Summer heat</p> <p>GoldFX - Bright summer heat</p> <p>GoldFX - Hot summer heat</p> <p>TechnicalFX - Backlight filter</p> <p>ZilverFX - B&W Solarization</p> <p>ZilverFX - Infrared</p> <p>ZilverFX - Vintage B&W</p>
Strength (%) / Strength_	Double	100	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Normalize colors / Normalize_colors	Choice	None	<p>None</p> <p>Pre-process</p> <p>Post-process</p> <p>Both</p>

Continued on next page

Table 408 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 408 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.223 G'MIC PictureFX collage node

This documentation is for version 1.0 of G'MIC PictureFX collage.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Image size / Image_size	Double	512	
Columns for collage / Columns_for_collage	Integer	4	
Label size / Label_size	Integer	16	
Output as / Output_as	Choice	Table	Table Multiple layers

Continued on next page

Table 409 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.224 G'MIC Pixel sort node

This documentation is for version 1.0 of G'MIC Pixel sort.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Order / Order	Choice	Increasing	Decreasing Increasing
Axis / Axis	Choice	X-axis	X-axis Y-axis X-axis then Y-axis Y-axis then X-axis
Sorting criterion / Sorting_criterion	Choice	Red	Red Green Blue Intensity Luminance Lightness Hue Saturation Minimum Maximum Random
Mask by / Mask_by	Choice	Criterion	Bottom layer Criterion Contours Random
Lower mask threshold (%) / Lower_mask_threshold_	Double	0	
Higher mask threshold (%) / Higher_mask_threshold_	Double	100	
Mask smoothness (%) / Mask_smoothness_	Double	0	
Invert mask / Invert_mask	Boolean	Off	
Preview mask / Preview_mask	Boolean	Off	

Continued on next page

Table 410 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.225 G'MIC Plaid node

This documentation is for version 1.0 of G'MIC Plaid.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Line / Line	Double	50	
Number of angles / Number_of_angles	Integer	2	
Starting angle / Starting_angle	Double	0	
Angle range / Angle_range	Double	90	
Smoothness / Smoothness	Double	1	
Sharpen / Sharpen	Double	300	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.226 G'MIC Plasma node

This documentation is for version 1.0 of G'MIC Plasma.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Alpha / Alpha	Double	0.5	
Beta / Beta	Double	0	
Scale / Scale	Integer	8	
Randomize / Randomize	Boolean	Off	
Transparency / Transparency	Boolean	Off	
Color balance / Color_balance	Color	r: 0.501961 g: 0.501961 b: 0.501961 a: 0.501961	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 412 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.227 G'MIC Polar transform node

This documentation is for version 1.0 of G'MIC Polar transform.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Preset / Preset	Choice	Custom transform	Custom transform Inverse radius Swap radius/angle
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Radius / Radius	String	r + R/10*cos(a*5)	
Angle / Angle	String	a	

Continued on next page

Table 413 – continued from previous page

Parameter / script name	Type	Default	Function
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.228 G'MIC Polaroid node

This documentation is for version 1.0 of G'MIC Polaroid.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Frame size / Frame_size	Integer	10	
Bottom size / Bottom_size	Integer	20	
X-shadow / Xshadow	Double	0	
Y-shadow / Yshadow	Double	0	
Smoothness / Smoothness	Double	3	
Curvature / Curvature	Double	0	
Angle / Angle	Double	20	
Vignette strength / Vignette_strength	Double	50	
Vignette min radius / Vignette_min_radius	Double	70	
Vignette max radius / Vignette_max_radius	Double	95	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 414 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.229 G'MIC Polka dots node

This documentation is for version 1.0 of G'MIC Polka dots.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Size / Size	Double	80	
Density / Density	Double	20	
First offset / First_offset	Double	50	
Second offset / Second_offset	Double	50	
Angle / Angle	Double	0	
Aliasing / Aliasing	Double	0.5	
Shading / Shading	Double	0.1	
Opacity / Opacity	Double	1	
Color / Color	Color	r: 1 g: 0 b: 0 a: 0	

Continued on next page

Table 415 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.230 G'MIC Polygonize delaunay node

This documentation is for version 1.0 of G'MIC Polygonize delaunay.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density (%) / Density_	Double	40	
Edges / Edges	Double	5	
Boundaries (%) / Boundaries_	Double	75	
Smoothness / Smoothness	Double	0.5	
Filling / Filling	Choice	Average	Black White Random Average Linear
Outline (%) / Outline_	Double	50	
Outline color / Outline_color	Color	r: 0 g: 0 b: 0 a: 0	
Anti-aliasing / Antialiasing	Boolean	On	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerred Checkerred inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 416 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.231 G'MIC Polygonize energy node

This documentation is for version 1.0 of G'MIC Polygonize energy.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Integer	300	
Smoothness / Smoothness	Double	10	
Minimal area / Minimal_area	Double	10	
X-resolution / Xresolution	Double	10	
Y-resolution / Yresolution	Double	10	
Outline color / Outline_color	Color	r: 0 g: 0 b: 0 a: 0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 417 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.232 G'MIC Pop shadows node

This documentation is for version 1.0 of G'MIC Pop shadows.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Strength / Strength	Double	0.75	
Scale / Scale	Double	5	
Post-normalize / Postnormalize	Boolean	On	

Continued on next page

Table 418 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 418 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.233 G'MIC Poster edges node

This documentation is for version 1.0 of G'MIC Poster edges.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Image smoothness / Image_smoothness	Double	20	
Edge threshold / Edge_threshold	Double	60	
Edge shade / Edge_shade	Double	5	
Edge thickness / Edge_thickness	Double	0	
Edge antialiasing / Edge_antialiasing	Double	10	
Posterization level / Posterization_level	Integer	0	
Posterization antialiasing / Posterization_antialiasing	Double	0	

Continued on next page

Table 419 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 419 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.234 G'MIC Posterize node

This documentation is for version 1.0 of G'MIC Posterize.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	150	
Edges (%) / Edges_	Double	30	
Paint / Paint	Double	1	
Colors / Colors	Integer	12	
Minimal area / Minimal_area	Integer	0	
Outline (%) / Outline_	Double	0	
Normalize colors / Normalize_colors	Boolean	Off	

Continued on next page

Table 420 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 420 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.235 G'MIC Print films node

This documentation is for version 1.0 of G'MIC Print films.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Preset / Preset	Choice	None	None Fuji 3510 (Constlclip) Fuji 3510 (Constlmap) Fuji 3510 (Cuspclip) Fuji 3513 (Constlclip) Fuji 3513 (Constlmap) Fuji 3513 (Cuspclip) Kodak 2383 (Constlclip) Kodak 2383 (Constlmap) Kodak 2383 (Cuspclip) Kodak 2393 (Constlclip) Kodak 2393 (Constlmap) Kodak 2393 (Cuspclip)
Strength (%) / Strength_	Double	100	
Brightness (%) / Brightness_	Double	0	

Continued on next page

Table 421 – continued from previous page

Parameter / script name	Type	Default	Function
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Normalize colors / Normalize_colors	Choice	None	None Pre-process Post-process Both
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 421 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.236 G'MIC Print films collage node

This documentation is for version 1.0 of G'MIC Print films collage.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Image size / Image_size	Double	512	
Columns for collage / Columns_for_collage	Integer	4	
Label size / Label_size	Integer	16	

Continued on next page

Table 422 – continued from previous page

Parameter / script name	Type	Default	Function
Output as / Output_as	Choice	Table	Table Multiple layers
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.237 G'MIC Puzzle node

This documentation is for version 1.0 of G'MIC Puzzle.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-tiles / Xtiles	Integer	5	
Y-tiles / Ytiles	Integer	5	
Curvature / Curvature	Double	0.5	
Connectors centering / Connectors_centering	Double	0	
Connectors variability / Connectors_variability	Double	0	
Relief smoothness / Relief_smoothness	Double	0.3	
Relief contrast / Relief_contrast	Double	100	
Outline smoothness / Outline_smoothness	Double	0.2	
Outline contrast / Outline_contrast	Double	255	
Scale / Scale	Double	100	
Scale variations / Scale_variations	Double	0	
Angle / Angle	Double	0	
Angle variations / Angle_variations	Double	0	
Shuffle pieces / Shuffle_pieces	Boolean	Off	
Additional outline / Additional_outline	Boolean	Off	
Output each piece on a different layer / Output_each_piece_on_a_different_layer	Boolean	Off	

Continued on next page

Table 423 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.238 G'MIC Quadrangle node

This documentation is for version 1.0 of G'MIC Quadrangle.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Top-left vertex (%) / Topleft_vertex_	Double	x: 0.05 y: 0.05	
Top-right vertex (%) / Topright_vertex_	Double	x: 0.95 y: 0.25	
Bottom-right vertex (%) / Bottomright_vertex_	Double	x: 0.6 y: 0.95	
Bottom-left vertex (%) / Bottomleft_vertex_	Double	x: 0.4 y: 0.95	
Interpolation / Interpolation	Choice	Linear	Nearest neighbor Linear
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror
Preview type / Preview_type	Choice	Output	Input Output Both
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 424 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.239 G'MIC Quadtree variations node

This documentation is for version 1.0 of G'MIC Quadtree variations.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Mode / Mode	Choice	Squares	Squares Sierpinski design Ellipse painting
Precision / Precision	Integer	1024	

Continued on next page

Table 425 – continued from previous page

Parameter / script name	Type	Default	Function
Homogeneity / Homogeneity	Double	0.5	
Outline / Outline	Integer	0	
Primary radius / Primary_radius	Double	3	
Secondary radius / Secondary_radius	Double	1.5	
Anisotropy / Anisotropy	Double	1	
Only leafs / Only_leafs	Boolean	On	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 425 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.240 G'MIC Quick copyright node

This documentation is for version 1.0 of G'MIC Quick copyright.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Text / Text	String	\251 G'MIC	
Size / Size	Integer	27	
Color / Color	Color	r: 1 g: 1 b: 1 a: 1	
Outline / Outline	Integer	1	

Continued on next page

Table 426 – continued from previous page

Parameter / script name	Type	Default	Function
Position / Position	Choice	Bottom-right	Up-left Up-right Bottom-left Bottom-right
Offset / Offset	Integer	5	
Orientation / Orientation	Choice	0 deg.	-90 deg. 0 deg. +90 deg. +180 deg.
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.241 G'MIC Rain & snow node

This documentation is for version 1.0 of G'MIC Rain & snow.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Angle / Angle	Double	65	
Speed / Speed	Double	10	
Density (%) / Density_	Double	50	
Radius / Radius	Double	0.1	
Gamma / Gamma	Double	1	
Opacity / Opacity	Double	1	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 427 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.242 G'MIC Rainbow node

This documentation is for version 1.0 of G'MIC Rainbow.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Left position / Left_position	Double	80	
Right position / Right_position	Double	80	
Left slope / Left_slope	Double	175	
Right slope / Right_slope	Double	175	
Thinness / Thinness	Double	3	
Opacity / Opacity	Double	80	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.243 G'MIC Random color ellipses node

This documentation is for version 1.0 of G'MIC Random color ellipses.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Integer	400	
Radius / Radius	Double	8	
Opacity / Opacity	Double	0.1	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.244 G'MIC Random shade stripes node

This documentation is for version 1.0 of G'MIC Random shade stripes.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Frequency / Frequency	Double	30	
Orientation / Orientation	Choice	Vertical	Horizontal Vertical
Darkness / Darkness	Double	0.8	
Lightness / Lightness	Double	1.3	

Continued on next page

Table 430 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Value action / Value_action	Choice	None	<p> None Cut Normalize </p>

Continued on next page

Table 430 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 430 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.245 G'MIC Reflection node

This documentation is for version 1.0 of G'MIC Reflection.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	80	
Density / Density	Double	0.1	
Wavelength / Wavelength	Double	1	
Merging steps / Merging_steps	Integer	0	
Amplitude_2 / Amplitude_2	Double	10	
Amplitude_3 / Amplitude_3	Double	10	
Bandwidth / Bandwidth	Double	20	
Shape / Shape	Choice	Sine	Bloc Triangle Sine Sine+ Random
Angle / Angle	Double	0	

Continued on next page

Table 431 – continued from previous page

Parameter / script name	Type	Default	Function
Offset / Offset	Double	0	
Height / Height	Double	50	
Attenuation / Attenuation	Double	1	
Color / Color	Color	r: 0.431373 g: 0.627451 b: 0.745098 a: 0.745098	
Waves amplitude / Waves_amplitude	Double	0	
Waves smoothness / Waves_smoothness	Double	1.5	
X-angle / Xangle	Double	0	
Y-angle / Yangle	Double	-3.3	
Focale / Focale	Double	7	
Zoom / Zoom	Double	1.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 431 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.246 G'MIC Relief light node

This documentation is for version 1.0 of G'MIC Relief light.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Ambient lightness / Ambient_lightness	Double	0.3	
Specular lightness / Specular_lightness	Double	0.2	
Specular size / Specular_size	Double	0.2	
Darkness / Darkness	Double	0	
Light smoothness / Light_smoothness	Double	1	
XY-light / XYlight	Double	x: 0.5 y: 0.5	
Z-light / Zlight	Double	5	
Z-scale / Zscale	Double	0.5	
Opacity as heightmap / Opacity_as_heightmap	Boolean	Off	
Image smoothness / Image_smoothness	Double	0	

Continued on next page

Table 432 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.247 G'MIC Remove hot pixels node

This documentation is for version 1.0 of G'MIC Remove hot pixels.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Threshold / Threshold	Double	75	
Smoothness / Smoothness	Double	3.5	
Factor / Factor	Double	0.1	
Mask size / Mask_size	Integer	3	
Threshold_2 / Threshold_2	Double	10	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 433 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.248 G'MIC Resynthesize texture FFT node

This documentation is for version 1.0 of G'MIC Resynthesize texture FFT.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Width / Width	Integer	1024	
Height / Height	Integer	1024	
Equalize light / Equalize_light	Double	0	

Continued on next page

Table 434 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 434 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.249 G'MIC Resynthesize texture patch-based node

This documentation is for version 1.0 of G'MIC Resynthesize texture patch-based.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Width / Width	Integer	512	
Height / Height	Integer	512	
Number of scales / Number_of_scales	Integer	0	
Patch size / Patch_size	Integer	7	
Blending size / Blending_size	Integer	5	
Precision / Precision	Double	1	
Equalize light / Equalize_light	Double	0	

Continued on next page

Table 435 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 435 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.250 G'MIC Retinex node

This documentation is for version 1.0 of G'MIC Retinex.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Strength (%) / Strength_	Double	75	
Value offset / Value_offset	Double	16	
Colorspace / Colorspace	Choice	HSV	HSI HSV Lab Linear RGB RGB YCbCr
Min cut (%) / Min_cut_	Double	1	
Max cut (%) / Max_cut_	Double	1	
Regularization / Regularization	Double	5	
Low scale / Low_scale	Double	15	

Continued on next page

Table 436 – continued from previous page

Parameter / script name	Type	Default	Function
Middle scale / Middle_scale	Double	80	
High scale / High_scale	Double	250	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 436 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.251 G'MIC Retro fade node

This documentation is for version 1.0 of G'MIC Retro fade.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Iterations / Iterations	Integer	20	
Colors / Colors	Integer	6	
Grain / Grain	Double	40	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse

Continued on next page

Table 437 – continued from previous page

Parameter / script name	Type	Default	Function
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.252 G'MIC Rodilius node

This documentation is for version 1.0 of G'MIC Rodilius.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	10	
Thickness / Thickness	Double	10	
Sharpness / Sharpness	Double	300	
Orientations / Orientations	Integer	5	
Offset / Offset	Double	30	
Smoothness / Smoothness	Integer	0	
Color mode / Color_mode	Choice	Lighter	Darker Lighter

Continued on next page

Table 438 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Value action / Value_action	Choice	None	<p> None Cut Normalize </p>

Continued on next page

Table 438 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 438 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.253 G'MIC Rorschach node

This documentation is for version 1.0 of G'MIC Rorschach.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Scale / Scale	Double	3	
Mirror / Mirror	Choice	X-axis	None X-axis Y-axis XY-axes
Stencil type / Stencil_type	Choice	Color	Black & White RGB Color

Continued on next page

Table 439 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.254 G'MIC Sample image node

This documentation is for version 1.0 of G'MIC Sample image.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Input / Input	Choice	Random	<p> Random Apples Barbara Boats Bottles Butterfly Cameraman Car Cat Cliff David Dog Duck Eagle Elephant Earth Flower Fruits Greece Gummy House Inside Landscape Leaf Lena Leno Lion Mandrill Mona Lisa Monkey Parrots Pencils Peppers Rooster Rose Square Teddy Tiger Wall Waterfall Zelda </p>
Width / Width	Integer	0	
Height / Height	Integer	0	

Continued on next page

Table 440 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.255 G'MIC Satin node

This documentation is for version 1.0 of G'MIC Satin.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Iterations / Iterations	Integer	20	
Smoothness (%) / Smoothness_	Double	1	
Seed / Seed	Integer	0	
Dark color / Dark_color	Color	r: 0 g: 0 b: 0 a: 0	
Light color / Light_color	Color	r: 1 g: 1 b: 1 a: 1	
Stretch contrast / Stretch_contrast	Boolean	Off	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	-50	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 441 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.256 G'MIC Scanlines node

This documentation is for version 1.0 of G'MIC Scanlines.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	60	
Bandwidth / Bandwidth	Double	2	
Shape / Shape	Choice	Bloc	Bloc Triangle Sine Sine+ Random
Angle / Angle	Double	0	
Offset / Offset	Double	0	

Continued on next page

Table 442 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Value action / Value_action	Choice	None	<p> None Cut Normalize </p>

Continued on next page

Table 442 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 442 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.257 G'MIC Seamcarve node

This documentation is for version 1.0 of G'MIC Seamcarve.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Width (%) / Width_	Double	85	
Height (%) / Height_	Double	100	
Maximal seams per iteration (%) / Maximal_seams_per_iteration_	Double	15	
Use top layer as a priority mask / Use_top_layer_as_a_priority_mask	Boolean	Off	
Antialiasing / Antialiasing	Boolean	On	

Continued on next page

Table 443 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.258 G'MIC Seamless turbulence node

This documentation is for version 1.0 of G'MIC Seamless turbulence.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	15	
Smoothness / Smoothness	Double	20	
Orientation / Orientation	Double	0	
Deviation / Deviation	Double	1	
Contrast / Contrast	Double	3	
Color rendering / Color_rendering	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.259 G'MIC Segmentation node

This documentation is for version 1.0 of G'MIC Segmentation.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Edge threshold / Edge_threshold	Double	2	
Smoothness / Smoothness	Double	1	

Continued on next page

Table 445 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Value action / Value_action	Choice	None	<p> None Cut Normalize </p>

Continued on next page

Table 445 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 445 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.260 G'MIC Select-replace color node

This documentation is for version 1.0 of G'MIC Select-replace color.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Similarity space / Similarity_space	Choice	RGB[A]	RGB[A] RGB YCbCr Red Green Blue Opacity Luminance Blue & Red chrominances Hue Saturation
Tolerance / Tolerance	Double	20	
Smoothness / Smoothness	Double	0	
Fill holes / Fill_holes	Integer	0	

Continued on next page

Table 446 – continued from previous page

Parameter / script name	Type	Default	Function
Selected color / Selected_color	Color	r: 1 g: 1 b: 1 a: 1	
Output as / Output_as	Choice	Selected colors	Selected colors Selected mask Rejected colors Rejected mask Replaced color
Replacement color / Replacement_color	Color	r: 1 g: 0 b: 0 a: 0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerred Checkerred inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 446 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.261 G'MIC Selective desaturation node

This documentation is for version 1.0 of G'MIC Selective desaturation.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Reference color / Reference_color	Color	r: 1 g: 1 b: 1 a: 1	
Desaturate / Desaturate	Choice	Reference color	Reference color All but reference color
Strength / Strength	Double	3	

Continued on next page

Table 447 – continued from previous page

Parameter / script name	Type	Default	Function
Regularization / Regularization	Integer	0	
Maximum saturation / Maximum_saturation	Choice	From input	From input From reference color Maximum value
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 447 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.262 G'MIC Sepia node

This documentation is for version 1.0 of G'MIC Sepia.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	

Continued on next page

Table 448 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 448 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.263 G'MIC Shade bobs node

This documentation is for version 1.0 of G'MIC Shade bobs.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Integer	50	
Radius / Radius	Integer	5	
Duration / Duration	Integer	200	
Velocity / Velocity	Double	1	
Rx / Rx	Double	-1	
Ry / Ry	Double	2	
Rz / Rz	Double	1	
Rt / Rt	Double	0.8	
Rcx / Rcx	Double	0	
Colormap / Colormap	Choice	Cube	Grayscale Standard HSV Lines Hot Cool Jet Flag Cube

Continued on next page

Table 449 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.264 G'MIC Shadow patch node

This documentation is for version 1.0 of G'MIC Shadow patch.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Opacity / Opacity	Double	0.7	
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 450 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.265 G'MIC Shapeism node

This documentation is for version 1.0 of G'MIC Shapeism.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Shape / Shape	Choice	Circles	Squares Triangles Circles Diamond Hexagon Octagon Stars Custom
Branches / Branches	Integer	7	
Thickness / Thickness	Double	0.38	
Angle / Angle	Double	0	
Antialiasing / Antialiasing	Boolean	On	
Scales / Scales	Integer	5	
Maximal size / Maximal_size	Integer	32	
Minimal size / Minimal_size	Integer	8	
Allow angle / Allow_angle	Choice	Any	0 deg. 180 deg. 90 deg. Any
Spacing / Spacing	Integer	1	
Precision / Precision	Integer	5	
Edges / Edges	Double	0.5	
Smoothness / Smoothness	Double	1	
Background / Background	Color	r: 0 g: 0 b: 0 a: 0	

Continued on next page

Table 451 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.266 G'MIC Sharp abstract node

This documentation is for version 1.0 of G'MIC Sharp abstract.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Spatial scale / Spatial_scale	Double	4	
Value scale / Value_scale	Double	10	
Precision / Precision	Double	0.5	
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 452 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 452 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.267 G'MIC Sharpen Gold-Meinel node

This documentation is for version 1.0 of G'MIC Sharpen Gold-Meinel.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Sigma / Sigma	Double	1	
Iterations / Iterations	Integer	5	
Acceleration / Acceleration	Double	1	
Blur / Blur	Choice	Gaussian	Exponential Gaussian
Cut / Cut	Boolean	On	

Continued on next page

Table 453 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	YCbCr [luminance]	All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas]
Parallel processing / Parallel_processing	Choice	Auto	Auto One thread Two threads Four threads Eight threads Sixteen threads
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 453 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 453 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.268 G'MIC Sharpen Richardson-Lucy node

This documentation is for version 1.0 of G'MIC Sharpen Richardson-Lucy.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Sigma / Sigma	Double	1	
Iterations / Iterations	Integer	10	
Blur / Blur	Choice	Gaussian	Exponential Gaussian
Cut / Cut	Boolean	On	

Continued on next page

Table 454 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 454 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.269 G'MIC Sharpen deblur node

This documentation is for version 1.0 of G'MIC Sharpen deblur.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Radius / Radius	Double	2	
Iterations / Iterations	Integer	10	
Time step / Time_step	Double	20	
Smoothness / Smoothness	Double	0.1	
Regularization / Regularization	Choice	Mean curva- ture	Tikhonov Mean curvature Total variation

Continued on next page

Table 455 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	YCbCr [luminance]	All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas]
Parallel processing / Parallel_processing	Choice	Auto	Auto One thread Two threads Four threads Eight threads Sixteen threads
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 455 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 455 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.270 G'MIC Sharpen inv. diffusion node

This documentation is for version 1.0 of G'MIC Sharpen inv. diffusion.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	50	
Iterations / Iterations	Integer	2	

Continued on next page

Table 456 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	YCbCr [luminance]	All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas]
Parallel processing / Parallel_processing	Choice	Auto	Auto One thread Two threads Four threads Eight threads Sixteen threads
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 456 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 456 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.271 G'MIC Sharpen octave sharpening node

This documentation is for version 1.0 of G'MIC Sharpen octave sharpening.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Scales / Scales	Integer	4	
Maximal radius / Maximal_radius	Double	5	
Amount / Amount	Double	3	
Threshold / Threshold	Double	0	

Continued on next page

Table 457 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Parallel processing / Parallel_processing	Choice	Auto	<p> Auto One thread Two threads Four threads Eight threads Sixteen threads </p>
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 457 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 457 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.272 G'MIC Sharpen shock filters node

This documentation is for version 1.0 of G'MIC Sharpen shock filters.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	150	
Edge threshold / Edge_threshold	Double	0.1	
Gradient smoothness / Gradient_smoothness	Double	0.8	
Tensor smoothness / Tensor_smoothness	Double	1.1	
Iterations / Iterations	Integer	1	

Continued on next page

Table 458 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Parallel processing / Parallel_processing	Choice	Auto	<p> Auto One thread Two threads Four threads Eight threads Sixteen threads </p>
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 458 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 458 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.273 G'MIC Sharpen texture node

This documentation is for version 1.0 of G'MIC Sharpen texture.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Strength / Strength	Double	1	
Radius / Radius	Double	4	

Continued on next page

Table 459 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	Lab [lightness]	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 459 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 459 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.274 G'MIC Sharpen unsharp mask node

This documentation is for version 1.0 of G'MIC Sharpen unsharp mask.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Sharpening type / Sharpening_type	Choice	Bilateral	Gaussian Bilateral
Spatial radius / Spatial_radius	Double	1.25	
Bilateral radius / Bilateral_radius	Double	10	
Amount / Amount	Double	2	
Threshold / Threshold	Double	0	
Darkness level / Darkness_level	Double	1	
Lightness level / Lightness_level	Double	1	
Iterations / Iterations	Integer	1	
Negative effect / Negative_effect	Boolean	Off	

Continued on next page

Table 460 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 460 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 460 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.275 G'MIC Shock waves node

This documentation is for version 1.0 of G'MIC Shock waves.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	10	
Low frequency / Low_frequency	Double	10	
Frequency range / Frequency_range	Double	20	

Continued on next page

Table 461 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 461 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 461 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.276 G'MIC Sierpinski triangle node

This documentation is for version 1.0 of G'MIC Sierpinski triangle.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Recursions / Recursions	Integer	6	
1st X-coord / p1st_Xcoord	Double	50	
1st Y-coord / p1st_Ycoord	Double	0	
2nd X-coord / p2nd_Xcoord	Double	0	
2nd Y-coord / p2nd_Ycoord	Double	100	
3rd X-coord / p3rd_Xcoord	Double	100	
3rd Y-coord / p3rd_Ycoord	Double	100	
Color / Color	Color	r: 0.501961 g: 0.501961 b: 0.501961 a: 0.501961	
Opacity / Opacity	Double	1	

Continued on next page

Table 462 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.277 G'MIC Skeleton node

This documentation is for version 1.0 of G'MIC Skeleton.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Method / Method	Choice	Distance (fast)	Distance (fast) Thinning (slow)
Smoothness / Smoothness	Double	0	
Curviness / Curviness	Double	0	
Multiple channels / Multiple_channels	Boolean	On	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 463 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.278 G'MIC Sketch node

This documentation is for version 1.0 of G'MIC Sketch.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of orientations / Number_of_orientations	Integer	3	
Starting angle / Starting_angle	Double	45	
Angle range / Angle_range	Double	180	
Stroke length / Stroke_length	Double	30	

Continued on next page

Table 464 – continued from previous page

Parameter / script name	Type	Default	Function
Contour threshold / Contour_threshold	Double	1.75	
Opacity / Opacity	Double	0.02	
Background intensity / Background_intensity	Double	0.5	
Density / Density	Double	0.75	
Sharpness / Sharpness	Double	0.1	
Anisotropy / Anisotropy	Double	0.7	
Smoothness / Smoothness	Double	3	
Coherence / Coherence	Double	6	
Boost stroke / Boost_stroke	Boolean	Off	
Curved stroke / Curved_stroke	Boolean	On	
Color model / Color_model	Choice	Color on white	Black on white White on black Black on transparent white White on transparent black Color on white
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 464 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.279 G'MIC Slice luminosity node

This documentation is for version 1.0 of G'MIC Slice luminosity.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Luminosity type / Luminosity_type	Choice	Luminance	Average RGB Luminance Lightness Value
Output as / Output_as	Choice	Masked image	Mask Masked image
Preview type / Preview_type	Choice	Image	Mask Mask + background Image Image + background
Activate slice 1 / Activate_slice_1	Boolean	On	
Starting value / Starting_value	Integer	0	
Ending value / Ending_value	Integer	64	
Starting feathering / Starting_feathering	Integer	0	
Ending feathering / Ending_feathering	Integer	0	
Activate slice 2 / Activate_slice_2	Boolean	On	
Starting value_2 / Starting_value_2	Integer	64	
Ending value_2 / Ending_value_2	Integer	128	
Starting feathering_2 / Starting_feathering_2	Integer	0	
Ending feathering_2 / Ending_feathering_2	Integer	0	
Activate slice 3 / Activate_slice_3	Boolean	Off	
Starting value_3 / Starting_value_3	Integer	128	
Ending value_3 / Ending_value_3	Integer	192	
Starting feathering_3 / Starting_feathering_3	Integer	0	
Ending feathering_3 / Ending_feathering_3	Integer	0	

Continued on next page

Table 465 – continued from previous page

Parameter / script name	Type	Default	Function
Activate slice 4 / Activate_slice_4	Boolean	Off	
Starting value_4 / Starting_value_4	Integer	192	
Ending value_4 / Ending_value_4	Integer	255	
Starting feathering_4 / Starting_feathering_4	Double	0	
Ending feathering_4 / Ending_feathering_4	Double	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.280 G'MIC Slide collage color node

This documentation is for version 1.0 of G'MIC Slide collage color.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Image size / Image_size	Double	512	
Columns for collage / Columns_for_collage	Integer	4	
Label size / Label_size	Integer	16	
Output as / Output_as	Choice	Table	Table Multiple layers
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 466 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.281 G'MIC Slide color node

This documentation is for version 1.0 of G'MIC Slide color.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Preset / Preset	Choice	None	<p>None</p> <p>Agfa Precisa 100</p> <p>Fuji Astia 100F</p> <p>Fuji FP 100C</p> <p>Fuji Provia 100F</p> <p>Fuji Provia 400F</p> <p>Fuji Provia 400X</p> <p>Fuji Sensia 100</p> <p>Fuji Superia 200 XPRO</p> <p>Fuji Velvia 50</p> <p>Generic Fuji Astia 100</p> <p>Generic Fuji Provia 100</p> <p>Generic Fuji Velvia 100</p> <p>Generic Kodachrome 64</p> <p>Generic Kodak Ektachrome 100 VS</p> <p>Kodak E-100 GX Ektachrome 100</p> <p>Kodak Ektachrome 100 VS</p> <p>Kodak Elite Chrome 200</p> <p>Kodak Elite Chrome 400</p> <p>Kodak Elite ExtraColor 100</p> <p>Kodak Kodachrome 200</p> <p>Kodak Kodachrome 25</p> <p>Kodak Kodachrome 64</p> <p>Lomography X-Pro Slide 200</p> <p>Polaroid 669</p> <p>Polaroid 690</p> <p>Polaroid Polachrome</p>
Strength (%) / Strength_	Double	100	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Normalize colors / Normalize_colors	Choice	None	<p>None</p> <p>Pre-process</p> <p>Post-process</p> <p>Both</p>

Continued on next page

Table 467 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 467 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.282 G'MIC Smooth abstract node

This documentation is for version 1.0 of G'MIC Smooth abstract.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness (%) / Smoothness_	Double	75	
Regularization / Regularization	Choice	Isotropic	Isotropic Delaunay-oriented Edge-oriented
Regularization iterations / Regularization_iterations	Integer	20	
Geometry / Geometry	Double	1	
Details / Details	Double	30	

Continued on next page

Table 468 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 468 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.283 G'MIC Smooth anisotropic node

This documentation is for version 1.0 of G'MIC Smooth anisotropic.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	60	
Sharpness / Sharpness	Double	0.7	
Anisotropy / Anisotropy	Double	0.3	
Gradient smoothness / Gradient_smoothness	Double	0.6	
Tensor smoothness / Tensor_smoothness	Double	1.1	
Spatial precision / Spatial_precision	Double	0.8	
Angular precision / Angular_precision	Double	30	
Value precision / Value_precision	Double	2	
Interpolation / Interpolation	Choice	Nearest neighbor	Nearest neighbor Linear Runge-Kutta

Continued on next page

Table 469 – continued from previous page

Parameter / script name	Type	Default	Function
Fast approximation / Fast_approximation	Boolean	On	
Iterations / Iterations	Integer	1	
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 469 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 469 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.284 G'MIC Smooth antialias node

This documentation is for version 1.0 of G'MIC Smooth antialias.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	5	
Edge threshold (%) / Edge_threshold_	Double	10	
Smoothness / Smoothness	Double	0.8	

Continued on next page

Table 470 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 470 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.285 G'MIC Smooth bilateral node

This documentation is for version 1.0 of G'MIC Smooth bilateral.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Spatial variance / Spatial_variance	Double	10	
Value variance / Value_variance	Double	7	
Iterations / Iterations	Integer	2	

Continued on next page

Table 471 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 471 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 471 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.286 G'MIC Smooth diffusion node

This documentation is for version 1.0 of G'MIC Smooth diffusion.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Sharpness / Sharpness	Double	0.7	
Anisotropy / Anisotropy	Double	0.3	
Gradient smoothness / Gradient_smoothness	Double	0.6	
Tensor smoothness / Tensor_smoothness	Double	1.1	
Time step / Time_step	Double	15	
Iterations / Iterations	Integer	8	

Continued on next page

Table 472 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Parallel processing / Parallel_processing	Choice	Auto	<p> Auto One thread Two threads Four threads Eight threads Sixteen threads </p>
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 472 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 472 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.287 G'MIC Smooth guided node

This documentation is for version 1.0 of G'MIC Smooth guided.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Radius / Radius	Integer	5	
Smoothness / Smoothness	Double	30	
Iterations / Iterations	Integer	1	

Continued on next page

Table 473 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 473 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 473 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.288 G'MIC Smooth mean-curvature node

This documentation is for version 1.0 of G'MIC Smooth mean-curvature.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Time step / Time_step	Double	30	
Iterations / Iterations	Integer	4	
Keep iterations as different layers / Keep_iterations_as_different_layers	Boolean	Off	

Continued on next page

Table 474 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Parallel processing / Parallel_processing	Choice	Auto	<p> Auto One thread Two threads Four threads Eight threads Sixteen threads </p>
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 474 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 474 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.289 G'MIC Smooth median node

This documentation is for version 1.0 of G'MIC Smooth median.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Radius / Radius	Integer	3	
Threshold / Threshold	Double	255	

Continued on next page

Table 475 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 475 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 475 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.290 G'MIC Smooth nlmeans node

This documentation is for version 1.0 of G'MIC Smooth nlmeans.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Patch size / Patch_size	Double	4	
Spatial bandwidth / Spatial_bandwidth	Integer	4	
Tonal bandwidth / Tonal_bandwidth	Double	10	
Patch measure / Patch_measure	Choice	Luminance	Linf-norm L1-norm L2-norm Luminance Lightness RGB

Continued on next page

Table 476 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Parallel processing / Parallel_processing	Choice	Auto	<p> Auto One thread Two threads Four threads Eight threads Sixteen threads </p>
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 476 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 476 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.291 G'MIC Smooth patch-based node

This documentation is for version 1.0 of G'MIC Smooth patch-based.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Spatial variance / Spatial_variance	Double	10	
Patch variance / Patch_variance	Double	10	
Patch size / Patch_size	Integer	3	
Lookup size / Lookup_size	Integer	5	
Patch smoothness / Patch_smoothness	Double	0	
Fast approximation / Fast_approximation	Boolean	On	
Iterations / Iterations	Integer	1	

Continued on next page

Table 477 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Parallel processing / Parallel_processing	Choice	Auto	<p> Auto One thread Two threads Four threads Eight threads Sixteen threads </p>
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 477 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 477 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.292 G'MIC Smooth patch-pca node

This documentation is for version 1.0 of G'MIC Smooth patch-pca.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Strength / Strength	Double	4	
Patch size / Patch_size	Integer	7	
Lookup size / Lookup_size	Integer	11	
Spatial sampling / Spatial_sampling	Integer	7	

Continued on next page

Table 478 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 478 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 478 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.293 G'MIC Smooth perona-malik node

This documentation is for version 1.0 of G'MIC Smooth perona-malik.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
K-factor / Kfactor	Double	20	
Time step / Time_step	Double	5	
Iterations / Iterations	Integer	5	
Keep iterations as different layers / Keep_iterations_as_different_layers	Boolean	Off	

Continued on next page

Table 479 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Parallel processing / Parallel_processing	Choice	Auto	<p> Auto One thread Two threads Four threads Eight threads Sixteen threads </p>
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 479 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 479 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.294 G'MIC Smooth selective gaussian node

This documentation is for version 1.0 of G'MIC Smooth selective gaussian.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	5	
Edges / Edges	Double	0.5	
Scales / Scales	Integer	5	
Iterations / Iterations	Integer	1	

Continued on next page

Table 480 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Parallel processing / Parallel_processing	Choice	Auto	<p> Auto One thread Two threads Four threads Eight threads Sixteen threads </p>
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 480 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkered Checkered inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 480 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.295 G'MIC Smooth skin node

This documentation is for version 1.0 of G'MIC Smooth skin.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Skin estimation / Skin_estimation	Choice	Automatic	None Manual Automatic
Tolerance / Tolerance	Double	0.5	
Smoothness / Smoothness	Double	1	
Threshold / Threshold	Double	1	
Pre-normalize image / Prenormalize_image	Boolean	On	
X-coordinate [manual] / Xcoordinate_manual	Double	50	
Y-coordinate [manual] / Ycoordinate_manual	Double	50	
Radius [manual] / Radius_manual	Double	5	

Continued on next page

Table 481 – continued from previous page

Parameter / script name	Type	Default	Function
Base scale / Base_scale	Double	2	
Fine scale / Fine_scale	Double	0.2	
Smoothness_2 / Smoothness_2	Double	3	
Smoothness type / Smoothness_type	Choice	Bilateral	Gaussian Bilateral
Gain / Gain	Double	0.05	
Preview data / Preview_data	Choice	Result image	Skin mask Base scale Medium scale (original) Medium scale (smoothed) Fine scale Result image
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	

Continued on next page

Table 481 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.296 G'MIC Smooth total variation node

This documentation is for version 1.0 of G'MIC Smooth total variation.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	60	
Sharpness / Sharpness	Double	0.9	
Anisotropy / Anisotropy	Double	0.64	
Gradient smoothness / Gradient_smoothness	Double	3.1	
Tensor smoothness / Tensor_smoothness	Double	1.1	
Spatial precision / Spatial_precision	Double	0.8	
Angular precision / Angular_precision	Double	30	
Value precision / Value_precision	Double	2	
Interpolation / Interpolation	Choice	Nearest neighbor	Nearest neighbor Linear Runge-Kutta
Fast approximation / Fast_approximation	Boolean	On	
Iterations / Iterations	Integer	1	
Channel(s) / Channels	Choice	RGB	RGB Luminance Blue & Red chrominances Blue chrominance Red chrominance

Continued on next page

Table 482 – continued from previous page

Parameter / script name	Type	Default	Function
Parallel processing / Parallel_processing	Choice	Auto	Auto One thread Two threads Four threads Eight threads Sixteen threads
Spatial overlap / Spatial_overlap	Integer	24	
Time step / Time_step	Double	30	
Iterations_2 / Iterations_2	Integer	10	
Keep iterations as different layers / Keep_iterations_as_different_layers	Boolean	Off	

Continued on next page

Table 482 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s)_2 / Channels_2	Choice	All	All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas]
Parallel processing_2 / Parallel_processing_2	Choice	Auto	Auto One thread Two threads Four threads Eight threads Sixteen threads
Spatial overlap_2 / Spatial_overlap_2	Integer	24	

Continued on next page

Table 482 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 482 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.297 G'MIC Smooth wavelets node

This documentation is for version 1.0 of G'MIC Smooth wavelets.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Threshold / Threshold	Double	1	
Iterations / Iterations	Integer	10	
Scales / Scales	Integer	10	

Continued on next page

Table 483 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Parallel processing / Parallel_processing	Choice	Auto	<p> Auto One thread Two threads Four threads Eight threads Sixteen threads </p>
Spatial overlap / Spatial_overlap	Integer	24	

Continued on next page

Table 483 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 483 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.298 G'MIC Snowflake node

This documentation is for version 1.0 of G'MIC Snowflake.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Recursions / Recursions	Integer	5	
Opacity / Opacity	Double	1	
Color / Color	Color	r: 1 g: 1 b: 1 a: 1	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 484 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.299 G'MIC Solidify node

This documentation is for version 1.0 of G'MIC Solidify.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness (%) / Smoothness_	Double	75	
Regularization / Regularization	Choice	Delaunay-oriented	Isotropic Delaunay-oriented Edge-oriented
Regularization iterations / Regularization_iterations	Integer	20	

Continued on next page

Table 485 – continued from previous page

Parameter / script name	Type	Default	Function
Dilation / erosion / Dilation__erosion	Integer	0	
Colorspace / Colorspace	Choice	Linear RGB	sRGB Linear RGB
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 485 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.300 G'MIC Solve maze node

This documentation is for version 1.0 of G'MIC Solve maze.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Starting point (%) / Starting_point_	Double	x: 0.05 y: 0.05	
Ending point (%) / Ending_point_	Double	x: 0.95 y: 0.95	
Smoothness / Smoothness	Double	0.1	
Thickness / Thickness	Integer	3	
Color / Color	Color	r: 1 g: 0 b: 0 a: 0	
Maze type / Maze_type	Choice	Dark walls	Dark walls White walls

Continued on next page

Table 486 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.301 G'MIC Spatial transition node

This documentation is for version 1.0 of G'MIC Spatial transition.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of added frames / Number_of_added_frames	Integer	10	
Shading (%) / Shading_	Double	0	
Transition shape / Transition_shape	Choice	Plasma	Bottom layer Top layer Custom formula Horizontal Vertical Angular Radial Plasma
Custom formula / Custom_formula	String	$\cos(x*y/(16+32*A))$	
A-value / Avalue	Double	0	
Preview type / Preview_type	Choice	Timed image	Transition map Timed image Sequence x4 Sequence x6 Sequence x8
Preview time / Preview_time	Double	0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 487 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.302 G'MIC Sphere node

This documentation is for version 1.0 of G'MIC Sphere.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Width / Width	Integer	512	
Height / Height	Integer	512	
Radius / Radius	Double	90	
Dilation / Dilation	Double	0.5	
Angle / Angle	Double	0	
Border smoothness / Border_smoothness	Double	0	
Border width / Border_width	Double	20	

Continued on next page

Table 488 – continued from previous page

Parameter / script name	Type	Default	Function
Orientation / Orientation	Choice	0 deg.	0 deg. 90 deg. 180 deg. 270 deg.
Background / Background	Choice	Transparent	Transparent Mean color
Fading / Fading	Double	0	
Fading shape / Fading_shape	Double	0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.303 G'MIC Spherize node

This documentation is for version 1.0 of G'MIC Spherize.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Radius (%) / Radius_	Double	50	
Strength / Strength	Double	1	
Smoothness (%) / Smoothness_	Double	0	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Ratio / Ratio	Double	0	
Angle / Angle	Double	0	
Interpolation / Interpolation	Choice	Cubic	Nearest neighbor Linear Cubic
Preview grid / Preview_grid	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 489 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.304 G'MIC Split details alpha node

This documentation is for version 1.0 of G'MIC Split details alpha.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of levels / Number_of_levels	Integer	6	
Base scale / Base_scale	Double	10	
Details scale / Details_scale	Double	1	
Opacity gain / Opacity_gain	Double	5	

Continued on next page

Table 490 – continued from previous page

Parameter / script name	Type	Default	Function
Preview without alpha / Preview_without_alpha	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.305 G'MIC Split details gaussian node

This documentation is for version 1.0 of G'MIC Split details gaussian.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Number of scales / Number_of_scales	Integer	6	
Base scale / Base_scale	Double	10	
Details scale / Details_scale	Double	1	
Sharpen details in preview / Sharpen_details_in_preview	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 491 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.306 G'MIC Split details wavelets node

This documentation is for version 1.0 of G'MIC Split details wavelets.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Number of scales / Number_of_scales	Integer	6	
Add alpha channels to detail scale layers / Add_alpha_channels_to_detail_scale_layers	Boolean	Off	
Sharpen details in preview / Sharpen_details_in_preview	Boolean	Off	

Continued on next page

Table 492 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.307 G'MIC Sponge node

This documentation is for version 1.0 of G'MIC Sponge.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Size / Size	Integer	13	
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 493 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 493 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.308 G'MIC Square to circle node

This documentation is for version 1.0 of G'MIC Square to circle.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Mode / Mode	Choice	Square to circle	Square to circle Circle to square
Interpolation / Interpolation	Choice	Linear	Nearest neighbor Linear
Boundary / Boundary	Choice	Transparent	Transparent Nearest Periodic Mirror
X-factor (%) / Xfactor_	Double	0	
Y-factor (%) / Yfactor_	Double	0	

Continued on next page

Table 494 – continued from previous page

Parameter / script name	Type	Default	Function
X-offset (%) / Xoffset_	Double	0	
Y-offset (%) / Yoffset_	Double	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.309 G'MIC Stained glass node

This documentation is for version 1.0 of G'MIC Stained glass.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Edges / Edges	Double	20	
Shading / Shading	Double	0.1	
Thin separators / Thin_separators	Boolean	On	
Equalize / Equalize	Boolean	On	
Colors / Colors	Double	1	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 495 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.310 G'MIC Stamp node

This documentation is for version 1.0 of G'MIC Stamp.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Auto-threshold / Autothreshold	Boolean	On	
Threshold / Threshold	Integer	50	
Smoothness / Smoothness	Double	0	
Sharpening / Sharpening	Double	0	
Grain / Grain	Double	0	

Continued on next page

Table 496 – continued from previous page

Parameter / script name	Type	Default	Function
Negative / Negative	Boolean	Off	
Anti-aliasing / Antialiasing	Boolean	On	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 496 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.311 G'MIC Stars node

This documentation is for version 1.0 of G'MIC Stars.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Double	10	
Depth / Depth	Double	0	
Size / Size	Integer	32	
Branches / Branches	Integer	5	
Thickness / Thickness	Double	0.38	
Smoothness / Smoothness	Double	0	
Color / Color	Color	r: 1 g: 1 b: 0.392157 a: 0.392157	

Continued on next page

Table 497 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.312 G'MIC Stencil node

This documentation is for version 1.0 of G'MIC Stencil.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Radius / Radius	Double	3	
Smoothness / Smoothness	Double	0	
Iterations / Iterations	Integer	8	
Aliasing / Aliasing	Double	0	
Stencil type / Stencil_type	Choice	Color	Black & White RGB Color
Transparency / Transparency	Boolean	Off	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 498 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.313 G'MIC Stereographic projection node

This documentation is for version 1.0 of G'MIC Stereographic projection.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Transform / Transform	Choice	Direct	Direct Inverse
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Radius / Angle / Radius__Angle	Double	x: 0.5 y: 0.75	

Continued on next page

Table 499 – continued from previous page

Parameter / script name	Type	Default	Function
Horizon leveling (deg.) / Horizon_leveling_deg	Double	0	
Left/right blur (%) / Leftright_blur_	Double	0	
Dilation / Dilation	Double	0	
Mirror / Mirror	Choice	None	None X-axis Y-axis XY-axis
Boundary / Boundary	Choice	Periodic	Transparent Nearest Periodic Mirror
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 499 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.314 G'MIC Streak node

This documentation is for version 1.0 of G'MIC Streak.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Mask color / Mask_color	Color	r: 1 g: 0 b: 0 a: 0	
Step (%) / Step_	Double	0	
Angle / Angle	Double	0	
Propagation / Propagation	Choice	Bidirectional [smooth]	Backward Forward Bidirectional [sharp] Bidirectional [smooth]

Continued on next page

Table 500 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.315 G'MIC Stroke node

This documentation is for version 1.0 of G'MIC Stroke.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Thickness (px) / Thickness_px	Integer	3	
Threshold (%) / Threshold_	Double	50	
Smoothness (px) / Smoothness_px	Double	0	
Shape / Shape	Choice	Round	Square Diamond Round
Direction / Direction	Choice	Outward	Inward Outward
Zoom (%) / Zoom_	Double	100	
X-shift (px) / Xshift_px	Integer	0	
Y-shift (px) / Yshift_px	Integer	0	
Starting color / Starting_color	Color	r: 1 g: 1 b: 1 a: 1	
Ending color / Ending_color	Color	r: 1 g: 1 b: 1 a: 1	
Inside color / Inside_color	Color	r: 0 g: 0 b: 0 a: 0	
Outside color / Outside_color	Color	r: 0 g: 0 b: 0 a: 0	
Output stroke layer on / Output_stroke_layer_on	Choice	Top	Bottom Top
Keep original image size / Keep_original_image_size	Boolean	Off	

Continued on next page

Table 501 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.316 G'MIC Super-pixels node

This documentation is for version 1.0 of G'MIC Super-pixels.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Size / Size	Integer	16	
Regularity / Regularity	Double	10	
Iterations / Iterations	Integer	5	
Colors / Colors	Choice	Average	Random Average
Border opacity / Border_opacity	Double	1	
Border color / Border_color	Color	r: 0 g: 0 b: 0 a: 0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 502 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.317 G'MIC Superformula node

This documentation is for version 1.0 of G'MIC Superformula.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Resolution / Resolution	Integer	4096	
X-size / Xsize	Double	0.9	
Y-size / Ysize	Double	0.9	
M / M	Integer	8	
N1 / N1	Double	1	
N2 / N2	Double	5	
N3 / N3	Double	8	
X-angle / Xangle	Double	0	

Continued on next page

Table 503 – continued from previous page

Parameter / script name	Type	Default	Function
Y-angle / Yangle	Double	0	
Z-angle / Zangle	Double	0	
Thickness / Thickness	Double	3	
Color / Color	Color	r: 0.501961 g: 1 b: 0.501961 a: 0.501961	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.318 G'MIC Symmetrize node

This documentation is for version 1.0 of G'MIC Symmetrize.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Point 1 / Point_1	Double	x: 0.5 y: 0.5	
Point 2 / Point_2	Double	x: 0.5 y: 0.75	
Angle / Angle	Double	0	
Boundary / Boundary	Choice	Transparent	Transparent Nearest Periodic Mirror
Type / Type	Choice	Symmetry	Symmetry Antisymmetry
Swap sides / Swap_sides	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16

Continued on next page

Table 504 – continued from previous page

Parameter / script name	Type	Default	Function
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.319 G'MIC Taquin node

This documentation is for version 1.0 of G'MIC Taquin.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-tiles / Xtiles	Integer	7	
Y-tiles / Ytiles	Integer	7	
Remove tile / Remove_tile	Choice	None	None First Last Random
Relief / Relief	Double	50	
Border thickness (%) / Border_thickness_	Double	5	
Border outline / Border_outline	Integer	0	
Outline color / Outline_color	Color	r: 0 g: 0 b: 0 a: 0	

Continued on next page

Table 505 – continued from previous page

Parameter / script name	Type	Default	Function
Random seed / Random_seed	Integer	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.320 G'MIC Tetris node

This documentation is for version 1.0 of G'MIC Tetris.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Scale / Scale	Integer	10	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.321 G'MIC Textured glass node

This documentation is for version 1.0 of G'MIC Textured glass.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-amplitude / Xamplitude	Double	40	
Y-amplitude / Yamplitude	Double	40	
X-smoothness / Xsmoothness	Double	1	
Y-smoothness / Ysmoothness	Double	1	
Edge attenuation / Edge_attenuation	Double	0	
Edge influence / Edge_influence	Double	2	
Noise scale / Noise_scale	Integer	0	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 507 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.322 G'MIC Thin edges node

This documentation is for version 1.0 of G'MIC Thin edges.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	0	
Threshold / Threshold	Double	15	
Negative colors / Negative_colors	Boolean	Off	

Continued on next page

Table 508 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 508 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.323 G'MIC Tileable rotation node

This documentation is for version 1.0 of G'MIC Tileable rotation.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Angle / Angle	Double	45	
Maximum size factor / Maximum_size_factor	Integer	8	
Array mode / Array_mode	Choice	None	None x-axis y-axis xy-axes 2xy-axes

Continued on next page

Table 509 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.324 G'MIC Tiled isolation node

This documentation is for version 1.0 of G'MIC Tiled isolation.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-size / Xsize	Double	10	
Y-size / Ysize	Double	10	
X-border / Xborder	Double	5	
Y-border / Yborder	Double	5	
Keep tiles square / Keep_tiles_square	Boolean	On	
Keep borders square / Keep_borders_square	Boolean	On	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.325 G'MIC Tiled normalization node

This documentation is for version 1.0 of G'MIC Tiled normalization.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-tiles / Xtiles	Integer	25	
Y-tiles / Ytiles	Integer	25	
Minimal value / Minimal_value	Double	0	
Maximal value / Maximal_value	Double	255	

Continued on next page

Table 511 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	YCbCr [luminance]	<p>All</p> <p>RGBA [all]</p> <p>RGB [all]</p> <p>RGB [red]</p> <p>RGB [green]</p> <p>RGB [blue]</p> <p>RGBA [alpha]</p> <p>Linear RGB [all]</p> <p>Linear RGB [red]</p> <p>Linear RGB [green]</p> <p>Linear RGB [blue]</p> <p>YCbCr [luminance]</p> <p>YCbCr [blue-red chrominances]</p> <p>YCbCr [blue chrominance]</p> <p>YCbCr [red chrominance]</p> <p>YCbCr [green chrominance]</p> <p>Lab [lightness]</p> <p>Lab [ab-chrominances]</p> <p>Lab [a-chrominance]</p> <p>Lab [b-chrominance]</p> <p>Lch [ch-chrominances]</p> <p>Lch [c-chrominance]</p> <p>Lch [h-chrominance]</p> <p>HSV [hue]</p> <p>HSV [saturation]</p> <p>HSV [value]</p> <p>HSI [intensity]</p> <p>HSL [lightness]</p> <p>CMYK [cyan]</p> <p>CMYK [magenta]</p> <p>CMYK [yellow]</p> <p>CMYK [key]</p> <p>YIQ [luma]</p> <p>YIQ [chromas]</p>

Continued on next page

Table 511 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.326 G'MIC Tiled parameterization node

This documentation is for version 1.0 of G'MIC Tiled parameterization.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-tiles / Xtiles	Integer	10	
Y-tiles / Ytiles	Integer	10	
Fitting function / Fitting_function	Choice	Linear	Linear Quadratic
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.327 G'MIC Tiled random shifts node

This documentation is for version 1.0 of G'MIC Tiled random shifts.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-tiles / Xtiles	Integer	10	
Y-tiles / Ytiles	Integer	10	
Amplitude / Amplitude	Double	10	
Opacity / Opacity	Double	1	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.328 G'MIC Tiled rotation node

This documentation is for version 1.0 of G'MIC Tiled rotation.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-tiles / Xtiles	Integer	5	
Y-tiles / Ytiles	Integer	5	
Angle / Angle	Double	15	
X-shadow / Xshadow	Double	3	
Y-shadow / Yshadow	Double	3	
Smoothness / Smoothness	Double	1.8	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 514 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.329 G'MIC Tiles to layers node

This documentation is for version 1.0 of G'MIC Tiles to layers.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
X-tiles / Xtiles	Integer	3	
Y-tiles / Ytiles	Integer	3	
Force tiles to have same size / Force_tiles_to_have_same_size	Boolean	Off	

Continued on next page

Table 515 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.330 G'MIC Tone mapping node

This documentation is for version 1.0 of G'MIC Tone mapping.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Threshold / Threshold	Double	0.5	
Gamma / Gamma	Double	0.7	
Smoothness / Smoothness	Double	0.1	
Iterations / Iterations	Integer	30	
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 516 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 516 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.331 G'MIC Tone mapping fast node

This documentation is for version 1.0 of G'MIC Tone mapping fast.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Radius / Radius	Double	3	
Power / Power	Double	0.5	

Continued on next page

Table 517 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	YCbCr [luminance]	<p>All</p> <p>RGBA [all]</p> <p>RGB [all]</p> <p>RGB [red]</p> <p>RGB [green]</p> <p>RGB [blue]</p> <p>RGBA [alpha]</p> <p>Linear RGB [all]</p> <p>Linear RGB [red]</p> <p>Linear RGB [green]</p> <p>Linear RGB [blue]</p> <p>YCbCr [luminance]</p> <p>YCbCr [blue-red chrominances]</p> <p>YCbCr [blue chrominance]</p> <p>YCbCr [red chrominance]</p> <p>YCbCr [green chrominance]</p> <p>Lab [lightness]</p> <p>Lab [ab-chrominances]</p> <p>Lab [a-chrominance]</p> <p>Lab [b-chrominance]</p> <p>Lch [ch-chrominances]</p> <p>Lch [c-chrominance]</p> <p>Lch [h-chrominance]</p> <p>HSV [hue]</p> <p>HSV [saturation]</p> <p>HSV [value]</p> <p>HSI [intensity]</p> <p>HSL [lightness]</p> <p>CMYK [cyan]</p> <p>CMYK [magenta]</p> <p>CMYK [yellow]</p> <p>CMYK [key]</p> <p>YIQ [luma]</p> <p>YIQ [chromas]</p>

Continued on next page

Table 517 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 517 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.332 G'MIC Tones to layers node

This documentation is for version 1.0 of G'MIC Tones to layers.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Number of tones / Number_of_tones	Integer	3	
Start of mid-tones / Start_of_midtones	Integer	85	
End of mid-tones / End_of_midtones	Integer	170	
Smoothness / Smoothness	Double	0.5	
Alpha / Alpha	Choice	Binary	Binary Scalar

Continued on next page

Table 518 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.333 G'MIC Transfer colors advanced node

This documentation is for version 1.0 of G'MIC Transfer colors advanced.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Regularization / Regularization	Integer	8	
Preserve luminance / Preserve_luminance	Double	0.2	
Precision / Precision	Choice	Normal	Low Normal High Very high
Reference colors / Reference_colors	Choice	Bottom layer	Bottom layer Top layer
Add user-defined constraints (interactive) / Add_userdefined_constraints_interactive	Boolean	Off	
Preview reference / Preview_reference	Choice	Up-left	None Up-left Up-right Bottom-left Bottom-right

Continued on next page

Table 519 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 519 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.334 G'MIC Transfer colors basic node

This documentation is for version 1.0 of G'MIC Transfer colors basic.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Transfer brightness / Transfer_brightness	Boolean	Off	
Revert layers / Revert_layers	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 520 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.335 G'MIC Truchet node

This documentation is for version 1.0 of G'MIC Truchet.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Scale / Scale	Integer	32	
Radius / Radius	Integer	5	
Smoothness / Smoothness	Double	1	
Type / Type	Choice	Curved	Straight Curved

Continued on next page

Table 521 – continued from previous page

Parameter / script name	Type	Default	Function
Color / Color	Choice	White on black	White on black Black on white White on transparent Black on transparent Transparent on white Transparent on black Random
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.336 G'MIC Tunnel node

This documentation is for version 1.0 of G'MIC Tunnel.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Depth / Depth	Integer	4	
Factor / Factor	Double	80	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Opacity / Opacity	Double	0.2	
Angle / Angle	Double	0	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	

Continued on next page

Table 522 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.337 G'MIC Turbulence node

This documentation is for version 1.0 of G'MIC Turbulence.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Radius / Radius	Double	128	
Octaves / Octaves	Integer	6	
Damping per octave / Damping_per_octave	Double	4	
Difference mixing / Difference_mixing	Double	0	
Mode / Mode	Choice	Turbulence	Turbulence Turbulence 2 Fractal Noise Fractured Clouds Stardust Pea Soup

Continued on next page

Table 523 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.338 G'MIC Twirl node

This documentation is for version 1.0 of G'MIC Twirl.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	1	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.339 G'MIC Upscale dccci2x node

This documentation is for version 1.0 of G'MIC Upscale dccci2x.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Threshold / Threshold	Double	1.15	
Exponent / Exponent	Integer	5	
Extend 1px / Extend_1px	Boolean	Off	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 525 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.340 G'MIC Upscale diffusion node

This documentation is for version 1.0 of G'MIC Upscale diffusion.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.redufx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Width / Width	String	200%	
Height / Height	String	200%	
Smoothness / Smoothness	Double	2	
Anisotropy / Anisotropy	Double	0.4	
Sharpness / Sharpness	Double	50	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 526 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.341 G'MIC Upscale scale2x node

This documentation is for version 1.0 of G'MIC Upscale scale2x.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernavy.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Scaling factor / Scaling_factor	Choice	x 2	x 2 x 3 x 4 x 6 x 8 x 9 x 12 x 16 x 18 x 27
Colorbase / Colorbase	Choice	RGB	RGB YCbCr Lab
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 527 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.342 G'MIC User-defined node

This documentation is for version 1.0 of G'MIC User-defined.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduffx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Red - green - blue - alpha / Red__green__blue__alpha	String	i	
Red - green - blue / Red__green__blue	String	$i + 90 \cdot (x/w) \cdot \cos(i/10)$	
Red / Red	String	i	
Green / Green	String	i	
Blue / Blue	String	i	
Alpha / Alpha	String	i	
Value normalization / Value_normalization	Choice	None	None RGB RGBA

Continued on next page

Table 528 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.343 G'MIC Various node

This documentation is for version 1.0 of G'MIC Various.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Continued on next page

Table 529 – continued from previous page

Parameter / script name	Type	Default	Function
Parameter / script name	Type	Default	Function
Preset / Preset	Choice	None	<p> None 60's 60's (faded) 60's (faded alt) Alien green Black & White Bleach bypass Blue mono Color (rich) Faded Faded (alt) Faded (analog) Faded (extreme) Faded (vivid) Expired (fade) Expired (polaroid) Extreme Fade Faux infrared Golden Golden (bright) Golden (fade) Golden (mono) Golden (vibrant) Green mono Hong Kong Light (blown) Lomo Mono tinted Muted fade Mute shift Natural (vivid) Nostalgic Orange tone Pink fade Purple Retro Rotate (muted) Rotate (vibrant) Smooth crome-ish Smooth fade Soft fade Solarize color Solarized color2 Summer Summer (alt) </p>
2.13. GMIC nodes			<p> Sunny Sunny (alt) Sunny (warm) Sunny (rich) </p>

Table 529 – continued from previous page

Parameter / script name	Type	Default	Function
Strength (%) / Strength_	Double	100	
Brightness (%) / Brightness_	Double	0	
Contrast (%) / Contrast_	Double	0	
Gamma (%) / Gamma_	Double	0	
Hue (%) / Hue_	Double	0	
Saturation (%) / Saturation_	Double	0	
Normalize colors / Normalize_colors	Choice	None	<p>None</p> <p>Pre-process</p> <p>Post-process</p> <p>Both</p>
Preview type / Preview_type	Choice	Full	<p>Full</p> <p>Forward horizontal</p> <p>Forward vertical</p> <p>Backward horizontal</p> <p>Backward vertical</p> <p>Duplicate top</p> <p>Duplicate left</p> <p>Duplicate bottom</p> <p>Duplicate right</p> <p>Duplicate horizontal</p> <p>Duplicate vertical</p> <p>Checkerboard</p> <p>Checkerboard inverse</p>
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	<p>Merged</p> <p>Layer 0</p> <p>Layer -1</p> <p>Layer -2</p> <p>Layer -3</p> <p>Layer -4</p> <p>Layer -5</p> <p>Layer -6</p> <p>Layer -7</p> <p>Layer -8</p> <p>Layer -9</p>

Continued on next page

Table 529 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.344 G'MIC Various collage node

This documentation is for version 1.0 of G'MIC Various collage.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No
Layer -1		Yes
Layer -2		Yes
Layer -3		Yes

Controls

Parameter / script name	Type	Default	Function
Image size / Image_size	Double	512	
Columns for collage / Columns_for_collage	Integer	4	
Label size / Label_size	Integer	16	

Continued on next page

Table 530 – continued from previous page

Parameter / script name	Type	Default	Function
Output as / Output_as	Choice	Table	Table Multiple layers
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.345 G'MIC Vector painting node

This documentation is for version 1.0 of G'MIC Vector painting.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Details / Details	Double	9	
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9

Continued on next page

Table 531 – continued from previous page

Parameter / script name	Type	Default	Function
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.346 G'MIC Vignette node

This documentation is for version 1.0 of G'MIC Vignette.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Strength / Strength	Double	70	
Min radius / Min_radius	Double	70	
Max radius / Max_radius	Double	95	
Color / Color	Color	r: 0 g: 0 b: 0 a: 0	

Continued on next page

Table 532 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.347 G'MIC Visible watermark node

This documentation is for version 1.0 of G'MIC Visible watermark.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Text / Text	String	\251 G'MIC	
Opacity / Opacity	Double	0.4	
Size / Size	Integer	50	
Angle / Angle	Double	25	
Smoothness / Smoothness	Double	0.5	
Lightness / Lightness	Choice	Brighter	Darker Brighter
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.348 G'MIC Warp by intensity node

This documentation is for version 1.0 of G'MIC Warp by intensity.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-factor / Xfactor	Double	0.04	
Y-factor / Yfactor	Double	0.04	
X-offset / Xoffset	Double	128	
Y-offset / Yoffset	Double	128	
Correlated channels / Correlated_channels	Boolean	Off	
Interpolation / Interpolation	Choice	Linear	Nearest neighbor Linear
Boundary / Boundary	Choice	Mirror	Transparent Nearest Periodic Mirror

Continued on next page

Table 534 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>

Continued on next page

Table 534 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 534 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.349 G'MIC Whirl drawing node

This documentation is for version 1.0 of G'MIC Whirl drawing.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
X-tiles / Xtiles	Integer	3	
Y-tiles / Ytiles	Integer	3	
Smoothness / Smoothness	Double	2	
Color / Color	Double	40	
Amplitude / Amplitude	Double	20	

Continued on next page

Table 535 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 535 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.350 G'MIC Whirls node

This documentation is for version 1.0 of G'MIC Whirls.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Density / Density	Integer	6	
Thickness / Thickness	Double	65	
Shadow / Shadow	Double	0	
Shading / Shading	Double	0.5	
Fibers amplitude / Fibers_amplitude	Double	0	
Fibers smoothness / Fibers_smoothness	Double	0	
Angle / Angle	Choice	0 deg.	0 deg. 22.5 deg. 45 deg. 67.5 deg.
X-curvature / Xcurvature	Double	0	
Y-curvature / Ycurvature	Double	0	
Density_2 / Density_2	Integer	7	

Continued on next page

Table 536 – continued from previous page

Parameter / script name	Type	Default	Function
Smoothness / Smoothness	Double	2	
Darkness / Darkness	Double	0.2	
Lightness / Lightness	Double	1.8	
Channel(s) / Channels	Choice	YCbCr [luminance]	<p>All</p> <p>RGBA [all]</p> <p>RGB [all]</p> <p>RGB [red]</p> <p>RGB [green]</p> <p>RGB [blue]</p> <p>RGBA [alpha]</p> <p>Linear RGB [all]</p> <p>Linear RGB [red]</p> <p>Linear RGB [green]</p> <p>Linear RGB [blue]</p> <p>YCbCr [luminance]</p> <p>YCbCr [blue-red chrominances]</p> <p>YCbCr [blue chrominance]</p> <p>YCbCr [red chrominance]</p> <p>YCbCr [green chrominance]</p> <p>Lab [lightness]</p> <p>Lab [ab-chrominances]</p> <p>Lab [a-chrominance]</p> <p>Lab [b-chrominance]</p> <p>Lch [ch-chrominances]</p> <p>Lch [c-chrominance]</p> <p>Lch [h-chrominance]</p> <p>HSV [hue]</p> <p>HSV [saturation]</p> <p>HSV [value]</p> <p>HSI [intensity]</p> <p>HSL [lightness]</p> <p>CMYK [cyan]</p> <p>CMYK [magenta]</p> <p>CMYK [yellow]</p> <p>CMYK [key]</p> <p>YIQ [luma]</p> <p>YIQ [chromas]</p>

Continued on next page

Table 536 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 536 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.351 G'MIC Wind node

This documentation is for version 1.0 of G'MIC Wind.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Amplitude / Amplitude	Double	30	
Smoothness / Smoothness	Double	1.5	
Angle / Angle	Double	45	
Amplitude_2 / Amplitude_2	Double	10	
Frequency / Frequency	Double	0.4	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Amplitude_3 / Amplitude_3	Integer	20	
Angle_2 / Angle_2	Double	0	
Attenuation / Attenuation	Double	0.7	
Threshold / Threshold	Double	20	
Mode / Mode	Choice	Brighter	Darker Brighter

Continued on next page

Table 537 – continued from previous page

Parameter / script name	Type	Default	Function
Channel(s) / Channels	Choice	All	<p> All RGBA [all] RGB [all] RGB [red] RGB [green] RGB [blue] RGBA [alpha] Linear RGB [all] Linear RGB [red] Linear RGB [green] Linear RGB [blue] YCbCr [luminance] YCbCr [blue-red chrominances] YCbCr [blue chrominance] YCbCr [red chrominance] YCbCr [green chrominance] Lab [lightness] Lab [ab-chrominances] Lab [a-chrominance] Lab [b-chrominance] Lch [ch-chrominances] Lch [c-chrominance] Lch [h-chrominance] HSV [hue] HSV [saturation] HSV [value] HSI [intensity] HSL [lightness] CMYK [cyan] CMYK [magenta] CMYK [yellow] CMYK [key] YIQ [luma] YIQ [chromas] </p>
Value action / Value_action	Choice	None	<p> None Cut Normalize </p>

Continued on next page

Table 537 – continued from previous page

Parameter / script name	Type	Default	Function
Preview type / Preview_type	Choice	Full	Full Forward horizontal Forward vertical Backward horizontal Backward vertical Duplicate top Duplicate left Duplicate bottom Duplicate right Duplicate horizontal Duplicate vertical Checkerboard Checkerboard inverse
Preview split / Preview_split	Double	x: 0.5 y: 0.5	
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Preview/Draft Mode / PreviewDraft_Mode	Boolean	Off	

Continued on next page

Table 537 – continued from previous page

Parameter / script name	Type	Default	Function
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.13.352 G'MIC Zoom node

This documentation is for version 1.0 of G'MIC Zoom.

Description

Wrapper for the G'MIC framework (<http://gmic.eu>) written by Tobias Fleischer (<http://www.reduxfx.com>) and Frederic Devernay.

Inputs

Input	Description	Optional
Source		No

Controls

Parameter / script name	Type	Default	Function
Factor / Factor	Double	2	
Center (%) / Center_	Double	x: 0.5 y: 0.5	
Boundary / Boundary	Choice	Transparent	Transparent Nearest Periodic Mirror

Continued on next page

Table 538 – continued from previous page

Parameter / script name	Type	Default	Function
Output Layer / Output_Layer	Choice	Layer 0	Merged Layer 0 Layer -1 Layer -2 Layer -3 Layer -4 Layer -5 Layer -6 Layer -7 Layer -8 Layer -9
Resize Mode / Resize_Mode	Choice	Dynamic	Fixed (Inplace) Dynamic Downsample 1/2 Downsample 1/4 Downsample 1/8 Downsample 1/16
Ignore Alpha / Ignore_Alpha	Boolean	Off	
Log Verbosity / Log_Verbosity	Choice	Off	Off Level 1 Level 2 Level 3

2.14 Extra nodes

The following sections contain documentation about every node in the Extra group. Node groups are available by clicking on buttons in the left toolbar, or by right-clicking the mouse in the Node Graph area.

2.14.1 Arc node



This documentation is for version 4.2 of Arc.

Description

Arc Distort transform node.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Angle / <i>angle</i>	Double	60	Arc angle
Rotate / <i>rotate</i>	Double	0	Arc rotate
Top radius / <i>top</i>	Double	0	Arc top radius
Bottom radius / <i>bottom</i>	Double	0	Arc bottom radius
Flip / <i>flip</i>	Boolean	Off	Flip image
Matte / <i>matte</i>	Boolean	Off	Merge Alpha before applying effect
Virtual Pixel / <i>pixel</i>	Choice	Transparent	Virtual Pixel Method Undefined Background Black CheckerTile Dither Edge Gray HorizontalTile HorizontalTileEdge Mirror Random Tile Transparent VerticalTile VerticalTileEdge White
OpenMP / <i>openmp</i>	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.

2.14.2 Charcoal node



This documentation is for version 2.2 of Charcoal.

Description

Charcoal effect node.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Radius / radius	Double	1	Adjust radius
Sigma / sigma	Double	0	Adjust sigma
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.

2.14.3 Edges node



This documentation is for version 2.0 of Edges.

Description

Edge extraction node.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Width / width	Double	2	Width of edges
Brightness / brightness	Double	5	Adjust edge brightness
Smoothing / smoothing	Double	1	Adjust edge smoothing
Grayscale / gray	Boolean	Off	Convert to grayscale before effect

Continued on next page

Table 541 – continued from previous page

Parameter / script name	Type	Default	Function
Kernel / kernel	Choice	Diamond	<p>Kernel</p> <p>Convolution Kernel</p> <p>BinomialKernel</p> <p>LaplacianKernel</p> <p>SobelKernel</p> <p>FreiChenKernel</p> <p>RobertsKernel</p> <p>PrewittKernel</p> <p>CompassKernel</p> <p>KirschKernel</p> <p>DiamondKernel</p> <p>SquareKernel</p> <p>RectangleKernel</p> <p>OctagonKernel</p> <p>DiskKernel</p> <p>PlusKernel</p> <p>CrossKernel</p> <p>RingKernel</p> <p>EdgesKernel</p> <p>CornersKernel</p> <p>DiagonalsKernel</p> <p>LineEndsKernel</p> <p>LineJunctionsKernel</p> <p>RidgesKernel</p> <p>ConvexHullKernel</p> <p>ThinSEKernel</p> <p>SkeletonKernel</p> <p>ChebyshevKernel</p> <p>ManhattanKernel</p> <p>OctagonalKernel</p> <p>EuclideanKernel</p>
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.

2.14.4 Implode node



This documentation is for version 2.3 of Implode.

Description

Implode transform node.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Factor / factor	Double	0.5	Implode image by factor
Swirl / swirl	Double	0	Swirl image by degree
Matte / matte	Boolean	Off	Merge Alpha before applying effect
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.

2.14.5 Modulate node



This documentation is for version 1.2 of Modulate.

Description

Modulate color node.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Brightness / brightness	Double	100	Adjust brightness (%)
Saturation / saturation	Double	100	Adjust saturation (%)
Hue / hue	Double	100	Adjust hue (%)
OpenMP / openmp	Boolean	On	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.
OpenCL / opencl	Boolean	Off	Enable/Disable OpenCL. This will enable the plugin to use supported GPU(s) for better performance.

2.14.6 Oilpaint node



This documentation is for version 2.1 of Oilpaint.

Description

Oilpaint filter node.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Radius / radius	Double	1	Adjust radius
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.

2.14.7 Polar node



This documentation is for version 4.3 of Polar.

Description

Polar Distort transform node.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Rotate / rotate	Double	0	Polar rotate
DePolar / dePolar	Boolean	Off	DePolar
Flip / flip	Boolean	Off	Polar Flip
Matte / matte	Boolean	Off	Merge Alpha before applying effect
Virtual Pixel / pixel	Choice	Transparent	Virtual Pixel Method Undefined Background Black CheckerTile Dither Edge Gray HorizontalTile HorizontalTileEdge Mirror Random Tile Transparent VerticalTile VerticalTileEdge White
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host. Note that this plugin is known to be unstable with this settings enabled, use at own risk.

2.14.8 Polaroid node



This documentation is for version 1.4 of Polaroid.

Description

Polaroid image effect node.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Angle / angle	Double	5	Adjust polaroid angle
Caption / caption	String	Enter text	Add caption to polaroid
Font family / font	Choice		The name of the font to be used
Font size / size	Integer	64	The height of the characters to render in pixels
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.

2.14.9 Reflection node



This documentation is for version 3.2 of Reflection.

Description

Mirror/Reflection transform node.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

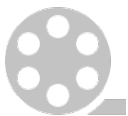
Parameter / script name	Type	Default	Function
Reflection offset / offset	Integer	0	Reflection offset
Reflection spacing / spacing	Integer	0	Space between image and reflection
Reflection / reflection	Boolean	On	Apply reflection
Matte / matte	Boolean	Off	Merge Alpha before applying effect

Continued on next page

Table 547 – continued from previous page

Parameter / script name	Type	Default	Function
Mirror / mirror	Choice	Undefined	Select mirror type Undefined North South East West NorthWest NorthEast SouthWest SouthEast Flip Flop Flip+Flop
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.

2.14.10 Roll node



This documentation is for version 2.9 of Roll.

Description

Roll effect using ImageMagick.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.
Matte / matte	Boolean	Off	Merge Alpha before applying effect.

Continued on next page

Table 548 – continued from previous page

Parameter / script name	Type	Default	Function
Virtual Pixel / vpixel	Choice	Transparent	Virtual Pixel Method. Undefined Background Black CheckerTile Dither Edge Gray HorizontalTile HorizontalTileEdge Mirror Random Tile Transparent VerticalTile VerticalTileEdge White
X / x	Double	0	Adjust roll X
Y / y	Double	0	Adjust roll Y

2.14.11 Sketch node



This documentation is for version 2.2 of Sketch.

Description

Sketch effect node.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Radius / radius	Double	1	Adjust radius
Sigma / sigma	Double	0	Adjust sigma

Continued on next page

Table 549 – continued from previous page

Parameter / script name	Type	Default	Function
Angle / angle	Double	0	Adjust angle
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.

2.14.12 Swirl node



This documentation is for version 2.9 of Swirl.

Description

Swirl effect using ImageMagick.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.
Matte / matte	Boolean	Off	Merge Alpha before applying effect.

Continued on next page

Table 550 – continued from previous page

Parameter / script name	Type	Default	Function
Virtual Pixel / vpixel	Choice	Transparent	Virtual Pixel Method. Undefined Background Black CheckerTile Dither Edge Gray HorizontalTile HorizontalTileEdge Mirror Random Tile Transparent VerticalTile VerticalTileEdge White
Amount / amount	Double	60	Swirl amount.

2.14.13 Texture node



This documentation is for version 3.7 of Texture.

Description

Texture/Background generator node.

Inputs

Input	Description	Optional
Source		Yes

Controls

Parameter / script name	Type	Default	Function
Background / background	Choice	Misc/Stripes	Background type Plasma/Regular Plasma/Fractal Noise/Gaussian Noise/Impulse Noise/Laplacian Misc/Checkerboard Misc/Stripes Gradient/Regular Gradient/Radial Misc/Loops 1 Misc/Loops 2 Misc/Loops 3
Seed / seed	Integer	0	Seed the random generator
Width / width	Integer	0	Set canvas width, default (0) is project format
Height / height	Integer	0	Set canvas height, default (0) is project format
Color from / fromColor	String		Set start color, you must set a end color for this to work. Valid values are: none (transparent), color name (red, blue etc) or hex colors
Color to / toColor	String		Set end color, you must set a start color for this to work. Valid values are : none (transparent), color name (red, blue etc) or hex colors
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.

2.14.14 Tile node



This documentation is for version 3.2 of Tile.

Description

Tile transform node.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
Rows / rows	Integer	2	Rows in grid

Continued on next page

Table 552 – continued from previous page

Parameter / script name	Type	Default	Function
Columns / cols	Integer	2	Columns in grid
Time Offset / offset	Integer	0	Set a time offset
Keep first frame / keepFirst	Boolean	On	Stay on first frame if offset
Matte / matte	Boolean	Off	Merge Alpha before applying effect
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.

2.14.15 Wave node



This documentation is for version 2.9 of Wave.

Description

Wave effect using ImageMagick.

Inputs

Input	Description	Optional
Source		No
Mask		Yes

Controls

Parameter / script name	Type	Default	Function
OpenMP / openmp	Boolean	Off	Enable/Disable OpenMP support. This will enable the plugin to use as many threads as allowed by host.
Matte / matte	Boolean	Off	Merge Alpha before applying effect.

Continued on next page

Table 553 – continued from previous page

Parameter / script name	Type	Default	Function
Virtual Pixel / vpixel	Choice	Transparent	Virtual Pixel Method. Undefined Background Black CheckerTile Dither Edge Gray HorizontalTile HorizontalTileEdge Mirror Random Tile Transparent VerticalTile VerticalTileEdge White
Amplitude / amp	Double	25	Adjust wave amplitude
Length / length	Double	150	Adjust wave length

3.1 Python API

All Python modules of the Natron API are referenced here.

3.1.1 NatronEngine

Detailed Description

Here are listed all classes being part of NatronEngine module. This module is always loaded by Natron natively, meaning access is granted to these classes in your scripts without importing anything.

AnimatedParam

Inherits *Param*

Inherited by: *StringParamBase*, *PathParam*, *OutputFileParam*, *FileParam*, *StringParam*, *BooleanParam*, *ChoiceParam*, *ColorParam*, *DoubleParam*, *Double2DParam*, *Double3DParam*, *IntParam*, *Int2DParam*, *Int3DParam*

Synopsis

This is the base class for all parameters which have the property *canAnimate* set to True. See the *detailed description* below

Functions

- `def deleteValueAtTime (time[, dimension=0])`
- `def getCurrentTime ()`
- `def getDerivativeAtTime (time[, dimension=0])`
- `def getExpression (dimension)`
- `def getIntegrateFromTimeToTime (time1, time2[, dimension=0])`

- `def getIsAnimated ([dimension=0])`
- `def getKeyIndex (time[, dimension=0])`
- `def getKeyTime (index, dimension)`
- `def getNumKeys ([dimension=0])`
- `def removeAnimation ([dimension=0])`
- `def setExpression (expr, hasRetVariable[, dimension=0])`
- `def setInterpolationAtTime (time, interpolation[, dimension=0])`

Detailed Description

Animating parameters have values that may change throughout the time. To enable animation the parameter should have at least 1 keyframe. Keyframes can be added in the derived class (since function signature is type specific) with the `setValueAtTime` function. Once 2 keyframes are active on the parameter, the value of the parameter will be interpolated automatically by Natron for a given time. You can control keyframes by adding, removing, changing their values and their `KeyFrameTypeEnum` type.

Note that by default new keyframes are always with a **Smooth** interpolation.

Moreover parameters can have Python expressions set on them to control their value. In that case, the expression takes precedence over any animation that the parameter may have, meaning that the value of the parameter would be computed using the expression provided.

Member functions description

`NatronEngine.AnimatedParam.deleteValueAtTime (time[, dimension=0])`

Parameters

- **time** – float
- **dimension** – int

Removes a keyframe at the given *time* and *dimension* for this parameter, if such keyframe exists.

`NatronEngine.AnimatedParam.getCurrentTime ()`

Return type

Convenience function: returns the current time on the timeline

`NatronEngine.AnimatedParam.getDerivativeAtTime (time[, dimension=0])`

Parameters

- **time** – float
- **dimension** – int

Return type

double

Returns the derivative of the parameter at the given *time* and for the given *dimension*. The derivative is computed on the animation curve of the parameter. This function is irrelevant for parameters that have an expression.

`NatronEngine.AnimatedParam.getExpression (dimension)`

Parameters

Return type

str

Returns the Python expression set on the parameter at the given dimension. When no expression is set, this function returns an empty string.

`NatronEngine.AnimatedParam.getIntegrateFromTimeToTime (time1, time2[, dimension=0])`

Parameters

- **time1** – float
- **time2** – float
- **dimension** – int

Return type float

Integrates the value of the parameter over the range [*time1* - *time2*]. This is done using the animation curve of the parameter of the given *dimension*. Note that if this parameter has an expression, the return value is irrelevant.

`NatronEngine.AnimatedParam.getIsAnimated ([dimension=0])`

Parameters **dimension** – int

Return type bool

Returns whether the given *dimension* has an animation or not. This returns true if the underlying animation curve has 1 or more keyframes.

`NatronEngine.AnimatedParam.getKeyIndex (time[, dimension=0])`

Parameters

- **time** – float
- **dimension** – int

Return type int

Returns the index of the keyframe at the given *time* for the animation curve at the given *dimension*, or -1 if no such keyframe could be found.

`NatronEngine.AnimatedParam.getKeyTime (index, dimension)`

Parameters

- **index** – int
- **dimension** – int

Return type tuple

Returns a tuple [bool,float] where the first member is True if a keyframe exists at the given *index* for the animation curve at the given *dimension*. The second *float* member is the keyframe exact time.

`NatronEngine.AnimatedParam.getNumKeys ([dimension=0])`

Parameters **dimension** – int

Return type int

Returns the number of keyframes for the animation curve at the given *dimension*.

`NatronEngine.AnimatedParam.removeAnimation ([dimension=0])`

Parameters **dimension** – int

Removes all animation for the animation curve at the given *dimension*. Note that this will not remove any expression set.

`NatronEngine.AnimatedParam.setExpression (expr, hasRetVariable[, dimension=0])`

Parameters

- **expr** – str
- **hasRetVariable** – bool
- **dimension** – int

Return type `bool`

Set the Python expression *expr* on the parameter at the given *dimension*. If *hasRetVariable* is `True`, then *expr* is assumed to have a variable *ret* declared. Otherwise, Natron will declare the *ret* variable itself.

`NatronEngine.AnimatedParam.setInterpolationAtTime (time, interpolation[, dimension=0])`

Parameters

- **time** – float
- **interpolation** – `KeyFrameTypeEnum`
- **dimension** – int

Return type `bool`

Set the interpolation of the animation curve of the given dimension at the given keyframe. If no such keyframe could be found, this method returns `False`. Upon success, this method returns `True`.

Example:

```
appl.Blur2.size.setInterpolationAtTime(56, NatronEngine.Natron.KeyframeTypeEnum.  
↪eKeyframeTypeConstant, 0)
```

App

Inherits *Group*

Inherited by: *GuiApp*

Synopsis

The App object represents one instance of a project. See [detailed](#) description. . .

Functions

- def *addProjectLayer* (layer)
- def *addFormat* (formatSpec)
- def *createNode* (pluginID[, majorVersion=-1[, group=None] [, properties=None]])
- def *createReader* (filename[, group=None] [, properties=None])
- def *createWriter* (filename[, group=None] [, properties=None])
- def *getAppID* ()
- def *getProjectParam* (name)
- def *getViewNames* ()
- def *render* (effect, firstFrame, lastFrame[, frameStep])
- def *render* (tasks)
- def *saveTempProject* (filename)
- def *saveProject* (filename)
- def *saveProjectAs* (filename)
- def *loadProject* (filename)
- def *resetProject* ()

- `def closeProject ()`
- `def newProject ()`
- `def timelineGetLeftBound ()`
- `def timelineGetRightBound ()`
- `def timelineGetTime ()`
- `def writeToScriptEditor (message)`

Detailed Description

An App object is created automatically every times a new project is opened. For each instance of Natron opened, there's a new instance of App. You never create an App object by yourself, instead you can access them with variables that Natron pre-declared for you: The first instance will be named `app1`, the second `app2`, etc... See [this section](#) for an explanation of auto-declared variables.

When in background mode, (interpreter or render mode) there will always ever be a single App instance, so Natron will make the following assignment before running any other script:

```
app = app1
```

So you don't have to bother on which instance you're in. For [Group](#) Python plug-ins exported from Natron, they have a function with the following signature:

```
def createInstance (app, group) :
```

So you don't have to bother again on which App instance your script is run. You should only ever need to refer to the `app1`, `app2`... variables when using the Script Editor.

Finally, you can always access the App object of any instance by calling the following function when your script is for command line (background mode):

```
natron.getInstance (index)
```

Or the following function when you want to use GUI functionalities:

```
natron.getGuiInstance (index)
```

Warning: Note that in both cases, *index* is a 0-based number. So to retrieve *app1* you would need to call the function with *index* = 0.

Creating nodes

The App object is responsible for creating new nodes. To create a node, you need to specify which plug-in you want to instantiate and optionally specify which major version should your node instantiate if the plug-in has multiple versions. For instance we could create a new Reader node this way:

```
reader = app.createNode ("fr.inria.openfx.ReadOIIIO")
```

You can also specify the group into which the node should be created, None being the project's top level:

```
group = app.createNode ("fr.inria.built-in.Group")
reader = app.createNode ("fr.inria.openfx.ReadOIIIO", -1, group)
```

For convenience, small wrapper functions have been made to directly create a Reader or Writer given a filename:

```
reader = app.createReader("/Users/me/Pictures/mySequence###.exr")
writer = app.createWriter("/Users/me/Pictures/myVideo.mov")
```

In case 2 plug-ins can decode/encode the same format, e.g. ReadPSD and ReadOIIO can both read .psd files, internally Natron picks the “best” OpenFX plug-in to decode/encode the image sequence/video according to the settings in the Preferences of Natron. If however you need a specific decoder/encoder to decode/encode the file format, you can use the `getSettings()` function with the exact plug-in ID.

In Natron you can call the following function to get a sequence with all plug-in IDs currently available:

```
natron.getPluginIDs()
```

You can also get a sub-set of those plug-ins with the `getPluginIDs(filter)` which returns only plug-in IDs containing the given filter (compared without case sensitivity).

Accessing the settings of Natron

To modify the parameters in the *Preferences* of Natron, you can call the `getSettings()` function to get an object containing all the *parameters* of the preferences.

Accessing the project settings

You can get a specific *parameter* of the project settings with the `getProjectParam(name)` function.

Member functions description

`NatronEngine.App.addProjectLayer(layer)`

Parameters `layer` – ImageLayer

Appends a new project-wide layer. It will be available to all layer menus of all nodes. Each layer menu must be refreshed individually with either a right click on the menu or by changing nodes connections to get access to the new layer. Layer names are unique: even if you add duplicates to the layers list, only the first one in the list with that name will be available in the menus.

`NatronEngine.App.addFormat(formatSpec)`

Parameters `formatSpec` – str

Attempts to add a new format to the project’s formats list. The *formatSpec* parameter must follow this spec: First the name of the format, without any spaces and without any non Python compliant characters; followed by a space and then the size of the format, in the form *width*x*height*; followed by a space and then the pixel aspect ratio of the format. For instance:

```
HD 1920x1080 1
```

Wrongly formatted format will be omitted and a warning will be printed in the *ScriptEditor*.

`NatronEngine.App.createNode(pluginID[, majorVersion=-1[, group=None] [, properties=None]])`

Parameters

- **pluginID** – str
- **majorVersion** – int
- **group** – Group
- **properties** – Dict

Return type Effect

Creates a new node instantiating the plug-in specified with the given *pluginID* at the given *majorVersion*. If *majorVersion* is -1, the highest version of the plug-in will be instantiated. The optional *group* parameter can be used to specify into which *group* the node should be created, *None* meaning the project's top level.

In Natron you can call the following function to get a sequence with all plug-in IDs currently available:

```
natron.getPluginIDs()
```

The optional parameter *properties* is a dictionary containing properties that may modify the creation of the node, such as hiding the node GUI, disabling auto-connection in the NodeGraph, etc...

The properties are values of type Bool, Int, Float or String and are mapped against a unique *key* identifying them.

Most properties have a default value and don't need to be specified, except the pluginID property.

Below is a list of all the properties available that are recognized by Natron. If you specify an unknown property, Natron will print a warning in the Script Editor.

All properties type have been wrapped to Natron types:

- A boolean property is represented by the **BoolNodeCreationProperty** class
- An int property is represented by the **IntNodeCreationProperty** class
- A float property is represented by the **FloatNodeCreationProperty** class
- A string property is represented by the **StringNodeCreationProperty** class

Here is an example on how to pass properties to the createNode function:

```
app.createNode("net.sf.cimg.CImgBlur", -1, app, dict([ (
↪ "CreateNodeArgsPropSettingsOpened", NatronEngine.BoolNodeCreationProperty(True)),
↪ ("CreateNodeArgsPropNodeInitialParamValues", NatronEngine.
↪ StringNodeCreationProperty("size")), ("CreateNodeArgsPropParamValue_size",
↪ NatronEngine.FloatNodeCreationProperty([2.3, 5.1])) ]))
```

- **Name: CreateNodeArgsPropPluginID**

Dimension: 1

Type: string

Default: None

Description: Indicates the ID of the plug-in to create. This property is mandatory. It is set automatically by passing the pluginID to the createNode function

- **Name: CreateNodeArgsPropPluginVersion**

Dimension: 2

Type: int

Default: -1,-1

Description: Indicates the version of the plug-in to create. With the value (-1,-1) Natron will load the highest possible version available for that plug-in.

- **Name: CreateNodeArgsPropNodeInitialPosition**

Dimension: 2

Type: float

Default: None

Description: Indicates the initial position of the node in the nodegraph. By default Natron will position the node according to the state of the interface (current selection, position of the viewport, etc...)

- **Name: CreateNodeArgsPropNodeInitialName**

Dimension: 1

Type: string

Default: None

Description: Indicates the initial *script-name* of the node By default Natron will name the node according to the plug-in label and will add a digit afterwards depending on the current number of instances of that plug-in.

- **Name: CreateNodeArgsPropNodeInitialParamValues**

Dimension: N

Type: string

Default: None

Description: Contains a sequence of parameter script-names for which a default value is specified by a property. Each default value must be specified by a property whose name is in the form *CreateNodeArgsPropParamValue_PARAMETERNAME* where *PARAMETERNAME* must be replaced by the *script-name* of the parameter. The property must have the same type as the data-type of the parameter (e.g. int for IntParam, float for FloatParam, bool for BooleanParam, String for StringParam).

- **Name: CreateNodeArgsPropOutOfProject**

Dimension: 1

Type: bool

Default: False

Description: When True the node will not be part of the project. The node can be used for internal used, e.g. in a Python script but will not appear to the user. It will also not be saved in the project.

- **Name: CreateNodeArgsPropNoNodeGUI**

Dimension: 1

Type: bool

Default: False

Description: * If True, the node will not have any GUI created. The property CreateNodeArgsPropOutOfProject set to True implies this.

- **Name: CreateNodeArgsPropSettingsOpened**

Dimension: 1

Type: bool

Default: False

Description: * If True, the node settings panel will not be opened by default when created. If the property CreateNodeArgsPropNoNodeGUI is set to true or CreateNodeArgsPropOutOfProject is set to true, this property has no effect.

- **Name: CreateNodeArgsPropAutoConnect**

Dimension: 1

Type: bool

Default: False

Description: * If True, Natron will try to automatically connect the node to others depending on the user selection. If the property CreateNodeArgsPropNoNodeGUI is set to true or CreateNodeArgsPropOutOfProject is set to true, this property has no effect.

- **Name: CreateNodeArgsPropAddUndoRedoCommand**

Dimension: 1

Type: bool

Default: False

Description: Natron will push a undo/redo command to the stack when creating this node. If the property CreateNodeArgsPropNoNodeGUI is set to true or CreateNodeArgsPropOutOfProject is set to true, this property has no effect.

- **Name: CreateNodeArgsPropSilent**

Dimension: 1

Type: bool

Default: True

Description: When set to True, Natron will not show any information, error, warning, question or file dialog when creating the node.

NatronEngine.App.**createReader** (*filename* [, *group*=None] [, *properties*=None])

Parameters

- **filename** – str
- **group** – Group

Return type Effect

Creates a reader to decode the given *filename*. The optional *group* parameter can be used to specify into which *group* the node should be created, *None* meaning the project's top level.

In case 2 plug-ins can decode the same format, e.g. ReadPSD and ReadOIIO can both read .psd files, internally Natron picks the “best” OpenFX plug-in to decode the image sequence/video according to the settings in the Preferences of Natron. If however you need a specific decoder to decode the file format, you can use the *getSettings()* function with the exact plug-in ID.

NatronEngine.App.**createWriter** (*filename* [, *group*=None] [, *properties*=None])

Parameters

- **filename** – str
- **group** – Group

Return type Effect

Creates a reader to decode the given *filename*. The optional *group* parameter can be used to specify into which *group* the node should be created, *None* meaning the project's top level.

In case 2 plug-ins can encode the same format, e.g. WritePFM and WriteOIIO can both write .pfm files, internally Natron picks the “best” OpenFX plug-in to encode the image sequence/video according to the settings in the Preferences of Natron. If however you need a specific decoder to encode the file format, you can use the *getSettings()* function with the exact plug-in ID.

NatronEngine.App.**getAppID** ()

Return type int

Returns the **zero-based** ID of the App instance. *app1* would have the AppID 0, *app2* would have the AppID 1, and so on...

NatronEngine.App.**getProjectParam** (*name*)

Parameters *name* – str

Return type Param

Returns a project *Param* given its *name* (script-name). See [this section](#) for an explanation of *script-name* vs. *label*.

NatronEngine.App.getViewNames()

Return type Sequence

Returns a sequence with the name of all the views in the project as setup by the user in the “Views” tab of the Project Settings.

NatronEngine.App.render(effect, firstFrame, lastFrame[, frameStep])

Parameters

- **effect** – Effect
- **firstFrame** – int
- **lastFrame** – int
- **frameStep** – int

Starts rendering the given *effect* on the frame-range defined by [*firstFrame*,**lastFrame**]. The *frameStep* parameter indicates how many frames the timeline should step after rendering each frame. The value must be greater or equal to 1. The *frameStep* parameter is optional and if not given will default to the value of the **Frame Increment** parameter in the Write node.

For instance:

```
render(effect, 1, 10, 2)
```

Would render the frames 1,3,5,7,9

This is a blocking function only in background mode. A blocking render means that this function returns only when the render finishes (from failure or success).

This function should only be used to render with a Write node or DiskCache node.

NatronEngine.App.render(tasks)

Parameters tasks – sequence

This function takes a sequence of tuples of the form (*effect*,*firstFrame*,*lastFrame*[,*frameStep*]) The *frameStep* is optional in the tuple and if not set will default to the value of the **Frame Increment** parameter in the Write node.

This is an overloaded function. Same as *render(effect, firstFrame, lastFrame, frameStep)* but all *tasks* will be rendered concurrently.

This function is called when rendering a script in background mode with multiple writers.

This is a blocking call only in background mode.

NatronEngine.App.timelineGetLeftBound()

Return type int

Returns the *left bound* of the timeline, that is, the first member of the project’s frame-range parameter

NatronEngine.App.timelineGetRightBound()

Return type int

Returns the *right bound* of the timeline, that is, the second member of the project’s frame-range parameter

NatronEngine.App.timelineGetTime()

Return type int

Get the timeline’s current time. In Natron there’s only a single internal timeline and all Viewers are synchronised on that timeline. If the user seeks a specific frames, then all Viewers will render that frame.

NatronEngine.App.writeToScriptEditor(message)

Parameters `message` – str

Writes the given *message* to the Script Editor panel of Natron. This can be useful to inform the user of various informations, warnings or errors.

`NatronEngine.App.saveProject (filename)`

Parameters `filename` – str

Return type bool

Saves the current project under the current project name. If the project has never been saved so far, this function e saves the project to the file indicated by the *filename* parameter. In GUI mode, if *filename* is empty, it asks the user where to save the project in GUI mode.

This function returns *True* if it saved successfully, *False* otherwise.

`NatronEngine.App.saveProjectAs (filename)`

Parameters `filename` – str

Return type bool

Save the project under the given *filename*. In GUI mode, if *filename* is empty, it prompts the user where to save the project.

This function returns *True* if it saved successfully, *False* otherwise.

`NatronEngine.App.saveTempProject (filename)`

Parameters `filename` – str

Return type bool

Saves a copy of the project to the given *filename* without updating project properties such as the project path, last save time etc... This function returns *True* if it saved successfully, *False* otherwise.

`NatronEngine.App.loadProject (filename)`

Parameters `filename` – str

Return type App

Loads the project indicated by *filename*. In GUI mode, this will open a new window only if the current window has modifications. In background mode this will close the current project of this App and open the project indicated by *filename* in it. This function returns the App object upon success, *None* otherwise.

`NatronEngine.App.resetProject ()`

Return type bool

Attempts to close the current project, without wiping the window. In GUI mode, the user is first prompted to saved his/her changes and can abort the reset, in which case this function will return *False*. In background mode this function always succeeds, hence always returns *True*. this always succeed.

`NatronEngine.App.closeProject ()`

Return type bool

Same as `resetProject ()` except that the window will close in GUI mode. Also, if this is the last App alive, Natron will close.

`NatronEngine.App.newProject ()`

Return type App

Creates a new App. In GUI mode, this will open a new window. Upon success, the App object is returned, otherwise *None* is returned.

AppSettings

Synopsis

This class gathers all settings of Natron. You can access them exactly like you would for the *Effect* class.

Functions

- def *getParam* (scriptName)
- def *getParams* ()
- def *restoreDefaultSettings* ()
- def *saveSettings* ()

Member functions description

NatronEngine.AppSettings.**getParam** (*scriptName*)

Parameters *scriptName* – str

Return type Param

Returns a *Param* by its *scriptName*. See *this* section for a detailed explanation of what is the *script-name*.

NatronEngine.AppSettings.**getParams** ()

Return type sequence

Returns a sequence with all *Param* composing the settings.

NatronEngine.AppSettings.**restoreDefaultSettings** ()

Restores all settings to their default value shipped with Natron.

NatronEngine.AppSettings.**saveSettings** ()

Saves all the settings on disk so that they will be restored with their current value on the following runs of Natron.

BezierCurve

Inherits *ItemBase*

Synopsis

A BezierCurve is the class used for beziers, ellipses and rectangles. See *detailed* description. ...

Functions

- def *addControlPoint* (x, y)
- def *addControlPointOnSegment* (index, t)
- def *getActivatedParam* ()
- def *getColor* (time)
- def *getColorParam* ()

- `def getCompositingOperator ()`
- `def getCompositingOperatorParam ()`
- `def getControlPointPosition (index,time)`
- `def getFeatherDistance (time)`
- `def getFeatherDistanceParam ()`
- `def getFeatherFalloff (time)`
- `def getFeatherFalloffParam ()`
- `def getFeatherPointPosition (index,time)`
- `def getIsActivated (time)`
- `def getKeyframes ()`
- `def getNumControlPoints ()`
- `def getOpacity (time)`
- `def getOpacityParam ()`
- `def getOverlayColor ()`
- `def isCurveFinished ()`
- `def moveFeatherByIndex (index, time, dx, dy)`
- `def moveLeftBezierPoint (index, time, dx, dy)`
- `def movePointByIndex (index, time, dx, dy)`
- `def moveRightBezierPoint (index, time, dx, dy)`
- `def removeControlPointByIndex (index)`
- `def setActivated (time, activated)`
- `def setColor (time, r, g, b)`
- `def setCompositingOperator (op)`
- `def setCurveFinished (finished)`
- `def setFeatherDistance (dist, time)`
- `def setFeatherFalloff (falloff, time)`
- `def setFeatherPointAtIndex (index, time, x, y, lx, ly, rx, ry)`
- `def setOpacity (opacity, time)`
- `def setOverlayColor (r, g, b)`
- `def setPointAtIndex (index, time, x, y, lx, ly, rx, ry)`

Detailed Description

Almost all functionalities available to the user have been made available to the Python API, although in practise making a shape just by calling functions might be tedious due to the potential huge number of control points and keyframes. You can use the Natron Group node's export functionality to generate automatically a script from a Roto node within that group.

A Bezier initially is in an *opened* state, meaning it doesn't produce a shape yet. At this stage you can then add control points using the `:func'addControlPoint(x,y)<NatronEngine.BezierCurve.addControlPoint>'` function. Once you're one adding control points, call the function `setCurveFinished(finished)` to close the shape by connecting the last control point with the first.

Once finished, you can refine the Bezier curve by adding control points with the `addControlPointOnSegment(index, t)` function. You can then move and remove control points of the Bezier.

To get the position of the control points of the Bezier as well as the position of the feather points, use the functions `getControlPointPosition` and `getFeatherPointPosition`. The *index* passed to the function must be between 0 and `getNumControlPoints - 1`.

The *time* passed to the function corresponds to a time on the timeline's in frames. If it lands on a keyframe of the Bezier shape, then the position at that keyframe is returned, otherwise the position is sampled between the surrounding keyframes.

To get a list of all keyframes time for a Bezier call the function `getKeyframes()`.

A Bezier curve has several parameters that the API allows you to modify:

- opacity
- color
- feather distance
- feather fall-off
- enable state
- overlay color
- compositing operator

Each of them is a regular parameter that you can access to modify or query its properties. All parameters can be retrieved with their *script-name* with the function `getParam(scriptName)`.

Member functions description

NatronEngine.BezierCurve.CairoOperatorEnum

This enumeration represents the different blending modes of a shape. See the user interface for the different modes, or type `help(NatronEngine.BezierCurve.CairoOperatorEnum)` to see the different values.

NatronEngine.BezierCurve.addControlPoint(x, y)

Parameters

- **x** – float
- **y** – float

Adds a new control point to an *opened* shape (see `isCurveFinished()`) at coordinates (x,y). By default the feather point attached to this point will be equivalent to the control point. If the auto-keying is enabled in the user interface, then this function will set a keyframe at the timeline's current time for this shape.

NatronEngine.BezierCurve.addControlPointOnSegment(index, t)

Parameters

- **index** – PySide.QtCore.int
- **t** – PySide.QtCore.double

Adds a new control point to a *closed* shape (see `isCurveFinished()`). The *index* is the index of the Bezier segment linking the control points at *index* and *index + 1*. *t* is a value between [0,1] indicating the distance from the control point *index* the new control point should be. The closer to 1 *t* is, the closer the new control point will be to the control point at *index + 1*. By default the feather point attached to this point will be equivalent to the control point.

If the auto-keying is enabled in the user interface, then this function will set a keyframe at the timeline's current time for this shape.

NatronEngine.BezierCurve.getActivatedParam()

Return type BooleanParam

Returns the *Param* controlling the enabled state of the Bezier.

NatronEngine.BezierCurve.**getColor** (*time*)

Parameters *time* – int

Return type ColorTuple

Returns the value of the color parameter at the given time as an [R,G,B,A] tuple. Note that alpha will always be 1.

NatronEngine.BezierCurve.**getColorParam** ()

Return type ColorParam

Returns the *Param* controlling the color of the Bezier.

NatronEngine.BezierCurve.**getCompositingOperator** ()

Return type NatronEngine.BezierCurve.CairoOperatorEnum

Returns the blending mode for this shape. Type help(NatronEngine.BezierCurve.CairoOperatorEnum) to see the different values possible.

NatronEngine.BezierCurve.**getCompositingOperatorParam** ()

Return type NatronEngine.ChoiceParam

Returns the *Param* controlling the blending mode of the Bezier.

NatronEngine.BezierCurve.**getControlPointPosition** (*index*, *time*)

Parameters

- *index* – int
- *time* – float

Return type PyTuple

Returns a tuple with the position of the control point at the given *index* as well as the position of its left and right tangents.

The tuple is encoded as such:

```
(x,y, leftTangentX, leftTangentY, rightTangentX, rightTangentY)
```

The position of the left and right tangents is absolute and not relative to (x,y).

The *index* passed to the function must be between 0 and *getNumControlPoints* -1. The *time* passed to the function corresponds to a time on the timeline's in frames. If it lands on a keyframe of the Bezier shape, then the position at that keyframe is returned, otherwise the position is sampled between the surrounding keyframes.

To get a list of all keyframes time for a Bezier call the function *getKeyframes* ().

NatronEngine.BezierCurve.**getFeatherDistance** (*time*)

Parameters *time* – int

Return type float

Returns the feather distance of this shape at the given *time*.

NatronEngine.BezierCurve.**getFeatherDistanceParam** ()

Return type NatronEngine.DoubleParam

Returns the *Param* controlling the feather distance of the Bezier.

NatronEngine.BezierCurve.**getFeatherFalloff** (*time*)

Parameters *time* – int

Return type float

Returns the feather fall-off of this shape at the given *time*.

`NatronEngine.BezierCurve.getFeatherFallOffParam()`

Return type DoubleParam

Returns the *Param* controlling the color of the Bezier.

`NatronEngine.BezierCurve.getFeatherPointPosition(index, time)`

Parameters

- **index** – int
- **time** – float

Return type PyTuple

Returns a tuple with the position of the feather point at the given *index* as well as the position of its left and right tangents.

The tuple is encoded as such:

```
(x, y, leftTangentX, leftTangentY, rightTangentX, rightTangentY)
```

The position of the left and right tangents is absolute and not relative to (x,y).

The *index* passed to the function must be between 0 and `getNumControlPoints`-1. The *time* passed to the function corresponds to a time on the timeline's in frames. If it lands on a keyframe of the Bezier shape, then the position at that keyframe is returned, otherwise the position is sampled between the surrounding keyframes.

To get a list of all keyframes time for a Bezier call the function `getKeyframes()`.

`NatronEngine.BezierCurve.getIsActivated(time)`

Parameters **time** – int

Return type bool

Returns whether the curve is enabled or not at the given *time*. When not activated the curve will not be rendered at all in the image.

`NatronEngine.BezierCurve.getKeyframes()`

Return type PyList

Returns a list of all keyframes set on the Bezier animation.

`NatronEngine.BezierCurve.getNumControlPoints()`

Return type int

Returns the number of control points for this shape.

`NatronEngine.BezierCurve.getOpacity(time)`

Parameters **time** – int

Return type float

Returns the opacity of the curve at the given *time*.

`NatronEngine.BezierCurve.getOpacityParam()`

Return type DoubleParam

Returns the *Param* controlling the opacity of the Bezier.

`NatronEngine.BezierCurve.getOverlayColor()`

Return type ColorTuple

Returns the overlay color of this shape as a [R,G,B,A] tuple. Alpha will always be 1.

`NatronEngine.BezierCurve.isCurveFinished()`

Return type `bool`

Returns whether the curve is finished or not. A finished curve will have a Bezier segment between the last control point and the first control point and the Bezier will be rendered in the image.

`NatronEngine.BezierCurve.moveFeatherByIndex(index, time, dx, dy)`

Parameters

- **index** – `int`
- **time** – `int`
- **dx** – `float`
- **dy** – `float`

Moves the feather point at the given *index* (zero-based) by the given delta (dx,dy). The *time* parameter is given so that if auto-keying is enabled a new keyframe will be set.

`NatronEngine.BezierCurve.moveLeftBezierPoint(index, time, dx, dy)`

Parameters

- **index** – `int`
- **time** – `int`
- **dx** – `float`
- **dy** – `float`

Moves the left Bezier point of the control point at the given *index* by the given delta. The *time* parameter is given so that if auto-keying is enabled a new keyframe will be set.

`NatronEngine.BezierCurve.movePointByIndex(index, time, dx, dy)`

Parameters

- **index** – `int`
- **time** – `int`
- **dx** – `float`
- **dy** – `float`

Moves the point at the given *index* (zero-based) by the given delta (dx,dy). The *time* parameter is given so that if auto-keying is enabled a new keyframe will be set.

`NatronEngine.BezierCurve.moveRightBezierPoint(index, time, dx, dy)`

Parameters

- **index** – `int`
- **time** – `int`
- **dx** – `float`
- **dy** – `float`

Moves the right Bezier point at the given *index* (zero-based) by the given delta (dx,dy). The *time* parameter is given so that if auto-keying is enabled a new keyframe will be set.

`NatronEngine.BezierCurve.removeControlPointByIndex(index)`

Parameters **index** – `int`

Removes the control point at the given *index* (zero-based).

`NatronEngine.BezierCurve.setActivated(time, activated)`

Parameters

- **time** – int
- **activated** – bool

Set a new keyframe for the *activated* parameter at the given *time*

NatronEngine.BezierCurve.**setColor** (*time*, *r*, *g*, *b*)

Parameters

- **time** – int
- **r** – float
- **g** – float
- **b** – float

Set a new keyframe for the *color* parameter at the given *time*

NatronEngine.BezierCurve.**setCompositingOperator** (*op*)

Parameters **op** – *NatronEngine.BezierCurve.CairoOperatorEnum*

Set the compositing operator for this shape.

NatronEngine.BezierCurve.**setCurveFinished** (*finished*)

Parameters **finished** – bool

Set whether the curve should be finished or not. See *isCurveFinished()*

NatronEngine.BezierCurve.**setFeatherDistance** (*dist*, *time*)

Parameters

- **dist** – float
- **time** – int

Set a new keyframe for the *feather distance* parameter at the given *time*

NatronEngine.BezierCurve.**setFeatherFalloff** (*falloff*, *time*)

Parameters

- **falloff** – float
- **time** – int

Set a new keyframe for the *feather fall-off* parameter at the given *time*

NatronEngine.BezierCurve.**setFeatherPointAtIndex** (*index*, *time*, *x*, *y*, *lx*, *ly*, *rx*, *ry*)

Parameters

- **index** – int
- **time** – int
- **x** – float
- **y** – float
- **lx** – float
- **ly** – float
- **rx** – float
- **ry** – float

Set the feather point at the given *index* at the position (x,y) with the left Bezier point at (lx,ly) and right Bezier point at (rx,ry).

The *time* parameter is given so that if auto-keying is enabled a new keyframe will be set.

`NatronEngine.BezierCurve.setOpacity (opacity, time)`

Parameters

- **opacity** – float
- **time** – int

Set a new keyframe for the *opacity* parameter at the given *time*

`NatronEngine.BezierCurve.setOverlayColor (r, g, b)`

Parameters

- **r** – float
- **g** – float
- **b** – float

Set the overlay color of this shape

`NatronEngine.BezierCurve.setPointAtIndex (index, time, x, y, lx, ly, rx, ry)`

Parameters

- **index** – int
- **time** – int
- **x** – float
- **y** – float
- **lx** – float
- **ly** – float
- **rx** – float
- **ry** – float

Set the point at the given *index* at the position (x,y) with the left Bezier point at (lx,ly) and right Bezier point at (rx,ry).

The *time* parameter is given so that if auto-keying is enabled a new keyframe will be set.

BooleanParam

Inherits *AnimatedParam*

Synopsis

A parameter that contains a boolean value. See *detailed* description below

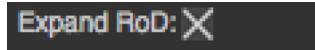
Functions

- def *get* ()
- def *get* (frame)
- def *getDefault*Value ()
- def *getValue* ()
- def *getValueAtTime* (time)
- def *restoreDefault*Value ()

- `def set (x)`
- `def set (x, frame)`
- `def setDefaultValue (value)`
- `def setValue (value)`
- `def setValueAtTime (value, time)`

Detailed Description

A BooleanParam looks like a checkbox in the user interface.



Member functions description

`NatronEngine.BooleanParam.get ()`

Return type `bool`

Returns the value of the parameter at the current timeline's time.

`NatronEngine.BooleanParam.get (frame)`

Parameters `frame` – float

Return type `bool`

Returns the value of the parameter at the given *frame*. This value may be interpolated given the *interpolation* of the underlying animation curve.

`NatronEngine.BooleanParam.getDefaultValue ()`

Return type `bool`

Returns the default value for this parameter.

`NatronEngine.BooleanParam.getValue ()`

Return type `bool`

Same as `get ()`

`NatronEngine.BooleanParam.getValueAtTime (time)`

Parameters `time` – float

Return type `bool`

Same as `get (frame)`

`NatronEngine.BooleanParam.restoreDefaultValue ()`

Removes all animation and expression set on this parameter and set the value to be the default value.

`NatronEngine.BooleanParam.set (x)`

Parameters `x` – bool

Set the value of this parameter to be *x*. If this parameter is animated (see `getIsAnimated (dimension)`) then this function will automatically add a keyframe at the timeline's current time.

`NatronEngine.BooleanParam.set (x, frame)`

Parameters

- `x` – bool

- **frame** – float

Set a new keyframe on the parameter with the value x at the given *frame*.

`NatronEngine.BooleanParam.setDefaultValue (value)`

Parameters **value** – bool

Set the default *value* for this parameter.

`NatronEngine.BooleanParam.setValue (value)`

Parameters **value** – bool

Same as `set (value)`

`NatronEngine.BooleanParam.setValueAtTime (value, time)`

Parameters

- **value** – bool
- **time** – float

Same as `set (value, time)`

ButtonParam

Inherits *Param*

Synopsis

A button parameter that appears in the settings panel of the node.



To insert code to be executed upon a user click of the button, register a function to the `onParamChanged` callback on the node.

Functions

- `def trigger ()`

Member functions description

`NatronEngine.ButtonParam.trigger ()`

Triggers the button action as though the user had pressed it.

ChoiceParam

Inherits : *AnimatedParam*

Synopsis

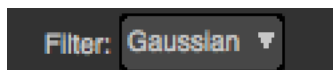
A choice parameter holds an integer value which corresponds to a choice. See *detailed description* below.

Functions

- `def addOption (option, help)`
- `def get ()`
- `def get (frame)`
- `def getDefaultValue ()`
- `def getOption (index)`
- `def getNumOptions ()`
- `def getOptions ()`
- `def getValue ()`
- `def getValueAtTime (time)`
- `def restoreDefaultValue ()`
- `def set (x)`
- `def set (x, frame)`
- `def set (label)`
- `def setDefaultValue (value)`
- `def setDefaultValue (label)`
- `def setOptions (options)`
- `def setValue (value)`
- `def setValueAtTime (value, time)`

Detailed Description

A choice is represented as a drop-down (combobox) in the user interface:



You can add options to the menu using the `addOption(option, help)` function. You can also set them all at once using the `setOptions(options)` function.

The value held internally is a 0-based index corresponding to an entry of the menu. the choice parameter behaves much like an *IntParam*.

Member functions description

`NatronEngine.ChoiceParam.addOption (option, help)`

Parameters

- **option** – str
- **help** – str

Adds a new *option* to the menu. If *help* is not empty, it will be displayed when the user hovers the entry with the mouse.

`NatronEngine.ChoiceParam.get (frame)`

Parameters **frame** – float

Return type `int`

Get the value of the parameter at the given *frame*.

`NatronEngine.ChoiceParam.get()`

Return type `int`

Get the value of the parameter at the current timeline's time.

`NatronEngine.ChoiceParam.getDefaultValue()`

Return type `int`

Get the default value for this parameter.

`NatronEngine.ChoiceParam.getOption(index)`

Parameters `index` – `int`

Return type `str`

Get the menu entry at the given *index*.

`NatronEngine.ChoiceParam.getNumOptions()`

Return type `int`

Returns the number of menu entries.

`NatronEngine.ChoiceParam.getOptions()`

Return type `sequence`

Returns a sequence of string with all menu entries from top to bottom.

`NatronEngine.ChoiceParam.getValue()`

Return type `int`

Same as `get()`

`NatronEngine.ChoiceParam.getValueAtTime(time)`

Parameters `time` – `float`

Return type `float`

Same as `get(frame)`

`NatronEngine.ChoiceParam.restoreDefaultValue()`

Removes all animation and expression set on this parameter and set the value to be the default value.

`NatronEngine.ChoiceParam.set(x)`

Parameters `x` – `int`

Set the value of this parameter to be *x*. If this parameter is animated (see `getIsAnimated(dimension)`) then this function will automatically add a keyframe at the timeline's current time.

`NatronEngine.ChoiceParam.set(x, frame)`

Parameters

- `x` – `int`
- `frame` – `float`

Set a new keyframe on the parameter with the value *x* at the given *frame*.

`NatronEngine.ChoiceParam.set(label)`

Parameters `label` – `str`

Set the value of this parameter given a *label*. The *label* must match an existing option. Strings will be compared without case sensitivity. If not found, nothing happens.

`NatronEngine.ChoiceParam.setDefaultValue (value)`

Parameters `value` – int

Set the default *value* for this parameter.

`NatronEngine.ChoiceParam.setDefaultValue (label)`

Parameters `label` – str

Set the default value from the *label* for this parameter. The *label* must match an existing option. Strings will be compared without case sensitivity. If not found, nothing happens.

`NatronEngine.ChoiceParam.setOptions (options)`

Parameters `options` – class::sequence

Clears all existing entries in the menu and add all entries contained in *options* to the menu.

`NatronEngine.ChoiceParam.setValue (value)`

Parameters `value` – int

Same as `set`

`NatronEngine.ChoiceParam.setValueAtTime (value, time)`

Parameters

- `value` – int
- `time` – int

Same as `set (time) <NatronEngine.ChoiceParam.set ()`

ColorParam

Inherits [AnimatedParam](#)

Synopsis

A color parameter is a RGB[A] value that can be animated throughout the time. See [detailed](#) description...

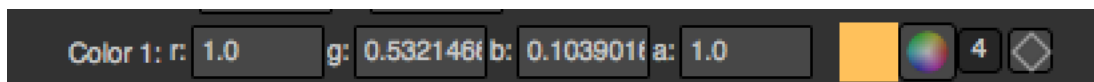
Functions

- def `get ()`
- def `get (frame)`
- def `getDefaultValue ([dimension=0])`
- def `getDisplayMaximum (dimension)`
- def `getDisplayMinimum (dimension)`
- def `getMaximum ([dimension=0])`
- def `getMinimum ([dimension=0])`
- def `getValue ([dimension=0])`
- def `getValueAtTime (time[, dimension=0])`
- def `restoreDefaultValue ([dimension=0])`
- def `set (r, g, b, a)`
- def `set (r, g, b, a, frame)`

- `def setDefaultValue (value[, dimension=0])`
- `def setDisplayMaximum (maximum[, dimension=0])`
- `def setDisplayMinimum (minimum[, dimension=0])`
- `def setMaximum (maximum[, dimension=0])`
- `def setMinimum (minimum[, dimension=0])`
- `def setValue (value[, dimension=0])`
- `def setValueAtTime (value, time[, dimension=0])`

Detailed Description

A color parameter can either be of dimension 3 (RGB) or dimension 4 (RGBA). The user interface for this parameter looks like this:



This parameter type is very similar to a [Double3DParam](#) except that it can have 4 dimensions and has some more controls.

Member functions description

`NatronEngine.ColorParam.get (frame)`

Parameters `frame` – float

Return type `ColorTuple`

Returns a `ColorTuple` of the color held by the parameter at the given *frame*.

`NatronEngine.ColorParam.get ()`

Return type `ColorTuple`

Returns a `ColorTuple` of the color held by the parameter at the current timeline's time.

`NatronEngine.ColorParam.getDefaultValue ([dimension=0])`

Parameters `dimension` – int

Return type float

Returns the default value for this parameter at the given *dimension*.

`NatronEngine.ColorParam.getDisplayMaximum (dimension)`

Parameters `dimension` – int

Return type float

Returns the display maximum for this parameter at the given *dimension*. The display maximum is the maximum value visible on the slider, internally the value can exceed this range.

`NatronEngine.ColorParam.getDisplayMinimum (dimension)`

Parameters `dimension` – int

Return type float

Returns the display minimum for this parameter at the given *dimension*. The display minimum is the minimum value visible on the slider, internally the value can exceed this range.

`NatronEngine.ColorParam.getMaximum ([dimension=0])`

Parameters `dimension` – int

Return type float

Returns the maximum for this parameter at the given *dimension*. The maximum value cannot be exceeded and any higher value will be clamped to this value.

`NatronEngine.ColorParam.getMinimum([dimension=0])`

Parameters `dimension` – int

Return type float

Returns the minimum for this parameter at the given *dimension*. The minimum value cannot be exceeded and any lower value will be clamped to this value.

`NatronEngine.ColorParam.getValue([dimension=0])`

Parameters `dimension` – int

Return type float

Returns the value of this parameter at the given *dimension* at the current timeline's time.

`NatronEngine.ColorParam.getValueAtTime(time[, dimension=0])`

Parameters

- `time` – float
- `dimension` – int

Return type float

Returns the value of this parameter at the given *dimension* at the given *time*.

`NatronEngine.ColorParam.restoreDefaultValue([dimension=0])`

Parameters `dimension` – int

Removes all animation and expression set on this parameter and set the value to be the default value.

`NatronEngine.ColorParam.set(r, g, b, a, frame)`

Parameters

- `r` – float
- `g` – float
- `b` – float
- `a` – float
- `frame` – float

Set a keyframe on each of the 4 animations curves at [r,g,b,a] for the given *frame*. If this parameter is 3-dimensional, the *a* value is ignored.

`NatronEngine.ColorParam.set(r, g, b, a)`

Parameters

- `r` – float
- `g` – float
- `b` – float
- `a` – float

Set the value of this parameter to be `[r,*g*,*b*,*a*]`. If this parameter is animated (see `getIsAnimated(dimension)`) then this function will automatically add a keyframe at the timeline's current time.

`NatronEngine.ColorParam.setDefaultValue(value[, dimension=0])`

Parameters

- **value** – float
- **dimension** – int

Set the default value of this parameter at the given *dimension* to be *value*.

`NatronEngine.ColorParam.setDisplayMaximum (maximum[, dimension=0])`

Parameters

- **maximum** – float
- **dimension** – int

Set the display maximum of the parameter to be *maximum* for the given *dimension*. See `getDisplayMaximum`

`NatronEngine.ColorParam.setDisplayMinimum (minimum[, dimension=0])`

Parameters

- **minimum** – float
- **dimension** – int

Set the display minimum of the parameter to be *minnum* for the given *dimension*. See `getDisplayMinimum`

`NatronEngine.ColorParam.setMaximum (maximum[, dimension=0])`

Parameters

- **maximum** – float
- **dimension** – int

Set the maximum of the parameter to be *maximum* for the given *dimension*. See `getMaximum`

`NatronEngine.ColorParam.setMinimum (minimum[, dimension=0])`

Parameters

- **minimum** – float
- **dimension** – int

Set the minimum of the parameter to be *minimum* for the given *dimension*. See `getMinimum`

`NatronEngine.ColorParam.setValue (value[, dimension=0])`

Parameters

- **value** – float
- **dimension** – int

Set the value of this parameter at the given *dimension* to be *value*. If this parameter is animated (see `getIsAnimated(dimension)`) then this function will automatically add a keyframe at the timeline's current time.

`NatronEngine.ColorParam.setValueAtTime (value, time[, dimension=0])`

Parameters

- **value** – float
- **time** – int
- **dimension** – int

Set a keyframe on each of the animation curve at the given *dimension*. The keyframe will be at the given *time* with the given *value*.

ColorTuple

Synopsis

Utility class used to return [R,G,B,[A]] values.

Functions

- `def __getitem__ (arg__1)`

class NatronEngine.ColorTuple

NatronEngine.ColorTuple.g

NatronEngine.ColorTuple.r

NatronEngine.ColorTuple.a

NatronEngine.ColorTuple.b

NatronEngine.ColorTuple.__getitem__ (*index*)

Parameters `arg__1` – int

Return type float

Returns the item at the given index. This is the bracket operator `[]`

Double2DParam

Inherits *DoubleParam*

Inherited by: *Double3DParam*

Synopsis

See *DoubleParam* for more informations on this class.

Functions

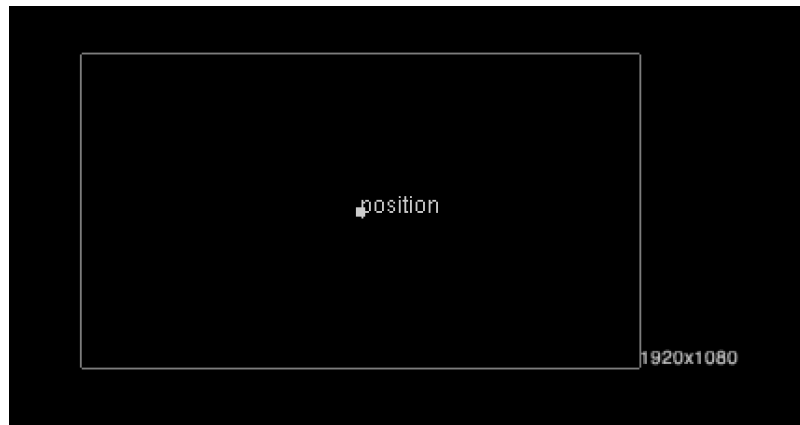
- `def setUsePointInteract<NatronEngine.Double2DParam.setUsePointInteract () (enabled)`
- `def setCanAutoFoldDimensions<NatronEngine.Double2DParam.setCanAutoFoldDimensions () (enabled)`
- `def get ()`
- `def get (frame)`
- `def set (x, y)`
- `def set (x, y, frame)`

Member functions description

NatronEngine.Double2DParam.**setUsePointInteract** (*enabled*)

Parameters `enabled` – bool

When called, the parameter will have its own overlay interact on the viewer as a point that the user can select and drag.



`NatronEngine.Double2DParam.setCanAutoFoldDimensions(enabled)`

Parameters `enabled` – bool

Sets whether all dimensions should be presented as a single value/slider whenever they are equal.

`NatronEngine.Double2DParam.get()`

Return type `Double2DTuple`

Returns a `Double2DTuple` with the [x,y] values for this parameter at the current timeline's time.

`NatronEngine.Double2DParam.get(frame)`

Parameters `frame` – float

Return type `Double2DTuple`

Returns a `Double2DTuple` with the [x,y] values for this parameter at the given *frame*.

`NatronEngine.Double2DParam.set(x, y, frame)`

Parameters

- `x` – float
- `y` – float
- `frame` – float

Same as `set(x, frame)` but for 2-dimensional doubles.

`NatronEngine.Double2DParam.set(x, y)`

Parameters

- `x` – float
- `y` – float

Same as `set(x)` but for 2-dimensional doubles.

Double2DTuple

Synopsis

Utility class to return pair of floating point values.

Functions

- `def __getitem__ (arg__1)`

Detailed Description

class `NatronEngine.Double2DTuple`

`NatronEngine.Double2DTuple.x`

`NatronEngine.Double2DTuple.y`

`NatronEngine.Double2DTuple.__getitem__ (index)`

Parameters `index` – int

Return type float

Returns the item at the given index. This is the bracket operator `[]`

Double3DParam

Inherits *Double2DParam*

Synopsis

See *DoubleParam* for more informations on this class.

Functions

- `def get ()`
- `def get (frame)`
- `def set (x, y, z)`
- `def set (x, y, z, frame)`

Member functions description

`NatronEngine.Double3DParam.get ()`

Return type *Double3DTuple*

Returns a *Double3DTuple* with the [x,y,z] values for this parameter at the current timeline's time.

`NatronEngine.Double3DParam.get (frame)`

Parameters `frame` – float

Return type *Double3DTuple*

Returns a *Double3DTuple* with the [x,y,z] values for this parameter at the given *frame*.

`NatronEngine.Double3DParam.set (x, y, z, frame)`

Parameters

- `x` – float
- `y` – float
- `z` – float

- **frame** – `PySide.QtCore.int`

Same as `set(x, frame)` but for 3-dimensional doubles.

`NatronEngine.Double3DParam.set(x, y, z)`

Parameters

- **x** – float
- **y** – float
- **z** – float

Same as `set(x)` but for 3-dimensional doubles.

Double3DTuple

Synopsis

Utility class to return pair of floating point values.

Functions

- `def __getitem__(index)`

Detailed Description

class `NatronEngine.Double3DTuple`

`NatronEngine.Double3DTuple.x`

`NatronEngine.Double3DTuple.y`

`NatronEngine.Double3DTuple.z`

`NatronEngine.Double3DTuple.__getitem__(index)`

Parameters `index` – int

Return type float

Returns the item at the given index. This is the bracket operator `[]`

DoubleParam

Inherits *AnimatedParam*

Inherited by: *Double2DParam*, *Double3DParam*

Synopsis

A double param can contain one or multiple floating point values. See *detailed* description. . .

Functions

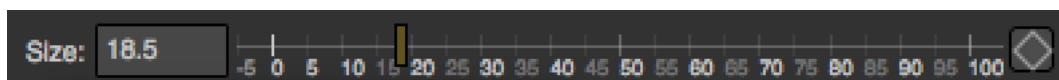
- `def get ()`
- `def get (frame)`
- `def getDefaultValue ([dimension=0])`
- `def getDisplayMaximum (dimension)`
- `def getDisplayMinimum (dimension)`
- `def getMaximum ([dimension=0])`
- `def getMinimum ([dimension=0])`
- `def getValue ([dimension=0])`
- `def getValueAtTime (time[, dimension=0])`
- `def restoreDefaultValue ([dimension=0])`
- `def set (x)`
- `def set (x, frame)`
- `def setDefaultValue (value[, dimension=0])`
- `def setDisplayMaximum (maximum[, dimension=0])`
- `def setDisplayMinimum (minimum[, dimension=0])`
- `def setMaximum (maximum[, dimension=0])`
- `def setMinimum (minimum[, dimension=0])`
- `def setValue (value[, dimension=0])`
- `def setValueAtTime (value, time[, dimension=0])`

Detailed Description

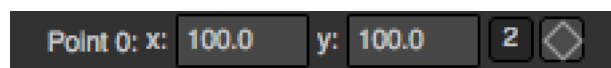
A double param can have 1 to 3 dimensions. (See [Double2DParam](#) and [Double3DParam](#)). Usually this is used to represent a single floating point value that may animate over time.

The user interface for them varies depending on the number of dimensions.

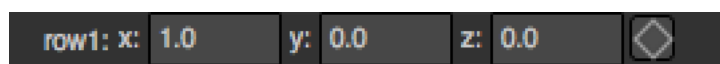
A 1-dimensional [DoubleParam](#)



A 2-dimensional [Double2DParam](#)



A 3-dimensional [Double3DParam](#)



Member functions description

`NatronEngine.DoubleParam.get (frame)`

Parameters `frame` – float

Return type float

Returns the value of this parameter at the given *frame*. If the animation curve has an animation (see `getIsAnimated`) then the value will be interpolated using the *interpolation* chosen by the user for the curve.

`NatronEngine.DoubleParam.get ()`

Return type float

Returns the value of this parameter at the given current timeline's time.

`NatronEngine.DoubleParam.getDefaultValue ([dimension=0])`

Parameters `dimension` – int

Return type float

Returns the default value for this parameter. *dimension* is meaningless for the `DoubleParam` class because it is 1-dimensional, but is useful for inherited classes *Double2DParam* and *Double3DParam*

`NatronEngine.DoubleParam.getDisplayMaximum (dimension)`

Parameters `dimension` – int

Return type double

Returns the display maximum for this parameter at the given *dimension*. The display maximum is the maximum value visible on the slider, internally the value can exceed this range.

`NatronEngine.DoubleParam.getDisplayMinimum (dimension)`

Parameters `dimension` – int

Return type float

Returns the display minimum for this parameter at the given *dimension*. The display minimum is the minimum value visible on the slider, internally the value can exceed this range.

`NatronEngine.DoubleParam.getMaximum ([dimension=0])`

Parameters `dimension` – int

Return type float

Returns the maximum for this parameter at the given *dimension*. The maximum value cannot be exceeded and any higher value will be clamped to this value.

`NatronEngine.DoubleParam.getMinimum ([dimension=0])`

Parameters `dimension` – int

Return type float

Returns the minimum for this parameter at the given *dimension*. The minimum value cannot be exceeded and any lower value will be clamped to this value.

`NatronEngine.DoubleParam.getValue ([dimension=0])`

Parameters `dimension` – int

Return type float

Returns the value of this parameter at the given *dimension* at the current timeline's time.

`NatronEngine.DoubleParam.getValueAtTime (time[, dimension=0])`

Parameters

- **time** – float
- **dimension** – int

Return type float

Returns the value of this parameter at the given *dimension* at the given *time*.

If the animation curve has an animation (see `getIsAnimated`) then the value will be interpolated using the *interpolation* chosen by the user for the curve.

`NatronEngine.DoubleParam.restoreDefaultValue ([dimension=0])`

Parameters **dimension** – int

Returns the value of this parameter at the given *dimension* at the given *time*.

`NatronEngine.DoubleParam.set (x, frame)`

Parameters

- **x** – float
- **frame** – float

Set a new keyframe on the parameter with the value *x* at the given *frame*.

`NatronEngine.DoubleParam.set (x)`

Parameters **x** – float

Set the value of this parameter to be *x*. If this parameter is animated (see `getIsAnimated (dimension)`) then this function will automatically add a keyframe at the timeline's current time.

`NatronEngine.DoubleParam.setDefaultValue (value[, dimension=0])`

Parameters

- **value** – float
- **dimension** – int

Set the default *value* for this parameter at the given *dimension*.

`NatronEngine.DoubleParam.setDisplayMaximum (maximum[, dimension=0])`

Parameters

- **maximum** – float
- **dimension** – int

Set the display maximum of the parameter to be *maximum* for the given *dimension*. See `getDisplayMaximum`

`NatronEngine.DoubleParam.setDisplayMinimum (minimum[, dimension=0])`

Parameters

- **minimum** – float
- **dimension** – int

Set the display minimum of the parameter to be *minimum* for the given *dimension*. See `getDisplayMinimum`

`NatronEngine.DoubleParam.setMaximum (maximum[, dimension=0])`

Parameters

- **maximum** – float
- **dimension** – int

Set the maximum of the parameter to be *maximum* for the given *dimension*. See `getMaximum`

`NatronEngine.DoubleParam.setMinimum (minimum[, dimension=0])`

Parameters

- **minimum** – float
- **dimension** – int<PySide.QtCore.int<

Set the minimum of the parameter to be *minimum* for the given *dimension*. See `getMinimum`

`NatronEngine.DoubleParam.setValue (value[, dimension=0])`

Parameters

- **value** – float
- **dimension** – int

Same as `set (value, dimension)`

`NatronEngine.DoubleParam.setValueAtTime (value, time[, dimension=0])`

Parameters

- **value** – float
- **time** – float
- **dimension** – int

Same as `set (value, time, dimension)`

Effect

Inherits: `Group`, `UserParamHolder`

Synopsis

This object represents a single node in Natron, that is: an instance of a plug-in. See [Detailed Description](#)

Functions

- `def addUserPlane (planeName, channels)`
- `def endChanges ()`
- `def beginChanges ()`
- `def canConnectInput (inputNumber, node)`
- `def connectInput (inputNumber, input)`
- `def destroy ([autoReconnect=true])`
- `def disconnectInput (inputNumber)`
- `def getAvailableLayers ()`
- `def getBitDepth ()`
- `def getColor ()`
- `def getCurrentTime ()`
- `def getOutputFormat ()`
- `def getFrameRate ()`
- `def getInput (inputNumber)`
- `def getInput (inputName)`

- `def getLabel ()`
- `def getInputLabel (inputNumber)`
- `def getMaxInputCount ()`
- `def getParam (name)`
- `def getParams ()`
- `def getPluginID ()`
- `def getPosition ()`
- `def getPremult ()`
- `def getPixelAspectRatio ()`
- `def getRegionOfDefinition (time,view)`
- `def getRotoContext ()`
- `def getTrackerContext ()`
- `def getScriptName ()`
- `def getSize ()`
- `def getUserPageParam ()`
- `def isUserSelected ()`
- `def isReaderNode ()`
- `def isWriterNode ()`
- `def setColor (r, g, b)`
- `def setLabel (name)`
- `def setPosition (x, y)`
- `def setScriptName (scriptName)`
- `def setSize (w, h)`
- `def setSubGraphEditable (editable)`
- `def setPagesOrder (pages)`

Detailed Description

The Effect object can be used to operate with a single node in Natron. To create a new Effect, use the `app.createNode(pluginID)` function.

Natron automatically declares a variable to Python when a new Effect is created. This variable will have a script-name determined by Natron as explained in the *Python Auto-declared variables* section.

Once an Effect is instantiated, it declares all its *Param* and inputs. See how to *manage* user parameters below

To get a specific *Param* by script-name, call the `getParam(name)` function

Input effects are mapped against a zero-based index. To retrieve an input Effect given an index, you can use the `getInput(inputNumber)` function.

To manage inputs, you can connect them and disconnect them with respect to their input index with the `connectInput(inputNumber, input)` and then `disconnectInput(inputNumber)` functions.

If you need to destroy permanently the Effect, just call `destroy()`.

For convenience some GUI functionalities have been made accessible via the Effect class to control the GUI of the node (on the node graph):

- Get/Set the node position with the `setPosition(x, y)` and `getPosition()` functions
- Get/Set the node size with the `setSize(width, height)` and `getSize()` functions
- Get/Set the node color with the `setColor(r, g, b)` and `getColor()` functions

Creating user parameters

See [this section](#)

Member functions description

`NatronEngine.Effect.addUserPlane(planeName, channels)`

Parameters

- **planeName** – str
- **channels** – sequence

Return type bool

Adds a new plane to the Channels selector of the node in its settings panel. When selected, the end-user can choose to output the result of the node to this new custom plane. The *planeName* will identify the plane uniquely and must not contain spaces or non python compliant characters. The *channels* are a sequence of channel names, e.g.:

```
addUserPlane("MyLayer",["R", "G", "B", "A"])
```

Note: A plane cannot contain more than 4 channels and must at least have 1 channel.

This function returns *True* if the layer was added successfully, *False* otherwise.

`NatronEngine.Effect.beginChanges()`

Starts a begin/End bracket, blocking all evaluation (=renders and callback onParamChanged) that would be issued due to a call to `setValue` on any parameter of the Effect.

Similarly all input changes will not be evaluated until `endChanges()` is called.

Typically to change several values at once we bracket the changes like this:

```
node.beginChanges()
param1.setValue(...)
param2.setValue(...)
param3.setValue(...)
param4.setValue(...)
node.endChanges() # This triggers a new render
```

A more complex call:

```
node.beginChanges()    node.connectInput(0,otherNode)    node.connectInput(1,thirdNode)
param1.setValue(...) node.endChanges() # This triggers a new render
```

`NatronEngine.Effect.endChanges()`

Ends a begin/end bracket. If the begin/end bracket recursion reaches 0 and there were calls made to `setValue` this function will effectively compress all evaluations into a single one. See [beginChanges\(\)](#)

`NatronEngine.Effect.canConnectInput(inputNumber, node)`

Parameters

- **inputNumber** – int

- **node** – Effect

Return type bool

Returns whether the given *node* can be connected at the given *inputNumber* of this Effect. This function could return False for one of the following reasons:

- The Effect already has an input at the given *inputNumber*
- The *node* is None
- The given *inputNumber* is out of range
- The *node* cannot have any node connected to it (such as a BackDrop or an Output)
- This Effect or the given *node* is a child of another node (for trackers only)
- Connecting *node* would create a cycle in the graph implying that it would create infinite recursions

`NatronEngine.Effect.connectInput (inputNumber, input)`

Parameters

- **inputNumber** – int
- **input** – Effect

Return type bool

Connects *input* to the given *inputNumber* of this Effect. This function calls internally `canConnectInput ()` to determine if a connection is possible.

`NatronEngine.Effect.destroy ([autoReconnect=true])`

Parameters **autoReconnect** – bool

Removes this Effect from the current project definitively. If *autoReconnect* is True then any nodes connected to this node will try to connect their input to the input of this node instead.

`NatronEngine.Effect.disconnectInput (inputNumber)`

Parameters **inputNumber** – int

Removes any input Effect connected to the given *inputNumber* of this node.

`NatronEngine.Effect.getAvailableLayers ()`

Return type dict

Returns the layer available on this node. This is a dict with a ImageLayer as key and Effect as value. The Effect is the closest node in the upstream tree (including this node) that produced that layer.

For example, in a simple graph Read → Blur, if the Read node has a layer available named “RenderLayer.combined” but Blur is set to process only the color layer (RGBA), then calling this function on the Blur will return a dict containing for key “RenderLayer.combined” the Read node, whereas the dict will have for the key “RGBA” the Blur node.

`NatronEngine.Effect.getBitDepth ()`

Return type ImageBitDepthEnum

Returns the bit-depth of the image in output of this node.

`NatronEngine.Effect.getColor ()`

Return type tuple

Returns the color of this node as it appears on the node graph as [R,G,B] 3-dimensional tuple.

`NatronEngine.Effect.getCurrentTime ()`

Return type int

Returns the current time of timeline if this node is currently rendering, otherwise it returns the current time at which the node is currently rendering for the caller thread.

`NatronEngine.Effect.getOutputFormat()`

Return type `RectI`

Returns the output format of this node in pixel units.

`NatronEngine.Effect.getFrameRate()`

Return type `float`

Returns the frame-rate of the sequence in output of this node.

`NatronEngine.Effect.getInput(inputNumber)`

Parameters `inputNumber` – `int`

Return type `Effect`

Returns the node connected at the given *inputNumber*.

`NatronEngine.Effect.getInput(inputName)`

param `inputName` `str`

rtype `Effect`

Same as `getInput(inputNumber)` except that the parameter in input is the name of the input as displayed on the node-graph. This function is made available for convenience.

`NatronEngine.Effect.getLabel()`

Return type `str`

Returns the *label* of the node. See [this section](#) for a discussion of the *label* vs the *script-name*.

`NatronEngine.Effect.getInputLabel(inputNumber)`

Parameters `inputNumber` – `int`

Return type `str`

Returns the label of the input at the given *inputNumber*. It corresponds to the label displayed on the arrow of the input in the node graph.

`NatronEngine.Effect.getMaxInputCount()`

Return type `int`

Returns the number of inputs for the node. Graphically this corresponds to the number of arrows in input.

`NatronEngine.Effect.getParam(name)`

Parameters `name` – `str`

Return type `Param`

Returns a *parameter* by its script-name or `None` if no such parameter exists.

`NatronEngine.Effect.getParams()`

Return type `sequence`

Returns all the parameters of this Effect as a sequence.

`NatronEngine.Effect.getPluginID()`

Return type `str`

Returns the ID of the plug-in that this node instantiate.

`NatronEngine.Effect.getPosition()`

Return type `tuple`

Returns the current position of the node on the node-graph. This is a 2 dimensional [X,Y] tuple. Note that in background mode, if used, this function will always return [0,0] and should NOT be used.

`NatronEngine.Effect.getPremult()`

Return type `ImagePremultiplicationEnum`

Returns the alpha premultiplication state of the image in output of this node.

`NatronEngine.Effect.getPixelAspectRatio()`

Return type `float`

Returns the pixel aspect ratio of the image in output of this node.

`NatronEngine.Effect.getRegionOfDefinition(time, view)`

Parameters

- **time** – `float`
- **view** – `int`

Return type `RectD`

Returns the bounding box of the image produced by this effect in canonical coordinates. This is exactly the value displayed in the “Info” tab of the settings panel of the node for the “Output”. This can be useful for example to set the position of a point parameter to the center of the region of definition.

`NatronEngine.Effect.getRotoContext()`

Return type `Roto`

Returns the roto context for this node. Currently only the Roto node has a roto context. The roto context is in charge of maintaining all informations relative to *Beziers* and *Layers*. Most of the nodes don’t have a roto context though and this function will return `None`.

`NatronEngine.Effect.getTrackerContext()`

Return type `Tracker`

Returns the tracker context for this node. Currently only the Tracker node has a tracker context. The tracker context is in charge of maintaining all informations relative to *Tracks*. Most of the nodes don’t have a tracker context though and this function will return `None`.

`NatronEngine.Effect.getScriptName()`

Return type `str`

Returns the script-name of this Effect. See [this](#) section for more information about the script-name.

`NatronEngine.Effect.getSize()`

Return type `tuple`

Returns the size of this node on the node-graph as a 2 dimensional [Width,Height] tuple. Note that calling this function will in background mode will always return [0,0] and should not be used.

`NatronEngine.Effect.getUserPageParam()`

Return type `PageParam`

Convenience function to return the user page parameter if this Effect has one.

`NatronEngine.Effect.isUserSelected()`

Return type `bool`

Returns true if this node is selected in its containing nodegraph.

`NatronEngine.Effect.isReaderNode()`

Return type `bool`

Returns True if this node is a reader node

`NatronEngine.Effect.isWriterNode()`

Return type `bool`

Returns True if this node is a writer node

`NatronEngine.Effect.setColor(r, g, b)`

Parameters

- **r** – float
- **g** – float
- **b** – float

Set the color of the node as it appears on the node graph. Note that calling this function will in background mode will do nothing and should not be used.

`NatronEngine.Effect.setLabel(name)`

Parameters **name** – str

Set the label of the node as it appears in the user interface. See [this](#) section for an explanation of the difference between the *label* and the *script-name*.

`NatronEngine.Effect.setPosition(x, y)`

Parameters

- **x** – float
- **y** – float

Set the position of the node as it appears on the node graph. Note that calling this function will in background mode will do nothing and should not be used.

`NatronEngine.Effect.setScriptName(scriptName)`

Parameters **scriptName** – str

Return type `bool`

Set the script-name of the node as used internally by Natron. See [this](#) section for an explanation of the difference between the *label* and the *script-name*.

Warning: Using this function will remove any previous variable declared using the old script-name and will create a new variable with the new script name if valid.

If your script was using for instance a node named:

```
app1.Blur1
```

and you renamed it BlurOne, it should now be available to Python this way:

```
app1.BlurOne
```

but using `app1.Blur1` would report the following error:

```
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
NameError: name 'Blur1' is not defined
```

`NatronEngine.Effect.setSize(w, h)`

Parameters

- **w** – float
- **h** – float

Set the size of the node as it appears on the node graph. Note that calling this function will in background mode will do nothing and should not be used.

`NatronEngine.Effect.setSubGraphEditable` (*editable*)

Parameters `editable` – bool

Can be called to disable editing of the group via Natron’s graphical user interface. This is handy to prevent users from accidentally breaking the sub-graph. This can always be reverted by editing the python script associated. The user will still be able to see the internal node graph but will not be able to unlock it.

`NatronEngine.Effect.setPagesOrder` (*pages*)

Parameters `pages` – sequence

Given the string list *pages* try to find the corresponding pages by their-script name and order them in the given order.

ExprUtils

Inherits *Double2DParam*

Synopsis

Various functions useful for expressions. Most noise functions have been taken from the Walt Disney Animation Studio SeExpr library.

Functions

- `def boxstep` (*x,a*)
- `def linearstep` (*x,a,b*)
- `def smoothstep` (*x,a,b*)
- `def gaussstep` (*x,a,b*)
- `def remap` (*x,source,range,falloff,interp*)
- `def mix` (*x,y,alpha*)
- `def hash` (*args*)
- `def noise` (*x*)
- `def noise` (*p*)
- `def noise` (*p*)
- `def noise` (*p*)
- `def snoise` (*p*)
- `def vnoise` (*p*)
- `def cnoise` (*p*)
- `def snoise4` (*p*)
- `def vnoise4` (*p*)
- `def cnoise4` (*p*)
- `def turbulence` (*p[,ocaves=6, lacunarity=2, gain=0.5]*)
- `def vturbulence` (*p[,ocaves=6, lacunarity=2, gain=0.5]*)
- `def cturbulence` (*p[,ocaves=6, lacunarity=2, gain=0.5]*)

- `def fbm (p[,ocaves=6, lacunarity=2, gain=0.5])`
- `def vfbm (p[,ocaves=6, lacunarity=2, gain=0.5])`
- `def fbm4 (p[,ocaves=6, lacunarity=2, gain=0.5])`
- `def vfbm4 (p[,ocaves=6, lacunarity=2, gain=0.5])`
- `def cfbm (p[,ocaves=6, lacunarity=2, gain=0.5])`
- `def cfbm4 (p[,ocaves=6, lacunarity=2, gain=0.5])`
- `def cellnoise (p)`
- `def ccellnoise (p)`
- `def pnoise (p, period)`

Member functions description

`NatronEngine.ExprUtils.bboxstep (x, a)`

Parameters

- **x** – float
- **a** – float

Return type

float

if $x < a$ then 0 otherwise 1

`NatronEngine.ExprUtils.linearstep (x, a, b)`

Parameters

- **x** – float
- **a** – float
- **b** – float

Return type

float

Transitions linearly when $a < x < b$

`NatronEngine.ExprUtils.bboxstep (x, a, b)`

Parameters

- **x** – float
- **a** – float
- **b** – float

Return type

float

Transitions smoothly (cubic) when $a < x < b$

`NatronEngine.ExprUtils.gaussstep (x, a, b)`

Parameters

- **x** – float
- **a** – float
- **b** – float

Return type

float

Transitions smoothly (exponentially) when $a < x < b$

`NatronEngine.ExprUtils.remap(x, source, range, falloff, interp)`

Parameters

- **x** – float
- **source** – float
- **range** – float
- **falloff** – float
- **interp** – float

Return type float

General remapping function. When **x** is within +/- **range** of **source**, the result is 1. The result falls to 0 beyond that range over **falloff** distance. The falloff shape is controlled by **interp**: linear = 0 smooth = 1 gaussian = 2

`NatronEngine.ExprUtils.mix(x, y, alpha)`

Parameters

- **x** – float
- **y** – float
- **alpha** – float

Return type float

Linear interpolation of a and b according to alpha

`NatronEngine.ExprUtils.hash(args)`

Parameters **args** – Sequence

Return type float

Like random, but with no internal seeds. Any number of seeds may be given and the result will be a random function based on all the seeds.

`NatronEngine.ExprUtils.noise(x)`

Parameters **x** – float

Return type float

Original perlin noise at location (C2 interpolant)

`NatronEngine.ExprUtils.noise(p)`

Parameters **p** – *Double2DTuple*

Return type float

Original perlin noise at location (C2 interpolant)

`NatronEngine.ExprUtils.noise(p)`

Parameters **p** – *Double3DTuple*

Return type float

Original perlin noise at location (C2 interpolant)

`NatronEngine.ExprUtils.noise(p)`

Parameters **p** – *ColorTuple*

Return type float

Original perlin noise at location (C2 interpolant)

NatronEngine.ExprUtils.**snoise** (*p*)

Parameters *p* – *Double3DTuple*

Return type float

Signed noise w/ range -1 to 1 formed with original perlin noise at location (C2 interpolant)

NatronEngine.ExprUtils.**vnoise** (*p*)

Parameters *p* – *Double3DTuple*

Return type *Double3DTuple*

Vector noise formed with original perlin noise at location (C2 interpolant)

NatronEngine.ExprUtils.**cnoise** (*p*)

Parameters *p* – *Double3DTuple*

Return type *Double3DTuple*

Color noise formed with original perlin noise at location (C2 interpolant)

NatronEngine.ExprUtils.**snoise4** (*p*)

Parameters *p* – *ColorTuple*

Return type float

4D signed noise w/ range -1 to 1 formed with original perlin noise at location (C2 interpolant)

NatronEngine.ExprUtils.**vnoise4** (*p*)

Parameters *p* – *ColorTuple*

Return type *Double3DTuple*

4D vector noise formed with original perlin noise at location (C2 interpolant)

NatronEngine.ExprUtils.**cnoise4** (*p*)

Parameters *p* – *ColorTuple*

Return type *Double3DTuple*

4D color noise formed with original perlin noise at location (C2 interpolant)”

NatronEngine.ExprUtils.**turbulence** (*p*[, *ocaves*=6, *lacunarity*=2, *gain*=0.5])

Parameters

- *p* – *Double3DTuple*
- **octaves** – int
- **lacunarity** – float
- **gain** – float

Return type

float

FBM (Fractal Brownian Motion) is a multi-frequency noise function. The base frequency is the same as the noise function. The total number of frequencies is controlled by **octaves**. The **lacunarity** is the spacing between the frequencies - A value of 2 means each octave is twice the previous frequency. The **gain** controls how much each frequency is scaled relative to the previous frequency.

NatronEngine.ExprUtils.**vturbulence** (*p*[, *ocaves*=6, *lacunarity*=2, *gain*=0.5])

Parameters

- *p* – *Double3DTuple*

- **octaves** – int
- **lacunarity** – float
- **gain** – float

Return type

Double3DTuple

FBM (Fractal Brownian Motion) is a multi-frequency noise function. The base frequency is the same as the noise function. The total number of frequencies is controlled by **octaves**. The **lacunarity** is the spacing between the frequencies - A value of 2 means each octave is twice the previous frequency. The **gain** controls how much each frequency is scaled relative to the previous frequency.

`NatronEngine.ExprUtils.cturbulence(p[, ocaves=6, lacunarity=2, gain=0.5])`

Parameters

- **p** – *Double3DTuple*
- **octaves** – int
- **lacunarity** – float
- **gain** – float

Return type

Double3DTuple

FBM (Fractal Brownian Motion) is a multi-frequency noise function. The base frequency is the same as the noise function. The total number of frequencies is controlled by **octaves**. The **lacunarity** is the spacing between the frequencies - A value of 2 means each octave is twice the previous frequency. The **gain** controls how much each frequency is scaled relative to the previous frequency.

`NatronEngine.ExprUtils.fbm(p[, ocaves=6, lacunarity=2, gain=0.5])`

Parameters

- **p** – *Double3DTuple*
- **octaves** – int
- **lacunarity** – float
- **gain** – float

Return type

float

FBM (Fractal Brownian Motion) is a multi-frequency noise function. The base frequency is the same as the noise function. The total number of frequencies is controlled by **octaves**. The **lacunarity** is the spacing between the frequencies - A value of 2 means each octave is twice the previous frequency. The **gain** controls how much each frequency is scaled relative to the previous frequency.

`NatronEngine.ExprUtils.vfbm(p[, ocaves=6, lacunarity=2, gain=0.5])`

Parameters

- **p** – *Double3DTuple*
- **octaves** – int
- **lacunarity** – float
- **gain** – float

Return type

Double3DTuple

FBM (Fractal Brownian Motion) is a multi-frequency noise function. The base frequency is the same as the noise function. The total number of frequencies is controlled by **octaves**. The **lacunarity** is the spacing between the frequencies - A value of 2 means each octave is twice the previous frequency. The **gain** controls how much each frequency is scaled relative to the previous frequency.

`NatronEngine.ExprUtils.fbm4(p[, ocaves=6, lacunarity=2, gain=0.5])`

Parameters

- **p** – *Double3DTuple*
- **octaves** – int
- **lacunarity** – float
- **gain** – float

Return type

float

FBM (Fractal Brownian Motion) is a multi-frequency noise function. The base frequency is the same as the noise function. The total number of frequencies is controlled by **octaves**. The **lacunarity** is the spacing between the frequencies - A value of 2 means each octave is twice the previous frequency. The **gain** controls how much each frequency is scaled relative to the previous frequency.

`NatronEngine.ExprUtils.vfbm4(p[, ocaves=6, lacunarity=2, gain=0.5])`

Parameters

- **p** – *Double3DTuple*
- **octaves** – int
- **lacunarity** – float
- **gain** – float

Return type

Double3DTuple

FBM (Fractal Brownian Motion) is a multi-frequency noise function. The base frequency is the same as the noise function. The total number of frequencies is controlled by **octaves**. The **lacunarity** is the spacing between the frequencies - A value of 2 means each octave is twice the previous frequency. The **gain** controls how much each frequency is scaled relative to the previous frequency.

`NatronEngine.ExprUtils.cfbm(p[, ocaves=6, lacunarity=2, gain=0.5])`

Parameters

- **p** – *Double3DTuple*
- **octaves** – int
- **lacunarity** – float
- **gain** – float

Return type

Double3DTuple

FBM (Fractal Brownian Motion) is a multi-frequency noise function. The base frequency is the same as the noise function. The total number of frequencies is controlled by **octaves**. The **lacunarity** is the spacing between the frequencies - A value of 2 means each octave is

twice the previous frequency. The **gain** controls how much each frequency is scaled relative to the previous frequency.

`NatronEngine.ExprUtils.cfbm4 (p[, ocaves=6, lacunarity=2, gain=0.5])`

Parameters

- **p** – *Double3DTuple*
- **octaves** – int
- **lacunarity** – float
- **gain** – float

Return type

Double3DTuple

FBM (Fractal Brownian Motion) is a multi-frequency noise function. The base frequency is the same as the noise function. The total number of frequencies is controlled by **octaves**. The **lacunarity** is the spacing between the frequencies - A value of 2 means each octave is twice the previous frequency. The **gain** controls how much each frequency is scaled relative to the previous frequency.

`NatronEngine.ExprUtils.cellnoise (p)`

Parameters **p** – *Double3DTuple*

Return type

float

cellnoise generates a field of constant colored cubes based on the integer location This is the same as the prman cellnoise function

`NatronEngine.ExprUtils.ccellnoise (p)`

Parameters **p** – *Double3DTuple*

Return type *Double3DTuple*

cellnoise generates a field of constant colored cubes based on the integer location This is the same as the prman cellnoise function

`NatronEngine.ExprUtils.pnoise (p, period)`

Parameters

- **p** – *Double3DTuple*
- **period** – *Double3DTuple*

Return type float

Periodic noise

FileParam

Inherits *StringParamBase*

Synopsis

This parameter is used to specify an input file (i.e: a file that already exist).

Functions

- `def openFile()`
- `def reloadFile()`
- `def setSequenceEnabled(enabled)`

Member functions description

`NatronEngine.FileParam.openFile()`

When called in GUI mode, this will open a file dialog for the user. Does nothing in background mode.

`NatronEngine.FileParam.reloadFile()`

Force a refresh of the data read from the file. Any cached data associated to the file will be discarded.

`NatronEngine.FileParam.setSequenceEnabled(enabled)`

Parameters `enabled` – bool

Determines whether the file dialog opened by `openFile()` should have support for file sequences or not.

Group

Inherited by: *Effect*, *App*, *GuiApp*

Synopsis

Base class for *Effect* and *App*. See *detailed* description below.

Functions

- `def getChildren()`
- `def getNode(fullyQualifiedName)`

Detailed Description

This is an abstract class, it is derived by 2 different classes:

- *App* which represents an instance of Natron, or more specifically the current project.
- *Effect* which represents a node in the node graph.

The `getNode(fullyQualifiedName)` can be used to retrieve a node in the project, although all nodes already have an *auto-declared* variable by Natron.

Member functions description

`NatronEngine.Group.getChildren()`

Return type `sequence`

Returns a sequence with all nodes in the group. Note that this function is not recursive and you'd have to call `getChildren()` on all sub-groups to retrieve their children, etc...

`NatronEngine.Group.getNode(fullyQualifiedName)`

Parameters `fullySpecifiedName` – str

Return type `Effect`

Retrieves a node in the group with its *fully qualified name*. The fully qualified name of a node is the *script-name* of the node prefixed by all the group hierarchy into which it is, e.g.:

Blur1 # the node is a top level node

Group1.Group2.Blur1 # the node is inside Group2 which is inside Group1

Basically you should never call this function because Natron already pre-declares a variable for each node upon its creation. If you were to create a new node named “Blur1”, you could the access it in the Script Editor the following way:

```
app1.Blur1
```

GroupParam

Inherits *Param*

Synopsis

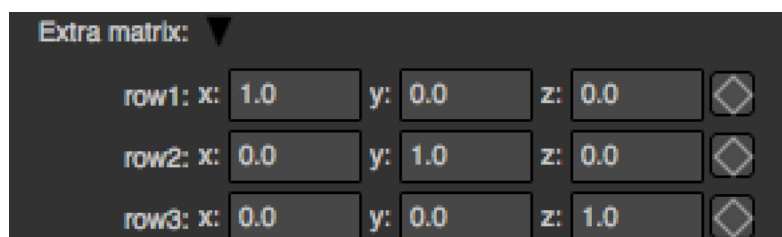
A group param is a container for other parameters. See *detailed* description.

Functions

- def *addParam* (param)
- def *getIsOpened* ()
- def *setAsTab* ()
- def *setOpened* (opened)

Detailed Description

A group param does not hold any relevant value. Rather this is a purely graphical element that is used to gather multiple parameters under a group. On the graphical interface a GroupParam looks like this:



When a *Param* is under a group, the *getParent* () will return the group as parent.

Member functions description

`NatronEngine.GroupParam.addParam (param)`

Parameters `param` – Param

Adds *param* into the group.

Warning: Note that this function cannot be called on groups that are not user parameters (i.e: created either by script or by the “Manage user parameters” user interface)

Warning: Once called, you should call `refreshUserParamsGUI()` to update the user interface.

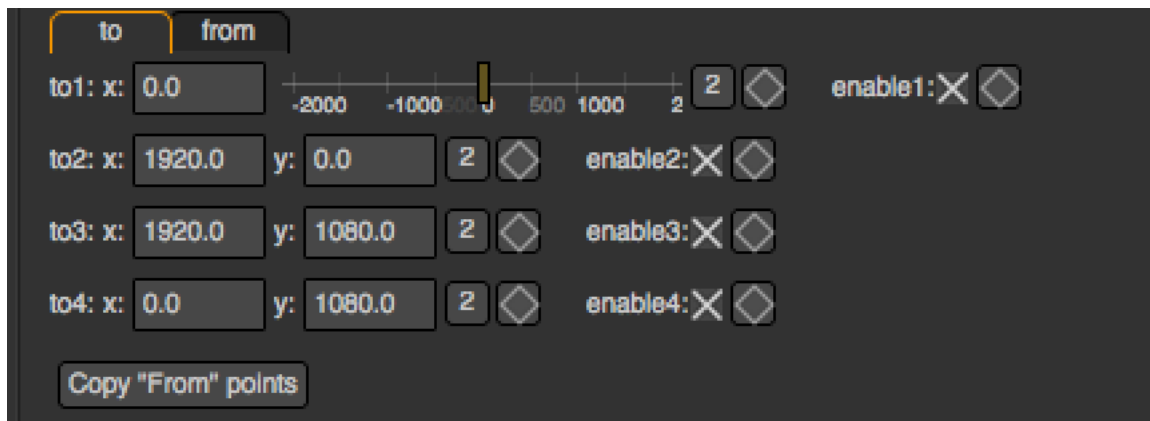
`NatronEngine.GroupParam.getIsOpened()`

Return type `bool`

Returns whether the group is currently expanded (True) or folded (False).

`NatronEngine.GroupParam.setAsTab()`

Set this group as a tab. When set as a tab, it will be inserted into a special `TabWidget` of the Effect. For instance, on the following screenshot, *to* and *from* are 2 groups on which `setAsTab()` has been called.



`NatronEngine.GroupParam.setOpened(opened)`

Parameters `opened` – `bool`

Set this group to be expanded (*opened* = True) or folded (*opened* = False)

ImageLayer

Synopsis

A small object representing a layer of an image. For example, the base image layer is the color layer, or sometimes called “RGBA”. Some other default layers include `ForwardMotion`, `BackwardMotion`, `DisparityLeft`, `DisparityRight`, etc. . .

See [detailed](#) description. . .

Functions

- `def ImageLayer(layerName,componentsPrettyName,componentsName)`
- `def isColorPlane()`
- `def getNumComponents()`
- `def getLayerName()`

- `def getComponentsNames ()`
- `def getComponentsPrettyName ()`
- `def getNoneComponents ()`
- `def getRGBAComponents ()`
- `def getRGBComponents ()`
- `def getAlphaComponents ()`
- `def getBackwardMotionComponents ()`
- `def getForwardMotionComponents ()`
- `def getDisparityLeftComponents ()`
- `def getDisparityRightComponents ()`

Detailed Description

A Layer is constituted of a layer *name* and a set of channel names (also called components). You can get a sequence with all the channels in the layer with the function `getComponentsNames ()`. For some default layers, the components may be represented by a prettier name for the end-user, such as *DisparityLeft* instead of XY. When the ImageLayer does not have a pretty name, its pretty name will just be a concatenation of all channel names in order.

There is one special layer in Natron: the color layer. It be represented as 3 different types: RGBA, RGB or Alpha. If the ImageLayer is a color layer, the method `isColorPlane ()` will return True

Member functions description

`NatronEngine.ImageLayer.ImageLayer (layerName, componentsPrettyName, componentsName)`

Parameters `layerName` – str

Make a new image layer with the given layer name, optional components pretty name and the set of channels (also called components) in the layer.

`NatronEngine.ImageLayer.isColorPlane ()`

Return type bool

Returns True if this layer is a color layer, i.e: it is RGBA, RGB or alpha. The color layer is what is output by default by all nodes in Natron.

`NatronEngine.ImageLayer.getNumComponents ()`

Return type int

Returns the number of channels in this layer. Can be between 0 and 4 included.

`NatronEngine.ImageLayer.getLayerName ()`

Return type str

Returns the layer name

`NatronEngine.ImageLayer.getComponentsNames ()`

Return type Sequence

Returns a sequence with all channels in this layer in order

`NatronEngine.ImageLayer.getComponentsPrettyName ()`

Return type str

Returns the channels pretty name. E.g: DisparityLeft instead of XY

`NatronEngine.ImageLayer.getNoneComponents()`

Return type `ImageLayer`

Returns the default “none” layer

`NatronEngine.ImageLayer.getRGBAComponents()`

Return type `ImageLayer`

Returns the default “RGBA” layer

`NatronEngine.ImageLayer.getRGBComponents()`

Return type `ImageLayer`

Returns the default “RGB” layer

`NatronEngine.ImageLayer.getAlphaComponents()`

Return type `ImageLayer`

Returns the default “Alpha” layer

`NatronEngine.ImageLayer.getBackwardMotionComponents()`

Return type `ImageLayer`

Returns the default “Backward” layer

`NatronEngine.ImageLayer.getForwardMotionComponents()`

Return type `ImageLayer`

Returns the default “Forward” layer

`NatronEngine.ImageLayer.getDisparityLeftComponents()`

Return type `ImageLayer`

Returns the default “DisparityLeft” layer

`NatronEngine.ImageLayer.getDisparityRightComponents()`

Return type `ImageLayer`

Returns the default “DisparityRight” layer

Int2DParam

Inherits *IntParam*

Inherited by: *Int3DParam*

Synopsis

See *IntParam* for more details.

Functions

- def *get* ()
- def *get* (frame)
- def *set* (x, y)
- def *set* (x, y, frame)

Detailed Description

`NatronEngine.Int2DParam.get()`

Return type

class *Int2DTuple*

Returns a *Int2DTuple* containing the [x,y] value of this parameter at the timeline's current time.

`NatronEngine.Int2DParam.get(frame)`

Param float

Return type

class *Int2DTuple*

Returns a *Int2DTuple* containing the [x,y] value of this parameter at the given *frame*.

`NatronEngine.Int2DParam.set(x, y)`

Parameters

- **x** – int
- **y** – int

Same as `set(x)` but for 2-dimensional integers.

`NatronEngine.Int2DParam.set(x, y, frame)`

Parameters

- **x** – int
- **y** – int
- **frame** – float

Same as `set(x, frame)` but for 2-dimensional integers.

Int2DTuple

Synopsis

Utility class to return pair of integers values.

Functions

- `def __getitem__(index)`

Detailed Description

class `NatronEngine.Int2DTuple`

`NatronEngine.Int2DTuple.x`

`NatronEngine.Int2DTuple.y`

`NatronEngine.Int2DTuple.__getitem__(index)`

Parameters **index** – int

Return type PyObject

Returns the item at the given index. This is the bracket operator `[]`

Int3DParam

Inherits *Int2DParam*

Synopsis

See *IntParam* for more details.

Functions

- `def set ()`
- `def set (frame)`
- `def set (x, y, z)`
- `def set (x, y, z, frame)`

Detailed Description

`NatronEngine.Int3DParam.get ()`

Return type `<Int3DTuple>`

Returns a *Int3DTuple* containing the [x,y,z] value of this parameter at the timeline's current time.

`NatronEngine.Int3DParam.get (frame)`

Parameters `frame` – float

Return type `<Int3DTuple>`

Returns a *Int3DTuple* containing the [x,y,z] value of this parameter at the given *frame*

`NatronEngine.Int3DParam.set (x, y, z)`

Parameters

- `x` – int
- `y` – int
- `z` – int

Same as `set (x)` but for 3-dimensional integers.

`NatronEngine.Int3DParam.set (x, y, z, frame)`

Parameters

- `x` – int
- `y` – int
- `z` – int
- `frame` – float

Same as `set (x, frame)` but for 3-dimensional integers.

Int3DTuple

Synopsis

Utility class to return pair of integers values.

Functions

- `def __getitem__ (index)`

Detailed Description

class NatronEngine.Int3DTuple

NatronEngine.Int3DTuple.**x**

NatronEngine.Int3DTuple.**y**

NatronEngine.Int3DTuple.**z**

NatronEngine.Int3DTuple.**__getitem__** (*index*)

param *index* PySide.QtCore.int

rtype PyObject

Returns the item at the given index. This is the bracket operator `[]`

IntParam

Inherits AnimatedParam

Inherited by: *Int2DParam, Int3DParam*

Synopsis

An IntParam can contain one or multiple int values. See *detailed* description...

Functions

- `def get ()`
- `def get (frame)`
- `def getDefaultValues ([dimension=0])`
- `def getDisplayMaximum (dimension)`
- `def getDisplayMinimum (dimension)`
- `def getMaximum ([dimension=0])`
- `def getMinimum ([dimension=0])`
- `def getValue ([dimension=0])`
- `def getValueAtTime (time[, dimension=0])`
- `def restoreDefaultValues ([dimension=0])`
- `def set (x)`
- `def set (x, frame)`
- `def setDefaultValues (value[, dimension=0])`
- `def setDisplayMaximum (maximum[, dimension=0])`
- `def setDisplayMinimum (minimum[, dimension=0])`
- `def setMaximum (maximum[, dimension=0])`

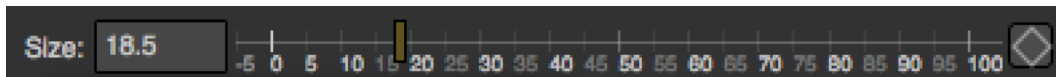
- `def setMinimum (minimum[, dimension=0])`
- `def setValue (value[, dimension=0])`
- `def setValueAtTime (value, time[, dimension=0])`

Detailed Description

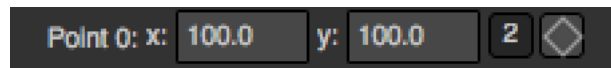
An int param can have 1 to 3 dimensions. (See [Int2DParam](#) and [Int3DParam](#)). Usually this is used to represent a single integer value that may animate over time.

The user interface for them varies depending on the number of dimensions. *Screenshots are the same than for the :doc:'DoubleParam' because the user interface is the same*

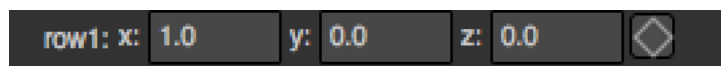
A 1-dimensional [IntParam](#)



A 2-dimensional [Int2DParam](#)



A 3-dimensional [Int3DParam](#)



Member functions description

`NatronEngine.IntParam.get (frame)`

Parameters `frame` – float

Return type `int`

Returns the value of this parameter at the given *frame*. If the animation curve has an animation (see `getIsAnimated`) then the value will be interpolated using the *interpolation* chosen by the user for the curve.

`NatronEngine.IntParam.get ()`

Return type `int`

Returns the value of this parameter at the given current timeline's time.

`NatronEngine.IntParam.getDefaultValue ([dimension=0])`

Parameters `dimension` – int

Return type `int`

Returns the default value for this parameter. *dimension* is meaningless for the `IntParam` class because it is 1-dimensional, but is useful for inherited classes [Int2DParam](#) and [Int3DParam](#)

`NatronEngine.IntParam.getDisplayMaximum (dimension)`

Parameters `dimension` – int

Return type `int`

Returns the display maximum for this parameter at the given *dimension*. The display maximum is the maximum value visible on the slider, internally the value can exceed this range.

`NatronEngine.IntParam.getDisplayMinimum (dimension)`

Parameters `dimension` – int

Return type int

Returns the display minimum for this parameter at the given *dimension*. The display minimum is the minimum value visible on the slider, internally the value can exceed this range.

`NatronEngine.IntParam.setMaximum([dimension=0])`

Parameters `dimension` – int

Return type int

Returns the maximum for this parameter at the given *dimension*. The maximum value cannot be exceeded and any higher value will be clamped to this value.

`NatronEngine.IntParam.getMinimum([dimension=0])`

Parameters `dimension` – int

Return type int

Returns the minimum for this parameter at the given *dimension*. The minimum value cannot be exceeded and any lower value will be clamped to this value.

`NatronEngine.IntParam.getValue([dimension=0])`

Parameters `dimension` – int

Return type int

Returns the value of this parameter at the given *dimension* at the current timeline's time.

`NatronEngine.IntParam.getValueAtTime(time[, dimension=0])`

Parameters

- `time` – float
- `dimension` – int

Return type int

Returns the value of this parameter at the given *dimension* at the given *time*.

If the animation curve has an animation (see `getIsAnimated`) then the value will be interpolated using the *interpolation* chosen by the user for the curve.

`NatronEngine.IntParam.restoreDefaultValue([dimension=0])`

Parameters `dimension` – int

Returns the value of this parameter at the given *dimension* at the given *time*.

`NatronEngine.IntParam.set(x, frame)`

Parameters

- `x` – int
- `frame` – float

Set a new keyframe on the parameter with the value *x* at the given *frame*.

`NatronEngine.IntParam.set(x)`

Parameters `x` – int

Set the value of this parameter to be *x*. If this parameter is animated (see `getIsAnimated(dimension)`) then this function will automatically add a keyframe at the timeline's current time.

`NatronEngine.IntParam.setDefaultValue(value[, dimension=0])`

Parameters

- `value` – int

- **dimension** – int

Set the default *value* for this parameter at the given *dimension*.

`NatronEngine.IntParam.setDisplayMaximum(maximum[, dimension=0])`

Parameters

- **maximum** – int
- **dimension** – int

Set the display maximum of the parameter to be *maximum* for the given *dimension*. See [getDisplayMaximum](#)

`NatronEngine.IntParam.setDisplayMinimum(minimum[, dimension=0])`

Parameters

- **minimum** – int
- **dimension** – int

Set the display minimum of the parameter to be *minnum* for the given *dimension*. See [getDisplayMinimum](#)

`NatronEngine.IntParam.setMaximum(maximum[, dimension=0])`

Parameters

- **maximum** – int
- **dimension** – int

Set the maximum of the parameter to be *maximum* for the given *dimension*. See [getMaximum](#)

`NatronEngine.IntParam.setMinimum(minimum[, dimension=0])`

Parameters

- **minimum** – int
- **dimension** – int

Set the minimum of the parameter to be *minimum* for the given *dimension*. See [getMinimum](#)

`NatronEngine.IntParam.setValue(value[, dimension=0])`

Parameters

- **value** – int
- **dimension** – int

Same as [set\(value, dimension\)](#)

`NatronEngine.IntParam.setValueAtTime(value, time[, dimension=0])`

Parameters

- **value** – int
- **time** – float
- **dimension** – int

Same as [set\(value, time, dimension\)](#)

ItemBase

Inherited by: [BezierCurve](#), [Layer](#)

Synopsis

This is an abstract class that serves as a base class for both *Layer* and *BezierCurve*. See *detailed* description...

Functions

- `def getLabel ()`
- `def getLocked ()`
- `def getLockedRecursive ()`
- `def getParentLayer ()`
- `def getParam (name)`
- `def getScriptName ()`
- `def getVisible ()`
- `def setLabel (name)`
- `def setLocked (locked)`
- `def setScriptName (name)`
- `def setVisible (activated)`

Detailed Description

This class gathers all common functions to both *layers* and *beziers*. An item has both a *script-name* and *label*. The *script-name* uniquely identifies an item within a roto node, while several items can have the same *label*.

Member functions description

`NatronEngine.ItemBase.getLabel ()`

Return type `str`

Returns the label of the item, has visible in the table of the settings panel.

`NatronEngine.ItemBase.getLocked ()`

Return type `bool`

Returns whether this item is locked or not. When locked the item is no longer editable by the user.

`NatronEngine.ItemBase.getLockedRecursive ()`

Return type `bool`

Returns whether this item is locked or not. Unlike `getLocked ()` this function looks parent layers recursively to find out if the item should be locked.

`NatronEngine.ItemBase.getParentLayer ()`

Return type `Layer`

Returns the parent *layer* of the item. All items must have a parent layer, except the base layer.

`NatronEngine.ItemBase.getParam (name)`

Parameters `name` – `str`

Return type `Param`

Returns a *parameter* by its script-name or None if no such parameter exists.

`NatronEngine.ItemBase.getScriptName()`

Return type `str`

Returns the *script-name* of the item. The script-name is unique for each items in a roto node.

`NatronEngine.ItemBase.getVisible()`

Return type `bool`

Returns whether the item is visible or not. On the user interface, this corresponds to the small *eye*. When hidden, an item will no longer have its overlay painted on the viewer, but it will still render in the image.

`NatronEngine.ItemBase.setLabel(name)`

Parameters `name` – `str`

Set the item's label.

`NatronEngine.ItemBase.setLocked(locked)`

Parameters `locked` – `bool`

Set whether the item should be locked or not. See `getLocked()`.

`NatronEngine.ItemBase.setScriptName(name)`

Parameters `name` – `str`

Return type `bool`

Set the script-name of the item. You should never call it yourself as Natron chooses automatically a unique script-name for each item. However this function is made available for internal technicalities, but be aware that changing the script-name of an item can potentially break other scripts relying on it.

`NatronEngine.ItemBase.setVisible(activated)`

Parameters `activated` – `bool`

Set whether the item should be visible in the Viewer. See `getVisible()`.

Layer

Inherits *ItemBase*

Synopsis

This class is used to group several shapes together and to organize them so they are rendered in a specific order. See *detailed* description...

Functions

- def `addItem(item)`
- def `getChildren()`
- def `insertItem(pos, item)`
- def `removeItem(item)`

Detailed Description

Currently a layer acts only as a group so that you can organize shapes and control in which order they are rendered. To add a new *item* to the layer, use the `addItem(item)` or the `insertItem(item)` function.

To remove an item from the layer, use the `removeItem(item)` function.

Items in a layer are rendered from top to bottom, meaning the bottom-most items will always be drawn on top of other items.

Member functions description

`NatronEngine.Layer.addItem(item)`

Parameters `item` – `ItemBase`

Adds a new item at the bottom of the layer.

`NatronEngine.Layer.getChildren()`

Return type `sequence`

Returns a sequence with all *items* in the layer.

`NatronEngine.Layer.insertItem(pos, item)`

Parameters

- `pos` – `int`
- `item` – `ItemBase`

Inserts a new item at the given *pos* (0 based index) in the layer. If *pos* is out of range, it will be inserted at the bottom of the layer.

`NatronEngine.Layer.removeItem(item)`

Parameters `item` – `ItemBase`

Removes the *item* from the layer.

Natron

Detailed Description

This class contains enumerations that are used by some functions of the API to return status that are more complicated than a simple boolean value.

`NatronEngine.Natron.StandardButtonEnum`

Can have the following values:

- `eStandardButtonNoButton = 0x00000000,`
- `eStandardButtonEscape = 0x00000200, // obsolete`
- `eStandardButtonOk = 0x00000400,`
- `eStandardButtonSave = 0x00000800,`
- `eStandardButtonSaveAll = 0x00001000,`
- `eStandardButtonOpen = 0x00002000,`
- `eStandardButtonYes = 0x00004000,`
- `eStandardButtonYesToAll = 0x00008000,`
- `eStandardButtonNo = 0x00010000,`

- `eStandardButtonNoToAll = 0x00020000`,
- `eStandardButtonAbort = 0x00040000`,
- `eStandardButtonRetry = 0x00080000`,
- `eStandardButtonIgnore = 0x00100000`,
- `eStandardButtonClose = 0x00200000`,
- `eStandardButtonCancel = 0x00400000`,
- `eStandardButtonDiscard = 0x00800000`,
- `eStandardButtonHelp = 0x01000000`,
- `eStandardButtonApply = 0x02000000`,
- `eStandardButtonReset = 0x04000000`,
- `eStandardButtonRestoreDefaults = 0x08000000`

`NatronEngine.Natron.ImagePlaneDescEnum`

Can have the following values:

- `eImageComponentNone = 0`,
- `eImageComponentAlpha`,
- `eImageComponentRGB`,
- `eImageComponentRGBA`

`NatronEngine.Natron.ImageBitDepthEnum`

Can have the following values:

- `eImageBitDepthNone = 0`,
- `eImageBitDepthByte`,
- `eImageBitDepthShort`,
- `eImageBitDepthFloat`

`NatronEngine.Natron.KeyframeTypeEnum`

Can have the following values:

- `eKeyframeTypeConstant = 0`,
- `eKeyframeTypeLinear = 1`,
- `eKeyframeTypeSmooth = 2`,
- `eKeyframeTypeCatmullRom = 3`,
- `eKeyframeTypeCubic = 4`,
- `eKeyframeTypeHorizontal = 5`,
- `eKeyframeTypeFree = 6`,
- `eKeyframeTypeBroken = 7`,
- `eKeyframeTypeNone = 8`

`NatronEngine.Natron.ValueChangedReasonEnum`

Can have the following values:

- `eValueChangedReasonUserEdited = 0`, A user change to the param triggered the call, gui will not be refreshed but `onParamChanged` will be called
- `eValueChangedReasonPluginEdited`, A plugin change triggered the call, gui will be refreshed but `onParamChanged` not called

- `eValueChangedReasonNatronGuiEdited`, Natron gui called `setValue` itself, `onParamChanged` will be called (with a reason of User edited) AND param gui refreshed
- `eValueChangedReasonNatronInternalEdited`, Natron engine called `setValue` itself, `onParamChanged` will be called (with a reason of plugin edited) AND param gui refreshed
- `eValueChangedReasonTimeChanged`, A time-line seek changed the call, called when timeline time changes
- `eValueChangedReasonSlaveRefresh`, A master parameter ordered the slave to refresh its value
- `eValueChangedReasonRestoreDefault`, The param value has been restored to its defaults

NatronEngine.Natron.AnimationLevelEnum

Can have the following values:

- `eAnimationLevelNone` = 0,
- `eAnimationLevelInterpolatedValue` = 1,
- `eAnimationLevelOnKeyframe` = 2

NatronEngine.Natron.OrientationEnum

Can have the following values:

- `eOrientationHorizontal` = 0x1,
- `eOrientationVertical` = 0x2

NatronEngine.Natron.ImagePremultiplicationEnum

Can have the following values:

- `eImagePremultiplicationOpaque` = 0,
- `eImagePremultiplicationPremultiplied`,
- `eImagePremultiplicationUnPremultiplied`,

NatronEngine.Natron.StatusEnum

Can have the following values:

- `eStatusOK` = 0,
- `eStatusFailed` = 1,
- `eStatusReplyDefault` = 14

NatronEngine.Natron.ViewerCompositingOperatorEnum

Can have the following values:

- `eViewerCompositingOperatorNone`,
- `eViewerCompositingOperatorOver`,
- `eViewerCompositingOperatorMinus`,
- `eViewerCompositingOperatorUnder`,
- `eViewerCompositingOperatorWipe`

NatronEngine.Natron.PlaybackModeEnum

Can have the following values:

- `ePlaybackModeLoop` = 0,
- `ePlaybackModeBounce`,
- `ePlaybackModeOnce`

NatronEngine.Natron.PixmapEnum

See [here](#) for potential values of this enumeration.

NatronEngine.Natron.ViewerColorSpaceEnum

Can have the following values:

- `eViewerColorSpaceSRGB = 0`,
- `eViewerColorSpaceLinear`,
- `eViewerColorSpaceRec709`

OutputFileParam

Inherits *StringParamBase*

Synopsis

This parameter is used to specify an output file

Functions

- `def openFile()`
- `def setSequenceEnabled(enabled)`

Member functions description

`NatronEngine.OutputFileParam.openFile()`

When called in GUI mode, this will open a file dialog for the user. Does nothing in background mode.

`NatronEngine.OutputFileParam.setSequenceEnabled(enabled)`

param enabled bool

Determines whether the file dialog opened by `openFile()` should have support for file sequences or not.

PageParam

Inherits *Param*

Synopsis

A page param is a container for other parameters. See *detailed* description.

Functions

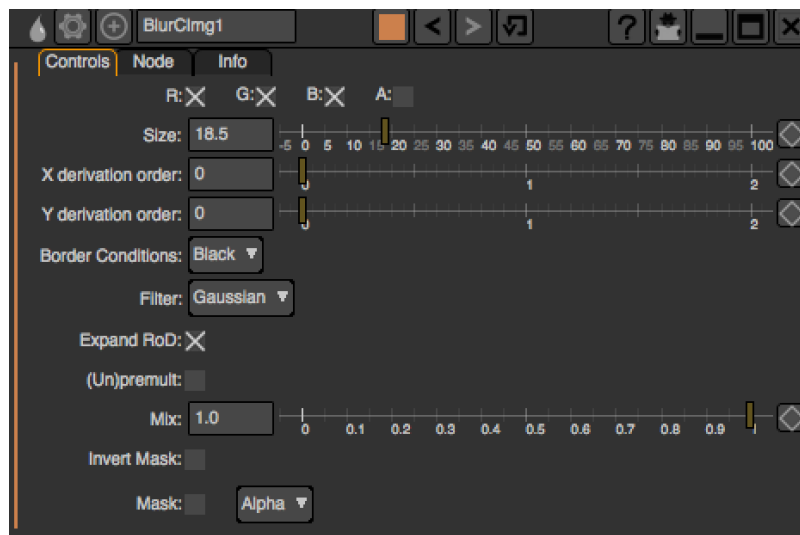
- `def addParam(param)`

Detailed Description

A page param does not hold any relevant value. Rather this is a purely graphical element that is used to gather parameters under a tab. On the graphical interface a PageParam looks like this (e.g. the *Controls* tab of the panel)

Warning: All parameters **MUST** be in a container, being a *group* or a *page*. If a *Param* is not added into any container, Natron will add it by default to the *User* page.

`NatronEngine.PageParam.addParam(param)`



param param Param

Adds *param* into the page.

Warning: Note that this function cannot be called on pages that are not user parameters (i.e: created either by script or by the “Manage user parameters” user interface)

Warning: Once called, you should call `refreshUserParamsGUI()` to update the user interface.

Param

Inherited by: *ParametricParam*, *PageParam*, *GroupParam*, *ButtonParam*, *AnimatedParam*, *StringParamBase*, *PathParam*, *OutputFileParam*, *FileParam*, *StringParam*, *BooleanParam*, *ChoiceParam*, *ColorParam*, *DoubleParam*, *Double2DParam*, *Double3DParam*, *IntParam*, *Int2DParam*, *Int3DParam*

Synopsis

This is the base class for all parameters. Parameters are the controls found in the settings panel of a node. See [details here](#).

Functions

- `def copy (param[, dimension=-1])`
- `def curve (time[, dimension=-1])`
- `def getAddNewLine ()`
- `def getCanAnimate ()`
- `def getEvaluateOnChange ()`
- `def getHelp ()`
- `def getIsAnimationEnabled ()`
- `def getIsEnabled ([dimension=0])`

- `def getIsPersistent ()`
- `def getIsVisible ()`
- `def getLabel ()`
- `def getNumDimensions ()`
- `def getParent ()`
- `def getScriptName ()`
- `def getTypeName ()`
- `def random<NatronEngine.Param.random() ([min=0.,max=1.]`
- `def random<NatronEngine.Param.random() (seed)`
- `def randomInt<NatronEngine.Param.randomInt () (min,max)`
- `def randomInt<NatronEngine.Param.randomInt () (seed)`
- `def setAddNewLine (a)`
- `def setAnimationEnabled (e)`
- `def setEnabled (enabled[, dimension=0])`
- `def setEnabledByDefault (enabled)`
- `def setEvaluateOnChange (eval)`
- `def setIconFilePath (icon)`
- `def setHelp (help)`
- `def setPersistent (persistent)`
- `def setVisible (visible)`
- `def setVisibleByDefault (visible)`
- `def setAsAlias (otherParam)`
- `def slaveTo (otherParam, thisDimension, otherDimension)`
- `def unslave (dimension)`

Detailed Description

The Param object can be used to control a specific parameter of a node. There are different types of parameters, ranging from the single checkbox (boolean) to parametric curves. Each type of parameter has specific functions to control the parameter according to its internal value type. In this base class, all common functionalities for parameters have been gathered.

Warning: Note that since each child class has a different value type, all the functions to set/get values, and set/get keyframes are specific for each class.

A Param can have several functions to control some properties, namely:

- `addNewLine`: When True, the next parameter declared will be on the same line as this parameter
- `canAnimate`: This is a static property that you cannot control which tells whether animation can be enabled for a specific type of parameter
- `animationEnabled`: For all parameters that have `canAnimate=True`, this property controls whether this parameter should be able to animate (= have keyframes) or not

- `evaluateOnChange`: This property controls whether a new render should be issues when the value of this parameter changes
- `help`: This is the tooltip visible when hovering the parameter with the mouse
- `enabled`: Should this parameter be editable by the user or not. Generally, disabled parameters have their text in painted in black.
- `visible`: Should this parameter be visible in the user interface or not
- `persistant`: If true then the parameter value will be saved in the project
- `dimension`: How many dimensions this parameter has. For instance a *Double3DParam* has 3 dimensions. A *ParametricParam* has as many dimensions as there are curves.

Note that most of the functions in the API of Params take a *dimension* parameter. This is a 0-based index of the dimension on which to operate.

The following table sums up the different properties for all parameters including type-specific properties not listed above.

Note that most of the properties are not dynamic: they need to be set before calling `refreshUserParamsGUI()` which will create the GUI for these parameters.

Warning: A non-dynamic property can no longer be changed once `refreshUserParamsGUI()` has been called.

For non *user-parameters* (i.e: parameters that were defined by the underlying OpenFX plug-in), only their **dynamic** properties can be changed since `refreshUserParamsGUI()` will only refresh user parameters.

If a Setter function contains a (*) that means it can only be called for user parameters, it has no effect on already declared non-user parameters.

Name:	Type:	Dynamic:	Setter:	Getter:	Default:
name	string	no	None	getScriptName	""
label	string	no	None	getLabel	""
help	string	yes	setHelp(*)	getHelp	""
addNewLine	bool	no	setAddNewLine(*)	getAddNewLine	True
persistent	bool	yes	setPersistant(*)	getIsPersistant	True
evaluatesOnChange	bool	yes	setEvaluateOnChange(*)	getEvaluateOnChange	True
animates	bool	no	setAnimationEnabled(*)	getIsAnimationEnabled	See (1)
visible	bool	yes	setVisible	getIsVisible	True
enabled	bool	yes	setEnabled	getIsEnabled	True
<i>Properties on IntParam, Int2DParam, Int3DParam, DoubleParam, Double2DParam, Double3DParam, ColorParam only:</i>					
min	int/double	yes	setMinimum(*)	getMinimum	INT_MIN
max	int/double	yes	setMaximum(*)	getMaximum	INT_MAX
displayMin	int/double	yes	setDisplayMinimum(*)	getDisplayMinimum	INT_MIN
displayMax	int/double	yes	setDisplayMaximum(*)	getDisplayMaximum	INT_MAX
<i>Properties on ChoiceParam only:</i>					
options	list<string>	yes	setOptions/addOption(*)	getOption	empty list
<i>Properties on FileParam, OutputFileParam only:</i>					
sequenceDialog	bool	yes	setSequenceEnabled(*)	None	False
<i>Properties on StringParam only:</i>					
type	TypeEnum	no	setType(*)	None	eStringType-Default
<i>Properties on PathParam only:</i>					
multi-PathTable	bool	no	setAsMulti-PathTable(*)	None	False
<i>Properties on GroupParam only:</i>					
isTab	bool	no	setAsTab(*)	None	False

(1): animates is set to True by default only if it is one of the following parameters: IntParam Int2DParam Int3DParam DoubleParam Double2DParam Double3DParam ColorParam

Note that ParametricParam , GroupParam, PageParam, ButtonParam, FileParam, Output-FileParam, PathParam cannot animate at all.

Member functions description

`NatronEngine.Param.copy (other[, dimension=-1])`

Parameters

- **other** – Param
- **dimension** – int

Return type bool

Copies the *other* parameter values, animation and expressions at the given *dimension*. If *dimension* is -1, all dimensions in `min(getNumDimensions(), other.getNumDimensions())` will be copied.

Note: Note that types must be convertible:

IntParam, DoubleParam, ChoiceParam, ColorParam and BooleanParam can convert between types but StringParam cannot.

Warning: When copying a parameter, only values are copied, not properties, hence if copying a choice parameter, make sure that the value you copy has a meaning to the receiver otherwise you might end-up with an undefined behaviour, e.g.:

If ChoiceParam1 has 3 entries and the current index is 2 and ChoiceParam2 has 15 entries and current index is 10, copying ChoiceParam2 to ChoiceParam1 will end-up in undefined behaviour.

This function returns **True** upon success and **False** otherwise.

NatronEngine.Param.**curve** (*time* [, *dimension=-1*])

Parameters

- **time** – float
- **dimension** – int

Return type float

If this parameter has an animation curve on the given *dimension*, then the value of that curve at the given *time* is returned. If the parameter has an expression on top of the animation curve, the expression will be ignored, ie.g. the value of the animation curve will still be returned. This is useful to write custom expressions for motion design such as looping, reversing, etc...

NatronEngine.Param.**getAddNewLine** ()

Return type bool

Returns whether the parameter is on a new line or not.

NatronEngine.Param.**getCanAnimate** ()

Return type bool

Returns whether this class can have any animation or not. This cannot be changed. calling *setAnimationEnabled(True)* will not enable animation for parameters that cannot animate.

NatronEngine.Param.**getEvaluateOnChange** ()

Return type bool

Returns whether this parameter can evaluate on change. A parameter evaluating on change means that a new render will be triggered when its value changes due to a call of one of the setValue functions.

NatronEngine.Param.**getHelp** ()

Return type str

Returns the help tooltip visible when hovering the parameter with the mouse on the GUI;

NatronEngine.Param.**getIsAnimationEnabled** ()

Return type bool

Returns whether animation is enabled for this parameter. This is dynamic and can be changed by *setAnimationEnabled(bool)* if the parameter *can animate*.

NatronEngine.Param.**getIsEnabled** ([*dimension=0*])

Parameters *dimension* – int

Return type bool

Returns whether the given *dimension* is enabled or not.

NatronEngine.Param.**getIsPersistent** ()

Return type `bool`

Returns whether this parameter should be persistent in the project or not. Non-persistent parameter will not have their value saved when saving a project.

`NatronEngine.Param.getIsVisible()`

Return type `bool`

Returns whether the parameter is visible on the user interface or not.

`NatronEngine.Param.getLabel()`

Return type `str`

Returns the *label* of the parameter. This is what is displayed in the settings panel of the node. See [this section](#) for an explanation of the difference between the *label* and the *script name*

`NatronEngine.Param.getNumDimensions()`

Return type `int`

Returns the number of dimensions. For example a *Double3DParam* has 3 dimensions. A *ParametricParam* has as many dimensions as there are curves.

`NatronEngine.Param.getParent()`

Return type `NatronEngine.Param`

If this param is within a *group*, then the parent will be the group. Otherwise the param's parent will be the:doc:page<PageParam> onto which the param appears in the settings panel.

`NatronEngine.Param.getScriptName()`

Return type `str`

Returns the *script-name* of the param as used internally. The script-name is visible in the tooltip of the parameter when hovering the mouse over it on the GUI. See [this section](#) for an explanation of the difference between the *label* and the *script name*

`NatronEngine.Param.getTypeName()`

Return type `str`

Returns the type-name of the parameter.

`NatronEngine.Param.random([min=0., max=1.])`

Parameters

- **min** – float
- **max** – float

Return type `float`

Returns a pseudo-random value in the interval $[min, max]$. The value is produced such that for a given parameter it will always be the same for a given time on the timeline, so that the value can be reproduced exactly.

Note: Note that if you are calling multiple times `random()` in the same parameter expression, each call would return a different value, but they would all return the same value again if the expressions is interpreted at the same time, e.g.:

```
# Would always return the same value at a given timeline's time. random() - random()
```

Note that you can ensure that `random()` returns a given value by calling the overloaded function `random(min, max, time, seed)` instead.

`NatronEngine.Param.random(min, max, time[, seed=0])`

Parameters

- **min** – float
- **max** – float
- **time** – float
- **seed** – unsigned int

Return type float

Same as `random()` but takes **time** and **seed** in parameters to control the value returned by the function. E.g:

```
ret = random(0,1,frame,2) - random(0,1,frame,2)
# ret == 0 always
```

NatronEngine.Param.**randomInt** (*min*, *max*)

Parameters

- **min** – int
- **max** – int

Return type int

Same as `random(min, max)` but returns an integer in the range [*min*, **max**[

NatronEngine.Param.**randomInt** (*min*, *max*, *time*[, *seed*=0])

Parameters

- **min** – int
- **max** – int
- **time** – float
- **seed** – unsigned int

Return type int

Same as `random(min, max, time, seed)` but returns an integer in the range [0, INT_MAX] instead.

NatronEngine.Param.**setAddNewLine** (*a*)

Parameters **a** – bool

Set whether the parameter should be on a new line or not. See `getAddNewLine()`

NatronEngine.Param.**setAnimationEnabled** (*e*)

Parameters **e** – bool

Set whether animation should be enabled (= can have keyframes). See `getIsAnimationEnabled()`

NatronEngine.Param.**setEnabled** (*enabled*[, *dimension*=0])

Parameters

- **enabled** – bool
- **dimension** – int

Set whether the given *dimension* of the parameter should be enabled or not. When disabled, the parameter will be displayed in black and the user will not be able to edit it. See `getIsEnabled(dimension)`

NatronEngine.Param.**setEnabledByDefault** (*enabled*)

Parameters **enabled** – bool

Set whether the parameter should be enabled or not by default. When disabled, the parameter will be displayed in black and the user will not be able to edit it.

NatronEngine.Param.**setEvaluateOnChange** (*eval*)

Parameters `eval` – bool

Set whether evaluation should be enabled for this parameter. When True, calling any function that change the value of the parameter will trigger a new render. See `getEvaluateOnChange()`

`NatronEngine.Param.setIconFilePath(icon)`

Parameters `icon` – str

Set here the icon file path for the label. This should be either an absolute path or a file-path relative to a path in the `NATRON_PLUGIN_PATH`. The icon will replace the label of the parameter.

`NatronEngine.Param.setHelp(help)`

Parameters `help` – str

Set the help tooltip of the parameter. See `getHelp()`

`NatronEngine.Param.setPersistant(persistant)`

Parameters `persistant` – bool

Set whether this parameter should be persistant or not. Non persistant parameter will not be saved in the project. See `getIsPersistant`

`NatronEngine.Param.setVisible(visible)`

Parameters `visible` – bool

Set whether this parameter should be visible or not to the user. See `getIsVisible()`

`NatronEngine.Param.setVisibleByDefault(visible)`

Parameters `visible` – bool

Set whether this parameter should be visible or not to the user in its default state.

`NatronEngine.Param.setAsAlias(otherParam)`

Parameters `otherParam` – Param

Return type bool

Set this parameter as an alias of `otherParam`. They need to be both of the same *type* and of the same *dimension*. This parameter will control `otherParam` entirely and in case of a choice param, its drop-down menu will be updated whenever the `otherParam` menu is updated.

This is used generally to make user parameters on groups with the “Pick” option of the “Manage User Parameters” dialog.

`NatronEngine.Param.slaveTo(otherParam, thisDimension, otherDimension)`

Parameters

- `otherParam` – Param
- `thisDimension` – int
- `otherDimension` – int

Return type bool

Set this parameter as a slave of `otherParam`. They need to be both of the same *type* but may vary in dimension, as long as `thisDimension` is valid according to the number of dimensions of this parameter and `otherDimension` is valid according to the number of dimensions of `otherParam`.

This parameter `thisDimension` will be controlled entirely by the `otherDimension` of `otherParam` until a call to `unslave(thisDimension)` is made

`NatronEngine.Param.unslave(dimension)`

Parameters `dimension` – int

If the given *dimension* of this parameter was previously slaved, then this function will remove the link between parameters, and the user will be free again to use this parameter as any other.

Note: The animation and values that were present before the link will remain.

ParametricParam

Inherits *Param*

Synopsis

A parametric param represents one or more parametric functions as curves. See *detailed* explanation below.

Functions

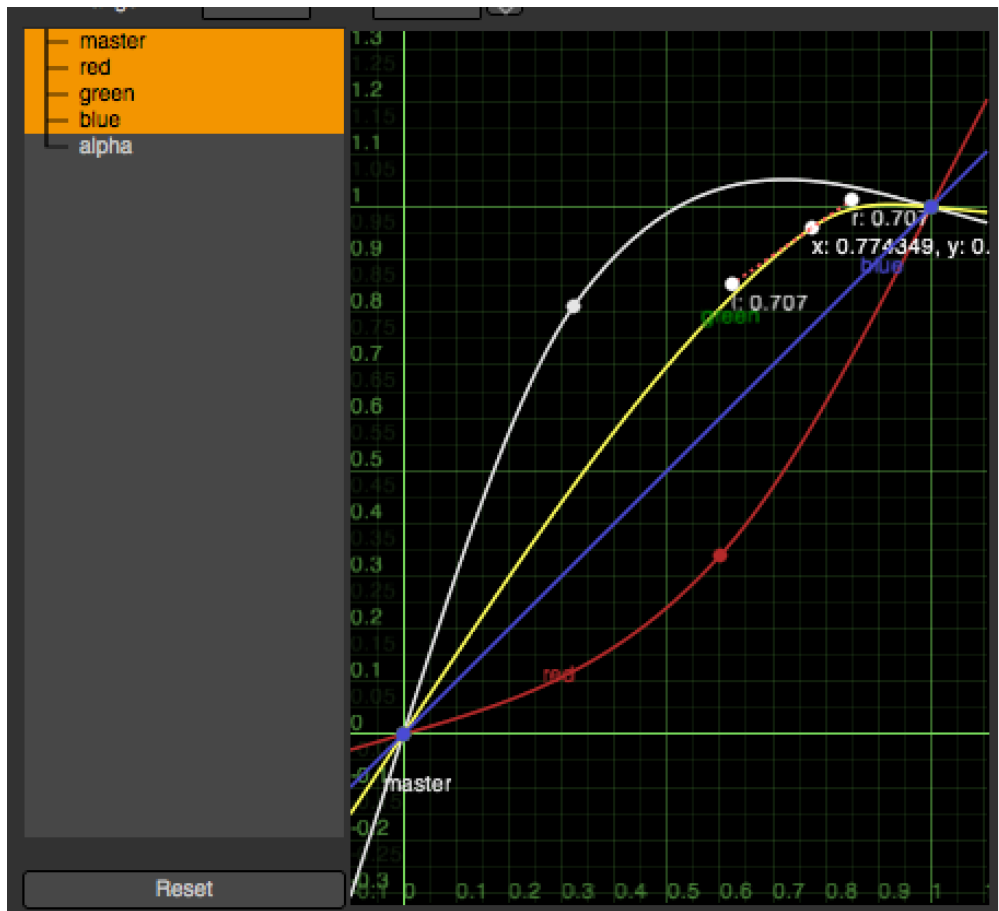
- def *addControlPoint* (dimension, key, value[,interpolation=NatronEngine.Natron.KeyframeTypeEnum.eKeyframeTypeSmooth])
- def *addControlPoint* (dimension, key, value, leftDerivative, rightDerivative, [,interpolation=NatronEngine.Natron.KeyframeTypeEnum.eKeyframeTypeSmooth])
- def *deleteAllControlPoints* (dimension)
- def *deleteControlPoint* (dimension, nthCtl)
- def *getCurveColor* (dimension)
- def *getNControlPoints* (dimension)
- def *getNthControlPoint* (dimension, nthCtl)
- def *getValue* (dimension, parametricPosition)
- def *setCurveColor* (dimension, r, g, b)
- def *setNthControlPoint* (dimension, nthCtl, key, value, leftDerivative, rightDerivative)
- def *setNthControlPointInterpolation* (dimension, nthCtl, interpolation)
- def :meth: *setDefaultCurvesFromCurrentCurves*<NatronEngine.ParametricParam.setDefaultCurvesFromCurrentCurves>()

Detailed Description

A parametric parameter has as many dimensions as there are curves. Currently the number of curves is static and you may only specify the number of curves via the *nbCurves* argument of the *createParametricParam(name, label, nbCurves)* function.

Parametric curves work almost the same way that animation curves do: you can add control points and remove them.

You can peak the value of the curve at a special *parametric position* with the *getValue(dimension, parametricPosition)* function. The *parametric position* is represented by the X axis on the graphical user interface.



Member functions description

`NatronEngine.ParametricParam.addControlPoint` (*dimension*, *key*, *value* [, *interpolation*=`NatronEngine.Natron.KeyframeTypeEnum.eKeyframeType`])

Parameters

- **dimension** – int
- **key** – float
- **value** – float
- **interpolation** – `KeyframeTypeEnum`

Return type `StatusEnum`

Attempts to add a new control point to the curve at the given *dimension*. The new point will have the coordinate (key,value). This function returns a `NatronEngine.Natron.StatusEnum.eStatusOK` upon success, otherwise `NatronEngine.Natron.StatusEnum.eStatusFailed` is returned upon failure.

`NatronEngine.ParametricParam.addControlPoint` (*dimension*, *key*, *value*, *leftDerivative*, *rightDerivative* [, *interpolation*=`NatronEngine.Natron.KeyframeTypeEnum.eKeyframeType`])

Parameters

- **dimension** – int
- **key** – float
- **value** – float

- **leftDerivative** – float
- **rightDerivative** – float
- **interpolation** – *KeyFrameTypeEnum*

Return type *StatusEnum*

Attempts to add a new control point to the curve at the given *dimension*. The new point will have the coordinate (key,value) and the derivatives (leftDerivative, rightDerivative). This function returns a `NatronEngine.Natron.StatusEnum.eStatusOK` upon success, otherwise `NatronEngine.Natron.StatusEnum.eStatusFailed` is returned upon failure.

`NatronEngine.ParametricParam.deleteAllControlPoints (dimension)`

Parameters **dimension** – int

Return type *StatusEnum*

Removes all control points of the curve at the given *dimension*. This function returns a `NatronEngine.Natron.StatusEnum.eStatusOK` upon success, otherwise `NatronEngine.Natron.StatusEnum.eStatusFailed` is returned upon failure.

`NatronEngine.ParametricParam.deleteControlPoint (dimension, nthCtl)`

Parameters

- **dimension** – int
- **nthCtl** – int

Return type *StatusEnum*

Attempts to remove the *nth* control point (sorted in increasing X order) of the parametric curve at the given *dimension*.

This function returns a `NatronEngine.Natron.StatusEnum.eStatusOK` upon success, otherwise `NatronEngine.Natron.StatusEnum.eStatusFailed` is returned upon failure.

`NatronEngine.ParametricParam.getCurveColor (dimension)`

Parameters **dimension** – *ColorTuple*

Returns a *ColorTuple* with the [R,G,B] color of the parametric curve at the given *dimension* on the graphical user interface.

`NatronEngine.ParametricParam.getNControlPoints (dimension)`

Parameters **dimension** – int

Return type int

Returns the number of control points of the curve at the given *dimension*.

`NatronEngine.ParametricParam.getNthControlPoint (dimension, nthCtl)`

Parameters

- **dimension** – int
- **nthCtl** – int

Return type tuple

Returns a *tuple* containing informations about the *nth* control point (sorted by increasing X order) control point of the curve at the given *dimension*. The tuple is composed of 5 members:

[status: *StatusEnum*, key : float, value: float, left derivative: float, right derivative: float]

This function returns in the status a `NatronEngine.Natron.StatusEnum.eStatusOK` upon success, otherwise `NatronEngine.Natron.StatusEnum.eStatusFailed` is returned upon failure.

`NatronEngine.ParametricParam.getValue (dimension, parametricPosition)`

Parameters

- **dimension** – int
- **parametricPosition** – double

Return type double

Returns the Y value of the curve at the given *parametricPosition* (on the X axis) of the curve at the given *dimension*.

NatronEngine.ParametricParam.**setCurveColor** (*dimension*, *r*, *g*, *b*)

Parameters

- **dimension** – int
- **r** – float
- **g** – float
- **b** – float

Set the color of the curve at the given *dimension*.

NatronEngine.ParametricParam.**setNthControlPoint** (*dimension*, *nthCtl*, *key*, *value*, *leftDerivative*, *rightDerivative*)

Parameters

- **dimension** – int
- **nthCtl** – int
- **key** – float
- **value** – float
- **leftDerivative** – float
- **rightDerivative** – float

Return type *StatusEnum*

Set the value of an existing control point on the curve at the given *dimension*. The *nthCtl* parameter is the (zero based) index of the control point (by increasing X order). The point will be placed at the coordinates defined by (key,value) and will have the derivatives given by *leftDerivative* and *rightDerivatives*.

This function returns a NatronEngine.Natron.StatusEnum.eStatusOK upon success, otherwise NatronEngine.Natron.StatusEnum.eStatusFailed is returned upon failure.

NatronEngine.ParametricParam.**setNthControlPointInterpolation** (*dimension*, *nthCtl*, *interpolation*)

Parameters

- **dimension** – int
- **nthCtl** – int
- **interpolation** – *KeyFrameTypeEnum*

Return type *StatusEnum*

Set the interpolation type of the curve surrounding the control point at the given index *nthCtl*.

NatronEngine.ParametricParam.**setDefaultCurvesFromCurrentCurves** ()

Set the default curves of the parameter from the current state of the curves. The default state will be used when the parameter is restored to default.

PathParam

Inherits *StringParamBase*

Synopsis

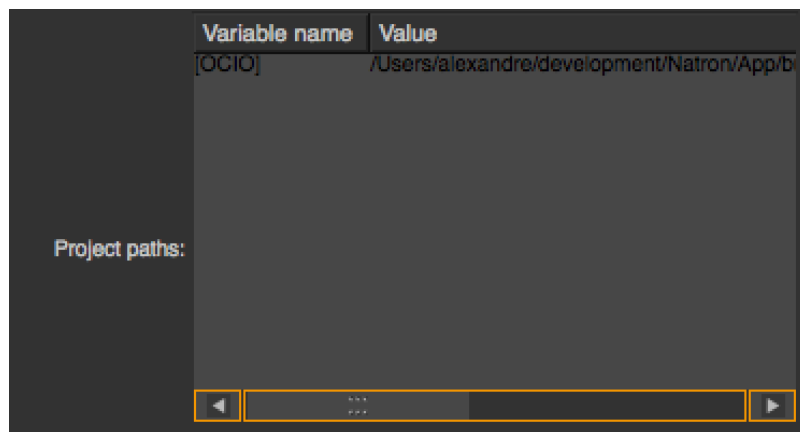
A path param is used to indicate the path to a directory. See *details...*

Functions

- def *setAsMultiPathTable* ()
- def *getTable* ()
- def *setTable* (table)

Detailed Description

By default the user can select a single directory as path, unless *setAsMultiPathTable* () is called in which case a table is presented to the user to specify multiple directories like this:



When using multiple paths, internally they are separated by a ; and the following characters are escaped as per the XML specification:

- < becomes <
- > becomes >
- & becomes &
- “ becomes "
- ‘ becomes '

Some more characters are escaped, you can see the full function in the source code of Natron [here](#)

Member functions description

`NatronEngine.PathParam.setAsMultiPathTable ()`

When called, the parameter will be able to store multiple paths.

`NatronEngine.PathParam.getTable ()`

Return type PySequence

Returns a list of list of strings. Each sub-list corresponds to a row in the table. Each elements of the row are the cell value for each column.

`NatronEngine.PathParam.setTable (table)`

Rparam table PySequence

Set the parameter to a list of list of strings. Each sub-list corresponds to a row in the table. Each elements of the row are the cell value for each column. An error will be invoked if the number of columns in the provided *table* do not match the number of columns of the parameter's table.

PyCoreApplication

Inherited by: *PyGuiApplication*

Synopsis

This object represents a background instance of Natron. See *detailed description...*

Functions

- def *appendToNatronPath* (path)
- def *getSettings* ()
- def *getBuildNumber* ()
- def *getInstance* (idx)
- def *getActiveInstance* ()
- def *getNatronDevelopmentStatus* ()
- def *getNatronPath* ()
- def *getNatronVersionEncoded* ()
- def *getNatronVersionMajor* ()
- def *getNatronVersionMinor* ()
- def *getNatronVersionRevision* ()
- def *getNatronVersionString* ()
- def *getNumCpus* ()
- def *getNumInstances* ()
- def *getPluginIDs* ()
- def *getPluginIDs* (filter)
- def *isBackground* ()
- def *is64Bit* ()
- def *isLinux* ()
- def *isMacOSX* ()
- def *isUnix* ()
- def *isWindows* ()
- def *setOnProjectCreatedCallback* (pythonFunctionName)
- def *setOnProjectLoadedCallback* (pythonFunctionName)

Detailed Description

When running Natron there's a **unique** instance of the *PyCoreApplication* object. It holds general informations about the process.

Generally, throughout your scripts, you can access this object with the variable *natron* that Natron pre-declared for you, e.g.:

```
natron.getPluginIDs()
```

Warning: The variable **natron** belongs to the module **NatronEngine**, hence make sure to make the following import:

```
from NatronEngine import *
```

Otherwise with a regular *import* you can still access **natron** by prepending the module:

```
NatronEngine.natron
```

Warning: The variable stored in the module **NatronEngine** contains a reference to a *PyCoreApplication*. If you need to have the GUI functionalities provided by *PyGuiApplication*, you must then use the variable **natron** belonging to the module **NatronGui**. Hence make sure to make the following import to have access to **natron**:

```
from NatronGui import *
```

With a regular import you can access it using **NatronGui.natron**.

Warning: Make sure to **not** make the 2 following imports, otherwise the **natron** variable will not point to something expected:

```
#This you should not do!
from NatronEngine import *
from NatronGui import *

#This is OK
import NatronEngine
import NatronGui

#This can also be done for convenience
from NatronEngine import NatronEngine.natron as NE
from NatronGui import NatronGui.natron as NG
```

This class is used only for background (command-line) runs of Natron, that is when you launch Natron in the following ways:

```
Natron -b ...
Natron -t
NatronRenderer
```

For interactive runs of Natron (with the user interface displayed), the derived class *PyGuiApplication* is used instead, which gives access to more GUI specific functionalities.

You should never need to make a new instance of this object yourself. Note that even if you did, internally the same object will be used and they will all refer to the same Natron application.

In GUI mode, a `:doc'PyGuiApplication'` can have several projects opened. For each project you can refer to them with pre-declared variables *app1* , *app2*, etc...

In background mode, there would be only 1 project opened, so Natron does the following assignment for you before calling any scripts:

```
app = app1
```

See [App](#) to access different opened projects.

Member functions description

class `NatronEngine.PyCoreApplication`

Defines a new variable pointing to the same underlying application that the *natron* variable points to. This is equivalent to calling:

```
myVar = natron
```

`NatronEngine.PyCoreApplication.appendToNatronPath(path)`

Parameters `path` – str

Adds a new path to the Natron search paths. See [this section](#) for a detailed explanation of Natron search paths.

`NatronEngine.PyCoreApplication.getSettings()`

Return type `AppSettings`

Returns an object containing all Natron settings. The settings are what can be found in the preferences of Natron.

`NatronEngine.PyCoreApplication.getBuildNumber()`

Return type `int`

Returns the build-number of the current version of Natron. Generally this is used for release candidates, e.g.:

Natron v1.0.0-RC1 : build number = 1 Natron v1.0.0-RC2 : build number = 2 Natron v1.0.0-RC3 :
build number = 3

`NatronEngine.PyCoreApplication.getInstance(idx)`

Parameters `idx` – int

Return type `App`

Returns the [App](#) instance at the given *idx*. Note that *idx* is 0-based, e.g.: 0 would return what's pointed to by *app1*.

`NatronEngine.PyCoreApplication.getActiveInstance()`

Return type `App`

Returns the [App](#) instance corresponding to the last project the user interacted with.

`NatronEngine.PyCoreApplication.getNatronDevelopmentStatus()`

Return type `str`

Returns a string describing the development status of Natron. This can be one of the following values:

- Alpha : Meaning the software has unimplemented functionalities and probably many bugs left
- Beta : Meaning the software has all features that were planned are implemented but there may be bugs
- RC : Meaning the software seems in a good shape and should be ready for release unless some last minute show-stoppers are found
- Release : Meaning the software is ready for production

`NatronEngine.PyCoreApplication.getNatronPath()`

Return type `sequence`

Returns a sequence of string with all natron [search paths](#).

`NatronEngine.PyCoreApplication.getNatronVersionEncoded()`

Return type `int`

Returns an *int* with the version of Natron encoded so that you can compare versions of Natron like this:

```
if natron.getNatronVersionEncoded() >= 20101:
    ...
```

In that example, Natron's version would be 2.1.1

`NatronEngine.PyCoreApplication.getNatronVersionMajor()`

Return type `int`

Returns the major version of Natron. If the version is 1.0.0, that would return 1.

`NatronEngine.PyCoreApplication.getNatronVersionMinor()`

Return type `int`

Get the minor version of Natron. If the version is 1.2.0, that would return 2.

`NatronEngine.PyCoreApplication.getNatronVersionRevision()`

Return type `int`

Returns the revision number of the version. If the version is 1.2.3, that would return 3.

`NatronEngine.PyCoreApplication.getNatronVersionString()`

Return type `str`

Returns the version of Natron as a string, e.g.: "1.1.0"

`NatronEngine.PyCoreApplication.getNumCpus()`

Return type `int`

Returns the maximum hardware concurrency of the computer. If the computer has 8 hyper-threaded cores, that would return 16.

`NatronEngine.PyCoreApplication.getNumInstances()`

Return type `int`

Returns the number of :doc:'App' instances currently active.

`NatronEngine.PyCoreApplication.getPluginIDs()`

Return type `sequence`

Returns a sequence of strings with all plugin-IDs currently loaded.

`NatronEngine.PyCoreApplication.getPluginIDs(filter)`

Parameters *filter* – `str`

Return type `sequence`

Same as `getPluginIDs()` but returns only plug-ins *containing* the given *filter*. Comparison is done **without** case-sensitivity.

`NatronEngine.PyCoreApplication.isBackground()`

Return type `bool`

Returns True if Natron is executed in background mode, i.e: from the command-line, without any graphical user interface displayed.

`NatronEngine.PyCoreApplication.is64Bit()`

Return type `bool`

Returns True if Natron is executed on a 64 bit computer.

`NatronEngine.PyCoreApplication.isLinux()`

Return type `bool`

Returns True if Natron is executed on a Linux or FreeBSD distribution.

`NatronEngine.PyCoreApplication.isMacOSX()`

Return type `bool`

Returns True if Natron is executed on MacOSX.

`NatronEngine.PyCoreApplication.isUnix()`

Return type `bool`

Returns True if Natron is executed on Unix. Basically this is equivalent to:

```
if natron.isLinux() or natron.isMacOSX():
```

`NatronEngine.PyCoreApplication.isWindows()`

Return type `bool`

Returns True if Natron is executed on Windows.

`NatronEngine.PyCoreApplication.setOnProjectCreatedCallback (pythonFunctionName)`

Param `str`

Convenience function to set the After Project Created callback. Note that this will override any callback set in the Preferences->Python->After Project created. This is exactly the same as calling:

```
NatronEngine.settings.afterProjectCreated.set (pythonFunctionName)
```

Note: Clever use of this function can be made in the `init.py` script to do generic stuff for all projects (whether they are new projects or loaded projects). For instance one might want to add a list of Formats to the project. See the example [here](#)

`NatronEngine.PyCoreApplication.setOnProjectLoadedCallback (pythonFunctionName)`

Param `str`

Convenience function to set the Default After Project Loaded callback. Note that this will override any callback set in the Preferences->Python->Default After Project Loaded. This is exactly the same as calling:

```
NatronEngine.settings.defOnProjectLoaded.set (pythonFunctionName)
```

RectD

Synopsis

A rectangle defined with floating point precision. See [detailed](#) description below

Functions

- `def area()`
- `def bottom()`
- `def clear()`
- `def contains (otherRect)`

- `def height ()`
- `def intersect (otherRect)`
- `def intersects (otherRect)`
- `def isInfinite ()`
- `def isNull ()`
- `def left ()`
- `def merge (otherRect)`
- `def right ()`
- `def set (x1,y1,x2,y2)`
- `def set_bottom (y1)`
- `def set_left (x1)`
- `def set_right (x2)`
- `def set_top (y2)`
- `def top ()`
- `def translate (dx,dy)`
- `def width ()`

Detailed Description

A rectangle where $x1 < x2$ and $y1 < y2$ such as `width() == (x2 - x1) && height() == (y2 - y1)` ($x1, y1$) is are the coordinates of the bottom left corner of the rectangle. The last element valid in the y dimension is $y2 - 1$ and the last valid in the x dimension is $x2 - 1$. $x1, x2, y1$ and $y2$ are with floating point precision.

Member functions description

`NatronEngine.RectD.area ()`

Return type `double`

Returns the area covered by the rectangle, that is: $(y2 - y1) * (x2 - x1)$

`NatronEngine.RectD.bottom ()`

Return type `double`

Returns the bottom edge, that is the

`NatronEngine.RectD.clear ()`

Same as `set (0,0,0,0)`

`NatronEngine.RectD.contains (otherRect)`

Parameters `otherRect` – `RectD`

Return type `bool`

Returns True if `otherRect` is contained in or equals this rectangle, that is if:

```
otherRect.x1 >= x1 and
otherRect.y1 >= y1 and
otherRect.x2 <= x2 and
otherRect.y2 <= y2
```

`NatronEngine.RectD.height ()`

Return type double

Returns the height of the rectangle, that is: $y_2 - y_1$

`NatronEngine.RectD.intersect (otherRect)`

Parameters `otherRect` – RectD

Return type RectD

Returns the intersection between this rectangle and *otherRect*. If the intersection is empty, the return value will have the `isNull()` function return True.

`NatronEngine.RectD.intersects (otherRect)`

Parameters `otherRect` – RectD

Return type bool

Returns True if rectangle and *otherRect* intersect.

`NatronEngine.RectD.isInfinite()`

Return type bool

Returns True if this rectangle is considered to cover an infinite area. Some generator effects use this to indicate that they can potentially generate an image of infinite size.

`NatronEngine.RectD.isNull()`

Return type bool

Returns true if $x_2 \leq x_1$ or $y_2 \leq y_1$

`NatronEngine.RectD.left()`

Return type double

Returns x_1 , that is the position of the left edge of the rectangle.

`NatronEngine.RectD.merge (otherRect)`

Parameters `otherRect` – RectD

Unions this rectangle with *otherRect*. In other words, this rectangle becomes the bounding box of this rectangle and *otherRect*.

`NatronEngine.RectD.left()`

Return type double

Returns x_1 , that is the position of the left edge of the rectangle.

`NatronEngine.RectD.right()`

Return type double

Returns x_2 , that is the position of the right edge of the rectangle. x_2 is considered to be the first element outside the rectangle.

`NatronEngine.RectD.set (x1, y1, x2, y2)`

Parameters

- **x1** – double
- **y1** – double
- **x2** – double
- **y2** – double

Set the x_1, y_1, x_2, y_2 coordinates of this rectangle.

`NatronEngine.RectD.set_bottom (y1)`

Parameters **y1** – double

Set y1

`NatronEngine.RectD.set_left (x1)`

Parameters **y1** – double

Set x1

`NatronEngine.RectD.set_right (x2)`

Parameters **x2** – double

Set x2

`NatronEngine.RectD.set_top (y2)`

Parameters **y2** – double

Set y2

`NatronEngine.RectD.top ()`

Return type double

Returns y2, that is the position of the top edge of the rectangle. y2 is considered to be the first element outside the rectangle.

`NatronEngine.RectD.translate (dx, dy)`

Parameters

- **dx** – double
- **dy** – double

Moves all edges of the rectangle by *dx*, *dy*, that is:

```
x1 += dx;
y1 += dy;
x2 += dx;
y2 += dy;
```

`NatronEngine.RectD.width ()`

Return type double

Returns the width of the rectangle, that is x2 - x1.

RectI

Synopsis

A rectangle defined with integer precision. See *detailed* description below

Functions

- def *bottom* ()
- def *clear* ()
- def *contains* (otherRect)
- def *height* ()
- def *intersect* (otherRect)
- def *intersects* (otherRect)

- `def isInfinite()`
- `def isNull()`
- `def left()`
- `def merge(otherRect)`
- `def right()`
- `def set(x1,y1,x2,y2)`
- `def set_bottom(y1)`
- `def set_left(x1)`
- `def set_right(x2)`
- `def set_top(y2)`
- `def top()`
- `def translate(dx,dy)`
- `def width()`

Detailed Description

A rectangle where $x1 < x2$ and $y1 < y2$ such as `width() == (x2 - x1) && height() == (y2 - y1)` ($x1, y1$) is are the coordinates of the bottom left corner of the rectangle. The last element valid in the y dimension is $y2 - 1$ and the last valid in the x dimension is $x2 - 1$. $x1, x2, y1$ and $y2$ are with integer precision.

Member functions description

`NatronEngine.RectI.bottom()`

Return type `int`

Returns the bottom edge, that is the

`NatronEngine.RectI.clear()`

Same as `set(0,0,0,0)`

`NatronEngine.RectI.contains(otherRect)`

Parameters `otherRect` – `RectI`

Return type `bool`

Returns True if `otherRect` is contained in or equals this rectangle, that is if:

```
otherRect.x1 >= x1 and
otherRect.y1 >= y1 and
otherRect.x2 <= x2 and
otherRect.y2 <= y2
```

`NatronEngine.RectI.height()`

Return type `int`

Returns the height of the rectangle, that is: $y2 - y1$

`NatronEngine.RectI.intersect(otherRect)`

Parameters `otherRect` – `RectI`

Return type `RectI`

Returns the intersection between this rectangle and *otherRect*. If the intersection is empty, the return value will have the `isNull()` function return True.

`NatronEngine.RectI.intersects (otherRect)`

Parameters `otherRect` – `RectI`

Return type `bool`

Returns True if rectangle and *otherRect* intersect.

`NatronEngine.RectI.isInfinite()`

Return type `bool`

Returns True if this rectangle is considered to cover an infinite area. Some generator effects use this to indicate that they can potentially generate an image of infinite size.

`NatronEngine.RectI.isNull()`

Return type `bool`

Returns true if `x2 <= x1` or `y2 <= y1`

`NatronEngine.RectI.left()`

Return type `int`

Returns `x1`, that is the position of the left edge of the rectangle.

`NatronEngine.RectI.merge (otherRect)`

Parameters `otherRect` – `RectI`

Unions this rectangle with *otherRect*. In other words, this rectangle becomes the bounding box of this rectangle and *otherRect*.

`NatronEngine.RectI.left()`

Return type `int`

Returns `x1`, that is the position of the left edge of the rectangle.

`NatronEngine.RectI.right()`

Return type `int`

Returns `x2`, that is the position of the right edge of the rectangle. `x2` is considered to be the first element outside the rectangle.

`NatronEngine.RectI.set (x1, y1, x2, y2)`

Parameters

- `x1` – `int`
- `y1` – `int`
- `x2` – `int`
- `y2` – `int`

Set the `x1`, `y1`, `x2`, `y2` coordinates of this rectangle.

`NatronEngine.RectI.set_bottom (y1)`

Parameters `y1` – `int`

Set `y1`

`NatronEngine.RectI.set_left (x1)`

Parameters `y1` – `int`

Set `x1`

`NatronEngine.RectI.set_right(x2)`

Parameters `x2` – int

Set x2

`NatronEngine.RectI.set_top(y2)`

Parameters `y2` – int

Set y2

`NatronEngine.RectI.top()`

Return type int

Returns y2, that is the position of the top edge of the rectangle. y2 is considered to be the first element outside the rectangle.

`NatronEngine.RectI.translate(dx, dy)`

Parameters

- `dx` – int
- `dy` – int

Moves all edges of the rectangle by `dx`, `dy`, that is:

```
x1 += dx;
y1 += dy;
x2 += dx;
y2 += dy;
```

`NatronEngine.RectI.width()`

Return type int

Returns the width of the rectangle, that is `x2 - x1`.

Roto

Synopsis

This class encapsulates all things related to the roto node. See detailed [description](#) below.

Functions

- def `createBezier`(x, y, time)
- def `createEllipse`(x, y, diameter, fromCenter, time)
- def `createLayer`()
- def `createRectangle`(x, y, size, time)
- def `getBaseLayer`()
- def `getItemByName`(name)

Detailed Description

The Roto class is uses for now in Natron exclusively by the roto node, but its functionalities could be re-used for other nodes as well. Its purpose is to manage all layers and shapes. You can create new shapes

with the `createBezier(x, y, time)`, `createEllipse(x, y, diameter, fromCenter, time)` and `createRectangle(x, y, size, time)` functions.

To create a new *Layer* you can use the `createLayer()` function.

As for other *auto-declared* variables, all shapes in the Roto objects can be accessed by their script-name, e.g.:

```
Roto1.roto.Layer1.Bezier1
```

Member functions description

`NatronEngine.Roto.createBezier(x, y, time)`

Parameters

- **x** – float
- **y** – float
- **time** – int

Return type `BezierCurve`

Creates a new *BezierCurve* with one control point at position (x,y) and a keyframe at the given *time*.

`NatronEngine.Roto.createEllipse(x, y, diameter, fromCenter, time)`

Parameters

- **x** – float
- **y** – float
- **diameter** – float
- **fromCenter** – bool
- **time** – int

Return type `BezierCurve`

Creates a new ellipse. This is a convenience function that uses `createBezier(x, y, time)` to create a new *BezierCurve* and then adds 3 other control points to the Bezier so that it forms an ellipse of the given *diameter*. A new keyframe will be set at the given *time*. If *fromCenter* is true, then (x,y) is understood to be the coordinates of the center of the ellipse, otherwise (x,y) is understood to be the position of the top-left point of the smallest enclosing rectangle of the ellipse.

`NatronEngine.Roto.createLayer()`

Return type `Layer`

Creates a new layer.

`NatronEngine.Roto.createRectangle(x, y, size, time)`

Parameters

- **x** – float
- **y** – float
- **size** – float
- **time** – int

Return type `BezierCurve`

Creates a new rectangle. This is a convenience function that uses `createBezier(x, y, time)` to create a new *BezierCurve* and then adds 3 other control points to the Bezier so that it forms a rectangle of the given *size* on each of its sides. A new keyframe will be set at the given *time*.

`NatronEngine.Roto.getBaseLayer()`

Return type `Layer`

Convenience function to access to the base [Layer](#). Note that all shapes should belong to a [Layer](#), the base layer being the top-level parent of all the hierarchy.

`NatronEngine.Roto.getItemByName(name)`

Parameters `name` – str

Return type `ItemBase`

Returns an item by its *script-name*. See [this section](#) for the details of what is the *script-name* of an item. E.g:

```
appl.Roto1.roto.Layer1.Bezier1 = appl.Roto1.roto.getItemByName("Bezier1")
```

StringParam

Inherits [StringParamBase](#)

Synopsis

This parameter is used to contain a string. See [here](#) for more details.

Functions

- def [setType](#)(type)

Detailed Description

A StringParam can have several forms on the user interface, depending on its type

Here are the different types of string parameters:

- A basic string that can be edited by the user .. figure:: stringParam.png

width 400px
align center
- A non animating label string that the user cannot edit .. figure:: stringLabel.png

width 400px
align center
- A multi-line string that the user can edit and animate .. figure:: multiLineString.png

width 400px
align center
- A multi-line string with rich text support with a [subset of html](#) .. figure:: multiLineRichTextParam.png

width 400px
align center

Member functions description

`NatronEngine.StringParam.setType (type)`

Parameters `type` – `NatronEngine.StringParam.TypeEnum`

Set the type of the `StringParam`. This should be called right away after creation time.

Warning: Once called, you should call `refreshUserParamsGUI ()` to update the user interface.

StringParamBase

Inherits *AnimatedParam*

Inherited by: *PathParam*, *OutputFileParam*, *FileParam*, *StringParam*

Synopsis

This is the base-class for all parameters holding a string. See [here](#) for more details.

Functions

- `def get ()`
- `def get (frame)`
- `def getDefaultValue ()`
- `def getValue ()`
- `def getValueAtTime (time)`
- `def restoreDefaultValue ()`
- `def set (x)`
- `def set (x, frame)`
- `def setDefaultValue (value)`
- `def setValue (value)`
- `def setValueAtTime (value, time)`

Detailed Description

A string parameter contains internally a string which can change over time. Much like keyframes for value parameters (like *IntParam* or *DoubleParam*) keyframes can be set on string params, though the interpolation will remain constant always.

Member functions description

`NatronEngine.StringParamBase.get ()`

Return type `str`

Get the value of the parameter at the current timeline's time

`NatronEngine.StringParamBase.get (frame)`

Parameters `frame` – float

Return type str

Get the value of the parameter at the given *frame*.

`NatronEngine.StringParamBase.getDefaultValue()`

Return type str

Get the default value for this parameter.

`NatronEngine.StringParamBase.getValue()`

Return type str

Same as `get()`

`NatronEngine.StringParamBase.getValueAtTime(time)`

Parameters `time` – float

Return type str

Same as `get(frame)`

`NatronEngine.StringParamBase.restoreDefaultValue()`

Removes all animation and expression set on this parameter and set the value to be the default value.

`NatronEngine.StringParamBase.set(x)`

Parameters `x` – str

Set the value of this parameter to be *x*. If this parameter is animated (see `getIsAnimated(dimension)`) then this function will automatically add a keyframe at the timeline's current time.

`NatronEngine.StringParamBase.set(x, frame)`

Parameters

- `x` – str
- `frame` – float

Set a new keyframe on the parameter with the value *x* at the given *frame*.

`NatronEngine.StringParamBase.setDefaultValue(value)`

Parameters `value` – str

Set the default *value* for this parameter.

`NatronEngine.StringParamBase.setValue(value)`

Parameters `value` – str

Same as `set`

`NatronEngine.StringParamBase.setValueAtTime(value, time)`

Parameters

- `value` – str
- `time` – float

Same as `set(time) < NatronEngine.StringParamBase.set()`

Track

Synopsis

This class represents one track marker as visible in the tracker node or on the viewer. It is available to Python to easily retrieve the tracked data. See *detailed* description below.

Functions

- def *setScriptName* (scriptName)
- def *getScriptName* ()
- def *getParam* (paramScriptName)
- def *getParams* ()
- def *reset* ()

Detailed Description

The track is internally represented by multiple *parameters* which holds animation curve for various data, such as: the track center, the pattern 4 corners, the error score, the search-window, etc... Each of them can be retrieved with the *getParam(scriptName)* function.

Here is an example briefly explaining how to retrieve the tracking data for a track:

```
myTrack = app.Tracker1.tracker.track1

keyframes = []

# get the number of keys for the X dimension only and try match the Y keyframes
nKeys = myTrack.centerPoint.getNumKeys(0)
for k in range(0,nKeys):

    # getKeyTime returns a tuple with a boolean value indicating if it succeeded,
    ↪and
    # the keyframe time

    gotXKeyTuple = myTrack.centerPoint.getKeyTime(k, 0)
    frame = gotXKeyTuple[1]

    # Only consider keyframes which have an X and Y value
    # If Y does not have a keyframe at this frame, ignore the keyframe
    # getKeyIndex returns a value >=0 if there is a keyframe
    yKeyIndex = myTrack.centerPoint.getKeyIndex(frame, 1)

    if yKeyIndex == -1:
        continue

    # Note that even if the x curve or y curve didn't have a keyframe we
    # could still call getValueAtTime but the value would be interpolated by
    # Natron with surrounding keyframes, which is not what we want.

    x = myTrack.centerPoint.getValueAtTime(frame, 0)
    y = myTrack.centerPoint.getValueAtTime(frame, 1)

    keyframes.append((x,y))

print keyframes
```

Member functions description

`NatronEngine.Track.setScriptName (scriptName)`

Parameters `scriptName` – str

Set the script-name of the track. It will then be accessible via a Python script as such:

```
Tracker1.tracker.MyTrackScriptName
```

`NatronEngine.Track.getScriptName ()`

Return type str

Get the script-name of the track

`NatronEngine.Track.getParam (paramScriptName)`

Return type Param

Get the Param with the given *paramScriptName*. The parameter can also be retrieved as an attribute of the *tracker* object like this:

```
Tracker1.tracker.center
```

`NatronEngine.Track.getParams ()`

Return type Param

Returns a list of all the Param for this track.

`NatronEngine.Track.reset ()`

Resets the track completely removing any animation on all parameters and any keyframe on the pattern.

Tracker

Synopsis

This class is a container for tracks See [detailed](#) description below.

Functions

- def *createTrack* ()
- def *getTrackByName* (scriptName)
- def *getAllTracks* ()
- def *getSelectedTracks* ()
- def *startTracking* (tracks, start, end, forward)
- def *stopTracking* ()

Detailed Description

The Tracker is a special class attached to effects that needs tracking capabilities. It contains all tracks for this node and also allow to start and stop tracking from a Python script.

Member functions description

`NatronEngine.Tracker.createTrack()`

Return type `Track`

Creates a new track in the tracker with default values

`NatronEngine.Tracker.getTrackByName(scriptName)`

Return type `Track`

Returns a track matching the given *scriptName* if any

`NatronEngine.Tracker.getAllTracks()`

Return type `sequence`

Returns all the tracks in this Tracker.

`NatronEngine.Tracker.getSelectedTracks()`

Return type `sequence`

Returns the user selected tracks

`NatronEngine.Tracker.startTracking(tracks, start, end, forward)`

Start tracking the given *tracks* from *start* frame to *end* frame (*end* frame will not be tracked) in the direction given by *forward*. If *forward* is **False**, then *end* is expected to be lesser than *start*.

`NatronEngine.Tracker.stopTracking()`

Stop any ongoing tracking for this Tracker.

UserParamHolder

Inherited by : *Effect*, *PyModalDialog*

Synopsis

This is an abstract class that serves as a base interface for all objects that can hold user parameters. See *Detailed Description*

Functions

- `def createBooleanParam(name, label)`
- `def createButtonParam(name, label)`
- `def createChoiceParam(name, label)`
- `def createColorParam(name, label, useAlpha)`
- `def createDouble2DParam(name, label)`
- `def createDouble3DParam(name, label)`
- `def createDoubleParam(name, label)`
- `def createFileParam(name, label)`
- `def createGroupParam(name, label)`
- `def createInt2DParam(name, label)`
- `def createInt3DParam(name, label)`
- `def createIntParam(name, label)`

- `def createOutputFileParam (name, label)`
- `def createPageParam (name, label)`
- `def createParametricParam (name, label, nbCurves)`
- `def createPathParam (name, label)`
- `def createStringParam (name, label)`
- `def removeParam (param)`
- `def refreshUserParamsGUI ()`

Detailed Description

To create a new user *parameter* on the object, use one of the **createXParam** function. To remove a user parameter created, use the `removeParam (param)` function. Note that this function can only be used to remove **user parameters** and cannot be used to remove parameters that were defined by the OpenFX plug-in.

Once you have made modifications to the user parameters, you must call the `refreshUserParamsGUI ()` function to notify the GUI, otherwise no change will appear on the GUI.

Member functions description

`NatronEngine.UserParamHolder.createBooleanParam (name, label)`

Parameters

- **name** – str
- **label** – str

Return type BooleanParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type boolean which will appear in the user interface as a checkbox.

Warning: After calling this function you should call `refreshUserParamsGUI ()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createButtonParam (name, label)`

Parameters

- **name** – str
- **label** – str

Return type ButtonParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type button which will appear as a push button. Use the `onParamChanged` callback of the Effect to handle user clicks.

Warning: After calling this function you should call `refreshUserParamsGUI ()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createChoiceParam (name, label)`

Parameters

- **name** – str
- **label** – str

Return type ChoiceParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type choice which will appear as a dropdown combobox.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

NatronEngine.UserParamHolder.**createColorParam**(*name*, *label*, *useAlpha*)

Parameters

- **name** – str
- **label** – str
- **useAlpha** – bool

Return type ColorParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type color.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

NatronEngine.UserParamHolder.**createDouble2DParam**(*name*, *label*)

Parameters

- **name** – str
- **label** – str

Return type Double2DParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type double with 2 dimensions.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

NatronEngine.UserParamHolder.**createDouble3DParam**(*name*, *label*)

Parameters

- **name** – str
- **label** – str

Return type Double3DParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type double with 3 dimensions.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createDoubleParam(name, label)`

Parameters

- **name** – str
- **label** – str

Return type DoubleParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type double with single dimension. A double is similar to a floating point value.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createFileParam(name, label)`

Parameters

- **name** – str
- **label** – str

Return type FileParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type double with 2 dimensions.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createGroupParam(name, label)`

Parameters

- **name** – str
- **label** – str

Return type GroupParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type group. It can contain other children parameters and can be expanded or folded.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createInt2DParam(name, label)`

Parameters

- **name** – str
- **label** – str

Return type Int2DParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type integer with 2 dimensions.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createInt3DParam(name, label)`

Parameters

- **name** – str
- **label** – str

Return type Int3DParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type integer with 3 dimensions.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createIntParam(name, label)`

Parameters

- **name** – str
- **label** – str

Return type IntParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type integer with a single dimension.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createOutputFileParam(name, label)`

Parameters

- **name** – str
- **label** – str

Return type OutputFileParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type string dedicated to specify paths to output files.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createPageParam(name, label)`

Parameters

- **name** – str
- **label** – str

Return type PageParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type page. A page is a tab within the settings panel of the node.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createParametricParam(name, label, nbCurves)`

Parameters

- **name** – str
- **label** – str
- **nbCurves** – int

Return type ParametricParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type parametric. A parametric parameter is what can be found in the ColorLookup node or in the Ranges tab of the ColorCorrect node.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createPathParam(name, label)`

Parameters

- **name** – str
- **label** – str

Return type PathParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type string. This parameter is dedicated to specify path to single or multiple directories.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

`NatronEngine.UserParamHolder.createStringParam(name, label)`

Parameters

- **name** – str
- **label** – str

Return type StringParam

Creates a new user *parameter* with the given *name* and *label*. See [here](#) for an explanation of the difference between the *name* and *label*. This function will return a new parameter of type string.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

NatronEngine.UserParamHolder.**removeParam**(*param*)

Parameters *param* – Param

Return type bool

Removes the given *param* from the parameters of this Effect. This function works only if *param* is a user parameter and does nothing otherwise. This function returns True upon success and False otherwise.

Warning: After calling this function you should call `refreshUserParamsGUI()` to refresh the user interface. The refreshing is done in a separate function because it may be expensive and thus allows you to make multiple changes to user parameters at once while keeping the user interface responsive.

NatronEngine.UserParamHolder.**refreshUserParamsGUI**()

This function must be called after new user parameter were created or removed. This will re-create the user interface for the parameters and can be expensive.

3.1.2 NatronGui

Detailed Description

Here are listed all classes being part of NatronEngine module. This module is loaded by Natron natively in GUI mode only. In that case, access is granted to these classes in your scripts without importing anything. Scripts that want to operate both in command line background mode and in GUI mode should poll the `NatronEngine.PyCoreApplication.isBackground()` function on the **natron** object before calling functions dependent on the module *NatronGui*. E.g:

```
if not NatronEngine.natron.isBackground():
    # do GUI only stuff here
```

GuiApp

Inherits App

Synopsis

This class is used for GUI application instances. See [detailed](#) description...

Functions

- `def createModalDialog ()`
- `def getFilenameDialog (filters[, location=None])`
- `def getSequenceDialog (filters[, location=None])`
- `def getDirectoryDialog ([location=None])`
- `def getRGBColorDialog ()`
- `def getTabWidget (scriptName)`
- `def getSelectedNodes ([group=None])`
- `def getViewer (scriptName)`
- `def getUserPanel (scriptName)`
- `def moveTab (tabScriptName,pane)`
- `def saveFilenameDialog (filters[, location=None])`
- `def saveSequenceDialog (filters[, location=None])`
- `def selectNode (node,clearPreviousSelection)`
- `def deselectNode (node)`
- `def setSelection (nodes)`
- `def selectAllNodes ([group=None])`
- `def clearSelection ([group=None])`
- `def registerPythonPanel (panel,pythonFunction)`
- `def unregisterPythonPanel (panel)`
- `def renderBlocking (effect,firstFrame,lastFrame,frameStep)`
- `def renderBlocking (tasks)`

Detailed Description

See App for the documentation of base fonctionnalities of this class.

To create a new modal dialog , use the `createModalDialog()` function.

Several functions are made available to pop dialogs to ask the user for filename(s) or colors. See `getFilenameDialog(filters, location)` and `getRGBColorDialog()`.

To create a new custom python panel, there are several ways to do it:

- Sub-class the `PyPanel` class and make your own PySide widget
- Create a `PyPanel` object and add controls using user parameters (as done for modal dialogs)

Once created, you can register the panel in the project so that it gets saved into the layout by calling `registerPythonPanel (panel, pythonFunction)`

Member functions description

`NatronGui.GuiApp.createModalDialog()`

Return type `PyModalDialog`

Creates a modal dialog : the control will not be returned to the user until the dialog is not closed. Once the dialog is created, you can enrich it with parameters or even raw PySide Qt widgets. To show the dialog call the function `exec()` on the dialog.

```
NatronGui.GuiApp.getFileDialog(filters[, location=None])
```

Parameters

- **filters** – sequence
- **location** – str

Return type str

Opens-up a file dialog to ask the user for a single filename which already exists.

filters is a list of file extensions that should be displayed in the file dialog.

location is the initial location the dialog should display, unless it is empty in which case the dialog will display the last location that was opened previously by a dialog.

```
NatronGui.GuiApp.getSequenceDialog(filters[, location=None])
```

Parameters

- **filters** – sequence
- **location** – str

Return type str

Same as `getFileDialog(filters, location)` but the dialog will accept sequence of files.

```
NatronGui.GuiApp.getDirectoryDialog([location=None])
```

Parameters location – str**Return type** str

Same as `getFileDialog(filters, location)` but the dialog will only accept directories as a result.

```
NatronGui.GuiApp.saveFileDialog(filters[, location=None])
```

Parameters

- **filters** – sequence
- **location** – str

Return type str

Opens-up a file dialog to ask the user for a single filename. If the file already exists, the user will be warned about potential overriding of the file.

filters is a list of file extensions that should be displayed in the file dialog.

location is the initial location the dialog should display, unless it is empty in which case the dialog will display the last location that was opened previously by a dialog.

```
NatronGui.GuiApp.saveSequenceDialog(filters[, location=None])
```

Parameters

- **filters** – sequence
- **location** – str

Return type str

Same as `saveFileDialog(filters, location)` but the dialog will accept sequence of files.

```
NatronGui.GuiApp.getRGBColorDialog()
```

Return type *ColorTuple*

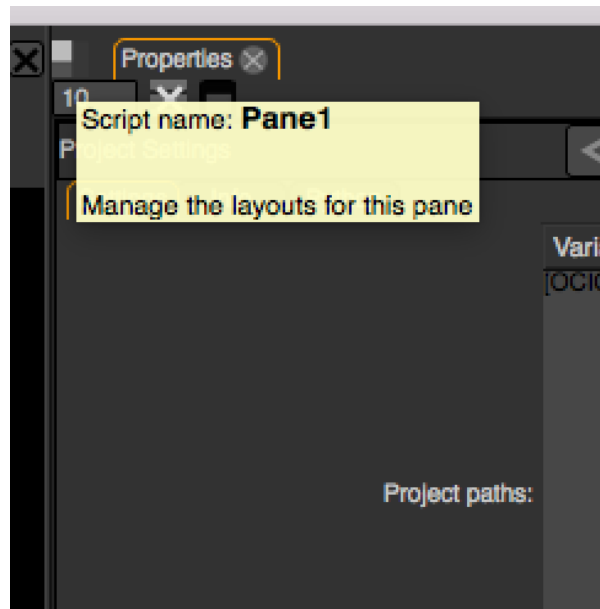
Opens-up a color dialog to ask the user for an RGB color.

`NatronGui.GuiApp.getTabWidget (scriptName)`

Parameters `scriptName` – str

Return type `PyTabWidget`

Returns the tab-widget with the given *scriptName*. The *scriptName* of a tab-widget can be found in the user interface when hovering with the mouse the “Manage layout” button (in the top left-hand corner of the pane)



`NatronGui.GuiApp.moveTab (tabScriptName, pane)`

Parameters

- `tabScriptName` – str
- `pane` – `PyTabWidget`

Return type `bool`

Attempts to move the tab with the given *tabScriptName* into the given *pane* and make it current in the *pane*. This function returns True upon success or False otherwise.

Warning: Moving tabs that are not registered to the application via `registerPythonPanel (panel, pythonFunction)` will not work.

`NatronGui.GuiApp.registerPythonPanel (panel, pythonFunction)`

Parameters

- `panel` – `PyPanel`
- `scriptName` – str

Registers the *given* panel into the project. When registered, the panel will be saved into the layout for the current project and a new entry in the “Panels” sub-menu of the “Manage layouts” button (in the top left-hand corner of each tab widget) will appear for this panel. *pythonFunction* is the name of a python-defined function that takes no argument that should be used to re-create the panel.

`NatronGui.GuiApp.unregisterPythonPanel (panel)`

Parameters `panel` – `PyPanel`

Unregisters a previously registered panel.

NatronGui.GuiApp.**getSelectedNodes** (*[group = None]*)

Return type sequence

Returns a sequence of *nodes* currently selected in the given *group*. You can pass the *app* object to get the top-level NodeGraph. If passing None, the last user-selected NodeGraph will be used:

```
topLevelSelection = app.getSelectedNodes()

group = app.createNode("fr.inria.built-in.Group")

groupSelection = app.getSelectedNodes(group)
```

NatronGui.GuiApp.**getViewer** (*scriptName*)

Parameters *scriptName* – str

Returns the viewer with the given *scriptName* if one can be found.

NatronGui.GuiApp.**getUserPanel** (*scriptName*)

Parameters *scriptName* – str

Returns a user panel matching the given *scriptName* if there is any.

NatronGui.GuiApp.**selectNode** (*node, clearPreviousSelection*)

Parameters

- **node** – Effect
- **clearPreviousSelection** – bool

Select the given *node* in its containing nodegraph. If *clearPreviousSelection* is set to *True*, all the current selection will be wiped prior to selecting the *node*; otherwise the *node* will just be added to the selection.

NatronGui.GuiApp.**deselectNode** (*node*)

Parameters *node* – Effect

Deselect the given *node* in its containing nodegraph. If the *node* is not selected, this function does nothing.

NatronGui.GuiApp.**setSelection** (*nodes*)

Parameters *nodes* – sequence

Set all the given *nodes* selected in the nodegraph containing them and wipe any current selection.

Note: All nodes must be part of the same nodegraph (group), otherwise this function will fail.

NatronGui.GuiApp.**selectAllNodes** (*[group=None]*)

Parameters *group* – Group

Select all nodes in the given *group*. You can pass the *app* object to get the top-level NodeGraph. If passing None, the last user-selected NodeGraph will be used.

NatronGui.GuiApp.**clearSelection** (*[group=None]*)

Wipe any current selection in the given *group*. You can pass the *app* object to get the top-level NodeGraph. If passing None, the last user-selected NodeGraph will be used.

NatronGui.GuiApp.**renderBlocking** (*effect, firstFrame, lastFrame, frameStep*)

Parameters

- **effect** – Effect
- **firstFrame** – int
- **lastFrame** – int

- **frameStep** – int

Starts rendering the given *effect* on the frame-range defined by [*firstFrame*,**lastFrame**]. The *frameStep* parameter indicates how many frames the timeline should step after rendering each frame. The value must be greater or equal to 1. The *frameStep* parameter is optional and if not given will default to the value of the **Frame Increment** parameter in the Write node.

For instance:

```
render(effect,1,10,2)
```

Would render the frames 1,3,5,7,9

This is a blocking function. A blocking render means that this function returns only when the render finishes (from failure or success).

This function should only be used to render with a Write node or DiskCache node.

NatronGui.GuiApp.**renderBlocking** (*tasks*)

Parameters *tasks* – sequence

This function takes a sequence of tuples of the form (*effect*,*firstFrame*,*lastFrame*[*frameStep*]) The *frameStep* is optional in the tuple and if not set will default to the value of the **Frame Increment** parameter in the Write node.

This is an overloaded function. Same as `render(effect, firstFrame, lastFrame, frameStep)` but all *tasks* will be rendered concurrently.

This function is called when rendering a script in background mode with multiple writers.

This is a blocking call.

PyGuiApplication

Inherits PyCoreApplication

Synopsis

See PyCoreApplication for a detailed explanation of the purpose of this object. This class is only used when Natron is run in GUI mode (with user interface). It gives you access to more GUI functionalities via the [GuiApp](#) class.

Functions

- def *addMenuCommand* (grouping,function)
- def *addMenuCommand* (grouping,function,key,modifiers)
- def *getGuiInstance* (idx)
- def *informationDialog* (title,message)
- def *warningDialog* (title,message)
- def *errorDialog* (title,message)
- def *questionDialog* (title,question)

Member functions description

class `NatronGui.PyGuiApplication`

See `PyCoreApplication()`

`NatronGui.PyGuiApplication.addMenuCommand(grouping, function)`

Parameters

- **grouping** – str
- **function** – str

Adds a new menu entry in the menubar of Natron. This should be used **exclusively** in the *initGui.py* initialisation script.

The *grouping* is a string indicating a specific menu entry where each submenu is separated from its parent menu with a `/`:

```
File/Do something special
MyStudio/Scripts/Our special trick
```

The *function* is the name of a python defined function.

Warning: If called anywhere but from the *initGui.py* script, this function will fail to dynamically add a new menu entry.

Example:

```
def printLala():
    print("Lala")

natron.addMenuCommand("Inria/Scripts/Print lala script", "printLala")
```

This registers in the menu *Inria->Scripts* an entry named *Print lala script* which will print *Lala* to the Script Editor when triggered.

`NatronGui.PyGuiApplication.addMenuCommand(grouping, function, key, modifiers)`

Parameters

- **grouping** – str
- **function** – str
- **key** – `PySide.QtCore.Qt.Key`
- **modifiers** – `PySide.QtCore.Qt.KeyboardModifiers`

Same as `addMenuCommand(grouping, function)` excepts that it accepts a default shortcut for the action. See `PySide` documentation for possible keys and modifiers.

The user will always be able to modify the shortcut from the built-in shortcut editor of Natron anyway.

`NatronGui.PyGuiApplication.getGuiInstance(idx)`

Parameters `idx` – int

Return type `GuiApp`

Same as `getInstance(idx)` but returns instead an instance of a GUI project.

Basically you should never call this function as Natron pre-declares all opened projects with the following variables: *app1* for the first opened project, *app2* for the second, and so on...

`NatronGui.PyGuiApplication.informationDialog(title, message)`

Parameters

- **title** – str
- **message** – str

Shows a modal information dialog to the user with the given window *title* and containing the given *message*.

`NatronGui.PyGuiApplication.warningDialog (title, message)`

Parameters

- **title** – str
- **message** – str

Shows a modal warning dialog to the user with the given window *title* and containing the given *message*.

`NatronGui.PyGuiApplication.errorDialog (title, message)`

Parameters

- **title** – str
- **message** – str

Shows a modal error dialog to the user with the given window *title* and containing the given *message*.

`NatronGui.PyGuiApplication.questionDialog (title, message)`

Parameters

- **title** – str
- **message** – str

Return type `NatronEngine.StandardButtonEnum`

Shows a modal question dialog to the user with the given window *title* and containing the given *message*. The dialog will be a “Yes” “No” dialog, and you can compare the result to the `NatronEngine.StandardButtonEnum` members.

PyModalDialog

Inherits `UserParamHolder QDialog`

Synopsis

A modal dialog to ask informations to the user or to warn about something. See [detailed](#) description...

Functions

- def `addWidget` (widget)
- def `getParam` (scriptName)
- def `insertWidget` (index, widget)
- def `setParamChangedCallback` (callback)

Detailed Description

The modal dialog is a way to ask the user for data or to inform him/her about something going on. A modal window means that control will not be returned to the user (i.e. no event will be processed) until the user closed the dialog.

If you are looking for a simple way to just ask a question or report an error, warning or even just a miscellaneous information, use the `informationDialog(title,message)` function.

To create a new *PyModalDialog*, just use the `createModalDialog()` function, e.g.:

In the Script Editor

```
dialog = app1.createModalDialog()
```

To show the dialog to the user, use the `exec_()` function inherited from `QDialog`

```
dialog.exec_()
```

Note that once `exec_()` is called, no instruction will be executed until the user closed the dialog.

The modal dialog always has *OK* and *Cancel* buttons. To query which button the user pressed, inspect the return value of the `exec_()` call:

```
if dialog.exec_():
    #The user pressed OK
    ...
else:
    #The user pressed Cancel or Escape
```

Adding user parameters:

You can start adding user parameters using all the `createXParam` functions inherited from the `UserParamHolder` class.

Once all your parameters are created, create the GUI for them using the `refreshUserParamsGUI()` function:

```
myInteger = dialog.createIntParam("myInt", "This is an integer very important")
myInteger.setAnimationEnabled(False)
myInteger.setAddNewLine(False)

#Create a boolean on the same line
myBoolean = dialog.createBooleanParam("myBool", "Yet another important boolean")

dialog.refreshUserParamsGUI()

dialog.exec_()
```

You can then retrieve the value of a parameter once the dialog is finished using the `getParam(scriptName)` function:

```
if dialog.exec_():
    intValue = dialog.getParam("myInt").get()
    boolValue = dialog.getParam("myBool").get()
```

Warning: Unlike the *Effect* class, parameters on modal dialogs are not automatically declared by Natron, which means you cannot do stuff like `dialog.intValue`

Member functions description

`NatronGui.PyModalDialog.addWidget(widget)`

Parameters `widget` – `PySide.QtGui.QWidget`

Append a `QWidget` inherited `widget` at the bottom of the dialog. This allows to add custom GUI created directly using `PySide` that will be inserted **after** any custom parameter.

`NatronGui.PyModalDialog.getParam(scriptName)`

Parameters `scriptName` – `str`

Return type `Param`

Returns the user parameter with the given `scriptName` if it exists or `None` otherwise.

`NatronGui.PyModalDialog.insertWidget(index, widget)`

Parameters

- `index` – `int`
- `widget` – `PySide.QtGui.QWidget`

Inserts a `QWidget` inherited `widget` at the given `index` of the layout in the dialog. This allows to add custom GUI created directly using `PySide`. The widget will always be inserted **after** any user parameter.

`NatronGui.PyModalDialog.setParamChangedCallback(callback)`

Parameters `callback` – `str`

Registers the given Python `callback` to be called whenever a user parameter changed. The `callback` should be the name of a Python defined function (taking no parameter).

The variable `paramName` will be declared upon calling the callback, referencing the script-name of the parameter that just changed. Example:

```
def myCallback():
    if paramName == "myInt":
        intValue = thisParam.get()
        if intValue > 0:
            myBoolean.setVisible(False)

dialog.setParamChangedCallback("myCallback")
```

PyPanel

Inherits `PySide.QtGui.QWidget` `NatronEngine.UserParamHolder`

Synopsis

A custom `PySide` pane that can be docked into `PyTabWidget`. See [detailed](#) description...

Functions

- def `PyPanel` (scriptName, label, useUserParameters, app)
- def `addWidget` (widget)
- def `getPanelLabel` ()
- def `getPanelScriptName` ()
- def `getParam` (scriptName)

- `def getParams ()`
- `def insertWidget (index,widget)`
- `def onUserDataChanged ()`
- `def setParamChangedCallback (callback)`
- `def save ()`
- `def setPanelLabel (label)`
- `def restore (data)`

Detailed Description

The *PyPanel* class can be used to implement custom PySide widgets that can then be inserted as tabs into *tab-widgets*.

There are 2 possible usage of this class:

- Sub-class it and create your own GUI using PySide
- Use the API proposed by *PyPanel* to add custom user parameters as done for *PyModalDialog*.

Sub-classing:

When sub-classing the *PyPanel* class, you should specify when calling the base class constructor that you do not want to use user parameters, as this might conflict with the layout that you will use:

```
class MyPanel (NatronGui.PyPanel) :
    def __init__(scriptName, label, app) :
        NatronGui.PyPanel.__init__(scriptName, label, False, app)
    ...
```

You're then free to use all features proposed by PySide in your class, including *signal/slots* See the following *example*.

Using the PyPanel API:

You can start adding user parameters using all the `createXParam` functions inherited from the `UserParamHolder` class.

Once all your parameters are created, create the GUI for them using the `refreshUserParamsGUI()` function:

```
panel = NatronGui.PyPanel("fr.inria.mypanel", "My Panel", True, app)
myInteger = panel.createIntParam("myInt", "This is an integer very important")
myInteger.setAnimationEnabled(False)
myInteger.setAddNewLine(False)

#Create a boolean on the same line
myBoolean = panel.createBooleanParam("myBool", "Yet another important boolean")

panel.refreshUserParamsGUI()
```

You can then retrieve the value of a parameter at any time using the `getParam(scriptName)` function:

```
intValue = panel.getParam("myInt").get()
boolValue = panel.getParam("myBool").get()
```

Warning: Unlike the *Effect* class, parameters on panels are not automatically declared by Natron, which means you cannot do stuff like *panel.intValue*

You can get notified when a parameter's value changed, by setting a callback using the *setParamChangedCallback(callback)* function that takes the name of a Python-defined function in parameters. The variable **thisParam** will be declared prior to calling the callback, referencing the parameter which just had its value changed.

Managing the panel:

Once created, you must add your panel to a *PyTabWidget* so it can be visible. Use the *getTabWidget(scriptName)* function to get a particular pane and then use the *appendTab(tab)* function to add this panel to the pane.

Warning: Note that the lifetime of the widget will be by default the same as the project's GUI because *PyPanel* is *auto-declared* by Natron.

```
panel = NatronGui.PyPanel("fr.inria.mypanel", "My Panel", True, app)
...
...
pane = app.getTabWidget("panel")
pane.appendTab(panel)
app.mypanel = panel
```

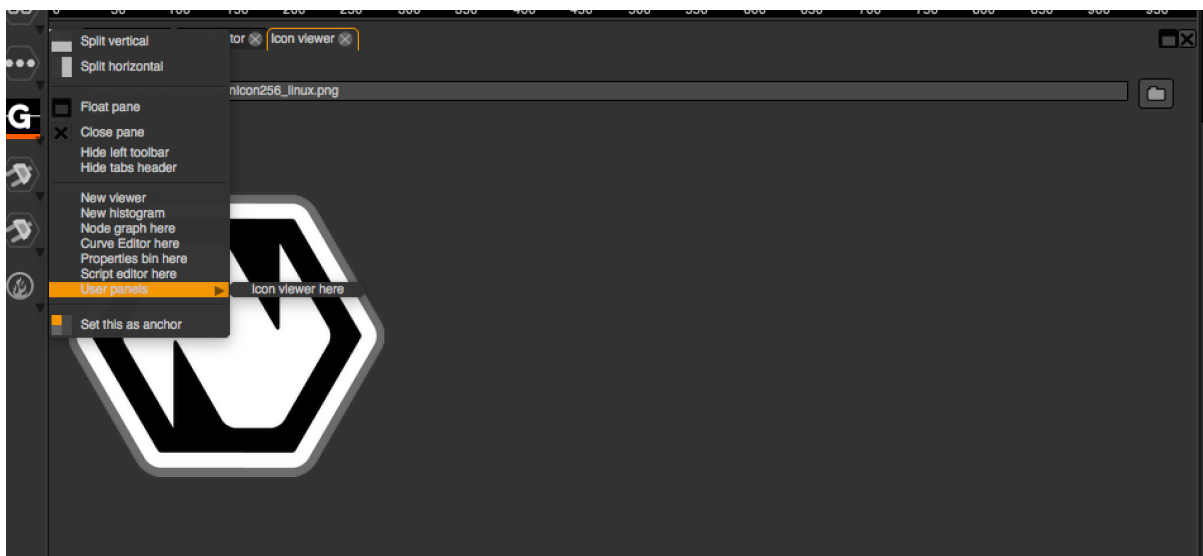
If you want the panel to persist in the project so that it gets recreated and placed at its original position when the user loads the project, you must use the *registerPythonPanel(panel, function)* function.

Note that the *function* parameter is the **name** of a Python-defined function that takes no parameter used to create the widget, e.g.:

```
def createMyPanel(): panel = NatronGui.PyPanel("MyPanel", True, app) ... #Make it live after the
    scope of the function app.mypanel = panel

app.registerPythonPanel(app.mypanel, "createMyPanel")
```

This function will also add a custom menu entry to the "Manage layout" button (located in the top-left hand corner of every pane) which the user can trigger to move the custom pane on the selected pane.



Saving and restoring state:

When the panel is registered in the project using the `registerPythonPanel(panel, function)` function, you may want to also save the state of your widgets and/or special values.

To do so, you must sub-class `PyPanel` and implement the `save()` and `restore(data)` functions.

Note: User parameters, if used, will be automatically saved and restored, you don't have to save it yourself. Hence if the panel is only composed of user parameters that you want to save, you do not need to sub-class `PyPanel` as it will be done automatically for you.

The function `save()` should return a string containing the serialization of your custom data.

The function `restore(data)` will be called upon loading of a project containing an instance of your panel. You should then restore the state of the panel from your custom serialized data.

Note that the auto-save of Natron occurs in a separate thread and for this reason it cannot call directly your `save()` function because it might create a race condition if the user is actively modifying the user interface using the main-thread.

To overcome this, Natron has an hidden thread-safe way to recover the data you have serialized using the `save()` function. The downside is that you have to call the `onUserDataChanged()` function whenever a value that you want to be persistent has changed (unless this is a user parameter in which case you do not need to call it).

Warning: If you do not call `onUserDataChanged()`, the `save()` function will never be called, and the data never serialized.

Member functions description

`NatronGui.PyPanel.PyPanel(label, useUserParameters, app)`

Parameters

- **label** – str
- **useUserParameters** – bool
- **app** – GuiApp

Make a new `PyPanel` with the given `label` that will be used to display in the tab header. If `useUserParameters` is True then user parameters support will be activated, attempting to modify the underlying layout in these circumstances will result in undefined behaviour.

`NatronGui.PyPanel.addWidget(widget)`

Parameters `widget` – PySide.QtGui.QWidget

Append a `QWidget` inherited `widget` at the bottom of the dialog. This allows to add custom GUI created directly using `PySide` that will be inserted **after** any custom parameter.

Warning: This function should be used exclusively when the widget was created using `useUserParameters = True`

`NatronGui.PyPanel.getParam(scriptName)`

Parameters `scriptName` – str

Return type Param

Returns the user parameter with the given `scriptName` if it exists or `None` otherwise.

Warning: This function should be used exclusively when the widget was created using *useUserParameters = True*

`NatronGui.PyPanel.getParams()`

Return type `sequence`

Returns all the user parameters used by the panel.

Warning: This function should be used exclusively when the widget was created using *useUserParameters = True*

`NatronGui.PyPanel.insertWidget(index, widget)`

Parameters

- **index** – `int`
- **widget** – `PySide.QtGui.QWidget`

Inserts a `QWidget` inherited *widget* at the given *index* of the layout in the dialog. This allows to add custom GUI created directly using `PySide`. The widget will always be inserted **after** any user parameter.

Warning: This function should be used exclusively when the widget was created using *useUserParameters = True*

`NatronGui.PyPanel.setParamChangedCallback(callback)`

Parameters **callback** – `str`

Registers the given Python *callback* to be called whenever a user parameter changed. The *callback* should be the name of a Python defined function (taking no parameter).

The variable **paramName** will be declared upon calling the callback, referencing the script name of the parameter that just changed. Example:

```
def myCallback():
    if paramName == "myInt":
        intValue = thisParam.get()
        if intValue > 0:
            myBoolean.setVisible(False)

panel.setParamChangedCallback("myCallback")
```

Warning: This function should be used exclusively when the widget was created using *useUserParameters = True*

`NatronGui.PyPanel.setPanelLabel(label)`

Parameters **callback** – `str`

Set the label of the panel as it will be displayed on the tab header of the *PyTabWidget*. This name should be unique.

`NatronGui.PyPanel.getPanelLabel()`

Return type `str`

Get the label of the panel as displayed on the tab header of the *PyTabWidget*.

`NatronGui.PyPanel.getPanelScriptName()`

Return type `str`

Get the script-name of the panel as used internally. This is a unique string identifying the tab in Natron.

`NatronGui.PyPanel.onUserDataChanged()`

Callback to be called whenever a parameter/value (that is not a user parameter) that you want to be saved has changed.

Warning: If you do not call `onUserDataChanged()`, the `save()` `NatronGui.PyPanel.save()` function will never be called, and the data never serialized.

Warning: This function should be used exclusively when the widget was created using `useUserParameters = True`

`NatronGui.PyPanel.save()`

Return type `str`

Warning: You should overload this function in a derived class. The base version does nothing.

Note: User parameters, if used, will be automatically saved and restored, you don't have to save it yourself. Hence if the panel is only composed of user parameters that you want to save, you do not need to sub-class `PyPanel` as it will be done automatically for you.

Returns a string with the serialization of your custom data you need to be persistent.

`NatronGui.PyPanel.restore(data)`

Parameters `data` – `str`

Warning: You should overload this function in a derived class. The base version does nothing.

This function should restore the state of your custom *PyPanel* using the custom *data* that you serialized. The *data* are exactly the return value that was returned from the `save()` function.

PyTabWidget

Synopsis

A `PyTabWidget` is one of the GUI pane onto which the user can dock tabs such as the `NodeGraph`, `CurveEditor`. . . See *detailed* description. . .

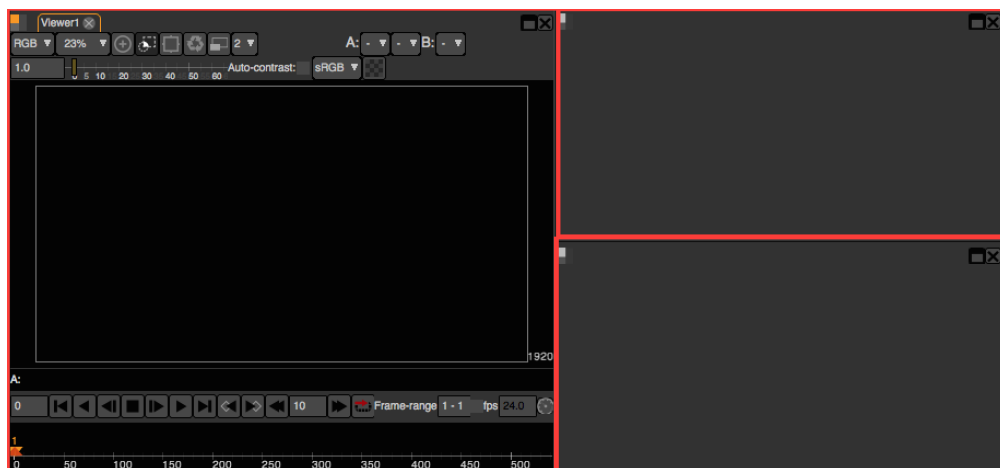
Functions

- def `appendTab` (tab)
- def `closeCurrentTab` ()
- def `closeTab` (index)
- def `closePane` ()
- def `count` ()

- `def currentWidget ()`
- `def floatCurrentTab ()`
- `def floatPane ()`
- `def getCurrentIndex ()`
- `def getScriptName ()`
- `def getTabLabel (index)`
- `def insertTab (index,tab)`
- `def removeTab (tab)`
- `def removeTab (index)`
- `def setCurrentIndex (index)`
- `def setNextTabCurrent ()`
- `def splitHorizontally ()`
- `def splitVertically ()`

Detailed Description

The *PyTabWidget* class is used to represent panes visible in the user interface:



On the screenshot above, each *PyTabWidget* is surrounded by a red box.

You cannot construct tab widgets on your own, you must call one of the *splitVertically()* or *splitHorizontally()* functions to make a new one based on another existing ones.

By default the GUI of Natron cannot have less than 1 tab widget active, hence you can always split it to make new panes.

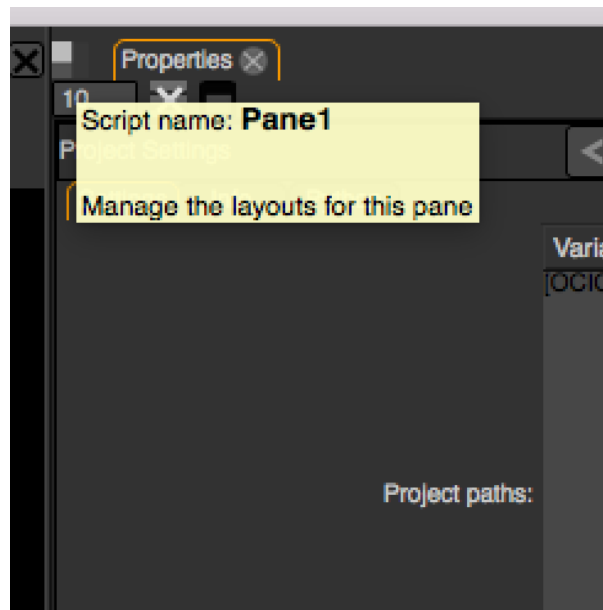
To retrieve an existing *PyTabWidget* you can call the *getTabWidget (scriptName)* function of *GuiApp*.

```
panel = app.getTabWidget("Panel")
```

Note that the script-name of a pane can be seen on the graphical user interface by hovering the mouse over the “Manage layout” button (in the top left hand corner of a pane).

Managing tabs

To insert tabs in the TabWidget you can call either *appendTab (tab)* or *insertTab (index, tab)*.



Warning: Note that to insert a tab, it must be first removed from the tab into which it was.

To remove a tab, use the function `removeTab(tab)` on the parent `PyTabWidget`

For convenience to move tabs around, there is a `moveTab(tab, pane)` function in `GuiApp`.

The function `closeTab(index)` can be used to close permanently a tab, effectively destroying it.

To change the current tab, you can use one of the following functions:

- `setCurrentIndex(index)<NatronGui.PyTabWidget.setCurrentIndex>`
- `setNextTabCurrent()<NatronGui.PyTabWidget.setNextTabCurrent>`

To float the current tab into a new floating window, use the `floatCurrentTab()<NatronGui.PyTabWidget.floatCurrentTab>` function.

Managing the pane

To close the pane permanently, use the `closePane()<NatronGui.PyTabWidget.closePane>` function. To float the pane into a new floating window with all its tabs, use the `floatPane()` function.

Member functions description

`NatronGui.PyTabWidget.appendTab(tab)`

Parameters `tab` – `PySide.QtGui.QWidget`

Appends a new tab to the tab widget and makes it current.

`NatronGui.PyTabWidget.closeCurrentTab()`

Closes the current tab, effectively destroying it.

`NatronGui.PyTabWidget.closeTab(index)`

Closes the tab at the given `index`, effectively destroying it.

`NatronGui.PyTabWidget.closePane()`

Closes this pane, effectively destroying it. Note that all tabs will not be destroyed but instead moved to another existing pane.

Warning: If this pane is the last one on the GUI, this function does nothing.

`NatronGui.PyTabWidget.count()`

Return type `int`

Returns the number of tabs in this pane.

`NatronGui.PyTabWidget.currentWidget()`

Return type `PySide.QtGui.QWidget`

Returns the current active tab.

`NatronGui.PyTabWidget.floatCurrentTab()`

Make a new floating window with a single pane and moves the current tab of this pane to the new pane of the floating window.

`NatronGui.PyTabWidget.floatPane()`

Make a new floating window and moves this pane to the new window (including all tabs).

`NatronGui.PyTabWidget.getCurrentIndex()`

Return type `int`

Returns the index of the current tab. This is 0-based (starting from the left).

`NatronGui.PyTabWidget.getScriptName()`

Return type `str`

Returns the script-name of the pane, as used by the `getTabWidget(scriptName)` function.

`NatronGui.PyTabWidget.getTabLabel(index)`

Parameters `index` – `int`

Return type `str`

Returns the name of the tab at the given `index` if it exists or an empty string otherwise.

`NatronGui.PyTabWidget.insertTab(index, tab)`

Parameters

- `tab` – `PySide.QtGui.QWidget`
- `index` – `int`

Inserts the given `tab` at the given `index` in this tab-widget.

`NatronGui.PyTabWidget.removeTab(tab)`

Parameters `tab` – `PySide.QtGui.QWidget`

Removes the given `tab` from this pane if it is found. Note that this function does not destroy the `tab`, unlike `closeTab(index)`.

This is used internally by `moveTab(tab, pane)`.

`NatronGui.PyTabWidget.removeTab(index)`

Parameters `index` – `int`

Same as `removeTab(tab)` but the `index` of a tab is given instead.

`NatronGui.PyTabWidget.setCurrentIndex(index)`

Parameters `index` – int

Makes the tab at the given *index* (0-based) the current one (if the index is valid).

`NatronGui.PyTabWidget.setNextTabCurrent()`

Set the tab at `getCurrentIndex() + 1` the current one. This functions cycles back to the first tab once the last tab is reached.

`NatronGui.PyTabWidget.splitHorizontally()`

Return type `PyTabWidget`

Splits this pane into 2 horizontally-separated panes. The new pane will be returned.

`NatronGui.PyTabWidget.splitVertically()`

Return type `PyTabWidget`

Splits this pane into 2 vertically-separated panes. The new pane will be returned.

PyViewer

Synopsis

A PyViewer is a wrapper around a Natron Viewer. See [detailed](#) description...

Functions

- `def seek (frame)`
- `def getCurrentFrame ()`
- `def startForward ()`
- `def startBackward ()`
- `def pause ()`
- `def redraw ()`
- `def renderCurrentFrame ([useCache=True])`
- `def setFrameRange (firstFrame,lastFrame)`
- `def getFrameRange ()`
- `def setPlaybackMode (mode)`
- `def getPlaybackMode ()`
- `def getCompositingOperator ()`
- `def setCompositingOperator (operator)`
- `def getAInput ()`
- `def setAInput (index)`
- `def getBInput ()`
- `def setBInput (index)`
- `def setChannels (channels)`
- `def getChannels ()`
- `def setProxyModeEnabled (enabled)`
- `def isProxyModeEnabled ()`

- `def setProxyIndex (index)`
- `def getProxyIndex ()`
- `def setCurrentView (viewIndex)`
- `def getCurrentView (channels)`

Detailed Description

This class is a wrapper around a Natron Viewer, exposing all functionalities available as user interaction to the Python API.

To get a *PyViewer*, use the `getViewer (scriptName)` function, passing it the *script-name* of a viewer node.

Member functions description

`NatronGui.PyTabWidget.seek (frame)`

Parameters `frame` – int

Seek the timeline to a particular frame. All other viewers in the project will be synchronized to that frame.

`NatronGui.PyTabWidget.getCurrentFrame ()`

Return type int

Returns the current frame on the timeline.

`NatronGui.PyTabWidget.startForward ()`

Starts playback, playing the video normally.

`NatronGui.PyTabWidget.startBackward ()`

Starts playback backward, like a rewind.

`NatronGui.PyTabWidget.pause ()`

Pauses the viewer if the playback is ongoing.

`NatronGui.PyTabWidget.redraw ()`

Redraws the OpenGL widget without actually re-rendering the internal image. This is provided for convenience as sometimes the viewer might need refreshing for OpenGL overlays.

`NatronGui.PyTabWidget.renderCurrentFrame ([useCache=True])`

Parameters `useCache` – bool

Renders the current frame on the timeline. If *useCache* is False, the cache will not be used and the frame will be completely re-rendered.

`NatronGui.PyTabWidget.setFrameRange (firstFrame, lastFrame)`

Parameters

- **firstFrame** – int
- **lastFrame** – int

Set the frame range on the Viewer to be [*firstFrame*, *lastFrame*] (included).

`NatronGui.PyTabWidget.getFrameRange ()`

Return type Tuple

Returns a 2-dimensional tuple of int containing [*firstFrame*, *lastFrame*].

`NatronGui.PyTabWidget.setPlaybackMode (mode)`

Parameters *mode* – NatronEngine.Natron.PlaybackModeEnum

Set the playback mode for the Viewer, it can be either **bouncing**, **looping** or **playing once**.

NatronGui.PyTabWidget.**getPlaybackMode**()

Return type NatronEngine.Natron.PlaybackModeEnum

Returns the playback mode for this Viewer.

NatronGui.PyTabWidget.**getCompositingOperator**()

Return type NatronEngine.Natron.ViewerCompositingOperatorEnum

Returns the current compositing operator applied by the Viewer.

NatronGui.PyTabWidget.**setCompositingOperator**(*operator*)

Parameters *operator* – NatronEngine.Natron.ViewerCompositingOperatorEnum

Set the current compositing operator applied by the Viewer.

NatronGui.PyTabWidget.**getAInput**()

Return type int

Returns the **index** of the input (the same index used by `getInput (index)`) used by the **A** choice of the Viewer.

NatronGui.PyTabWidget.**setAInput**(*index*)

Parameters *index* – int

Set the **index** of the input (the same index used by `getInput (index)`) used by the **A** choice of the Viewer.

NatronGui.PyTabWidget.**getBInput**()

Return type int

Returns the **index** of the input (the same index used by `getInput (index)`) used by the **B** choice of the Viewer.

NatronGui.PyTabWidget.**setBInput**(*index*)

Parameters *index* – int

Set the **index** of the input (the same index used by `getInput (index)`) used by the **B** choice of the Viewer.

NatronGui.PyTabWidget.**setChannels**(*channels*)

Parameters *channels* – NatronEngine.Natron.DisplayChannelsEnum

Set the *channels* to be displayed on the Viewer.

NatronGui.PyTabWidget.**getChannels**()

Return type NatronEngine.Natron.DisplayChannelsEnum

Returns the current *channels* displayed on the Viewer.

NatronGui.PyTabWidget.**setProxyModeEnabled**(*enabled*)

Parameters *enabled* – bool

Set the proxy mode *enabled*.

NatronGui.PyTabWidget.**isProxyModeEnabled**(*enabled*)

Return type bool

Returns whether the proxy mode is *enabled*.

NatronGui.PyTabWidget.**setProxyIndex**(*index*)

Parameters *index* – int

Set the *index* of the proxy to use. This is the index in the combobox on the graphical user interface, e.g. *index* = 0 will be 2

```
NatronGui.PyTabWidget.getProxyIndex()
```

Return type `int`

Returns the *index* of the proxy in use. This is the index in the combobox on the graphical user interface, e.g. *index* = 0 will be 2

```
NatronGui.PyTabWidget.setCurrentView(viewIndex)
```

Parameters `viewIndex` – `int`

Set the view to display the given *viewIndex*. This is the index in the multi-view combobox visible when the number of views in the project settings has been set to a value greater than 1.

```
NatronGui.PyTabWidget.getCurrentView()
```

Parameters `viewIndex` – `int`

Returns the currently displayed view index. This is the index in the multi-view combobox visible when the number of views in the project settings has been set to a value greater than 1.

3.2 Introduction

This section covers the basic principles for scripting in Python in Natron.

3.2.1 Natron plug-in paths

When looking for startup scripts or Python group plug-ins, Natron will look into the following search paths in order:

- The bundled plug-ins path. There are 2 kinds of plug-ins: PyPlugs and OpenFX plug-ins. The bundled OpenFX plug-ins are located in Plugins/OFX/Natron in your Natron installation and the bundled PyPlugs in the directory Plugins/PyPlugs.
- The standard user location for non OpenFX plug-ins (i.e. PyPlugs): that is the directory `.Natron` in the home directory, e.g.:

On Windows that would be:

```
C:\Users\<username>\.Natron
```

On OS X & Linux that would be:

```
~/ .Natron
```

- The standard system location for non OpenFX plug-ins (i.e. PyPlugs):

Windows:

```
C:\Program Files\Common Files\Natron\Plugins
```

OS X:

```
/Library/Application Support/Natron/Plugins
```

Linux:

```
/usr/share/Natron/Plugins
```

- All the paths indicated by the **NATRON_PLUGIN_PATH** environment variable. This environment variable should contain the separator `;` between each path, such as:

```
/home/<username>/NatronPluginsA; /home/<username>/NatronPluginsB
```

- The user extra search paths in the Plug-ins tab of the Preferences of Natron.

If the setting “Prefer bundled plug-ins over system-wide plug-ins” is checked in the preferences then Natron will first look into the bundled plug-ins before checking the standard location. Otherwise, Natron will check bundled plug-ins as the *last* location.

Note that if the “User bundled plug-ins” setting in the preferences is unchecked, Natron will not attempt to load any bundled plug-ins.

3.2.2 Python Auto-declared variables

A lot of Python variables are pre-declared by Natron upon the creation of specific objects. This applies currently to the following objects:

- *Effect*
- *Param*
- *Layer*
- *BezierCurve*
- *App*
- *Track*
- *PyCoreApplication*
- *PyTabWidget*
- *PyViewer*
- *PyPanel*

The idea is that it is simpler to access a simple variable like this:

```
node = appl.Blur1
```

rather than call a bunch of functions such as:

```
node = appl.getNode("appl.Blur1")
```

To achieve this, auto-declared objects must be named with a correct syntax in a python script. For instance, the following variable would not work in Python:

```
>>> my variable = 2
File "<stdin>", line 1
my variable = 2
      ^
SyntaxError: invalid syntax
```

But the following would work:

```
>>> myVariable = 2
```

To overcome this issue, all auto-declared variables in Natron have 2 names:

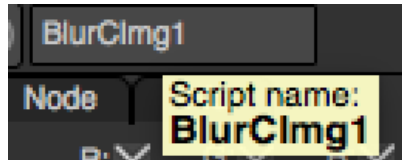
1. A script-name: The name that will be used to auto-declare the variable to Python. This name cannot be changed and is set once by Natron the first time the object is created. This name contains only alpha-numeric characters and does not start with a digit.
2. A label: The label is what is displayed on the graphical user interface. For example the node label is visible in the node graph. This label can contain any character without any restriction.

Basically there can only ever be one instance of an object with a *script-name* (so it is identified uniquely) but several instances of this object could have the same *label*.

Generally when calling a function which takes an object name in parameter, you pass it always the script-name of the object. See for example `getParam(name)`.

Knowing the script-name of a node:

The script-name of a node is visible in the graphical-user interface in the tool-tip of the widget to edit the *label* of the node:



For children nodes (like tracks) you can access their script-name from the table of the Tracker node:

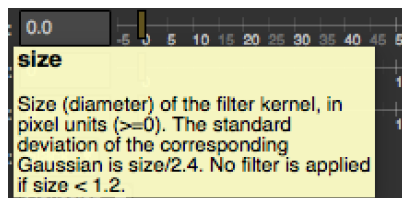
Enabled	Script-name	Track Name	Center x	Center y	Offset x	Offset y
X	TrackerPM1_1	TrackerPM1_1	1037.54	471.856	0	0

The script-name of the item as exposed to Python scripts

In command-line mode or via the *Script Editor*, you can also get the script-name of the node with the `getScriptName()` function of the *Effect* class.

Knowing the script-name of a parameter:

In the settings panel of a node, the script-name of a parameter is visible in *bold* in the tooltip displayed when hovering a parameter with the mouse:



In command-line mode or via the *Script Editor* you can also get the script-name of the parameter with the `getScriptName()` function of the *Param* class.

Knowing the script-name of an item of a Roto node:

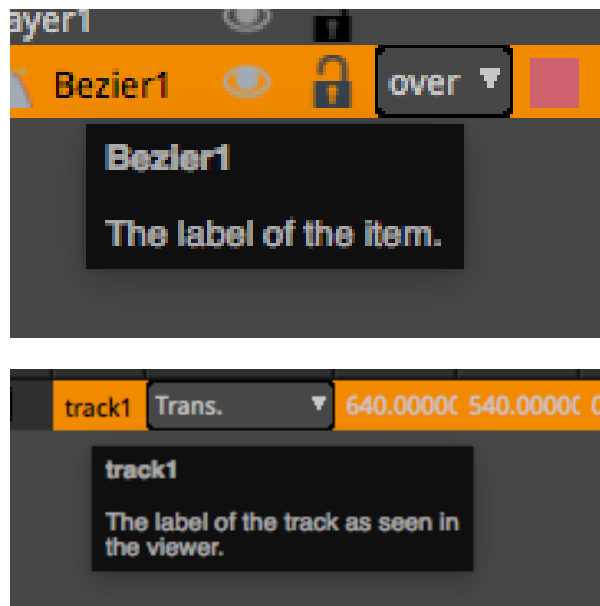
In the settings panel of a roto node, the script-name of an item is visible in the tooltip when hovering the mouse on the label of the item

In command-line mode or via the *Script Editor* you can also get the script-name of an item with the `getScriptName()` function of the *ItemBase* class.

Knowing the script-name of a track in a Tracker node:

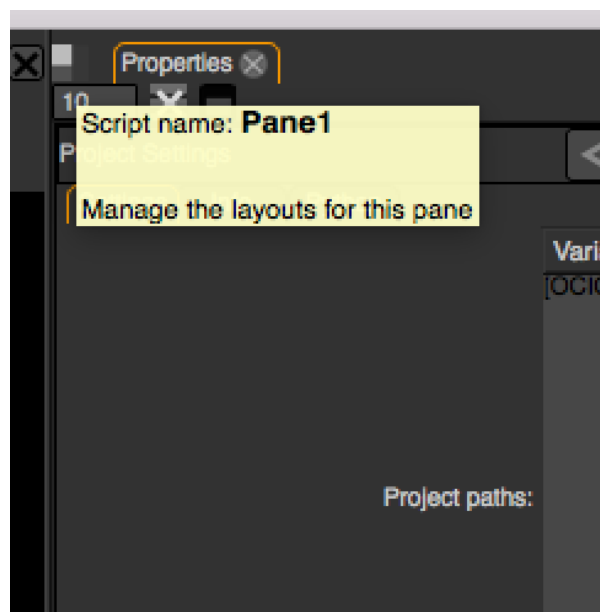
In the settings panel of a tracker node, the script-name of an item is visible in the tooltip when hovering the mouse on the label column.

In command-line mode or via the *Script Editor* you can also get the script-name of an item with the `getScriptName()` function of the *Track* class.



Knowing the script-name of a tab-widget:

The script-name of a pane can be seen on the graphical user interface by hovering the mouse over the “Manage layout” button (in the top left hand corner of a pane).



Knowing the script-name of a viewer:

The script-name of a viewer is the *script-name* of the node associated to it, e.g.:

```
app1.pane1.Viewer1
```

Knowing the script-name of a PyPanel:

The script-name of a *PyPanel* can be retrieved with the `getPanelScriptName()` function of the class.

3.2.3 Start-up scripts

On start-up Natron will run different start-up scripts to let you setup anything like callbacks, menus, etc...

There are 2 different initialization scripts that Natron will look for in the *search paths*.

- **init.py**

This script is always run and should only initialize non-GUI stuff. You may not use it to initialize e.g. new menus or windows. Generally this is a good place to initialize all the callbacks that you may want to use in your projects.

- **initGui.py**

This script is only run in GUI mode (that is with the user interface). It should initialize all gui-specific stuff like new menus or windows.

All the scripts with the above name found in the search paths will be run in the order of the *search paths*.

Warning: This is important that the 2 scripts above are named **init.py** and **initGui.py** otherwise they will not be loaded.

Warning: These scripts are run well before any *application instance* (i.e: project) is created. You should therefore not run any function directly that might rely on the *app* variable (or *app1*, etc...). However you're free to define classes and functions that may rely on these variable being declared, but that will be called only later on, when a project will actually be created.

3.2.4 Examples

initGui.py

A complete example of a **iniGui.py** can be found [here](#) .

init.py

Here is an example of a **init.py** script, featuring:

- Formats addition to the project
- Modifications of the default values of parameters for nodes
- PyPlug search paths modifications

```
#This Source Code Form is subject to the terms of the Mozilla Public
#License, v. 2.0. If a copy of the MPL was not distributed with this
#file, You can obtain one at http://mozilla.org/MPL/2.0/. */
#Created by Alexandre GAUTHIER-FOICHAT on 01/27/2015.

#To import the variable "natron"
import NatronEngine

def addFormats(app):

    app.addFormat ("720p 1280x720 1.0")
    app.addFormat ("2k_185 2048x1108 1.0")
```

(continues on next page)

(continued from previous page)

```
def afterNodeCreatedCallback(thisNode, app, userEdited):

    #Turn-off the Clamp black for new grade nodes
    if thisNode.getPluginID() == "net.sf.openfx.GradePlugin":
        thisNode.clampBlack.setDefaultValue(False)

    #Set the blur size to (3,3) upon creation
    elif thisNode.getPluginID() == "net.sf.cimg.CImgBlur":
        thisNode.size.setDefaultValue(3,0)
        thisNode.size.setDefaultValue(3,1)

#This will set the After Node Created callback on the project to tweak default_
↪values for parameters
def setNodeDefaults(app):
    app.afterNodeCreated.set("afterNodeCreatedCallback")

def setProjectDefaults(app):
    app.getProjectParam('autoPreviews').setValue(False)
    app.getProjectParam('outputFormat').setValue("2k_185")
    app.getProjectParam('frameRate').setValue(24)
    app.getProjectParam('frameRange').setValue(1, 0)
    app.getProjectParam('frameRange').setValue(30, 1)
    app.getProjectParam('lockRange').setValue(True)

def myCallback(app):
    addFormats(app)
    setNodeDefaults(app)
    setProjectDefaults(app)

#Set the After Project Created/Loaded callbacks
NatronEngine.natron.setOnProjectCreatedCallback("init.myCallback")
NatronEngine.natron.setOnProjectLoadedCallback("init.myCallback")

#Add this path to the Natron search paths so that our PyPlug can be found.
#Note that we could also set this from the NATRON_PLUGIN_PATH environment variable
#or even in the Preferences panel, Plug-ins tab, with the "Pyplugins search path"
NatronEngine.natron.appendToNatronPath("/Library/Natron/PyPlugs")
```

3.2.5 Natron in command-line

Natron has 3 different execution modes:

- The execution of Natron projects (.ntp)
- The execution of Python scripts that contain commands for Natron
- An interpreter mode where commands can be given directly to the Python interpreter

General options:

[--background] or [-b] enables background mode rendering. No graphical interface will be shown. When using *NatronRenderer* or the *-t* option this argument is implicit and you don't need to use it. If using Natron and this option is not specified then it will load the project as if opened from the file menu.

[-interpreter] or **[-t]** [optional] *<python script file path>* enables Python interpreter mode. Python commands can be given to the interpreter and executed on the fly. An optional Python script filename can be specified to source a script before the interpreter is made accessible. Note that Natron will not start rendering any Write node of the sourced script, you must explicitly start it. *NatronRenderer* and *Natron* will do the same thing in this mode, only the *init.py* script will be loaded.

Options for the execution of Natron projects:

```
Natron <project file path>
```

“-writer“ or **“-w“** *<Writer node script name>* [optional] *<filename>* [optional] *<frameRange>* specifies a Write node to render. When in background mode, the renderer will only try to render with the node script name following this argument. If no such node exists in the project file, the process will abort. Note that if you don’t pass the **–writer** argument, it will try to start rendering with all the writers in the project.

After the writer node script name you can pass an optional output filename and pass an optional frame range in the format firstFrame-lastFrame (e.g. 10-40).

Warning: You may only specify absolute file paths with the **-i** option, things like:

```
NatronRenderer -i MyReader ~/pictures.png -w MyWriter rendered###.exr
```

would not work. This would work on the other hand:

```
NatronRenderer -i MyReader /Users/me/Images/pictures.png -w MyWriter /Users/me/
↳Images/rendered###.exr
```

Note that several **“-w“** options can be set to specify multiple Write nodes to render.

Warning: Note that if specified, then the frame range will be the same for all Write nodes that will render.

“-reader“ or **“-i“** *<reader node script name>* *<filename>* : Specify the input file/sequence/video to load for the given Reader node. If the specified reader node cannot be found, the process will abort.

Warning: You may only specify absolute file paths with the **-i** option, things like:

```
NatronRenderer -i MyReader ~/pictures.png -w MyWriter rendered###.exr
```

would not work. This would work on the other hand:

```
NatronRenderer -i MyReader /Users/me/Images/pictures.png -w MyWriter /Users/me/
↳Images/rendered###.exr
```

“-onload“ or **“-l“** *<python script file path>* specifies a Python script to be executed after a project is created or loaded. Note that this will be executed in GUI mode or with *NatronRenderer* and it will be executed after any Python function set to the callback *onProjectLoaded* or *onProjectCreated*. The same rules apply to this script as the rules below on the execution of Python scripts.

“-render-stats“ or **“-s“** Enables render statistics that will be produced for each frame in form of a file located next to the image produced by the Writer node, with the same name and a *-stats.txt* extension. The breakdown contains informations about each nodes, render times, etc. This option is useful for debugging purposes or to control that a render is working correctly. **Please note** that it does not work when writing video files.

Some examples of usage of the tool:

```
Natron /Users/Me/MyNatronProjects/MyProject.ntp
```

(continues on next page)

(continued from previous page)

```
Natron -b -w MyWriter /Users/Me/MyNatronProjects/MyProject.ntp

NatronRenderer -w MyWriter /Users/Me/MyNatronProjects/MyProject.ntp

NatronRenderer -w MyWriter /FastDisk/Pictures/sequence###.exr 1-100 /Users/Me/
↳MyNatronProjects/MyProject.ntp

NatronRenderer -w MyWriter -w MySecondWriter 1-10 /Users/Me/MyNatronProjects/
↳MyProject.ntp

NatronRenderer -w MyWriter 1-10 -l /Users/Me/Scripts/onProjectLoaded.py /Users/Me/
↳MyNatronProjects/MyProject.ntp
```

Example of a script passed to `–onload`:

```
import NatronEngine

#Create a writer when loading/creating a project
writer = app.createNode("fr.inria.openfx.WriteOIO")
```

Options for the execution of Python scripts:

```
Natron <Python script path>
```

Note that the following does not apply if the `-t` option was given.

The script argument can either be the script of a Group that was exported from the graphical user interface or an exported project or even a script written by hand.

When executing a script, Natron first looks for a function with the following signature:

```
def createInstance(app, group) :
```

If this function is found, the script will be imported as a module and it will be executed.

Warning: Note that when imported, the script will not have access to any external variable declared by Natron except the variable passed to the `createInstance` function.

If this function is not found the whole content of the script will be interpreted as though it were given to Python natively.

Note: In this case the script **can** have access to the external variables declared by Natron.

Either cases, the “app” variable will always be defined and pointing to the correct application instance. Note that if you are using Natron in GUI mode, it will source the script before creating the graphical user interface and will not start rendering. When in command-line mode (`-b` option or `NatronRenderer`) you must specify the nodes to render. If nothing is specified, all Write nodes that were created in the Python script will be rendered.

You can render specific Write nodes either with the `-w` option as described above or with the following option:

`[–output] or [–o] <filename> <frameRange>` specifies an *Output* node in the script that should be replaced with a *Write* node.

The option looks for a node named *Output1* in the script and will replace it by a *Write* node much like when creating a *Write* node in the user interface.

A filename must be specified, it is the filename of the output files to render. Also a frame range must be specified if it was not specified earlier.

This option can also be used to render out multiple Output nodes, in which case it has to be used like this:

[-output1] or **[-o1]** looks for a node named *Output1* **[-output2]** or **[-o2]** looks for a node named *Output2*

etc...

-c or **[-cmd]** “PythonCommand” : Execute custom Python code passed as a script prior to executing the Python script passed in parameter. This option may be used multiple times and each python command will be executed in the order they were given to the command-line.

Some examples of usage of the tool:

```
Natron /Users/Me/MyNatronScripts/MyScript.py

Natron -b -w MyWriter /Users/Me/MyNatronScripts/MyScript.py

NatronRenderer -w MyWriter /Users/Me/MyNatronScripts/MyScript.py

NatronRenderer -o /FastDisk/Pictures/sequence###.exr 1-100 /Users/Me/
↳MyNatronScripts/MyScript.py

NatronRenderer -o1 /FastDisk/Pictures/sequence###.exr -o2 /FastDisk/Pictures/test##
↳#.exr 1-100 /Users/Me/MyNatronScripts/MyScript.py

NatronRenderer -w MyWriter -o /FastDisk/Pictures/sequence###.exr 1-100 /Users/Me/
↳MyNatronScripts/MyScript.py

NatronRenderer -w MyWriter /FastDisk/Pictures/sequence.mov 1-100 /Users/Me/
↳MyNatronScripts/MyScript.py -e "print \"Now executing MyScript.py...\""
```

Options for the execution of the interpreter mode:

```
Natron -t [optional] <Python script path>
```

Natron will first source the script passed in argument, if any and then return control to the user. In this mode, the user can freely input Python commands that will be interpreted by the Python interpreter shipped with Natron.

Some examples of usage of the tool:

```
Natron -t

NatronRenderer -t

NatronRenderer -t /Users/Me/MyNatronScripts/MyScript.py
```

3.2.6 Example

A typical example would be to convert an input image sequence to another format. There are multiple ways to do it from the command-line in Natron and we are going to show them all:

- Passing a .ntp file to the command line and passing the correct arguments
- Passing a Python script file to the command-line to setup the graph and render

With a Natron project (.ntp) file

With a Python script file

We would write a customized Python script that we pass to the command-line:

```
#This is the content of myStartupScript.py

reader = app.createReader("/Users/Toto/Sequences/Sequence_####.exr")
writer = app.createWriter("/Users/Toto/Sequences/Sequence.mov")

#The node will be accessible via app.MyWriter after this call
#We do this so that we can reference it from the command-line arguments
writer.setScriptName("MyWriter")

#The node will be accessible via app.MyReader after this call
reader.setScriptName("MyReader")

#Set the format type parameter of the Write node to Input Stream Format so that
↳the video
#is written to the size of the input images and not to the size of the project
formatType = writer.getParam("formatType")
formatType.setValue(0)

#Connect the Writer to the Reader
writer.connectInput(0, reader)

#When using Natron (Gui) then the render must explicitly be requested.
#Otherwise if using NatronRenderer or Natron -b the render will be automatically
↳started
#using the command-line arguments

#To use with Natron (Gui) to start render
#app.render(writer, 10, 20)
```

To launch this script in the background, you can do it like this:

```
NatronRenderer /path/to/myStartupScript.py -w MyWriter 10-20
```

For now the output filename and the input sequence are *static* and would need to be changed by hand to execute this script on another sequence.

We can customize the Reader filename and Writer filename parameters using the command-line arguments:

```
NatronRenderer /path/to/myStartupScript.py -i MyReader /Users/Toto/Sequences/
↳AnotherSequence_####.exr -w MyWriter /Users/Toto/Sequences/mySequence.mov 10-20
```

Let's imagine that now we would need to also set the frame-rate of the video in output and we would need it to vary for each different sequence we are going to transcode. This is for the sake of this example, you could also need to modify other parameters in a real use-case.

Since the fps cannot be specified from the command-line arguments, we could do it in Python with:

```
MyWriter.getParam("fps").set(48)
```

And change the value in the Python script for each call to the command-line, but that would require manual intervention.

That's where another option from the command-line comes into play: the “-c” option (or --cmd): It allows to pass custom Python code in form of a string that will be executed before the actual script.

To set the fps from the command-line we could do as such now:

```
NatronRenderer /path/to/myStartupScript.py -c "fpsValue=60" -w MyWriter 10-20
```

Which would require the following modifications to the Python script:

```
MyWriter.getParam("fps").set(fpsValue)
```

We could also set the same way the Reader and Writer file names:

```
NatronRenderer /path/to/myStartupScript.py -c "fpsValue=60; readFileName=\"/Users/
↳Toto/Sequences/AnotherSequence_####.exr\"; writeFileName=\"/Users/Toto/
↳Sequences/mySequence.mov\""
```

And modify the Python script to take into account the new *readFileName* and *writeFileName* parameters:

```
...
reader = app.createReader(readFileName)
writer = app.createNode(writeFileName)
...
```

The “-c” option can be given multiple times to the command-line and each command passed will be executed once, in the order they were given.

With a Natron project file:

Let’s suppose the user already setup the project via the GUI as such:

MyReader—>MyWriter

We can then launch the render from the command-line this way:

```
NatronRenderer /path/to/myProject.ntp -w MyWriter 10-20
```

We can customize the Reader filename and Writer filename parameters using the command-line arguments:

```
NatronRenderer /path/to/myProject.ntp -i MyReader /Users/Toto/Sequences/
↳AnotherSequence_####.exr -w MyWriter /Users/Toto/Sequences/mySequence.mov 10-20
```

3.2.7 Objects hierarchy Overview

When running Natron, several important objects are created automatically and interact at different levels of the application.

Natron is separated in 2 internal modules:

NatronEngine and *NatronGui*.

The latest is only available in **GUI** mode. You may access *globally* to the Natron process with either **NatronEngine.natron** or **NatronGui.natron**

NatronEngine.natron is of type PyCoreApplication and NatronGui.natron is of type PyGuiApplication. This is a singleton and there is only a **single** instance of that variable living throughout the execution of the Natron process.

When using with **NatronGui.natron** you get access to GUI functionalities in addition to the internal functionalities exposed by PyCoreApplication

Basically if using Natron in command-line you may only use **NatronEngine.natron**.

Note: You may want to use **natron** directly to avoid prefixing everything with *NatronEngine*. or *NatronGui*. by using a `from NatronEngine import *` statement. Be careful though as it then makes it more confusing for people reading the code as to which version of the **natron** variable you are using.

It handles all *application-wide* informations about plug-ins, environment, *application settings*... but also can hold one or multiple *application instance* which are made available to the global variables via the following variables:

```
app1 # References the first instance of the application (the first opened project)
app2 # The second project
...
```

Note that in background command-line mode, there would always be a single opened project so Natron does the following assignment for you:

```
app = app1
```

Warning: Note that when running scripts in the *Script Editor*, the application is running in GUI mode hence the *app* variable is not declared.

The *App* object is responsible for managing all informations relative to a project. This includes all the *nodes*, project settings and render controls. See [this section](#) to create and control nodes.

Each node can have *parameters* which are the controls found in the settings panel of the node.

The same *Param* class is also used for the project settings and the application settings (preferences).

3.3 Getting started

This section covers basic functionalities a Python script can do in Natron.

3.3.1 Creating and controlling nodes

Creating a new node:

To create a *node* in Natron, you would do so using the *app instance* via the function `createNode(pluginId, majorVersion, group)` like this:

```
app1.createNode("fr.inria.openfx.ReadOIIO")
```

In this line we specify that we want the first opened project to create a node instantiating the plug-in *ReadOIIO*. Note that if we were in background mode we could just write the following which would be equivalent:

```
app.createNode("fr.inria.openfx.ReadOIIO")
```

Since in command-line there is only a single project opened, Natron does the following assignment:

```
app = app1
```

If we were to create the node into a specific group, we would do so like this:

```
group = app.createNode("fr.inria.built-in.Group")
reader = app.createNode("fr.inria.openfx.ReadOIIO", -1, group)
```

Note that when passed the number -1, it specifies that we want to load the highest version of the plug-in found. This version parameter can be useful to load for example a specific version of a plug-in.

The *pluginID* passed to this function is a **unique** ID for each plug-in. If 2 plug-ins were to have the same ID, then Natron will create separate entries for each version.

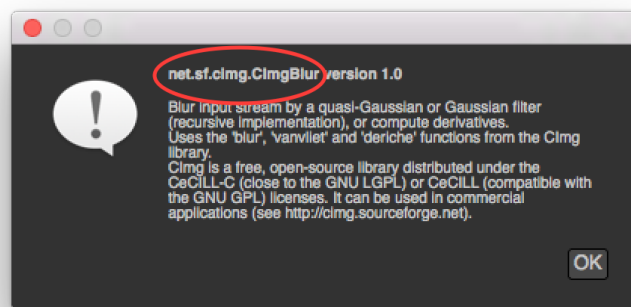
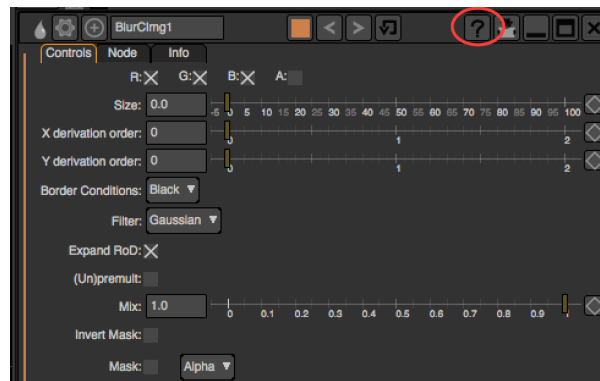
You can query all plug-ins available in Natron this way:

```
allPlugins = natron.getPluginIDs()
```

You can also filter out plug-ins that contain only a given *filter* name:

```
# Returns only plugin IDs containing ".inria" in it
filteredPlugins = natron.getPluginIDs(".inria.")
```

In the user interface, the plug-in ID can be found when pressing the ? button located in the top right-hand corner of the settings panel:



Connecting a node to other nodes:

To connect a node to the input of another node you can use the `connectInput(inputNumber, input)` function.

The *inputNumber* is a 0-based index specifying the input on which the function should connect the given *input Effect*.

You can query the input name at a specific index with the following function:

```
print(node.getInputLabel(i))
```

Here is a small example where we would create 3 nodes and connect them together:

```
#Create a write node
writer = app.createNode("fr.inria.openfx.WriteOIO")

#Create a blur
blur = app.createNode("net.sf.cimg.CimgBlur")

#Create a read node
```

(continues on next page)

(continued from previous page)

```
reader = app.createNode("fr.inria.openfx.ReadOIIIO")

#Connect the write node to the blur
writer.connectInput(0,blur)

#Connect the blur to the read node
blur.connectInput(0,reader)
```

Note that the following script would do the same since nodes are *auto-declared variables*

```
node = app.createNode("fr.inria.openfx.WriteOIIIO")
print(node.getScriptName()) # prints WriteOIIIO1

#The write node is now available via its script name app.WriteOIIIO1

node = app.createNode("net.sf.cimg.CImgBlur")
print(node.getScriptName()) # prints CImgBlur1

#The blur node is now available via its script name app.BlurCImg1

node = app.createNode("fr.inria.openfx.ReadOIIIO")
print(node.getScriptName()) # prints ReadOIIIO1

#The ReadOIIIO node is now available via its script name app.ReadOIIIO1

app.WriteOIIIO1.connectInput(0,app.BlurCImg1)
app.BlurCImg1.connectInput(0,app.ReadOIIIO1)
```

Note that not all connections are possible, and sometimes it may fail for some reasons explained in the documentation of the `connectInput(inputNumber, input)` function.

You should then check for errors this way:

```
if not app.WriteOIIIO1.connectInput(0,app.BlurCImg1):
    # Handle errors
```

You can check beforehand whether a subsequent *connectInput* call would succeed or not by calling the `canConnectInput(inputNumber, input)` which basically checks whether is is okay to do the connection or not. You can then safely write the following instructions:

```
if app.WriteOIIIO1.canConnectInput(0,app.BlurCImg1):
    app.WriteOIIIO1.connectInput(0,app.BlurCImg1)
else:
    # Handle errors
```

Note that internally *connectInput* calls *canConnectInput* to validate whether the connection is possible.

To disconnect an existing connection, you can use the `disconnectInput(inputNumber)` function.

3.3.2 Controlling parameters

Accessing a node's parameters:

As for nodes, *parameters* are *auto-declared* objects. You can access an existing parameter of a node by its *script-name*:

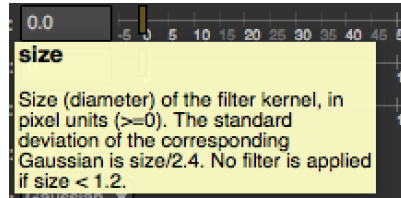
```
app.BlurCImg1.size
```

Note that you can also access a parameter with the `getParam(scriptName)` function:

```
param = app.BlurCimg1.getParam("size")
```

but you should not ever need it because Natron pre-declared all variables for you.

The *script-name* of a parameter is visible in the user interface when hovering the parameter in the settings panel with the mouse. This is the name in **bold**:



Parameters type:

Each parameter has a type to represent internally different data-types, here is a list of all existing parameters:

- *IntParam* to store 1-dimensional integers
- *Int2DParam* to store 2-dimensional integers
- *Int3DParam* to store 3-dimensional integers
- *DoubleParam* to store 1-dimensional floating point
- *Double2DParam* to store 2-dimensional floating point
- *Double3DParam* to store 3-dimensional floating point
- *BooleanParam* to store 1-dimensional boolean (checkbox)
- *ButtonParam* to add a push-button
- *ChoiceParam* a 1-dimensional drop-down (combobox)
- *StringParam* to store a 1-dimensional string
- *FileParam* to specify an input-file
- *OutputFileParam* to specify an output-file param
- *PathParam* to specify a path to a single or multiple directories
- *ParametricParam* to store N-dimensional parametric curves
- *GroupParam* to graphically gather parameters under a group
- *PageParam* to store parameters into a page

Retrieving a parameter's value:

Since each underlying type is different for parameters, each sub-class has its own version of the functions.

To get the value of the parameter at the timeline's current time, call the `get()` or `getValue()` function.

If the parameter is animated and you want to retrieve its value at a specific time on the timeline, you would use the `get(frame)` or `getValueAtTime(frame, dimension)` function.

Note that when animated and the given *frame* time is not a time at which a keyframe exists, Natron will interpolate the value of the parameter between surrounding keyframes with the interpolation filter selected (by default it is *smooth*).

Modifying a parameter's value:

You would set the parameter value by calling the `set(value)` or `setValue(value)` function. If the parameter is animated (= has 1 or more keyframe) then calling this function would create (or modify) a keyframe at the timeline's current time.

To add a new keyframe the `set(value, frame)` or `setValueAtTime(value, frame, dimension)` function can be used.

To remove a keyframe you can use the `deleteValueAtTime(frame, dimension)` function. If you want to remove all the animation on the parameter at a given *dimension*, use the `removeAnimation(dimension)` function.

Warning: Note that the dimension is a 0-based index referring to the dimension on which to operate. For instance a *Double2DParam* has 2 dimensions *x* and *y*. To set a value on *x* you would use *dimension = 0*, to set a value on *y* you would use *dimension = 1*.

Controlling other properties of parameters:

See the documentation for the *Param* class for a detailed explanation of other properties and how they affect the parameter.

Creating new parameters:

In Natron, the user has the possibility to add new parameters, called *User parameters*. They are pretty much the same than the parameters defined by the underlying OpenFX plug-in itself.

In the Python API, to create a new *user parameter*, you would need to call one of the `createXParam(name, label, ...)` of the *Effect* class.

These parameters can have their default values and properties changed as explained in the documentation page of the *Param* class.

To remove a user created parameter you would need to call the `removeParam(param)` function of the *Effect* class.

Warning: Only **user parameters** can be removed. Removing parameters defined by the OpenFX plug-in will not work.

3.3.3 Parameters expressions

The value of a *parameter* can be set by Python expressions. An expression is a line of code that can either reference the value of other parameters or apply mathematical functions to the current value.

The expression will be executed every times the value of the parameter is fetched from a call to `getValue(dimension)` or `get()`.

Warning: Note that when an expression is active, all animation is ignored and only the result of the expression will be used to return the value of the parameter.

When executing an expression, the expression itself has a **scope**. The **scope** of the expression defines all nodes and parameters that are possible to use in the expression in order to produce the output value.

Any node in the *scope* can has a variable declared corresponding to its script-name:


```
Blur1
```

You would then access a parameter of `Blur1` also by its script-name:

```
Blur1.size
Group1.Blur1.size
```

Warning: Referencing the value of the same parameter which expression is being edited can lead to an infinite recursion which Python should warn you about

In fact this is exactly like referencing *auto-declared* nodes via the *Script Editor* except that the *app* prefix was removed for nodes in the scope.

See [this section](#) to learn how to determine the *script-name* of a node.

See [this section](#) to learn how to determine the *script-name* of a parameter.

By default a parameter's expression can only refer to parameters of nodes belonging to the same Group, or to parameters belonging to the parent *Group* node.

Parameters of a Group node are also granted in the scope the parameters contained within that group.

For instance if your graph hierarchy looks like this:

```
Read1
Blur1
Group1/
  Input1
  Blur1
  Convolve1
  Roto1
  Output1
Viewer1
```

A parameter of *Read1* would be able to reference any parameter of *Read1*, *Blur1*, *Group1*, *Viewer1* but could not reference any parameter of the nodes within *Group1*.

Similarly, a parameter of *Group1.Blur1* would be able to reference any parameter of *Group1*, *Group1.Input1*, *Group1.Blur1*, *Group1.Convolve1*, *Group1.Roto1*, *Group1.Output1* but would not be able to reference any top-level node (*Read1*, *Blur1*, *Viewer1*) except the *Group1* node.

A parameter of *Group1* would on the other hand be able to reference any parameter in top-level nodes and in the nodes of *Group1*.

The *scope* was introduced to deal with problems where the user would write expressions referencing parameters that would probably no longer be referable when loading the script again in another project.

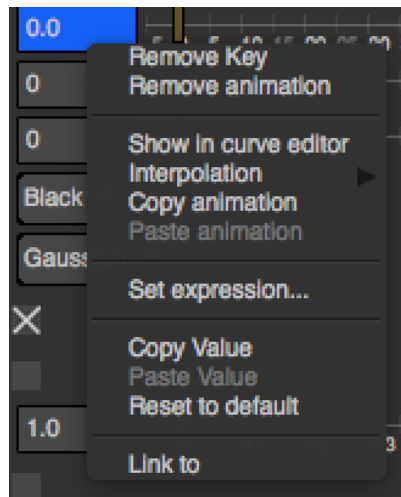
Warning: Note that you would still be able to reach any node or parameter in the project using the *app1* (or *app* prefix in command-line mode) but is not recommended to do so:

```
app1.Blur1.size
```

All functions available in the Python API are made available to expressions. Also for convenience the **math** Python module has been made available by default to expressions.

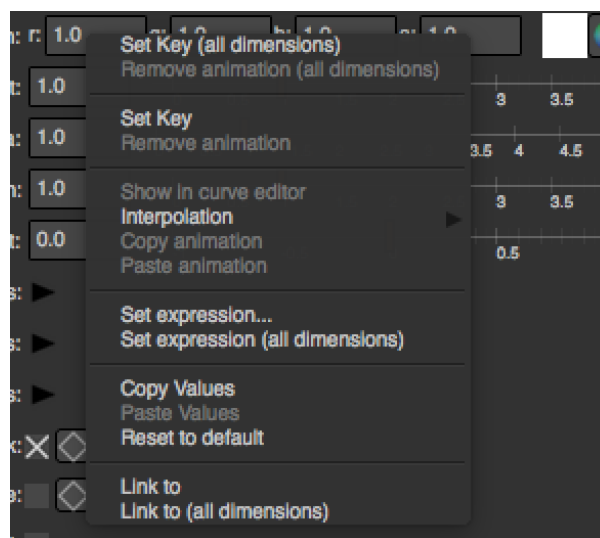
Setting an expression:

To create an expression from the user interface, right click a parameter and choose *Set Expression...*



Note that for multi-dimensional parameters such as *ColorParam*, the *Set Expression...* entry will **only set an expression for the right-clicked dimension**.

The *Set Expression (all dimensions)* entry will on the other hand set the same expression on all dimensions of the parameter at once.



A dialog will open where you can write the expression:

By default you do not have to assign any variable as the result of the expression, Natron will do it by itself:

```
#Expression for Blur1.size

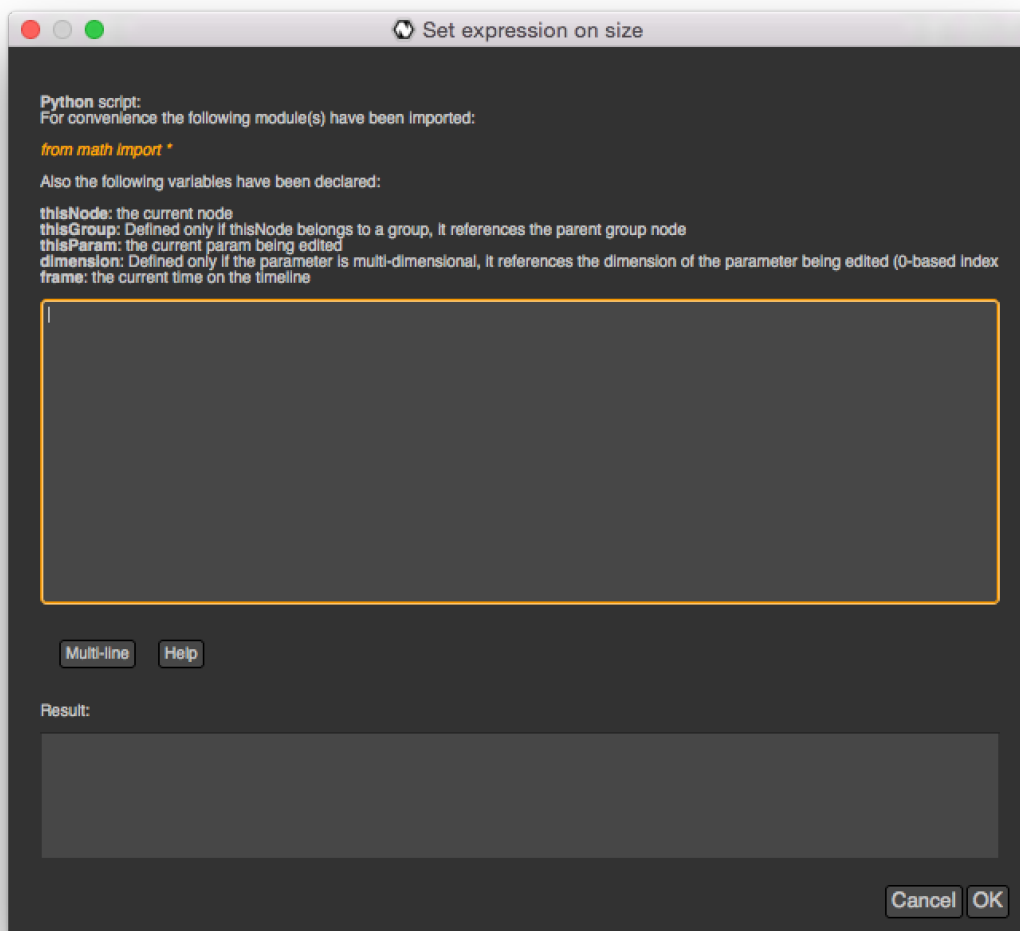
Transform1.translate.get[0]

#Will be expanded automatically by Natron to

ret = Transform1.translate.get[0]
```

However if you were to write an expression that spans over multiple lines you would need to specifically set the **ret** variable yourself and toggle-on the *multi-line* button:

```
a = acos(Transform1.translate.get[0])
b = sin(Transform1.rotate.get())
ret = (tan(a * b) / pi) + Group1.customParam.get()
```



You can also set an expression from a script using the `setExpression(expr, hasRetVariable, dimension)` function of [AnimatedParam](#).

Writing an expression:

For convenience the following variables have been declared to Python when executing the expression:

- **thisNode**: It references the node holding the parameter being edited
- **thisGroup**: It references the group containing *thisNode*
- **thisParam**: It references the param being edited
- **dimension**: Defined only for multi-dimensional parameters, it indicates the dimension (0-based index) of the parameter on which the expression has effect.
- **frame**: It references the current time on the timeline
- The **app** variable will be set so it points to the correct [application instance](#).

To reference the value of another parameter use the `get()` function which retrieves the value of the parameter at the current timeline's time. If the parameter is multi-dimensional, you need to use the subscript operator to retrieve the value of a particular dimension.

The `getValue(dimension)` does the same thing but takes a *dimension* parameter to retrieve the value of the parameter at a specific *dimension*. The following is equivalent:

```
ColorCorrect1.MasterSaturation.get()[dimension]
ColorCorrect1.MasterSaturation.getValue(dimension)
```

Note that for 1-dimensional parameter, the `get()` function cannot be used with subscript, e.g.:

```
Blur1.size.get()
```

To retrieve the value of the parameter at a specific *frame* because the parameter is animated, you can use the `get(frame)` function.

Again the `getValueAtTime(frame, dimension)` does the same thing but takes a *dimension* parameter to retrieve the value of the parameter at a specific *dimension*. The following lines are equivalent to the 2 lines above:

```
ColorCorrect1.MasterSaturation.get(frame)[dimension]
ColorCorrect1.MasterSaturation.getValueAtTime(frame, dimension)
```

We ask for the value of the *MasterSaturation* parameter of the *ColorCorrect1* node its value at the current *frame* and at the current *dimension*, which is the same as calling the `get()` function without a *frame* in parameter.

Copying another parameter through expressions:

If we want the value of the parameter **size** of the node *BlurCImg1* to copy the parameter **mix** of the node *DilateCImg1*, we would set the following expression on the **size** parameter of the node *BlurCImg1* (see [setting an expression](#)):

```
DilateCImg1.mix.get()
```

If mix has an animation and we wanted to get the value of the mix at the previous *frame*, the following code would work:

```
DilateCImg1.mix.get(frame - 1)
```

Note that when choosing the *Link to...* option when right-clicking a parameter, Natron writes automatically an expression to copy the parameter to link to for you.

Using random in expressions:

Sometimes it might be useful to add a random generator to produce noise on a value. However the noise produced must be reproducible such that when rendering multiple times the same frame or when loading the project again it would use the same value.

We had to add a specific random function in Natron that takes into account the state of a parameter and the current time on the timeline as a seed function to random.

Warning: We advise against using the functions provided by the module `random.py` of the Python standard library, as the values produced by these functions will not be reproducible between 2 runs of Natron for the same project.

The Natron specific random functions are exposed in the `Param` class.

When executing an expression, Natron pre-declares the **random()** function so that you do not have to do stuff like:

```
thisParam.random()
```

Instead you can just type the following in your expression:

```
myOtherNode.myOtherNodeParam.get() * random()
```

The `random(min = 0., max = 1.)` function also takes 2 optional arguments indicating the range into which the return value should fall in. The range is defined by `[min,max[`.

#Returns a random floating point value in the range [1., 10.[`random(1.,10.)`

For integers, use the `randomInt(min,max)` function instead:

```
#Returns a random integer in the range [1,100[
randomInt(1,100)

#Using the randomInt function with a given seed
seed = 5
randomInt(1,100,frame,seed)
```

Advanced expressions:

To write more advanced expressions based on fractal noise or perlin noise you may use the functions available in the `ExprUtils` class.

Expressions persistence

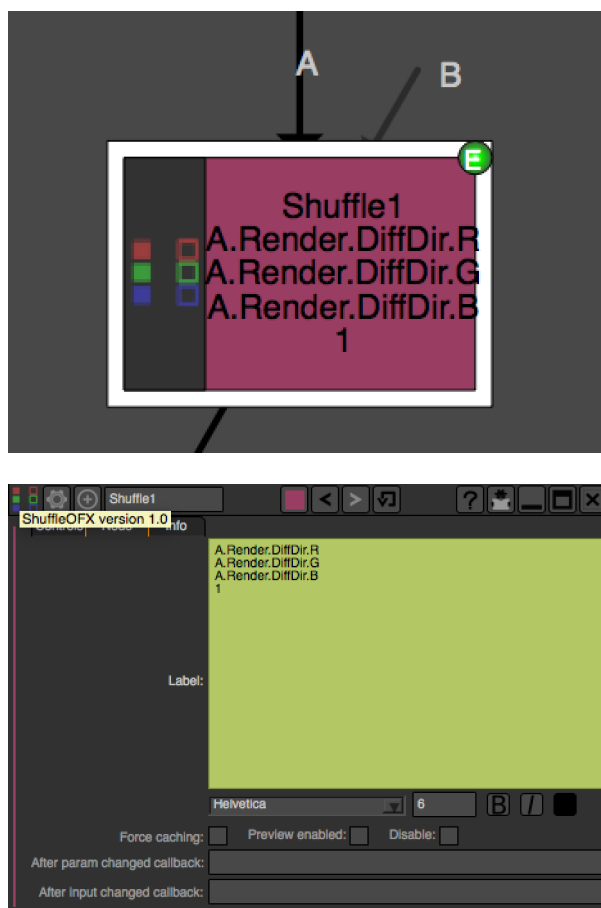
If you were to write a group plug-in and then want to have your expressions persist when your group will be instantiated, it is important to prefix the name of the nodes you reference in your expression by the **thisGroup.** prefix. Without it, Natron thinks you're referencing a top-level node, i.e: a node which belongs to the main node-graph, however, since you're using a group, all your nodes are no longer top-level and the expression will fail.

Examples

Setting the label of a Node so it displays the value of a parameter on the node-graph:

For instance, we may want to have on the shuffle node, the values of the output RGBA channels so we don't have to open the settings panel to understand what the node is doing.

To do so, we set an expression on the "Label" parameter located in the "Node" tab of the settings panel.



Set the following expression on the parameter

```
thisNode.outputR.getOption(thisNode.outputR.get()) + "\n" + thisNode.outputG.  
↪getOption(thisNode.outputG.get()) + "\n" + thisNode.outputB.getOption(thisNode.  
↪outputB.get()) + "\n" + thisNode.outputA.getOption(thisNode.outputA.get())
```

Generating custom animation for motion editing:

In this example we will demonstrate how to perform Loop, Negate and Reverse effects on an animation even though this is already available as a preset in Natron.

To do be able to do this we make use of the `curve(frame,dimension)` function of the [Param](#) class. This function returns the value of the animation curve (of the given dimension) at the given time.

If we were to write the following expression:

```
curve(frame)
```

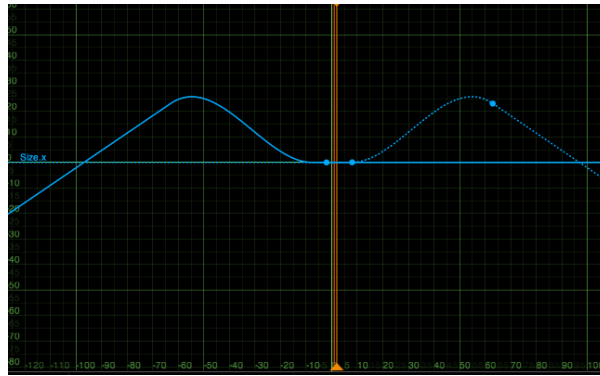
The result would be exactly the animation curve of the parameter.

On the other hand if we write:

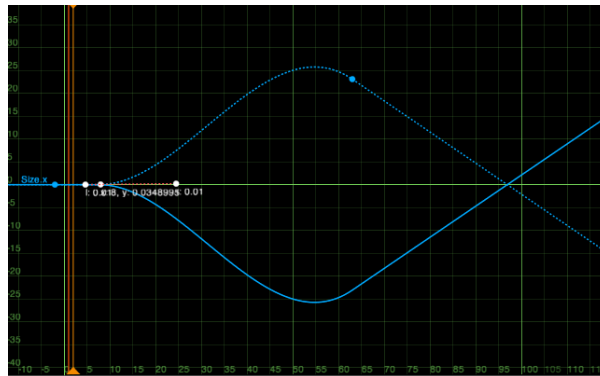
```
curve(-frame)
```

We have just reversed the curve, meaning that the actual result at the frame F will be in fact the value of the curve at the frame $-F$.

In the same way we can apply a negate effect:

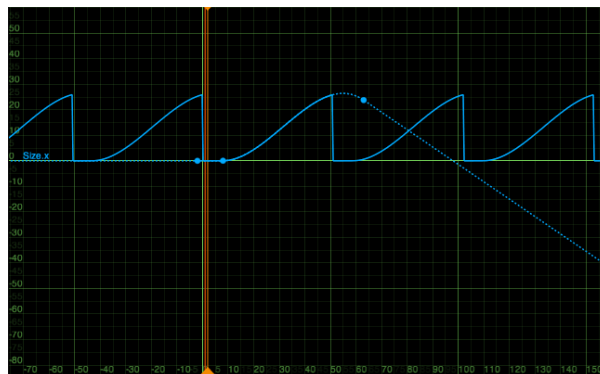


```
-curve(frame)
```



The loop effect is a bit more complicated and needs to have a frame-range in parameter:

```
firstFrame = 0
lastFrame = 10
curve(((frame - firstFrame) % (lastFrame - firstFrame + 1)) + firstFrame)
```

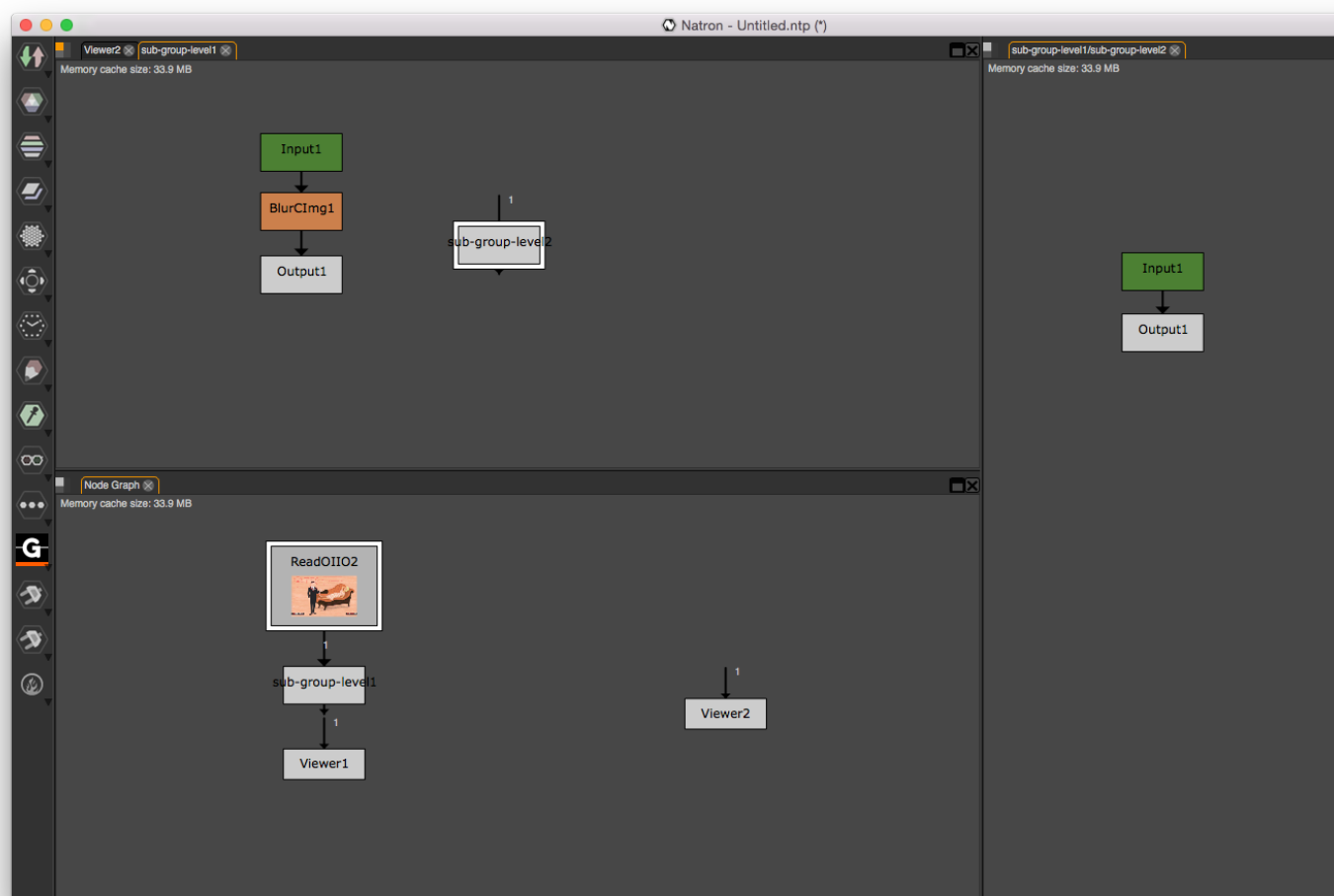


3.3.4 Working with groups

Groups in Natron are a complete sub-nodograph into which the user can manage nodes exactly like in the *main* nodograph, but everything in that sub-group will be referenced as 1 node in the hierarchy above, e.g.:

A group can be created like any other node in Natron and by default embeds already 2 nodes: The **Output** node and one **Input** node.

The **Output** node is used to reference what would be the output of the internal graph of the group. In Natron, **a node has necessarily a single output**, hence if you add several *Output* nodes to a group, **only the first Output node will be taken into account**.



Note that you can also add *Output* nodes to the top-level graph of Natron (the main Node Graph). They are useful if you need to export your project as a group.

When used in the top-level graph, there can be multiple *Output* nodes, which can then be used when launching Natron from the command-line to render the script, e.g.:

```
NatronRenderer -o1 /FastDisk/Pictures/sequence###.exr -o2 /FastDisk/Pictures/test###.exr 1-100
/Users/Me/MyNatronScripts/MyScript.py
```

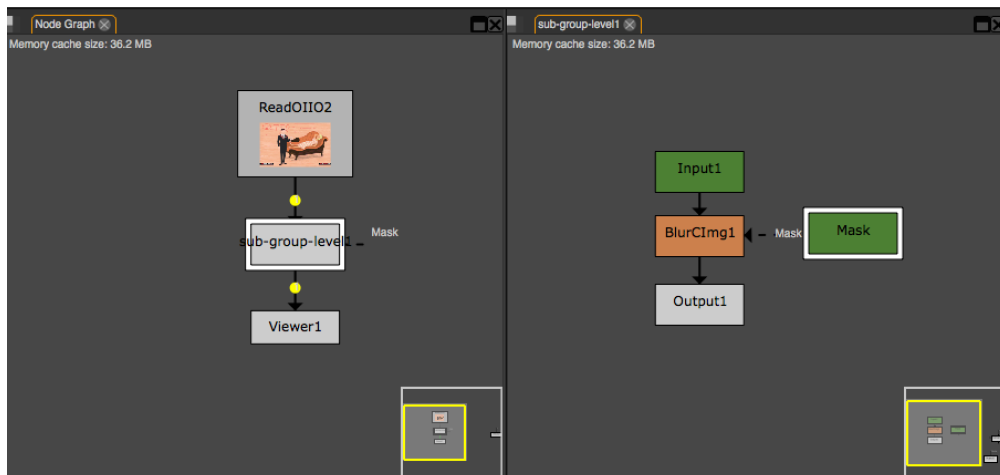
Where each argument *o1*, *o2* expand respectively the nodes *Output1* and *Output2*.

Warning: You should never attempt to change the script name of output nodes, otherwise Natron has no way to match the given command line arguments to the output nodes. In fact Natron will completely ignore your request if you explicitly try to set the script name of an *Output* node.

The **Input** node is not necessarily unique and represents 1 input arrow of the group node. You can also specify in the settings panel of the *Input* node whether this input should be considered as a mask or whether it should be optional.

Note: Note that the OpenFX standard specifies that Mask inputs must be optionals so when checking the mask parameter, this will automatically check the *optional* parameter.

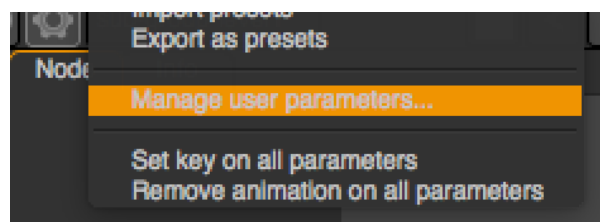
You can freely rename an **Input** node, effectively changing the label attached to the arrow on the group node.

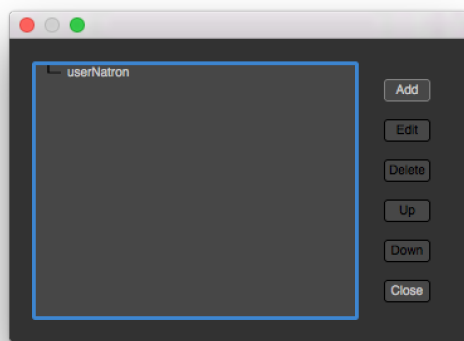


Parameters expressions and groups

A common task is to add parameters to the group node itself which directly interact to nodes parameters used internally by this group.

You can add a new parameter to the group node by clicking the “Settings and presets” button and clicking “Manage user parameters...”:





A dialog will popup on which you can manage all the parameters that you added. By default a page is added automatically that will contain user parameters.

To create a new parameter, click the add button, this brings up a new dialog:

In this dialog you can configure all the properties of the parameter exactly like you would do using the [Python API](#).

Once created, the new parameter can be found in the “User” page of the settings panel:

We can then set for instance an expression on the internal blur size parameter to copy the value of the blur size parameter we just added to the group node:

The expression is now visible in a green-ish color on the parameter in the settings panel and the node on the node-graph has a green “E” indicator.

Exporting a group

Once your group is setup correctly, you can *export* it as a Python script that Natron will generate automatically. We call them *PyPlugs*.

To do so, click the **Export as Python plug-in** button in the “Node” page of the settings panel of the Group node.

Exporting a group as a plug-in, means that it will create a Python script that will be able to re-create the group entirely and that will be loaded on startup like any other plug-in. That means that the group will also appear in the left toolbar of Natron and can potentially have an icon too.

The *Label* is the name of the plug-in as it will appear in the user interface. It should not contain spaces or non Python friendly characters as it is going to be used as variable names in several places.

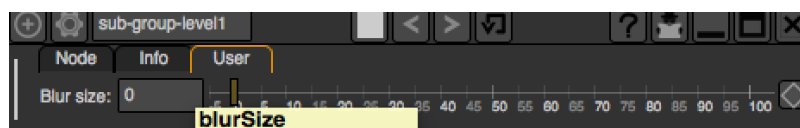
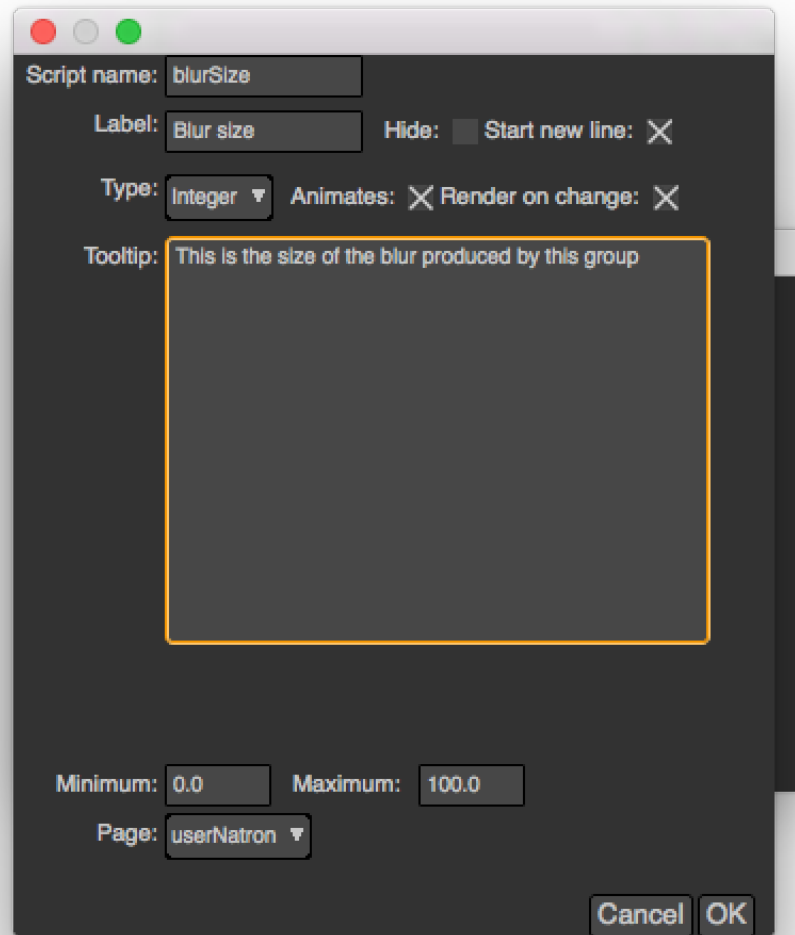
The *Grouping* is the tool-button under which the plug-in should appear. It accepts sub-menus notation like this: “Inria/StereoGroups”

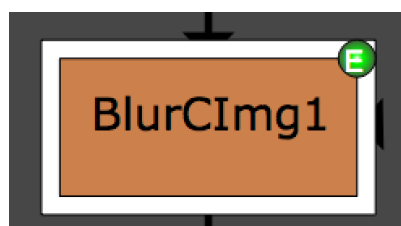
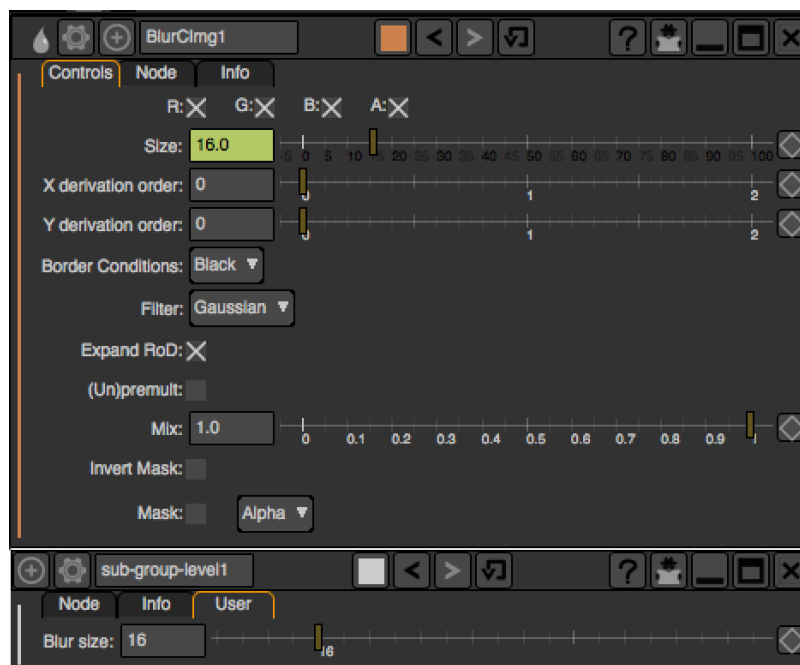
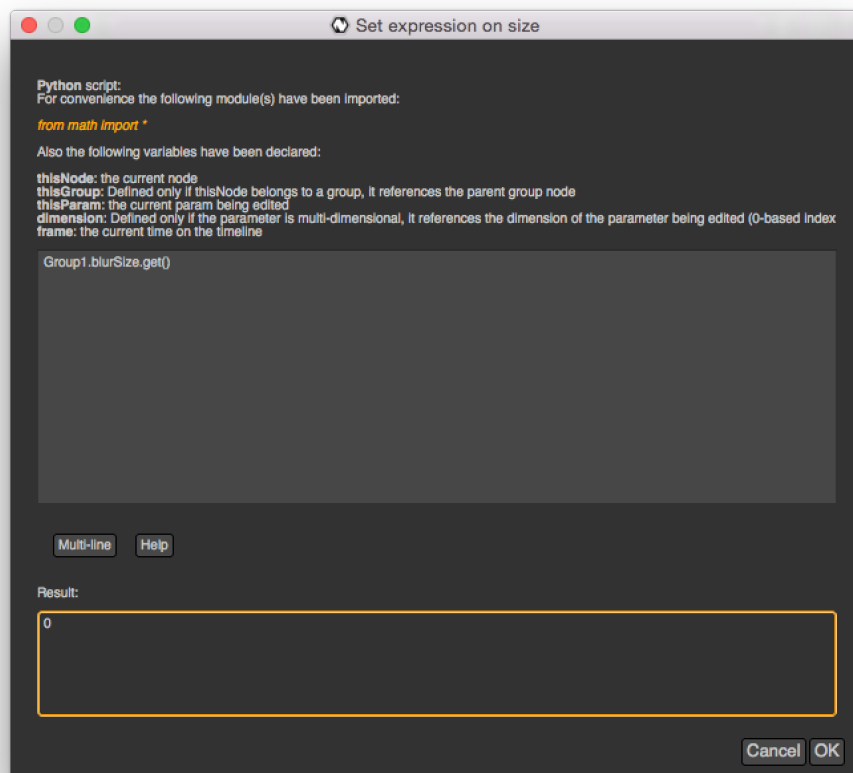
The *Icon relative path* is the filepath to an image which should be used as icon for the plug-in. Note that it is a relative path to the location of the python script.

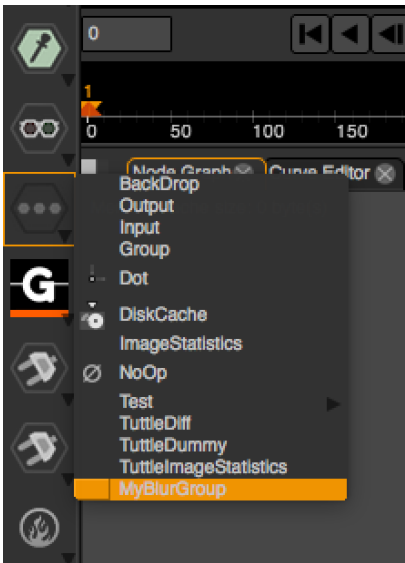
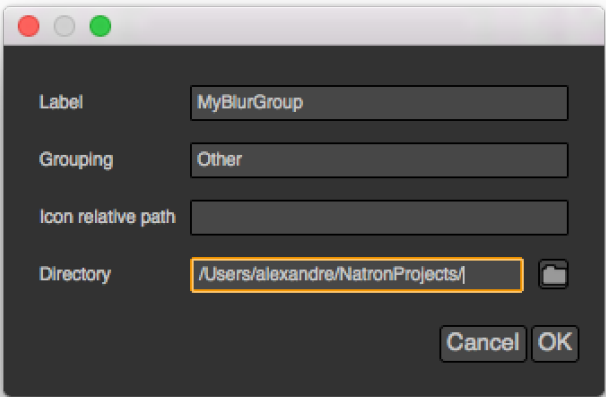
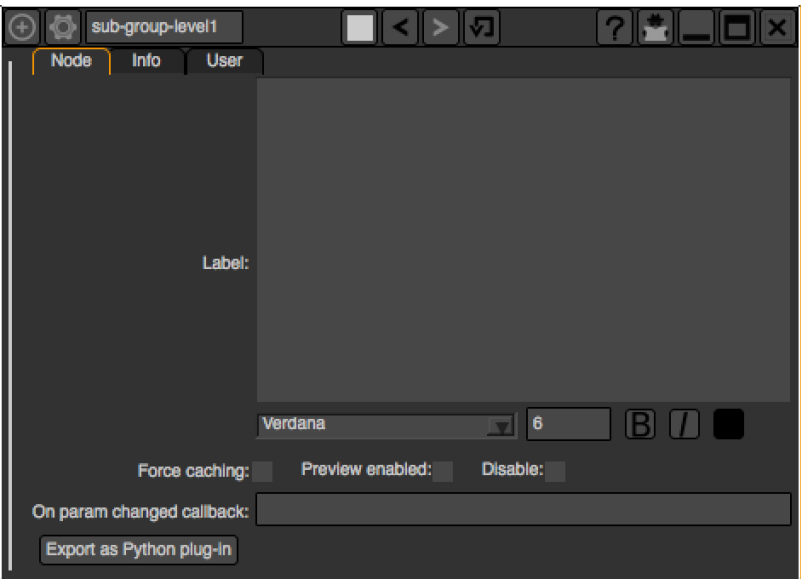
The *directory* is the location where the script should be written to. For the plug-in to be loaded by Natron, it should be in its [search-paths](#) hence if you select a directory that is not yet in the search-paths, it will prompt you to add it.

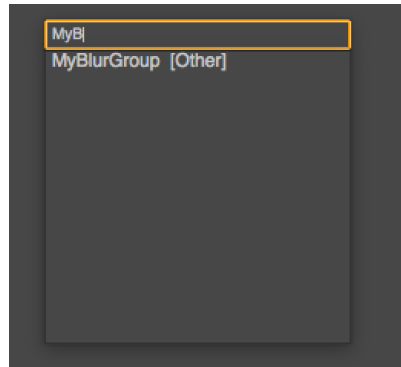
Note: A re-launch of Natron is required to re-scan the plug-ins and build the tool menus

Once restarted, the plug-in should now appear in the user interface and even in the tab menu of the node-graph:









Note: The plug-in ID of the group will be exactly the same as the *Label* you picked when exporting it, hence when creating a node using the group from a Python script, you would do so:

```
app.createNode("MyBlurGroup")
```

If several plug-ins have the same *pluginID*, Natron will then sort plug-ins by version.

The version of a plug-in by default when exporting it via Natron is 1.

Warning: If 2 plug-ins happen to have the same pluginID and version, Natron will then load the first one found in the search paths.

To change the **pluginID** and **version** of your group plug-in, you must implement the 2 following functions in the python script of the group:

```
# This function should return an int specifying the version of the plug-in
# If not implemented, Natron will use 1 by default
def getVersion():
    return VERSION

# This function should return a string specifying the ID of the plug-in, for_
↪example
# "fr.inria.groups.customBlur"
# If not implemented, Natron will use the label as a pluginID
def getPluginID():
    return UNIQUE_ID
```

Exporting a project as group

Similarly, Natron allows you to export the top-level node-graph as a Python group plug-in. From the “File” menu, select “Export project as group”.

Warning: To be exportable, your project should at least contain 1 output node.

Note: While this functionality is made for convenience, you should be cautious, as exporting a project containing Readers will probably not work very well in another project or computer because of file-paths no longer pointing to a valid location.

Warning: If you were to write a group plug-in and then want to have your expressions persist when your group will be instantiated, it is important to prefix the name of the nodes you reference in your expression by the **thisGroup.** prefix. Without it, Natron thinks you're referencing a top-level node, i.e: a node which belongs to the main node-graph, however, since you're using a group, all your nodes are no longer top-level and the expression will fail.

Moving nodes between groups

You can create a group from the selection in Natron by holding CTRL+SHIFT+G. This will effectively move all nodes selected into a new sub-group

You can also copy/cut/paste in-between groups and projects.

Creating a group by hand

You can also write a group plug-in by hand using the *Python API* of Natron.

To work as a plug-in, your script should implemented the following functions:

```
# This function is mandatory and should return the label of the plug-in as
# visible on the user interface
def getLabel():
    return LABEL

# This function should return an int specifying the version of the plug-in
# If not implemented, Natron will use 1 by default
def getVersion():
    return VERSION

# This function should return a string specifying the ID of the plug-in, for
↳example
# "fr.inria.groups.customBlur"
# If not implemented, Natron will use the label as a pluginID
def getPluginID():
    return UNIQUE_ID

# This function should return a string specifying the relative file path of an
↳image
# file relative to the location of this Python script.
# This function is optional.
def getIconPath():
    return ICON_PATH

# This function is mandatory and should return the plug-in grouping, e.g.:
# "Other/Groups"
def getGrouping():
    return GROUPING

# This function is optional and should return a string describing the plug-in to
↳the user.
# This is the text that will show up when the user press the "?" button on the
↳settings panel.
def getDescription():
    return DESCRIPTION

# This function is mandatory and should re-create all the nodes and parameters
↳state
# of the group.
# The group parameter is a group node that has been created by Natron and that
↳will host all
```

(continues on next page)

(continued from previous page)

```
# the internal nodes created by this function.
# The app parameter is for convenience to have access in a generic way to the app_
↪object,
# no matter in which project instance your script is invoked in.
def createInstance(app, group):
    ...
```

The Python group plug-ins generated automatically by Natron are a good start to figure out how to write scripts yourself.

Warning: Python group plug-ins should avoid using any functionality provided by the *NatronGui* module because it would then break their compatibility when working in command-line background mode. The reason behind this is that the Python module *NatronGui* is not imported in command-line mode because internally it relies on the *QtGui* library, which may not be present on some render-farms. Attempts to load PyPlugs relaying on the *NatronGui* module would then fail and the rendering would abort.

Warning: Note that PyPlugs are **imported** by Natron which means that the script will not have access to any external variable declared by Natron except the variables passed to the *createInstance* function or the attributes of the modules imported.

Adding hand-written code (callbacks, etc...)

It is common to add hand-written code to a PyPlug. When making changes to the PyPlug from the GUI of Natron, exporting it again will overwrite any change made to the python script of the PyPlug. In order to help development, all hand-written code can be written in a separate script with the **same** name of the original Python script but ending with *Ext.py*, e.g.:

MyPyPlugExt.py

This extension script can contain for example the definition of all callbacks used in the PyPlug. When calling the *createInstance(app,group)* function, the PyPlug will call right at the end of the function the *createInstanceExt(app,group)* function. You can define it in your *extension script* if you want to apply extra steps to the creation of the group. For example you might want to actually set the callbacks on the group:

```
#This is in MyPyPlugExt.py

def paramChangedCallback(thisParam, thisNode, thisGroup, app, userEdited):
    print thisParam.getScriptName()

def createInstanceExt(app,group):
    # Note that the callback belongs to the PyPlug to so we use it as prefix
    group.onParamChanged.set("MyPyPlug.paramChangedCallback")
```

Note: Note that callbacks don't have to be registered with the extension module prefix but just with the PyPlug's name prefix since the *"from ... import *"* statement is made to import the extensions script.

Starting Natron with a script in command line

Natron can be started with a Python script as argument.

When used in background mode (i.e: using *NatronRenderer* or *Natron* with the option **-b**) Natron will do the following steps:

- Source the script

- If found, run a function with the following signature *createInstance(app,group)*
- Start rendering the specified writer nodes (with the **-w** option) and/or the *Output* nodes (with the **-o** option)

This allows to pass a group plug-in to Natron and render it easily if needed. Also, it can take arbitrary scripts which are not necessarily group plug-ins.

When Natron is launched in GUI mode but with a Python script in argument, it will do the following steps:

- Source the script
- If found, run a function with the following signature *createInstance(app,group)*

Toolsets

Toolsets in Natron are a predefined set of actions that will be applied to the node-graph. They work exactly like PyPlugs except that no actual group node will be created, only the content of the *createInstance(app,group)* function will be executed.

This useful to create pre-defined graphs, for example like the Split and Join plug-in in the Views menu.

To be recognized as a toolset, your PyPlug must implement the following function:

```
def getIsToolset() :
    return True
```

Also the **group** parameter passed to the *createInstance(app,group)* function will be *None* because no group node is actually involved.

3.3.5 Using Callbacks

Callbacks are functions that are executed after or before a certain event in Natron. They are Python-defined methods that you declare yourself and then register to Natron in a different manner for each callback.

This document describes the signature that your different callbacks must have in order to work for each event. The parameters of your declaration must match exactly the same signature otherwise the function call will not work.

Warning: Note that callbacks will be called in background and GUI modes, hence you should wrap all GUI code by the following condition:

```
if not NatronEngine.natron.isBackground() :
    #...do gui stuff
```

Callback persistence

If you want your callback to persist 2 runs of Natron; it is necessary that you define it in a script that is loaded by Natron, that is, either the **init.py** script (or **initGui.py** if you want it only available in GUI mode) or the script of a Python group plug-in (or its extension script, see [here](#)). See [this section](#) for more infos.

Here is the list of the different callbacks:

The param changed callback

This function is called every times the value of a *parameter* changes. This callback is available for all objects that can hold parameters,namely:

- *Effect*
- *PyPanel*

- *PyModalDialog*

The signature of the callback used on the *Effect* is:

```
callback(thisParam, thisNode, thisGroup, app, userEdited)
```

- **thisParam** : This is a *Param* pointing to the parameter which just had its value changed.
- **thisNode** : This is a *Effect* pointing to the effect holding **thisParam**
- **thisGroup** : This is a *Effect* pointing to the group holding **thisNode** or **app** otherwise if the node is in the main node-graph.
- **app** : This variable will be set so it points to the correct *application instance*.
- **userEdited** : This indicates whether or not the parameter change is due to user interaction (i.e: because the user changed the value by herself/himself) or due to another parameter changing the value of the parameter via a derivative of the `setValue(value)` function.

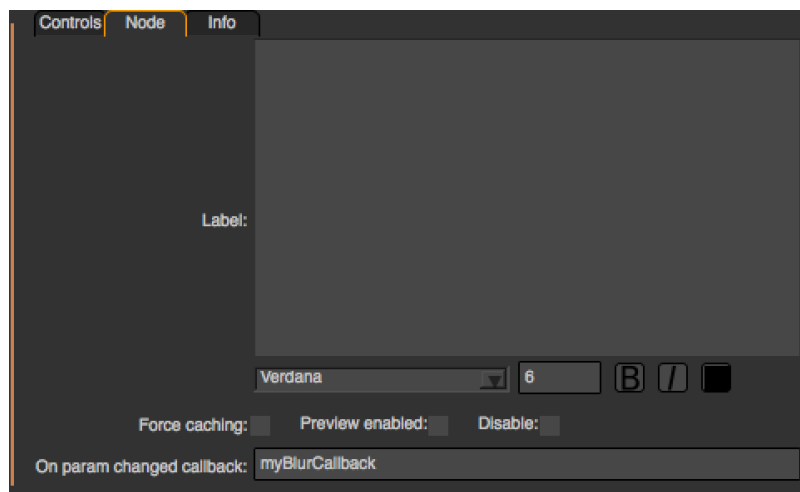
For the param changed callback of *PyPanel* and *PyModalDialog* on the other hand, Natron will define a string variable **paramName** indicating the *script-name* of the parameter which just had its value changed. The signature of the callback is then:

```
callback(paramName, app, userEdited)
```

Note: The difference between the callbacks on *PyPanel* and *PyModalDialog* and *Effect* is due to technical reasons: mainly because the parameters of the *PyPanel* class and *PyModalDialog* are not declared as attributes of the object.

Registering the param changed callback

To register the param changed callback of an *Effect*, you can do so in the settings panel of the node, in the “Node” tab, by entering the name of your Python function:



You can also set the callback directly from the script: The callback is just another *parameter* of the node, on which you can call `setValue(value)` to set the name of the callback

```
def myBlurCallback(thisParam, thisNode, thisGroup, app, userEdited):
    ...

app.BlurCimg1.onParamChanged.set("myBlurCallback")
```

Note: If the callback is defined in a separate python file, such as the python script of a python group plug-in, then do not forget the module prefix, e.g.:

```
app.MyPlugin1.BlurCImg1.onParamChanged.set("MyPlugin.myBlurCallback")
```

Example

```
# This simple callback just prints a string when the "size" parameter of the_
↳ BlurCImg
# node changes
def myBlurCallback(thisParam, thisNode, thisGroup, app, userEdited):
    if thisParam == thisNode.size:
        print("The size of the blur just changed!")

app.BlurCImg1.onParamChanged.set("myBlurCallback")
```

Using the param changed callback for PyModalDialog and PyModalDialog

To register the callback to the object, use the `setParamChangedCallback(pythonFunctionName)` function.

The following example is taken from the `initGui.py` script provided as example in [this section](#).

Example

```
#Callback called when a parameter of the player changes
#The variable paramName is declared by Natron; indicating the name of the_
↳ parameter which just had its value changed
def myPlayerParamChangedCallback(paramName, app, userEdited):

    viewer = app.getViewer("Viewer1")
    if viewer == None:
        return
    if paramName == "previous":
        viewer.seek(viewer.getCurrentFrame() - 1)
    elif paramName == "backward":
        viewer.startBackward()
    elif paramName == "forward":
        viewer.startForward()
    elif paramName == "next":
        viewer.seek(viewer.getCurrentFrame() + 1)
    elif paramName == "stop":
        viewer.pause()

def createMyPlayer():
    app.player = NatronGui.PyPanel("fr.inria.myplayer", "My Player", True, app)
    #...
    app.player.setParamChangedCallback("myPlayerParamChangedCallback")
```

The After input changed callback

Similarly to the param changed callback, this function is called whenever an input connection of the node is changed. The signature is:

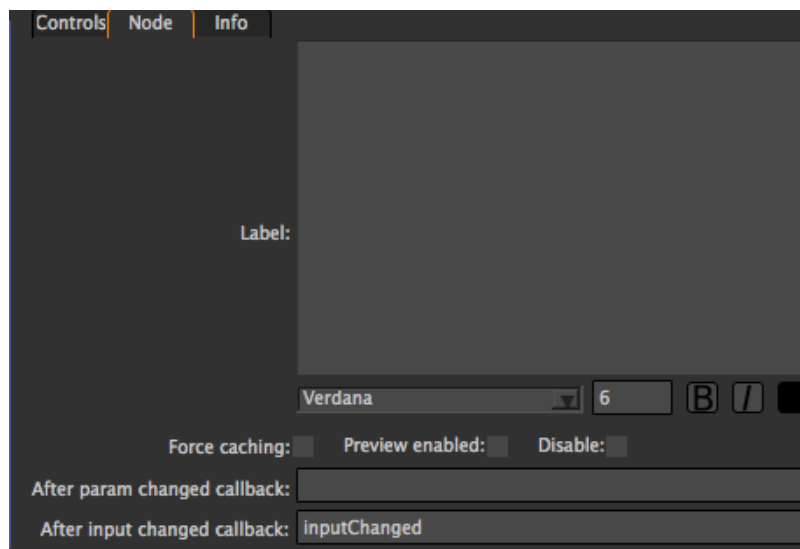
```
callback(inputIndex, thisNode, thisGroup, app)
```

Note: This function will be called even when loading a project

- **inputIndex** : This is the input which just got connected/disconnected. You can fetch the input at the given index with the `getInput (index)` function of the *Effect* class.
- **thisNode** : This is a *Effect* holding the input which just changed
- **thisGroup** : This is a *Effect* pointing to the group holding **thisNode**. Note that it will be declared only if **thisNode** is part of a group.
- **app** : points to the correct *application instance*.

Registering the input changed callback

To register the input changed callback of an *Effect*, you can do so in the settings panel of the node, in the “Node” tab, by entering the name of your Python function:



You can also set the callback directly from the script: The callback is just another *parameter* of the node, on which you can call `setValue (value)` to set the name of the callback

```
def inputChangedCallback(inputIndex, thisNode, thisGroup, app):
    ...

app.Merge1.onInputChanged.set ("inputChangedCallback")
```

Example

```
# This simple callback just prints the input node name if connected or "None"
↳ otherwise
# node changes
def inputChangedCallback(inputIndex, thisNode, thisGroup, app):
    inp = thisNode.getInput(inputIndex)
    if not inp is None:
        print("Input ", inputIndex, " is ", inp.getScriptName())
    else:
        print("Input ", inputIndex, " is None")
```

(continues on next page)

(continued from previous page)

```
app.Merge1.onInputChanged.set ("inputChangedCallback")
```

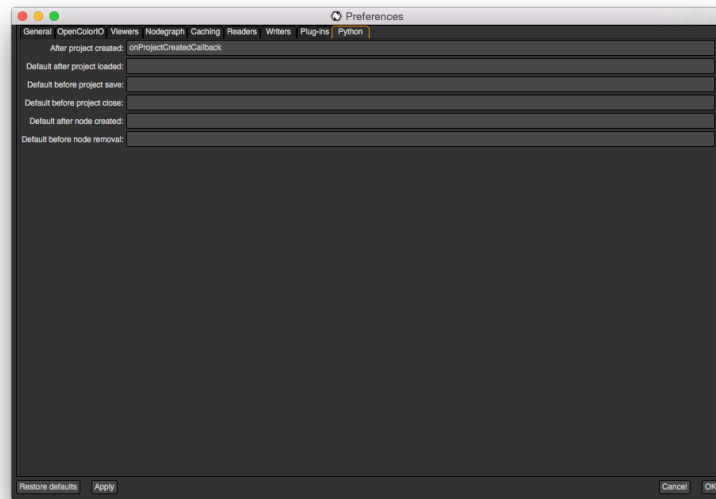
The After project created callback

This function is called whenever a new project is created, that is either when launching Natron without loading a project, or when clicking “Create a new project” or “Close project”.

Note: Note that this function is never called when a project is loaded either via an auto-save or from user interaction.

The **app** variable will be set so it points to the correct *application instance* being created.

You can set the callback via the *afterProjectCreated* parameter of the settings of Natron.



This is a good place to create custom panels and/or setup the node-graph with node presets.

Example, taken from the `initGui.py` script provided as example in [this section](#):

```
def onProjectCreated() :

    #Always create our icon viewer on project creation
    createIconViewer()

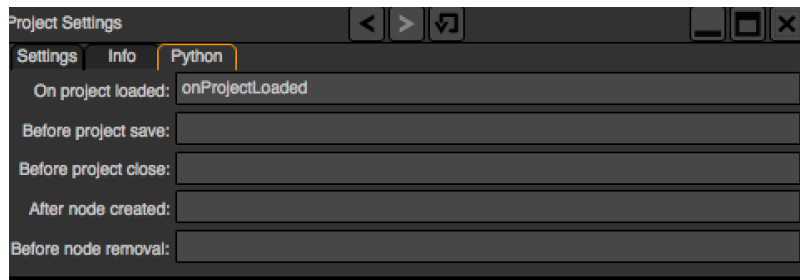
natron.settings.afterProjectCreated.set ("onProjectCreated")
```

The After project loaded callback

This function is very similar to the After project created callback but is a per-project callback, called only when a project is loaded from an auto-save or from user interaction. The signature is:

```
callback (app)
```

- **app** : points to the correct *application instance* being loaded.



You can set this callback in the project settings:

This is a good place to do some checks to opened projects or to setup something:

```
def onProjectLoaded(app) :

    if not natron.isBackground() :
        if app.getUserPanel("fr.inria.iconviewer") is None:
            createIconViewer()

app.afterProjectLoad.set("onProjectLoaded")
```

Note: You can set a default After project loaded callback for all new projects in the *Preferences*→*Python* tab.

The Before project save callback

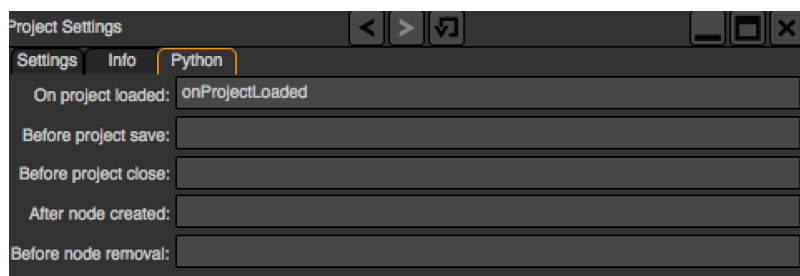
This function will be called prior to saving a project either via an auto-save or from user interaction. The signature is:

```
callback(filename, app, autoSave)
```

- **filename** : This is the file-path where the project is initially going to be saved.
- **app** : points to the correct *application instance* being created.
- **autoSave** : This indicates whether the save was originated from an auto-save or from user interaction.

Warning: This function should return the filename under which the project should really be saved.

You can set the callback from the project settings:



```
def beforeProjectSave(filename, app, autoSave):
    print("Saving project under: ",filename)
    return filename

app.beforeProjectSave.set("beforeProjectSave")
```

Note: You can set a default Before project save callback for all new projects in the *Preferences*→*Python* tab.

The Before project close callback

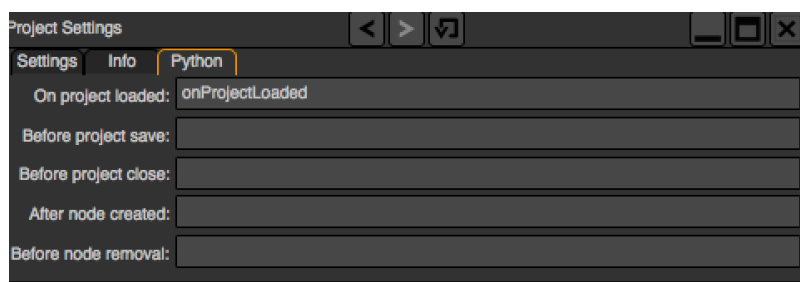
This function is called prior to closing a project either because the application is about to quit or because the user closed the project. The signature is:

```
callback (app)
```

- **app** : points to the correct *application instance* being closed.

This function can be used to synchronize any other device or piece of software communicating with Natron.

You can set the callback from the project settings:



```
def beforeProjectClose (app) :
    print ("Closing project")

app.beforeProjectClose.set ("beforeProjectClose")
```

Note: You can set a default Before project close callback for all new projects in the *Preferences*→*Python* tab.

The After node created callback

This function is called after creating a node in Natron. The signature is:

```
callback (thisNode, app, userEdited)
```

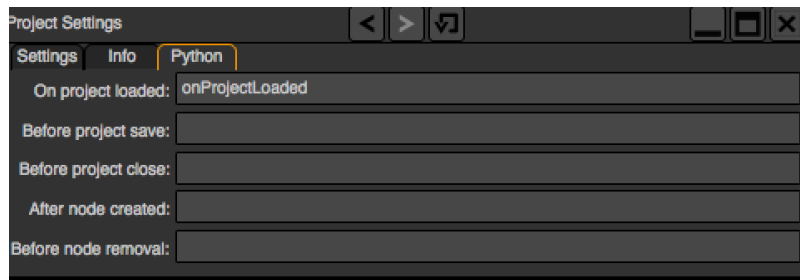
- **thisNode** points to the *node* that has been created.
- **app** points to the correct *application instance*.
- **userEdited** will be *True* if the node was created by the user (or by a script using the `createNode(pluginID, version, group)` function) or *False* if the node was created by actions such as pasting a node or when the project is loaded.

This is a good place to change default parameters values.

You can set the callback from the project settings:

```
def onNodeCreated (thisNode, app, userEdited) :
    print (thisNode.getScriptName(), " was just created")
    if userEdited:
        print (" due to user interaction")
    else:
        print (" due to project load or node pasting")

app.afterNodeCreated.set ("onNodeCreated")
```



Note: You can set a default After node created callback for all new projects in the *Preferences*→*Python* tab.

This callback can also be set in the *Node* tab of any **Group** node (or *PyPlug*). If set on the Group, the callback will be invoked for the *Group* node and all its direct children (not recursively).

The Before node removal callback:

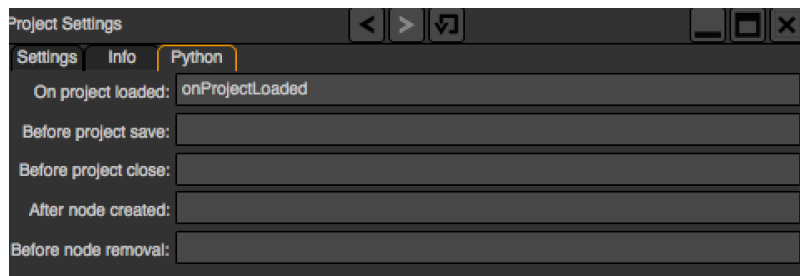
This function is called prior to deleting a node in Natron. The signature is:

```
callback(thisNode, app)
```

- **thisNode** : points to the *node* about to be deleted.
- **app** : points to the correct *application instance*.

Warning: This function will **NOT** be called when the project is closing

You can set the callback from the project settings:



```
def beforeNodeDeleted(thisNode, app):
    print(thisNode.getScriptName(), " is going to be destroyed")

app.beforeNodeRemoval.set("beforeNodeDeleted")
```

Note: You can set a default Before node removal callback for all new projects in the *Preferences*→*Python* tab.

This callback can also be set in the *Node* tab of any **Group** node (or *PyPlug*). If set on the Group, the callback will be invoked for the *Group* node and all its direct children (not recursively).

The Before frame render callback:

This function is called prior to rendering any frame with a Write node. The signature is:


```
callback(frame, thisNode, app)
```

- **thisNode** : points to the *write node*.
- **app** : points to the correct *application instance*.
- **frame**: The frame that is about to be rendered

To execute code specific when in background render mode or in GUI mode, use the following condition

```
if natron.isBackground():
    #We are in background mode
```

You can set the callback from the Write node settings panel in the “Python” tab.



This function can be used to communicate with external programs for example.

Warning: Any exception thrown in this callback will abort the render

The After frame rendered callback:

This function is called after each frame is finished rendering with a Write node. The signature is:

```
callback(frame, thisNode, app)
```

- **thisNode** : points to the *write node*.
- **app** : points to the correct *application instance*.
- **frame**: The frame that is about to be rendered

To execute code specific when in background render mode or in GUI mode, use the following condition

```
if natron.isBackground():
    #We are in background mode
```

You can set the callback from the Write node settings panel in the “Python” tab.



This function can be used to communicate with external programs for example.

Warning: Any exception thrown in this callback will abort the render

The Before render callback:

This function is called once before starting rendering the first frame of a sequence with the Write node. The signature is:

```
callback(frame, thisNode, app)
```

- **thisNode** : points to the *write node*.
- **app** : points to the correct *application instance*.

To execute code specific when in background render mode or in GUI mode, use the following condition

```
if natron.isBackground() :  
    #We are in background mode
```

You can set the callback from the Write node settings panel in the “Python” tab.



This function can be used to communicate with external programs for example.

Warning: Any exception thrown in this callback will abort the render

The After render callback:

This function is called once after the rendering of the last frame is finished with the Write node or if the render was aborted. The signature is:

```
callback(aborted, thisNode, app)
```

- **aborted** : *True* if the rendering was aborted or *False* otherwise.
- **thisNode** : points to the *write node*.
- **app** : points to the correct *application instance*.

To execute code specific when in background render mode or in GUI mode, use the following condition

```
if natron.isBackground() :  
    #We are in background mode
```

You can set the callback from the Write node settings panel in the “Python” tab.



This function can be used to communicate with external programs for example.

3.3.6 Rendering

To start rendering in Natron you need to use the `render(effect, firstFrame, lastFrame, frameStep)` or `render(tasks)` functions of the [App](#) class. The parameters passed are:

- The *writeNode*: This should point to the node you want to start rendering with
- The *firstFrame*: This is the first frame to render in the sequence
- The *lastFrame*: This is the last frame to render in the sequence
- The *frameStep*: This is the number of frames the timeline should step before rendering a new frame, e.g. To render frames 1,3,5,7,9, you can use a `frameStep` of 2

Natron always renders from the *firstFrame* to the *lastFrame*. Generally Natron uses multiple threads to render concurrently several frames, you can control this behaviour with the parameters in the [settings](#).

Let's imagine there's a node called **Write1** in your project and that you want to render frames 20 to 50 included, you would call it the following way:

```
app.render(app.Write1, 20, 50)
```

Note: Note that when the render is launched from a [GuiApp](#), it is not *blocking*, i.e: this function will return immediately even though the render is not finished.

On the other hand, if called from a [background application](#), this call will be blocking and return once the render is finished.

If you need to have a blocking render whilst using Natron Gui, you can use the `renderBlocking()` function but bear in mind that it will freeze the user interface until the render is finished.

This function can take an optional *frameStep* parameter:

```
#This will render frames 1,4,7,10,13,16,19
app.render(app.Write1, 1, 20, 3)
```

You can use the [after render callback](#) to call code to be run once the render is finished.

For convenience, the [App](#) class also have a `render(tasks)` function taking a sequence of tuples (Effect,int,int) (or (Effect,int,int,int) to specify a *frameStep*).

Let's imagine we were to render 2 write nodes concurrently, we could do the following call:

```
app.render([ (app.Write1, 1, 10),
            (app.WriteFFmpeg1, 1, 50, 2) ])
```

Note: The same restrictions apply to this variant of the render function: it is blocking in background mode and not blocking in GUI mode.

When executing multiple renders with the same call, each render is called concurrently from the others.

Using the DiskCache node

All the above can be applied to the **DiskCache** node to pre-render a sequence. Just pass the **DiskCache** node instead of the **Write** node to the render function.

3.3.7 Using the roto-scoping functionalities

All roto-scoping functionalities are gathered in the [Roto](#) class. For now, only the roto node can have a [Roto](#) object. The [Roto](#) object is *auto-declared* by Natron and can be accessed as an attribute of the roto node:

```
app.Roto1.roto
```

All the objects hierarchy in the *Roto* object is broken up in 2 classes:

- *BezierCurve*: This class represents a single bezier, may it be an ellipse, rectangle or bezier.
- *Layer* : This is a container for BezierCurves and Layers

Beziers and *layers* can be accessed via their script-name directly:

```
app.Roto1.roto.Layer1.Bezier1
```

The *script-name* of the roto items can be found in the *settings panel* of the Roto node.

Moving items within layers

In Natron, all the items in a layer are rendered from top to bottom, meaning the bottom-most items will always appear on top of the others.

You can re-organize the tree using the functions available in the *Layer* class.

Warning: Removing an item from a layer or inserting it in a layer will change the auto-declared variable, e.g.:

```
fromLayer = app.Roto1.roto.Layer1 toLayer = app.Roto1.roto.Layer2 item =
app.Roto1.roto.Layer1.Bezier1 toLayer.addItem(item)

#Now item is referenced from app.Roto1.roto.Layer2.Bezier1
```

Creating layers

To create a new *BezierCurve*, use the `createLayer()` function made available by the *Roto* class.

Creating shapes

To create a new *BezierCurve*, use one of the following functions made available by the *Roto* class:

- `createBezier(x,y,time)`
- `createEllipse(x,y,diameter,fromCenter,time)`
- `createRectangle(x,y,size,time)`

Once created, the Bezier will have at least 1 control point (4 for ellipses and rectangles) and one keyframe at the time specified in parameter.

A Bezier initially is in an *opened* state, meaning it doesn't produce a shape yet (unless it is a rectangle or ellipse). At this stage you can then add control points using the `func'addControlPoint(x,y)<NatronEngine.BezierCurve.addControlPoint>'` function. Once you're one adding control points, call the function `setCurveFinished(finished)` to close the shape by connecting the last control point with the first.

Once finished, you can refine the Bezier curve by adding control points with the `addControlPointOnSegment(index,t)` function. You can then move and remove control points of the Bezier.

You can also slave a control point to a track using the `slavePointToTrack(index,trackTime,trackCenter)` function.

A Bezier curve has several properties that the API allows you to modify:

- `opacity`

- color
- feather distance
- feather fall-off
- enable state
- overlay color
- compositing operator

Most of them are available via a *parameter*, e.g.:

```
colorParam = bezier.getColorParam() bezierColor = colorParam.get(time)
```

3.3.8 Using the tracker functionalities

All tracking functionalities are gathered in the *Tracker* class. For now, only the tracker node can have a *Tracker* object. The *Tracker* object is *auto-declared* by Natron and can be accessed as an attribute of the tracker node:

```
app.Tracker1.tracker
```

The tracker object itself is a container for *tracks*. The *Track* class represent one marker as visible by the user on the viewer.

Tracks can be accessed via their script-name directly:

```
app.Tracker1.tracker.track1
```

The *script-name* of the tracks can be found in the *settings panel* of the Tracker node.

Getting data out of the tracks:

In Natron, a *track* contains internally just *parameters* which can hold animated data just like regular parameters of the *effect class*

You can access the parameters directly with their script-name:

```
app.Tracker1.tracker.track1.centerPoint
```

Or you can use the `getParam(paramScriptName)` function:

```
app.Tracker1.tracker.track1.getParam("centerPoint")
```

Here is an example that retrieves all keyframes available on the center point for a given track:

```
myTrack = app.Tracker1.tracker.track1
keyframes = []

# get the number of keys for the X dimension only and try match the Y keyframes
nKeys = myTrack.centerPoint.getNumKeys(0)
for k in range(0,nKeys):

    # getKeyTime returns a tuple with a boolean value indicating if it succeeded,
    ↪and
    # the keyframe time

    gotXKeyTuple = myTrack.centerPoint.getKeyTime(k, 0)
    frame = gotXKeyTuple[1]

    # Only consider keyframes which have an X and Y value
```

(continues on next page)

(continued from previous page)

```
# If Y does not have a keyframe at this frame, ignore the keyframe
# getKeyIndex returns a value >=0 if there is a keyframe
yKeyIndex = myTrack.centerPoint.getKeyIndex(frame, 1)

if yKeyIndex == -1:
    continue

# Note that even if the x curve or y curve didn't have a keyframe we
# could still call getValueAtTime but the value would be interpolated by
# Natron with surrounding keyframes, which is not what we want.

x = myTrack.centerPoint.getValueAtTime(frame, 0)
y = myTrack.centerPoint.getValueAtTime(frame, 1)

keyframes.append((x,y))

print keyframes
```

Creating Tracks

To create a new *track*, use the `createTrack()` function made available by the *Tracker* class. You can then set values on parameters much like everything else in Natron.

3.3.9 Modal dialogs

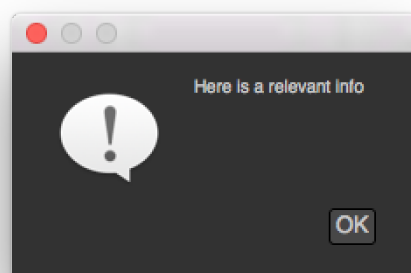
Modal dialogs are windows (or popup) that inform the user about something or ask for some informations and that does not allow any other action to be performed while the dialog is opened.

This can be used as a quick way to retrieve user inputs.

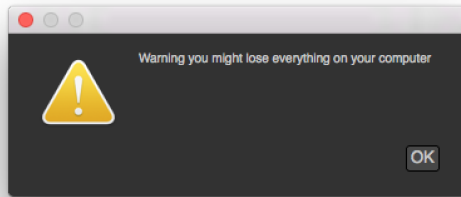
Simple dialogs

The most simple dialogs in Natron are the information/warning/error/question dialog which basically just take some text in input and may return a reply from the user

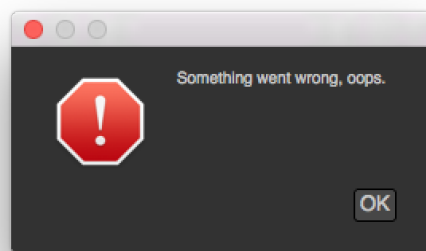
```
natron.informationDialog("Info", "Here is a relevant info")
```



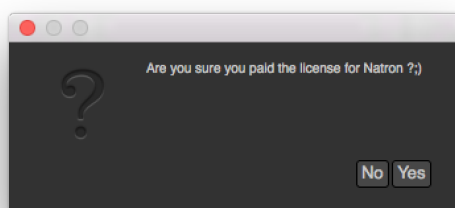
```
natron.warningDialog("Warning", "Warning you might lose everything on your computer  
↪")
```



```
natron.errorDialog("Error", "Something went wrong, oops.")
```



```
reply = natron.questionDialog("Question", "Are you sure you paid the license for_\n↩Natron ?;)")
if reply == NatronEngine.Natron.StandardButtonEnum.eStandardButtonNo:
    ...
elif reply == NatronEngine.Natron.StandardButtonEnum.eStandardButtonYes:
    ...
```



More refined dialogs

To create dialogs that may request some informations such as colors, frame range, coordinates or text input, you can create *modal dialogs*.

Basically you can add *user parameters*, and retrieve their value afterwards when the user pressed **OK**.

You can start adding user parameters using all the `createXParam` functions inherited from the `UserParamHolder` class. See the documentation of the *PyModalDialog* for more information:

```
dialog = app.createModalDialog() myInteger = dialog.createIntParam("myInt","This is an integer
very important") myInteger.setAnimationEnabled(False) myInteger.setAddNewLine(False)

#Create a boolean on the same line myBoolean = dialog.createBooleanParam("myBool","Yet another
important boolean")

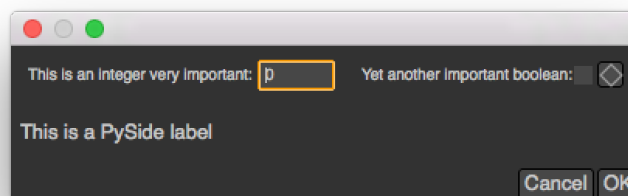
dialog.refreshUserParamsGUI()
```

You can also add custom PySide widgets that can be inserted **after** any user parameter(s) using the `addWidget(widget)` and `insertWidget(index,widget)` functions.

```
label = QLabel("This is a PySide label")
dialog.addWidget(label)
```

To make the dialog show-up, use the `exec()` function on the dialog. This function will return once the user pressed either "OK" or "Canceled":

```
if dialog.exec():
    #User pressed OK
```



You can add a custom callback when a parameter changes, for instance to hide another parameter:

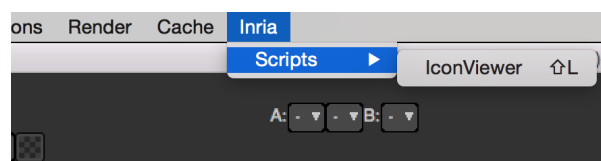
```
#Callback called when a parameter of changes
#The variable paramName is declared by Natron; indicating the name of the
↳parameter which just had its value changed
def paramChangedCallback():
    if paramName == "myBool":
        myInteger.setVisible(myBoolean.get())

dialog.setParamChangedCallback("paramChangedCallback")
```

3.3.10 User menu commands

In Natron you can add multiple menu commands that will then be available to the user via the menu. You can also assign it a shortcut and the user will be able to modify it via the shortcuts editor.

Project-wide menu commands:



To add a project-wipe menu command to the application's menu-bar, you need to use the `addMenuCommand(grouping,function,key,modifiers)` of the `PyGuiApplication` class to register it:


```
def createIconViewer() :
    ...

#Add a custom menu entry with a shortcut to create our icon viewer
NatronGui.natron.addMenuCommand("Inria/Scripts/IconViewer", "createIconViewer",
    ↪QtCore.Qt.Key.Key_L, QtCore.Qt.KeyboardModifier.ShiftModifier)
```

Note that this function is to be called on the whole **application** via the **natron** variable and is not *per-project* unlike most functions that are generally called on the **app** object.

Warning: This function can only be called in the startup script **init.py** and will have no effect otherwise. This is not a dynamic function and will not create menu entries on the fly.

3.3.11 PySide panels

To create a non-modal *panel* that can be saved in the project's layout and docked into the application's *tab-widgets*, there is 2 possible way of doing it:

- Sub-class *PyPanel* and create your own GUI using *PySide*
- Use the API proposed by *PyPanel* to add custom user *parameters* as done for *PyModalDialog*.

Generally you should define your panels in the **initGui.py** script (see *startup-scripts*). You can also define the panel in the *Script Editor* at run-time of Natron, though this will not persist when Natron is closed.

To make your panel be created upon new project created, register a Python callback in the *Preferences*→*Python* tab in the parameter *After project created*. This callback will not be called for project being loaded either via an auto-save or via a user action.

```
#This goes in initGui.py

def createMyPanel() :
    #Create panel
    ...

def onProjectCreatedCallback() :
    createMyPanel()
```

Warning: When the **initGui.py** script is executed, the *app* variable (or any derivative such as *app1 app2* etc...) does not exist since no project is instantiated yet. The purpose of the script is not to instantiate the GUI per-se but to define classes and functions that will be used later on by *application instances*.

Python panels can be re-created for existing projects using serialization functionalities explained [here](#) See the example below (the whole script is available attached below)

```
# We override the save() function and save the filename
def save(self) :
    return self.locationEdit.text()

# We override the restore(data) function and restore the current image
def restore(self, data) :

    self.locationEdit.setText(data)
    self.label.setPixmap(QPixmap(data))
```

The sole requirement to save a panel in the layout is to call the `registerPythonPanel(panel, function)` function of *GuiApp*:

```
app.registerPythonPanel (app.mypanel, "createIconViewer")
```

See the details of the *PyPanel* class for more explanation on how to sub-class it.

Also check-out the complete example *source code* below.

Using user parameters:

Let's assume we have no use to make our own widgets and want quick *parameters* fresh and ready, we just have to use the *PyPanel* class without sub-classing it:

```
#Callback called when a parameter of the player changes
#The variable paramName is declared by Natron; indicating the name of the
↪parameter which just had its value changed
def myPlayerParamChangedCallback():

    viewer = app.getViewer("Viewer1")
    if viewer == None:
        return
    if paramName == "previous":
        viewer.seek(viewer.getCurrentFrame() - 1)

def createMyPlayer():

    #Create a panel named "My Panel" that will use user parameters
    app.player = NatronGui.PyPanel("fr.inria.myplayer", "My Player", True, app)

    #Add a push-button parameter named "Previous"
    app.player.previousFrameButton = app.player.createButtonParam("previous",
↪"Previous")

    #Refresh user parameters GUI, necessary after changes to static properties of
↪parameters.
    #See the Param class documentation
    app.player.refreshUserParamsGUI()

    #Set a callback that will be called upon parameter change
    app.player.setParamChangedCallback("myPlayerParamChangedCallback")
```

Note: For convenience, there is a way to also add custom widgets to python panels that are using user parameters with the `addWidget(widget)` and `insertWidget(index, widget)` functions. However the widgets will be appended **after** any user parameter defined.

Managing panels and panes

Panels in Natron all have an underlying script-name, that is the one you gave as first parameter to the constructor of *PyPanel*.

You can then move the *PyPanel* between the application's *panes* by calling the function `moveTab(scriptName, pane)` of *GuiApp*.

Note: All application's panes are *auto-declared* by Natron and can be referenced directly by a variable, such as:

```
app.pane2
```

Panels also have a script-name but only *viewers* and *user panels* are auto-declared by Natron:

```
app.pane2.Viewer1
app.pane1.myPySidePanelScriptName
```

Source code of the example initGui.py

```
#This Source Code Form is subject to the terms of the Mozilla Public
#License, v. 2.0. If a copy of the MPL was not distributed with this
#file, You can obtain one at http://mozilla.org/MPL/2.0/. */
#Created by Alexandre GAUTHIER-FOICHAT on 01/27/2015.

#PySide is already imported by Natron, but we remove the cumbersome PySide.QtGui
↳and PySide.QtCore prefix
from PySide.QtGui import *
from PySide.QtCore import *

#To import the variable "natron"
from NatronGui import *

#Callback called when a parameter of the player changes
#The variable paramName is declared by Natron; indicating the name of the
↳parameter which just had its value changed
def myPlayerParamChangedCallback(paramName, app, userEdited):

    viewer = app.getViewer("Viewer1")
    if viewer == None:
        return
    if paramName == "previous":
        viewer.seek(viewer.getCurrentFrame() - 1)
    elif paramName == "backward":
        viewer.startBackward()
    elif paramName == "forward":
        viewer.startForward()
    elif paramName == "next":
        viewer.seek(viewer.getCurrentFrame() + 1)
    elif paramName == "stop":
        viewer.pause()

def createMyPlayer():

    app.player = NatronGui.PyPanel("fr.inria.myplayer", "My Player", True, app)
    app.player.previousFrameButton = app.player.createButtonParam("previous",
↳"Previous")
    app.player.previousFrameButton.setAddNewLine(False)

    app.player.playBackwardButton = app.player.createButtonParam("backward", "Rewind
↳")
    app.player.playBackwardButton.setAddNewLine(False)

    app.player.stopButton = app.player.createButtonParam("stop", "Pause")
    app.player.stopButton.setAddNewLine(False)

    app.player.playForwardButton = app.player.createButtonParam("forward", "Play")
    app.player.playForwardButton.setAddNewLine(False)

    app.player.nextFrameButton = app.player.createButtonParam("next", "Next")

    app.player.helpLabel = app.player.createStringParam("help", "Help")
    app.player.helpLabel.setType(NatronEngine.StringParam.TypeEnum.
↳eStringTypeLabel)
```

(continues on next page)

(continued from previous page)

```

app.player.helpLabel.set("<br><b>Previous:</b> Seek the previous frame on the_
↳ timeline</br>"
                                "<br><b>Rewind:</b> Play backward</br>"
                                "<br><b>Pause:</b> Pauses the playback</br>"
                                "<br><b>Play:</b> Play forward</br>"
                                "<br><b>Next:</b> Seek the next frame on the timeline</br>
↳ ")

app.player.refreshUserParamsGUI()
app.player.setParamChangedCallback("myPlayerParamChangedCallback")

#Add it to the "pane2" tab widget
app.pane2.appendTab(app.player);

#Register the tab to the application, so it is saved into the layout of the_
↳ project
#and can appear in the Panes sub-menu of the "Manage layout" button (in top_
↳ left-hand corner of each tab widget)
app.registerPythonPanel(app.player, "createMyPlayer")

#A small panel to load and visualize icons/images
class IconViewer(NatronGui.PyPanel):

    #Register a custom signal
    userFileChanged = QtCore.Signal()

    #Slots should be decorated:
    #http://qt-project.org/wiki/Signals_and_Slots_in_PySide

    #This is called upon a user click on the button
    @QtCore.Slot()
    def onButtonClicked(self):
        location = self.currentApp.getFilenameDialog(("jpg", "png", "bmp", "tif"))
        if location:
            self.locationEdit.setText(location)

            #Save the file
            self.onUserDataChanged()

            self.userFileChanged.emit()

    #This is called when the user finish editing of the line edit (when return is_
    ↳ pressed or focus out)
    @QtCore.Slot()
    def onLocationEditEditingFinished(self):
        #Save the file
        self.onUserDataChanged()
        self.userFileChanged.emit()

    #This is called when our custom userFileChanged signal is emitted
    @QtCore.Slot()
    def onFileChanged(self):
        self.label.setPixmap(QPixmap(self.locationEdit.text()))

    def __init__(self, scriptName, label, app):

        #Init base class, important! otherwise signals/slots won't work.
        NatronGui.PyPanel.__init__(self, scriptName, label, False, app)

```

(continues on next page)

(continued from previous page)

```

        #Store the current app as it might no longer be pointing to the app at the
↪time being called
        #when a slot will be invoked later on
        self.currentApp = app

        #Set the layout
        self.setLayout( QVBoxLayout() )

        #Create a widget container for the line edit + button
        fileContainer = QWidget(self)
        fileLayout = QHBoxLayout()
        fileContainer.setLayout(fileLayout)

        #Create the line edit, make it expand horizontally
        self.locationEdit = QLineEdit(fileContainer)
        self.locationEdit.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.
↪Preferred)

        #Create a pushbutton
        self.button = QPushButton(fileContainer)
        #Decorate it with the open-file pixmap built-in into Natron
        buttonPixmap = natron.getIcon(NatronEngine.Natron.PixmapEnum.NATRON_PIXMAP_
↪OPEN_FILE)
        self.button.setIcon(QIcon(buttonPixmap))

        #Add widgets to the layout
        fileLayout.addWidget(self.locationEdit)
        fileLayout.addWidget(self.button)

        #Use a QLabel to display the images
        self.label = QLabel(self)

        #Init the label with the icon of Natron
        natronPixmap = natron.getIcon(NatronEngine.Natron.PixmapEnum.NATRON_PIXMAP_
↪APP_ICON)
        self.label.setPixmap(natronPixmap)
        #Built-in icons of Natron are in the resources
        self.locationEdit.setText(":/Resources/Images/natronIcon256_linux.png")

        #Make it expand in both directions so it takes all space
        self.label.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Expanding)

        #Add widgets to the layout
        self.layout().addWidget(fileContainer)
        self.layout().addWidget(self.label)

        #Make signal/slot connections
        self.button.clicked.connect(self.onButtonClicked)
        self.locationEdit.editingFinished.connect(self.
↪onLocationEditEditingFinished)
        self.userFileChanged.connect(self.onFileChanged)

        # We override the save() function and save the filename
        def save(self):
            return self.locationEdit.text()

        # We override the restore(data) function and restore the current image
        def restore(self, data):

            self.locationEdit.setText(data)
            self.label.setPixmap(QPixmap(data))

```

(continues on next page)

(continued from previous page)

```
#To be called to create a new icon viewer panel:
#Note that *app* should be defined. Generally when called from
↳onProjectCreatedCallback
#this is set, but when called from the Script Editor you should set it yourself
↳beforehand:
#app = app1
#See http://natron.readthedocs.org/en/python/natronobjects.html for more info
def createIconViewer():

    if hasattr(app, "p"):
        #The icon viewer already exists, if we override the app.p variable, then
↳it will destroy the previous widget
        #and create a new one but we don't really need it

        #The warning will be displayed in the Script Editor
        print("Note for us developers: this widget already exists!")
        return

    #Create our icon viewer
    app.p = IconViewer("fr.inria.iconViewer", "Icon viewer", app)

    #Add it to the "pane2" tab widget
    app.pane2.appendTab(app.p);

    #Register the tab to the application, so it is saved into the layout of the
↳project
    #and can appear in the Panes sub-menu of the "Manage layout" button (in top
↳left-hand corner of each tab widget)
    app.registerPythonPanel(app.p, "createIconViewer")

#Callback set in the "After project created" parameter in the Preferences-->Python
↳tab of Natron
#This will automatically create our panels when a new project is created
def onProjectCreatedCallback(app):
    #Always create our icon viewer on project creation, you must register this
↳call-back in the
    #"After project created callback" parameter of the Preferences-->Python tab.
    createIconViewer()

    createMyPlayer()

#Add a custom menu entry with a shortcut to create our icon viewer
natron.addMenuCommand("Inria/Scripts/IconViewer", "createIconViewer", QtCore.Qt.Key.
↳Key_L, QtCore.Qt.KeyboardModifier.ShiftModifier)
```

3.3.12 Controlling the viewer

Natron exposes all functionalities available to the user in the Python API via the *PyViewer* class.

To retrieve a *PyViewer*, use the *auto-declared* variable:

```
app.pane2.Viewer1
```

or use the following function `getViewer(scriptName)`, passing it the *script-name* of a viewer node.

You can then control the player, the displayed channels, the current view, the current compositing operator, which are the input A and B, the frame-range, the proxy level and various other stuff.

Bibliography

- [Wikipedia] [Compositing](https://en.wikipedia.org/wiki/Compositing), in Wikipedia, retrieved Sep. 14, 2016 from <https://en.wikipedia.org/wiki/Compositing>
- [PorterDuff1984] Porter, Thomas; Tom Duff (1984). “[Compositing Digital Images](#)”. *Computer Graphics*. 18 (3): 253–259. doi:10.1145/800031.808606
- [Wallace1981] Wallace, Bruce A., [Merging and Transformation of Raster Images for Cartoon Animation](#), *Computer Graphics*, Vol 15, No 3, Aug 1981, 253-262. SIGGRAPH’81 Conference Proceedings, doi:10.1145/800224.806813.
- [Blinn1994a] Blinn, James F., [Jim Blinn’s Corner: Compositing Part 1: Theory](#), *IEEE Computer Graphics & Applications*, Sep 1994, 83-87, doi:10.1109/38.310740.
- [Blinn1994b] Blinn, James F., [Jim Blinn’s Corner: Compositing Part 2: Practice](#), *IEEE Computer Graphics & Applications*, Nov 1994, 78-82, doi:10.1109/38.329100.
- [Smith1995] Alvy Ray Smith, [Image Compositing Fundamentals](#), 1995.
- [Brinkmann2008] Ron Brinkmann, *The Art and Science of Digital Compositing*, 2nd Edition, 2008 (ISBN 0123706386)
- [Lanier2009] Lee Lanier, *Professional Digital Compositing: Essential Tools and Techniques*, 2009 (ISBN 0470452617)
- [Wright2010] Steve Wright, *Digital Compositing for Film and Video*, Third Edition, 2010 (ISBN 78-0-240-81309-7)
- [VES2014] *The VES Handbook of Visual Effects: Industry Standard VFX Practices and Procedures*, 2nd Edition (ISBN 0240825187)

n

NatronEngine, [1157](#)

NatronGui, [1258](#)

Symbols

`__getitem__()` (*NatronEngine.NatronEngine.ColorTuple* method), 1184
`__getitem__()` (*NatronEngine.NatronEngine.Double2DTuple* method), 1186
`__getitem__()` (*NatronEngine.NatronEngine.Double3DTuple* method), 1187
`__getitem__()` (*NatronEngine.NatronEngine.Int2DTuple* method), 1210
`__getitem__()` (*NatronEngine.NatronEngine.Int3DTuple* method), 1212

A

`a` (*NatronEngine.NatronEngine.ColorTuple* attribute), 1184
`addControlPoint()` (*NatronEngine.NatronEngine.BezierCurve* method), 1170
`addControlPoint()` (*NatronEngine.NatronEngine.ParametricParam* method), 1231
`addControlPointOnSegment()` (*NatronEngine.NatronEngine.BezierCurve* method), 1170
`addFormat()` (*NatronEngine.NatronEngine.App* method), 1162
`addItem()` (*NatronEngine.NatronEngine.Layer* method), 1218
`addMenuCommand()` (*NatronGui.NatronGui.PyGuiApplication* method), 1264
`addOption()` (*NatronEngine.NatronEngine.ChoiceParam* method), 1178
`addParam()` (*NatronEngine.NatronEngine.GroupParam* method), 1206
`addParam()` (*NatronEngine.NatronEngine.PageParam* method),

1221
`addProjectLayer()` (*NatronEngine.NatronEngine.App* method), 1162
`addUserPlane()` (*NatronEngine.NatronEngine.Effect* method), 1193
`addWidget()` (*NatronGui.NatronGui.PyModalDialog* method), 1267
`addWidget()` (*NatronGui.NatronGui.PyPanel* method), 1270
`AnimationLevelEnum` (*NatronEngine.NatronEngine.Natron* attribute), 1220
`appendTab()` (*NatronGui.NatronGui.PyTabWidget* method), 1274
`appendToNatronPath()` (*NatronEngine.NatronEngine.PyCoreApplication* method), 1237
`area()` (*NatronEngine.NatronEngine.RectD* method), 1240

B

`b` (*NatronEngine.NatronEngine.ColorTuple* attribute), 1184
`beginChanges()` (*NatronEngine.NatronEngine.Effect* method), 1193
`bottom()` (*NatronEngine.NatronEngine.RectD* method), 1240
`bottom()` (*NatronEngine.NatronEngine.RectI* method), 1243
`boxstep()` (*NatronEngine.NatronEngine.ExprUtils* method), 1199

C

`CairoOperatorEnum` (*NatronEngine.NatronEngine.BezierCurve* attribute), 1170
`canConnectInput()` (*NatronEngine.NatronEngine.Effect* method), 1193
`ccellnoise()` (*NatronEngine.NatronEngine.ExprUtils* method),

1204			
cellnoise()	(<i>NatronEngine.NatronEngine.ExprUtils</i> method), 1204	<i>nEngine.NatronEngine.UserParamHolder</i> method), 1254	
cfbm()	(<i>NatronEngine.NatronEngine.ExprUtils</i> method), 1203	<i>nEngine.NatronEngine.UserParamHolder</i> method), 1255	
cfbm4()	(<i>NatronEngine.NatronEngine.ExprUtils</i> method), 1204	createEllipse()	(<i>NatronEngine.NatronEngine.Roto</i> method), 1246
clear()	(<i>NatronEngine.NatronEngine.RectD</i> method), 1240	createFileParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1255
clear()	(<i>NatronEngine.NatronEngine.RectI</i> method), 1243	createGroupParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1255
clearSelection()	(<i>NatronGui.NatronGui.GuiApp</i> method), 1262	createInt2DParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1255
closeCurrentTab()	(<i>NatronGui.NatronGui.PyTabWidget</i> method), 1274	createInt3DParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1256
closePane()	(<i>NatronGui.NatronGui.PyTabWidget</i> method), 1274	createIntParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1256
closeProject()	(<i>NatronEngine.NatronEngine.App</i> method), 1167	createLayer()	(<i>NatronEngine.NatronEngine.Roto</i> method), 1246
closeTab()	(<i>NatronGui.NatronGui.PyTabWidget</i> method), 1274	createModalDialog()	(<i>NatronGui.NatronGui.GuiApp</i> method), 1259
cnoise()	(<i>NatronEngine.NatronEngine.ExprUtils</i> method), 1201	createNode()	(<i>NatronEngine.NatronEngine.App</i> method), 1162
cnoise4()	(<i>NatronEngine.NatronEngine.ExprUtils</i> method), 1201	createOutputFileParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1256
ColorTuple (class in <i>NatronEngine</i>), 1184		createPageParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1257
connectInput()	(<i>NatronEngine.NatronEngine.Effect</i> method), 1194	createParametricParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1257
contains()	(<i>NatronEngine.NatronEngine.RectD</i> method), 1240	createPathParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1257
contains()	(<i>NatronEngine.NatronEngine.RectI</i> method), 1243	createReader()	(<i>NatronEngine.NatronEngine.App</i> method), 1165
copy()	(<i>NatronEngine.NatronEngine.Param</i> method), 1225	createRectangle()	(<i>NatronEngine.NatronEngine.Roto</i> method), 1246
count()	(<i>NatronGui.NatronGui.PyTabWidget</i> method), 1275	createStringParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1257
createBezier()	(<i>NatronEngine.NatronEngine.Roto</i> method), 1246	createTrack()	(<i>NatronEngine.NatronEngine.Tracker</i> method), 1252
createBooleanParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1253	createWriter()	(<i>NatronEngine.NatronEngine.App</i> method), 1165
createButtonParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1253	cturbulence()	(<i>NatronEngine.NatronEngine.ExprUtils</i> method), 1202
createChoiceParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1253	currentWidget()	(<i>NatronEngine.NatronEngine.App</i> method), 1165
createColorParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1254		
createDouble2DParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1254		
createDouble3DParam()	(<i>NatronEngine.NatronEngine.UserParamHolder</i> method), 1254		

- tronGui.NatronGui.PyTabWidget* method), 1275
- curve()* (*NatronEngine.NatronEngine.Param* method), 1226
- ## D
- deleteAllControlPoints()* (*NatronEngine.NatronEngine.ParametricParam* method), 1232
- deleteControlPoint()* (*NatronEngine.NatronEngine.ParametricParam* method), 1232
- deleteValueAtTime()* (*NatronEngine.NatronEngine.AnimatedParam* method), 1158
- deselectNode()* (*NatronGui.NatronGui.GuiApp* method), 1262
- destroy()* (*NatronEngine.NatronEngine.Effect* method), 1194
- disconnectInput()* (*NatronEngine.NatronEngine.Effect* method), 1194
- Double2DTuple* (class in *NatronEngine*), 1186
- Double3DTuple* (class in *NatronEngine*), 1187
- ## E
- endChanges()* (*NatronEngine.NatronEngine.Effect* method), 1193
- errorDialog()* (*NatronGui.NatronGui.PyGuiApplication* method), 1265
- ## F
- fbm()* (*NatronEngine.NatronEngine.ExprUtils* method), 1202
- fbm4()* (*NatronEngine.NatronEngine.ExprUtils* method), 1203
- floatCurrentTab()* (*NatronGui.NatronGui.PyTabWidget* method), 1275
- floatPane()* (*NatronGui.NatronGui.PyTabWidget* method), 1275
- ## G
- g* (*NatronEngine.NatronEngine.ColorTuple* attribute), 1184
- gaussstep()* (*NatronEngine.NatronEngine.ExprUtils* method), 1199
- get()* (*NatronEngine.NatronEngine.BooleanParam* method), 1176
- get()* (*NatronEngine.NatronEngine.ChoiceParam* method), 1178, 1179
- get()* (*NatronEngine.NatronEngine.ColorParam* method), 1181
- get()* (*NatronEngine.NatronEngine.Double2DParam* method), 1185
- get()* (*NatronEngine.NatronEngine.Double3DParam* method), 1186
- get()* (*NatronEngine.NatronEngine.DoubleParam* method), 1189
- get()* (*NatronEngine.NatronEngine.Int2DParam* method), 1210
- get()* (*NatronEngine.NatronEngine.Int3DParam* method), 1211
- get()* (*NatronEngine.NatronEngine.IntParam* method), 1213
- get()* (*NatronEngine.NatronEngine.StringParamBase* method), 1248
- getActivatedParam()* (*NatronEngine.NatronEngine.BezierCurve* method), 1170
- getActiveInstance()* (*NatronEngine.NatronEngine.PyCoreApplication* method), 1237
- getAddNewLine()* (*NatronEngine.NatronEngine.Param* method), 1226
- getAInput()* (*NatronGui.NatronGui.PyTabWidget* method), 1278
- getAllTracks()* (*NatronEngine.NatronEngine.Tracker* method), 1252
- getAlphaComponents()* (*NatronEngine.NatronEngine.ImageLayer* method), 1209
- getAppID()* (*NatronEngine.NatronEngine.App* method), 1165
- getAvailableLayers()* (*NatronEngine.NatronEngine.Effect* method), 1194
- getBackwardMotionComponents()* (*NatronEngine.NatronEngine.ImageLayer* method), 1209
- getBaseLayer()* (*NatronEngine.NatronEngine.Roto* method), 1246
- getBInput()* (*NatronGui.NatronGui.PyTabWidget* method), 1278
- getBitDepth()* (*NatronEngine.NatronEngine.Effect* method), 1194
- getBuildNumber()* (*NatronEngine.NatronEngine.PyCoreApplication* method), 1237
- getCanAnimate()* (*NatronEngine.NatronEngine.Param* method), 1226
- getChannels()* (*NatronGui.NatronGui.PyTabWidget* method), 1278
- getChildren()* (*NatronEngine.NatronEngine.Group* method), 1205
- getChildren()* (*NatronEngine.NatronEngine.Group* method), 1205

<i>nEngine.NatronEngine.Layer</i> method), 1218	<i>getDefaultValue()</i> (<i>NatronEngine.NatronEngine.IntParam</i> method), 1213
<i>getColor()</i> (<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1171	<i>getDefaultValue()</i> (<i>NatronEngine.NatronEngine.StringParamBase</i> method), 1249
<i>getColor()</i> (<i>NatronEngine.NatronEngine.Effect</i> method), 1194	<i>getDerivativeAtTime()</i> (<i>NatronEngine.NatronEngine.AnimatedParam</i> method), 1158
<i>getColorParam()</i> (<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1171	<i>getDirectoryDialog()</i> (<i>NatronGui.NatronGui.GuiApp</i> method), 1260
<i>getComponentsNames()</i> (<i>NatronEngine.NatronEngine.ImageLayer</i> method), 1208	<i>getDisparityLeftComponents()</i> (<i>NatronEngine.NatronEngine.ImageLayer</i> method), 1209
<i>getComponentsPrettyName()</i> (<i>NatronEngine.NatronEngine.ImageLayer</i> method), 1208	<i>getDisparityRightComponents()</i> (<i>NatronEngine.NatronEngine.ImageLayer</i> method), 1209
<i>getCompositingOperator()</i> (<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1171	<i>getDisplayMaximum()</i> (<i>NatronEngine.NatronEngine.ColorParam</i> method), 1181
<i>getCompositingOperator()</i> (<i>NatronGui.NatronGui.PyTabWidget</i> method), 1278	<i>getDisplayMaximum()</i> (<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1189
<i>getCompositingOperatorParam()</i> (<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1171	<i>getDisplayMaximum()</i> (<i>NatronEngine.NatronEngine.IntParam</i> method), 1213
<i>getControlPointPosition()</i> (<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1171	<i>getDisplayMinimum()</i> (<i>NatronEngine.NatronEngine.ColorParam</i> method), 1181
<i>getCurrentFrame()</i> (<i>NatronGui.NatronGui.PyTabWidget</i> method), 1277	<i>getDisplayMinimum()</i> (<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1189
<i>getCurrentIndex()</i> (<i>NatronGui.NatronGui.PyTabWidget</i> method), 1275	<i>getDisplayMinimum()</i> (<i>NatronEngine.NatronEngine.IntParam</i> method), 1213
<i>getCurrentTime()</i> (<i>NatronEngine.NatronEngine.AnimatedParam</i> method), 1158	<i>getEvaluateOnChange()</i> (<i>NatronEngine.NatronEngine.Param</i> method), 1226
<i>getCurrentTime()</i> (<i>NatronEngine.NatronEngine.Effect</i> method), 1194	<i>getExpression()</i> (<i>NatronEngine.NatronEngine.AnimatedParam</i> method), 1158
<i>getCurrentView()</i> (<i>NatronGui.NatronGui.PyTabWidget</i> method), 1279	<i>getFeatherDistance()</i> (<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1171
<i>getCurveColor()</i> (<i>NatronEngine.NatronEngine.ParametricParam</i> method), 1232	<i>getFeatherDistanceParam()</i> (<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1171
<i>getDefaultValue()</i> (<i>NatronEngine.NatronEngine.BooleanParam</i> method), 1176	<i>getFeatherFallOff()</i> (<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1171
<i>getDefaultValue()</i> (<i>NatronEngine.NatronEngine.ChoiceParam</i> method), 1179	<i>getFeatherFallOffParam()</i> (<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1172
<i>getDefaultValue()</i> (<i>NatronEngine.NatronEngine.ColorParam</i> method), 1181	<i>getFeatherPointPosition()</i> (<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1172
<i>getDefaultValue()</i> (<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1189	<i>getFilenameDialog()</i> (<i>NatronGui.NatronGui.GuiApp</i> method), 1260

<code>getForwardMotionComponents()</code>	(<i>NatronEngine.NatronEngine.ImageLayer</i> method), 1209	<code>getLabel()</code>	(<i>NatronEngine.NatronEngine.Effect</i> method), 1195
<code>getFrameRange()</code>	(<i>NatronGui.NatronGui.PyTabWidget</i> method), 1277	<code>getLabel()</code>	(<i>NatronEngine.NatronEngine.ItemBase</i> method), 1216
<code>getFrameRate()</code>	(<i>NatronEngine.NatronEngine.Effect</i> method), 1195	<code>getLabel()</code>	(<i>NatronEngine.NatronEngine.Param</i> method), 1227
<code>getGuiInstance()</code>	(<i>NatronGui.NatronGui.PyGuiApplication</i> method), 1264	<code>getLayerName()</code>	(<i>NatronEngine.NatronEngine.ImageLayer</i> method), 1208
<code>getHelp()</code>	(<i>NatronEngine.NatronEngine.Param</i> method), 1226	<code>getLocked()</code>	(<i>NatronEngine.NatronEngine.ItemBase</i> method), 1216
<code>getInput()</code>	(<i>NatronEngine.NatronEngine.Effect</i> method), 1195	<code>getLockedRecursive()</code>	(<i>NatronEngine.NatronEngine.ItemBase</i> method), 1216
<code>getInputLabel()</code>	(<i>NatronEngine.NatronEngine.Effect</i> method), 1195	<code>getMaximum()</code>	(<i>NatronEngine.NatronEngine.ColorParam</i> method), 1181
<code>getInstance()</code>	(<i>NatronEngine.NatronEngine.PyCoreApplication</i> method), 1237	<code>getMaximum()</code>	(<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1189
<code>getIntegrateFromTimeToTime()</code>	(<i>NatronEngine.NatronEngine.AnimatedParam</i> method), 1158	<code>getMaximum()</code>	(<i>NatronEngine.NatronEngine.IntParam</i> method), 1214
<code>getIsActivated()</code>	(<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1172	<code>getMaxInputCount()</code>	(<i>NatronEngine.NatronEngine.Effect</i> method), 1195
<code>getIsAnimated()</code>	(<i>NatronEngine.NatronEngine.AnimatedParam</i> method), 1159	<code>getMinimum()</code>	(<i>NatronEngine.NatronEngine.ColorParam</i> method), 1182
<code>getIsAnimationEnabled()</code>	(<i>NatronEngine.NatronEngine.Param</i> method), 1226	<code>getMinimum()</code>	(<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1189
<code>getIsEnabled()</code>	(<i>NatronEngine.NatronEngine.Param</i> method), 1226	<code>getMinimum()</code>	(<i>NatronEngine.NatronEngine.IntParam</i> method), 1214
<code>getIsOpened()</code>	(<i>NatronEngine.NatronEngine.GroupParam</i> method), 1207	<code>getNatronDevelopmentStatus()</code>	(<i>NatronEngine.NatronEngine.PyCoreApplication</i> method), 1237
<code>getIsPersistant()</code>	(<i>NatronEngine.NatronEngine.Param</i> method), 1226	<code>getNatronPath()</code>	(<i>NatronEngine.NatronEngine.PyCoreApplication</i> method), 1237
<code>getIsVisible()</code>	(<i>NatronEngine.NatronEngine.Param</i> method), 1227	<code>getNatronVersionEncoded()</code>	(<i>NatronEngine.NatronEngine.PyCoreApplication</i> method), 1237
<code>getItemByName()</code>	(<i>NatronEngine.NatronEngine.Roto</i> method), 1247	<code>getNatronVersionMajor()</code>	(<i>NatronEngine.NatronEngine.PyCoreApplication</i> method), 1238
<code>getKeyframes()</code>	(<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1172	<code>getNatronVersionMinor()</code>	(<i>NatronEngine.NatronEngine.PyCoreApplication</i> method), 1238
<code>getKeyIndex()</code>	(<i>NatronEngine.NatronEngine.AnimatedParam</i> method), 1159	<code>getNatronVersionRevision()</code>	(<i>NatronEngine.NatronEngine.PyCoreApplication</i> method), 1238
<code>getKeyTime()</code>	(<i>NatronEngine.NatronEngine.AnimatedParam</i> method), 1159	<code>getNatronVersionString()</code>	(<i>NatronEngine.NatronEngine.PyCoreApplication</i> method), 1238
		<code>getNControlPoints()</code>	(<i>NatronEngine.NatronEngine.PyCoreApplication</i> method), 1238

nEngine.NatronEngine.ParametricParam method), 1232

`getNode()` (*NatronEngine.NatronEngine.Group method*), 1205

`getNoneComponents()` (*NatronEngine.NatronEngine.ImageLayer method*), 1209

`getNthControlPoint()` (*NatronEngine.NatronEngine.ParametricParam method*), 1232

`getNumComponents()` (*NatronEngine.NatronEngine.ImageLayer method*), 1208

`getNumControlPoints()` (*NatronEngine.NatronEngine.BezierCurve method*), 1172

`getNumCpus()` (*NatronEngine.NatronEngine.PyCoreApplication method*), 1238

`getNumDimensions()` (*NatronEngine.NatronEngine.Param method*), 1227

`getNumInstances()` (*NatronEngine.NatronEngine.PyCoreApplication method*), 1238

`getNumKeys()` (*NatronEngine.NatronEngine.AnimatedParam method*), 1159

`getNumOptions()` (*NatronEngine.NatronEngine.ChoiceParam method*), 1179

`getOpacity()` (*NatronEngine.NatronEngine.BezierCurve method*), 1172

`getOpacityParam()` (*NatronEngine.NatronEngine.BezierCurve method*), 1172

`getOption()` (*NatronEngine.NatronEngine.ChoiceParam method*), 1179

`getOptions()` (*NatronEngine.NatronEngine.ChoiceParam method*), 1179

`getOutputFormat()` (*NatronEngine.NatronEngine.Effect method*), 1194

`getOverlayColor()` (*NatronEngine.NatronEngine.BezierCurve method*), 1172

`getPanelLabel()` (*NatronGui.NatronGui.PyPanel method*), 1271

`getPanelScriptName()` (*NatronGui.NatronGui.PyPanel method*), 1271

`getParam()` (*NatronEngine.NatronEngine.AppSettings method*), 1168

`getParam()` (*NatronEngine.NatronEngine.Effect method*), 1195

`getParam()` (*NatronEngine.NatronEngine.ItemBase method*), 1216

`getParam()` (*NatronEngine.NatronEngine.Track method*), 1251

`getParam()` (*NatronGui.NatronGui.PyModalDialog method*), 1267

`getParam()` (*NatronGui.NatronGui.PyPanel method*), 1270

`getParams()` (*NatronEngine.NatronEngine.AppSettings method*), 1168

`getParams()` (*NatronEngine.NatronEngine.Effect method*), 1195

`getParams()` (*NatronEngine.NatronEngine.Track method*), 1251

`getParams()` (*NatronGui.NatronGui.PyPanel method*), 1271

`getParent()` (*NatronEngine.NatronEngine.Param method*), 1227

`getParentLayer()` (*NatronEngine.NatronEngine.ItemBase method*), 1216

`getPixelAspectRatio()` (*NatronEngine.NatronEngine.Effect method*), 1196

`getPlaybackMode()` (*NatronGui.NatronGui.PyTabWidget method*), 1278

`getPluginID()` (*NatronEngine.NatronEngine.Effect method*), 1195

`getPluginIDs()` (*NatronEngine.NatronEngine.PyCoreApplication method*), 1238

`getPosition()` (*NatronEngine.NatronEngine.Effect method*), 1195

`getPremult()` (*NatronEngine.NatronEngine.Effect method*), 1195

`getProjectParam()` (*NatronEngine.NatronEngine.AppSettings method*), 1165

`getProxyIndex()` (*NatronGui.NatronGui.PyTabWidget method*), 1279

`getRegionOfDefinition()` (*NatronEngine.NatronEngine.Effect method*), 1196

`getRGBAComponents()` (*NatronEngine.NatronEngine.ImageLayer method*), 1209

`getRGBColorDialog()` (*NatronGui.NatronGui.GuiApp method*), 1260

`getRGBComponents()` (*NatronEngine.NatronEngine.ImageLayer method*), 1209

`getRotoContext()` (*NatronEngine.NatronEngine.Effect method*),

1196			
getScriptName ()	(NatronEngine.NatronEngine.Effect method),	getValue ()	(NatronEngine.NatronEngine.DoubleParam method), 1189
1196			
getScriptName ()	(NatronEngine.NatronEngine.ItemBase method),	getValue ()	(NatronEngine.NatronEngine.IntParam method), 1214
1216			
getScriptName ()	(NatronEngine.NatronEngine.Param method),	getValue ()	(NatronEngine.NatronEngine.ParametricParam method), 1232
1227			
getScriptName ()	(NatronEngine.NatronEngine.Track method),	getValue ()	(NatronEngine.NatronEngine.StringParamBase method), 1249
1251			
getScriptName ()	(NatronGui.NatronGui.PyTabWidget method),	getValueAtTime ()	(NatronEngine.NatronEngine.BooleanParam method), 1176
1275			
getSelectedNodes ()	(NatronGui.NatronGui.GuiApp method),	getValueAtTime ()	(NatronEngine.NatronEngine.ChoiceParam method), 1179
1261			
getSelectedTracks ()	(NatronEngine.NatronEngine.Tracker method),	getValueAtTime ()	(NatronEngine.NatronEngine.ColorParam method),
1252			1182
getSequenceDialog ()	(NatronGui.NatronGui.GuiApp method),	getValueAtTime ()	(NatronEngine.NatronEngine.DoubleParam method), 1189
1260			
getSettings ()	(NatronEngine.NatronEngine.PyCoreApplication method),	getValueAtTime ()	(NatronEngine.NatronEngine.IntParam method),
1237			1214
getSize ()	(NatronEngine.NatronEngine.Effect method),	getValueAtTime ()	(NatronEngine.NatronEngine.StringParamBase method), 1249
1196			
getTabLabel ()	(NatronGui.NatronGui.PyTabWidget method),	getViewer ()	(NatronGui.NatronGui.GuiApp method), 1262
1275			
getTable ()	(NatronEngine.NatronEngine.PathParam method),	getViewNames ()	(NatronEngine.NatronEngine.App method), 1166
1234			
getTabWidget ()	(NatronGui.NatronGui.GuiApp method),	getVisible ()	(NatronEngine.NatronEngine.ItemBase method),
1261			1217
getTrackByName ()	(NatronEngine.NatronEngine.Tracker method),		
1252			
getTrackerContext ()	(NatronEngine.NatronEngine.Effect method),		
1196			
getTypeName ()	(NatronEngine.NatronEngine.Param method),		
1227			
getUserPageParam ()	(NatronEngine.NatronEngine.Effect method),		
1196			
getUserPanel ()	(NatronGui.NatronGui.GuiApp method),		
1262			
getValue ()	(NatronEngine.NatronEngine.BooleanParam method),		
1176			
getValue ()	(NatronEngine.NatronEngine.ChoiceParam method),		
1179			
getValue ()	(NatronEngine.NatronEngine.ColorParam method),		

H

hash ()	(NatronEngine.NatronEngine.ExprUtils method),
1200	
height ()	(NatronEngine.NatronEngine.RectD method),
1240	
height ()	(NatronEngine.NatronEngine.RectI method),
1243	

I

ImageBitDepthEnum	(NatronEngine.NatronEngine.Natron attribute),
1219	
ImageLayer ()	(NatronEngine.NatronEngine.ImageLayer method),
1208	
ImagePlaneDescEnum	(NatronEngine.NatronEngine.Natron attribute),
1219	
ImagePremultiplicationEnum	(NatronEngine.NatronEngine.Natron attribute),

- 1220
- `informationDialog()` (*NatronGui.NatronGui.PyGuiApplication method*), 1264
- `insertItem()` (*NatronEngine.NatronEngine.Layer method*), 1218
- `insertTab()` (*NatronGui.NatronGui.PyTabWidget method*), 1275
- `insertWidget()` (*NatronGui.NatronGui.PyModalDialog method*), 1267
- `insertWidget()` (*NatronGui.NatronGui.PyPanel method*), 1271
- `Int2DTuple` (class in *NatronEngine*), 1210
- `Int3DTuple` (class in *NatronEngine*), 1212
- `intersect()` (*NatronEngine.NatronEngine.RectD method*), 1241
- `intersect()` (*NatronEngine.NatronEngine.RectI method*), 1243
- `intersects()` (*NatronEngine.NatronEngine.RectD method*), 1241
- `intersects()` (*NatronEngine.NatronEngine.RectI method*), 1244
- `is64Bit()` (*NatronEngine.NatronEngine.PyCoreApplication method*), 1238
- `isBackground()` (*NatronEngine.NatronEngine.PyCoreApplication method*), 1238
- `isColorPlane()` (*NatronEngine.NatronEngine.ImageLayer method*), 1208
- `isCurveFinished()` (*NatronEngine.NatronEngine.BezierCurve method*), 1172
- `isInfinite()` (*NatronEngine.NatronEngine.RectD method*), 1241
- `isInfinite()` (*NatronEngine.NatronEngine.RectI method*), 1244
- `isLinux()` (*NatronEngine.NatronEngine.PyCoreApplication method*), 1238
- `isMacOSX()` (*NatronEngine.NatronEngine.PyCoreApplication method*), 1239
- `isNull()` (*NatronEngine.NatronEngine.RectD method*), 1241
- `isNull()` (*NatronEngine.NatronEngine.RectI method*), 1244
- `isProxyModeEnabled()` (*NatronGui.NatronGui.PyTabWidget method*), 1278
- `isReaderNode()` (*NatronEngine.NatronEngine.Effect method*), 1196
- `isUnix()` (*NatronEngine.NatronEngine.PyCoreApplication method*), 1239
- `isUserSelected()` (*NatronEngine.NatronEngine.Effect method*), 1196
- `isWindows()` (*NatronEngine.NatronEngine.PyCoreApplication method*), 1239
- `isWriterNode()` (*NatronEngine.NatronEngine.Effect method*), 1196
- ## K
- `KeyframeTypeEnum` (*NatronEngine.NatronEngine.Natron attribute*), 1219
- ## L
- `left()` (*NatronEngine.NatronEngine.RectD method*), 1241
- `left()` (*NatronEngine.NatronEngine.RectI method*), 1244
- `linearstep()` (*NatronEngine.NatronEngine.ExprUtils method*), 1199
- `loadProject()` (*NatronEngine.NatronEngine.App method*), 1167
- ## M
- `merge()` (*NatronEngine.NatronEngine.RectD method*), 1241
- `merge()` (*NatronEngine.NatronEngine.RectI method*), 1244
- `mix()` (*NatronEngine.NatronEngine.ExprUtils method*), 1200
- `moveFeatherByIndex()` (*NatronEngine.NatronEngine.BezierCurve method*), 1173
- `moveLeftBezierPoint()` (*NatronEngine.NatronEngine.BezierCurve method*), 1173
- `movePointByIndex()` (*NatronEngine.NatronEngine.BezierCurve method*), 1173
- `moveRightBezierPoint()` (*NatronEngine.NatronEngine.BezierCurve method*), 1173
- `moveTab()` (*NatronGui.NatronGui.GuiApp method*), 1261
- ## N
- `NatronEngine` (module), 1157, 1160, 1167, 1168, 1175, 1177, 1180, 1183–1187, 1191, 1198, 1204–1207, 1209–1212, 1215, 1217, 1218, 1221, 1222, 1230, 1233, 1235, 1239, 1242, 1245, 1247–1249, 1251, 1252
- `NatronGui` (module), 1258, 1263, 1265, 1267, 1272, 1276
- `newProject()` (*NatronEngine.NatronEngine.App method*), 1167

`noise()` (*NatronEngine.NatronEngine.ExprUtils method*), 1200

O

`onUserDataChanged()` (*NatronGui.NatronGui.PyPanel method*), 1272

`openFile()` (*NatronEngine.NatronEngine.FileParam method*), 1205

`openFile()` (*NatronEngine.NatronEngine.OutputFileParam method*), 1221

`OrientationEnum` (*NatronEngine.NatronEngine.Natron attribute*), 1220

P

`pause()` (*NatronGui.NatronGui.PyTabWidget method*), 1277

`PixmapEnum` (*NatronEngine.NatronEngine.Natron attribute*), 1220

`PlaybackModeEnum` (*NatronEngine.NatronEngine.Natron attribute*), 1220

`pnoise()` (*NatronEngine.NatronEngine.ExprUtils method*), 1204

`PyCoreApplication` (*class in NatronEngine*), 1237

`PyGuiApplication` (*class in NatronGui*), 1264

`PyPanel()` (*NatronGui.NatronGui.PyPanel method*), 1270

Q

`questionDialog()` (*NatronGui.NatronGui.PyGuiApplication method*), 1265

R

`r` (*NatronEngine.NatronEngine.ColorTuple attribute*), 1184

`random()` (*NatronEngine.NatronEngine.Param method*), 1227

`randomInt()` (*NatronEngine.NatronEngine.Param method*), 1228

`redraw()` (*NatronGui.NatronGui.PyTabWidget method*), 1277

`refreshUserParamsGUI()` (*NatronEngine.NatronEngine.UserParamHolder method*), 1258

`registerPythonPanel()` (*NatronGui.NatronGui.GuiApp method*), 1261

`reloadFile()` (*NatronEngine.NatronEngine.FileParam method*), 1205

`remap()` (*NatronEngine.NatronEngine.ExprUtils method*), 1200

`removeAnimation()` (*NatronEngine.NatronEngine.AnimatedParam method*), 1159

`removeControlPointByIndex()` (*NatronEngine.NatronEngine.BezierCurve method*), 1173

`removeItem()` (*NatronEngine.NatronEngine.Layer method*), 1218

`removeParam()` (*NatronEngine.NatronEngine.UserParamHolder method*), 1258

`removeTab()` (*NatronGui.NatronGui.PyTabWidget method*), 1275

`render()` (*NatronEngine.NatronEngine.App method*), 1166

`renderBlocking()` (*NatronGui.NatronGui.GuiApp method*), 1262, 1263

`renderCurrentFrame()` (*NatronGui.NatronGui.PyTabWidget method*), 1277

`reset()` (*NatronEngine.NatronEngine.Track method*), 1251

`resetProject()` (*NatronEngine.NatronEngine.App method*), 1167

`restore()` (*NatronGui.NatronGui.PyPanel method*), 1272

`restoreDefaultSettings()` (*NatronEngine.NatronEngine.AppSettings method*), 1168

`restoreDefaultValue()` (*NatronEngine.NatronEngine.BooleanParam method*), 1176

`restoreDefaultValue()` (*NatronEngine.NatronEngine.ChoiceParam method*), 1179

`restoreDefaultValue()` (*NatronEngine.NatronEngine.ColorParam method*), 1182

`restoreDefaultValue()` (*NatronEngine.NatronEngine.DoubleParam method*), 1190

`restoreDefaultValue()` (*NatronEngine.NatronEngine.IntParam method*), 1214

`restoreDefaultValue()` (*NatronEngine.NatronEngine.StringParamBase method*), 1249

`right()` (*NatronEngine.NatronEngine.RectD method*), 1241

`right()` (*NatronEngine.NatronEngine.RectI method*), 1244

S

`save()` (*NatronGui.NatronGui.PyPanel method*), 1272

`saveFilenameDialog()` (*NatronGui.NatronGui.GuiApp method*), 1260

`saveProject()` (*NatronEngine.NatronEngine.App method*), 1167
`saveProjectAs()` (*NatronEngine.NatronEngine.App method*), 1167
`saveSequenceDialog()` (*NatronGui.NatronGui.GuiApp method*), 1260
`saveSettings()` (*NatronEngine.NatronEngine.AppSettings method*), 1168
`saveTempProject()` (*NatronEngine.NatronEngine.App method*), 1167
`seek()` (*NatronGui.NatronGui.PyTabWidget method*), 1277
`selectAllNodes()` (*NatronGui.NatronGui.GuiApp method*), 1262
`selectNode()` (*NatronGui.NatronGui.GuiApp method*), 1262
`set()` (*NatronEngine.NatronEngine.BooleanParam method*), 1176
`set()` (*NatronEngine.NatronEngine.ChoiceParam method*), 1179
`set()` (*NatronEngine.NatronEngine.ColorParam method*), 1182
`set()` (*NatronEngine.NatronEngine.Double2DParam method*), 1185
`set()` (*NatronEngine.NatronEngine.Double3DParam method*), 1186, 1187
`set()` (*NatronEngine.NatronEngine.DoubleParam method*), 1190
`set()` (*NatronEngine.NatronEngine.Int2DParam method*), 1210
`set()` (*NatronEngine.NatronEngine.Int3DParam method*), 1211
`set()` (*NatronEngine.NatronEngine.IntParam method*), 1214
`set()` (*NatronEngine.NatronEngine.RectD method*), 1241
`set()` (*NatronEngine.NatronEngine.RectI method*), 1244
`set()` (*NatronEngine.NatronEngine.StringParamBase method*), 1249
`set_bottom()` (*NatronEngine.NatronEngine.RectD method*), 1241
`set_bottom()` (*NatronEngine.NatronEngine.RectI method*), 1244
`set_left()` (*NatronEngine.NatronEngine.RectD method*), 1242
`set_left()` (*NatronEngine.NatronEngine.RectI method*), 1244
`set_right()` (*NatronEngine.NatronEngine.RectD method*), 1242
`set_right()` (*NatronEngine.NatronEngine.RectI method*), 1244
`set_top()` (*NatronEngine.NatronEngine.RectD method*), 1242
`set_top()` (*NatronEngine.NatronEngine.RectI method*), 1245
`setActivated()` (*NatronEngine.NatronEngine.BezierCurve method*), 1173
`setAddNewLine()` (*NatronEngine.NatronEngine.Param method*), 1228
`setAInput()` (*NatronGui.NatronGui.PyTabWidget method*), 1278
`setAnimationEnabled()` (*NatronEngine.NatronEngine.Param method*), 1228
`setAsAlias()` (*NatronEngine.NatronEngine.Param method*), 1229
`setAsMultiPathTable()` (*NatronEngine.NatronEngine.PathParam method*), 1234
`setAsTab()` (*NatronEngine.NatronEngine.GroupParam method*), 1207
`setBInput()` (*NatronGui.NatronGui.PyTabWidget method*), 1278
`setCanAutoFoldDimensions()` (*NatronEngine.NatronEngine.Double2DParam method*), 1185
`setChannels()` (*NatronGui.NatronGui.PyTabWidget method*), 1278
`setColor()` (*NatronEngine.NatronEngine.BezierCurve method*), 1174
`setColor()` (*NatronEngine.NatronEngine.Effect method*), 1197
`setCompositingOperator()` (*NatronEngine.NatronEngine.BezierCurve method*), 1174
`setCompositingOperator()` (*NatronGui.NatronGui.PyTabWidget method*), 1278
`setCurrentIndex()` (*NatronGui.NatronGui.PyTabWidget method*), 1275
`setCurrentView()` (*NatronGui.NatronGui.PyTabWidget method*), 1279
`setCurveColor()` (*NatronEngine.NatronEngine.ParametricParam method*), 1233
`setCurveFinished()` (*NatronEngine.NatronEngine.BezierCurve method*), 1174
`setDefaultCurvesFromCurrentCurves()` (*NatronEngine.NatronEngine.ParametricParam method*), 1233
`setDefaultValue()` (*NatronEngine.NatronEngine.BooleanParam method*), 1177
`setDefaultValue()` (*NatronEngine.NatronEngine.ChoiceParam method*), 1179, 1180

<code>setDefaultValue()</code>	(<i>NatronEngine.NatronEngine.ColorParam</i> method), 1182	<code>setInterpolationAtTime()</code>	(<i>NatronEngine.NatronEngine.AnimatedParam</i> method), 1160
<code>setDefaultValue()</code>	(<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1190	<code>setLabel()</code>	(<i>NatronEngine.NatronEngine.Effect</i> method), 1197
<code>setDefaultValue()</code>	(<i>NatronEngine.NatronEngine.IntParam</i> method), 1214	<code>setLabel()</code>	(<i>NatronEngine.NatronEngine.ItemBase</i> method), 1217
<code>setDefaultValue()</code>	(<i>NatronEngine.NatronEngine.StringParamBase</i> method), 1249	<code>setLocked()</code>	(<i>NatronEngine.NatronEngine.ItemBase</i> method), 1217
<code>setDisplayMaximum()</code>	(<i>NatronEngine.NatronEngine.ColorParam</i> method), 1183	<code>setMaximum()</code>	(<i>NatronEngine.NatronEngine.ColorParam</i> method), 1183
<code>setDisplayMaximum()</code>	(<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1190	<code>setMaximum()</code>	(<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1190
<code>setDisplayMaximum()</code>	(<i>NatronEngine.NatronEngine.IntParam</i> method), 1215	<code>setMaximum()</code>	(<i>NatronEngine.NatronEngine.IntParam</i> method), 1215
<code>setDisplayMinimum()</code>	(<i>NatronEngine.NatronEngine.ColorParam</i> method), 1183	<code>setMinimum()</code>	(<i>NatronEngine.NatronEngine.ColorParam</i> method), 1183
<code>setDisplayMinimum()</code>	(<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1190	<code>setMinimum()</code>	(<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1190
<code>setDisplayMinimum()</code>	(<i>NatronEngine.NatronEngine.IntParam</i> method), 1215	<code>setMinimum()</code>	(<i>NatronEngine.NatronEngine.IntParam</i> method), 1215
<code>setEnabled()</code>	(<i>NatronEngine.NatronEngine.Param</i> method), 1228	<code>setNextTabCurrent()</code>	(<i>NatronGui.NatronGui.PyTabWidget</i> method), 1276
<code>setEnabledByDefault()</code>	(<i>NatronEngine.NatronEngine.Param</i> method), 1228	<code>setNthControlPoint()</code>	(<i>NatronEngine.NatronEngine.ParametricParam</i> method), 1233
<code>setEvaluateOnChange()</code>	(<i>NatronEngine.NatronEngine.Param</i> method), 1228	<code>setNthControlPointInterpolation()</code>	(<i>NatronEngine.NatronEngine.ParametricParam</i> method), 1233
<code>setExpression()</code>	(<i>NatronEngine.NatronEngine.AnimatedParam</i> method), 1159	<code>setOnProjectCreatedCallback()</code>	(<i>NatronEngine.NatronEngine.PyCoreApplication</i> method), 1239
<code>setFeatherDistance()</code>	(<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1174	<code>setOnProjectLoadedCallback()</code>	(<i>NatronEngine.NatronEngine.PyCoreApplication</i> method), 1239
<code>setFeatherFallOff()</code>	(<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1174	<code>setOpacity()</code>	(<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1174
<code>setFeatherPointAtIndex()</code>	(<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1174	<code>setOpened()</code>	(<i>NatronEngine.NatronEngine.GroupParam</i> method), 1207
<code>setFrameRange()</code>	(<i>NatronGui.NatronGui.PyTabWidget</i> method), 1277	<code>setOptions()</code>	(<i>NatronEngine.NatronEngine.ChoiceParam</i> method), 1180
<code>setHelp()</code>	(<i>NatronEngine.NatronEngine.Param</i> method), 1229	<code>setOverlayColor()</code>	(<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1175
<code>setIconFilePath()</code>	(<i>NatronEngine.NatronEngine.Param</i> method), 1229	<code>setPagesOrder()</code>	(<i>NatronEngine.NatronEngine.Effect</i> method), 1198

<code>setPanelLabel()</code> (<i>NatronGui.NatronGui.PyPanel</i> method), 1271	<code>nEngine.NatronEngine.BooleanParam</code> method), 1177
<code>setParamChangedCallback()</code> (<i>NatronGui.NatronGui.PyModalDialog</i> method), 1267	<code>setValue()</code> (<i>NatronEngine.NatronEngine.ChoiceParam</i> method), 1180
<code>setParamChangedCallback()</code> (<i>NatronGui.NatronGui.PyPanel</i> method), 1271	<code>setValue()</code> (<i>NatronEngine.NatronEngine.ColorParam</i> method), 1183
<code>setPersistant()</code> (<i>NatronEngine.NatronEngine.Param</i> method), 1229	<code>setValue()</code> (<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1191
<code>setPlaybackMode()</code> (<i>NatronGui.NatronGui.PyTabWidget</i> method), 1277	<code>setValue()</code> (<i>NatronEngine.NatronEngine.IntParam</i> method), 1215
<code>setPointAtIndex()</code> (<i>NatronEngine.NatronEngine.BezierCurve</i> method), 1175	<code>setValue()</code> (<i>NatronEngine.NatronEngine.StringParamBase</i> method), 1249
<code>setPosition()</code> (<i>NatronEngine.NatronEngine.Effect</i> method), 1197	<code>setValueAtTime()</code> (<i>NatronEngine.NatronEngine.BooleanParam</i> method), 1177
<code>setProxyIndex()</code> (<i>NatronGui.NatronGui.PyTabWidget</i> method), 1278	<code>setValueAtTime()</code> (<i>NatronEngine.NatronEngine.ChoiceParam</i> method), 1180
<code>setProxyModeEnabled()</code> (<i>NatronGui.NatronGui.PyTabWidget</i> method), 1278	<code>setValueAtTime()</code> (<i>NatronEngine.NatronEngine.ColorParam</i> method), 1183
<code>setScriptName()</code> (<i>NatronEngine.NatronEngine.Effect</i> method), 1197	<code>setValueAtTime()</code> (<i>NatronEngine.NatronEngine.DoubleParam</i> method), 1191
<code>setScriptName()</code> (<i>NatronEngine.NatronEngine.ItemBase</i> method), 1217	<code>setValueAtTime()</code> (<i>NatronEngine.NatronEngine.IntParam</i> method), 1215
<code>setScriptName()</code> (<i>NatronEngine.NatronEngine.Track</i> method), 1251	<code>setValueAtTime()</code> (<i>NatronEngine.NatronEngine.StringParamBase</i> method), 1249
<code>setSelection()</code> (<i>NatronGui.NatronGui.GuiApp</i> method), 1262	<code>setVisible()</code> (<i>NatronEngine.NatronEngine.ItemBase</i> method), 1217
<code>setSequenceEnabled()</code> (<i>NatronEngine.NatronEngine.FileParam</i> method), 1205	<code>setVisible()</code> (<i>NatronEngine.NatronEngine.Param</i> method), 1229
<code>setSequenceEnabled()</code> (<i>NatronEngine.NatronEngine.OutputFileParam</i> method), 1221	<code>setVisibleByDefault()</code> (<i>NatronEngine.NatronEngine.Param</i> method), 1229
<code>setSize()</code> (<i>NatronEngine.NatronEngine.Effect</i> method), 1197	<code>slaveTo()</code> (<i>NatronEngine.NatronEngine.Param</i> method), 1229
<code>setSubGraphEditable()</code> (<i>NatronEngine.NatronEngine.Effect</i> method), 1198	<code>snoise()</code> (<i>NatronEngine.NatronEngine.ExprUtils</i> method), 1200
<code>setTable()</code> (<i>NatronEngine.NatronEngine.PathParam</i> method), 1235	<code>snoise4()</code> (<i>NatronEngine.NatronEngine.ExprUtils</i> method), 1201
<code>setType()</code> (<i>NatronEngine.NatronEngine.StringParam</i> method), 1248	<code>splitHorizontally()</code> (<i>NatronGui.NatronGui.PyTabWidget</i> method), 1276
<code>setUsePointInteract()</code> (<i>NatronEngine.NatronEngine.Double2DParam</i> method), 1184	<code>splitVertically()</code> (<i>NatronGui.NatronGui.PyTabWidget</i> method), 1276
<code>setValue()</code> (<i>NatronEngine.NatronEngine.BooleanParam</i> method), 1177	<code>StandardButtonEnum</code> (<i>NatronEngine.NatronEngine.Natron</i> attribute), 1218
	<code>startBackward()</code> (Na-

`tronGui.NatronGui.PyTabWidget` method),
1277
`startForward()` (`NatronGui.NatronGui.PyTabWidget` method),
1277
`startTracking()` (`NatronEngine.NatronEngine.Tracker` method),
1252
`StatusEnum` (`NatronEngine.NatronEngine.Natron` attribute), 1220
`stopTracking()` (`NatronEngine.NatronEngine.Tracker` method),
1252

T

`timelineGetLeftBound()` (`NatronEngine.NatronEngine.App` method), 1166
`timelineGetRightBound()` (`NatronEngine.NatronEngine.App` method), 1166
`timelineGetTime()` (`NatronEngine.NatronEngine.App` method), 1166
`top()` (`NatronEngine.NatronEngine.RectD` method),
1242
`top()` (`NatronEngine.NatronEngine.RectI` method),
1245
`translate()` (`NatronEngine.NatronEngine.RectD` method), 1242
`translate()` (`NatronEngine.NatronEngine.RectI` method), 1245
`trigger()` (`NatronEngine.NatronEngine.ButtonParam` method), 1177
`turbulence()` (`NatronEngine.NatronEngine.ExprUtils` method),
1201

U

`unregisterPythonPanel()` (`NatronGui.NatronGui.GuiApp` method), 1261
`unslave()` (`NatronEngine.NatronEngine.Param` method), 1229

V

`ValueChangedReasonEnum` (`NatronEngine.NatronEngine.Natron` attribute),
1219
`vfbm()` (`NatronEngine.NatronEngine.ExprUtils` method), 1202
`vfbm4()` (`NatronEngine.NatronEngine.ExprUtils` method), 1203
`ViewerColorSpaceEnum` (`NatronEngine.NatronEngine.Natron` attribute),
1220
`ViewerCompositingOperatorEnum` (`NatronEngine.NatronEngine.Natron` attribute),
1220
`vnoise()` (`NatronEngine.NatronEngine.ExprUtils` method), 1201

`vnoise4()` (`NatronEngine.NatronEngine.ExprUtils` method), 1201
`vturbulence()` (`NatronEngine.NatronEngine.ExprUtils` method),
1201

W

`warningDialog()` (`NatronGui.NatronGui.PyGuiApplication` method), 1265
`width()` (`NatronEngine.NatronEngine.RectD` method), 1242
`width()` (`NatronEngine.NatronEngine.RectI` method), 1245
`writeToScriptEditor()` (`NatronEngine.NatronEngine.App` method), 1166

X

`x` (`NatronEngine.NatronEngine.Double2DTuple` attribute), 1186
`x` (`NatronEngine.NatronEngine.Double3DTuple` attribute), 1187
`x` (`NatronEngine.NatronEngine.Int2DTuple` attribute),
1210
`x` (`NatronEngine.NatronEngine.Int3DTuple` attribute),
1212

Y

`y` (`NatronEngine.NatronEngine.Double2DTuple` attribute), 1186
`y` (`NatronEngine.NatronEngine.Double3DTuple` attribute), 1187
`y` (`NatronEngine.NatronEngine.Int2DTuple` attribute),
1210
`y` (`NatronEngine.NatronEngine.Int3DTuple` attribute),
1212

Z

`z` (`NatronEngine.NatronEngine.Double3DTuple` attribute), 1187
`z` (`NatronEngine.NatronEngine.Int3DTuple` attribute),
1212