

MBS Real Studio ImageMagick Plugin Documentation

Christian Schmitz

May 15, 2012

0.1 Introduction

This is the PDF version of the documentation for the Real Studio Plug-in from Monkeybread Software Germany. Plugin part: MBS Real Studio ImageMagick Plugin

0.2 Content

- 1 List of all topics 3
- 2 All items in this plugin 29
- 3 List of all classes 337

Chapter 1

List of Topics

• 2 Image Magick	29
– 2.31 class QImageQ32MBS	197
* 2.31.1 AdaptiveThreshold(width as integer, height as integer, offset as integer) as QImageQ32MBS	197
* 2.31.1 AddNoise(NoiseType as integer) as QImageQ32MBS	197
* 2.31.1 AffineTransformImage(matrix as QImageAffineTransformQ32MBS) as QImageQ32MBS	198
* 2.31.1 AppendImageToList(img as QImageQ32MBS)	198
* 2.31.1 Average as QImageQ32MBS	198
* 2.31.1 BilevelChannel(channel as integer, threshold as double) as boolean	198
* 2.31.1 BlackThreshold(threshold as string) as boolean	199
* 2.31.1 BlobSize as integer	199
* 2.31.1 Blur(radius as double, sigma as double) as QImageQ32MBS	199
* 2.31.1 BlurImageChannel(channel as integer, radius as double, sigma as double) as QImageQ32MBS	200
* 2.31.1 BorderImage(x as integer, y as integer, width as integer, height as integer) as QImageQ32MBS	200
* 2.31.1 Charcoal(radius as double, sigma as double) as QImageQ32MBS	201
* 2.31.1 Chop(x as integer, y as integer, width as integer, height as integer) as QImageQ32MBS	201
* 2.31.1 ClipPath(path as string, inside as boolean) as boolean	201
* 2.31.1 Clone as QImageQ32MBS	201
* 2.31.1 CloneImageAttributes(image as QImageAttributeQ32MBS) as Boolean	202
* 2.31.1 CloneImageProfiles(SourceImage as QImageQ32MBS) as boolean	202
* 2.31.1 Close	202
* 2.31.1 CoalesceImages as QImageQ32MBS	202

* 2.31.1 Colorize(opacity as string, PenColorRed as integer, PenColorGreen as integer, PenColorBlue as integer, PenColorOpacity as integer) as QImageQ32MBS	203
* 2.31.1 Combine(channel as integer) as QImageQ32MBS	203
* 2.31.1 CompareImageLayers(ImageLayerMethod as integer) as QImageQ32MBS	203
* 2.31.1 Composite(ComposeOperator as integer, Image as QImageQ32MBS, x as integer, y as integer)	204
* 2.31.1 ConsolidateCMYKImages as QImageQ32MBS	204
* 2.31.1 CopyPicture as picture	205
* 2.31.1 CopyPicture(x as integer, y as integer, width as integer, height as integer) as picture	205
* 2.31.1 CopyPictureMask as picture	206
* 2.31.1 CopyPictureMask(x as integer, y as integer, width as integer, height as integer) as picture	206
* 2.31.1 CopyPixel(x as integer, y as integer) as QColorQ32MBS	207
* 2.31.1 Crop(x as integer, y as integer, width as integer, height as integer) as QImageQ32MBS	207
* 2.31.1 CropImageToTiles(CropGeometry as string) as QImageQ32MBS	207
* 2.31.1 CycleColormap(displace as integer) as boolean	207
* 2.31.1 DecipherImage(passkey as string) as boolean	208
* 2.31.1 DeconstructImages as QImageQ32MBS	208
* 2.31.1 DeleteImageAttribute(key as string) as Boolean	208
* 2.31.1 Despeckle() as QImageQ32MBS	208
* 2.31.1 DestroyImage	209
* 2.31.1 DestroyImageAttributes	209
* 2.31.1 DestroyImageList	209
* 2.31.1 DestroyImageProfiles	209
* 2.31.1 Edge(radius as double) as QImageQ32MBS	209
* 2.31.1 Emboss(radius as double, sigma as double) as QImageQ32MBS	210
* 2.31.1 EncipherImage(passkey as string) as boolean	210
* 2.31.1 ExcerptImage(x as integer, y as integer, width as integer, height as integer) as QImageQ32MBS	210
* 2.31.1 ExtentImage(x as integer, y as integer, width as integer, height as integer) as QImageQ32MBS	211
* 2.31.1 FlattenImages as QImageQ32MBS	211
* 2.31.1 Flip as QImageQ32MBS	211
* 2.31.1 Flop as QImageQ32MBS	211
* 2.31.1 FrameImage(x as integer, y as integer, width as integer, height as integer, innerBevel as integer, OuterBevel as integer) as QImageQ32MBS	212
* 2.31.1 FxImage(expression as string) as QImageQ32MBS	212
* 2.31.1 GaussianBlurChannel(channel as integer, radius as double, sigma as double) as QImageQ32MBS	212
* 2.31.1 GetImageAttribute(key as string) as QImageAttributeQ32MBS	213
* 2.31.1 GetImageClippingPathAttribute as QImageAttributeQ32MBS	213

* 2.31.1 GetImageProfile(name as string) as string	213
* 2.31.1 GetNextImageAttribute as IImageAttributeQ32MBS	213
* 2.31.1 GetNextImageProfile as string	213
* 2.31.1 HandleMemory as memoryblock	214
* 2.31.1 ImagesToBlob(info as IImageInfoQ32MBS) as String	214
* 2.31.1 ImageToBlob(info as IImageInfoQ32MBS) as String	214
* 2.31.1 Implode(factor as double) as IImageQ32MBS	216
* 2.31.1 IsBlobExempt as boolean	216
* 2.31.1 IsBlobSeekable as boolean	216
* 2.31.1 IsBlobTemporary as boolean	217
* 2.31.1 Magnify as IImageQ32MBS	217
* 2.31.1 MedianFilter(radius as double) as IImageQ32MBS	217
* 2.31.1 MergeImageLayers(ImageLayerMethod as integer) as IImageQ32MBS	217
* 2.31.1 Minify as IImageQ32MBS	218
* 2.31.1 MosaicImages as IImageQ32MBS	218
* 2.31.1 MotionBlur(radius as double, sigma as double, angle as double) as IImageQ32MBS	219
* 2.31.1 NewImage(info as IImageInfoQ32MBS, width as integer, height as integer, background as IMMagickPixelPacketQ32MBS) as boolean	219
* 2.31.1 OilPaint(radius as double) as IImageQ32MBS	220
* 2.31.1 OptimizeImageLayers as IImageQ32MBS	220
* 2.31.1 OptimizeImageTransparency	220
* 2.31.1 OptimizePlusImageLayers as IImageQ32MBS	221
* 2.31.1 ProfileImage(name as string, ProfileData as string) as boolean	221
* 2.31.1 RadialBlur(angle as double) as IImageQ32MBS	221
* 2.31.1 RaiseImage(x as integer, y as integer, width as integer, height as integer, raise as boolean) as boolean	222
* 2.31.1 RandomThresholdChannel(channel as integer, thresholds as string) as boolean	222
* 2.31.1 ReduceNoise(radius as double) as IImageQ32MBS	223
* 2.31.1 RemoveDuplicateLayers	223
* 2.31.1 RemoveFirstImageFromList as IImageQ32MBS	223
* 2.31.1 RemoveImageProfile(name as string) as string	224
* 2.31.1 RemoveZeroDelayLayers	224
* 2.31.1 ResetImageAttributeIterator	224
* 2.31.1 ResetImageProfileIterator	225
* 2.31.1 Resize(width as integer, height as integer, FilterID as integer, blur as double) as IImageQ32MBS	225
* 2.31.1 RGBTransformImage(Colorspace as integer) as boolean	225
* 2.31.1 Roll(x as integer, y as integer) as IImageQ32MBS	226
* 2.31.1 Rotate(degrees as double) as IImageQ32MBS	226
* 2.31.1 Sample(width as integer, height as integer) as IImageQ32MBS	226
* 2.31.1 Scale(width as integer, height as integer) as IImageQ32MBS	227

* 2.31.1 SetImageAttribute(key as string, value as string) as boolean	227
* 2.31.1 SetImageColorspace(Colorspace as integer) as boolean	228
* 2.31.1 SetImageProfile(name as string, ProfileData as string) as boolean	228
* 2.31.1 SetPicture(pic as picture, x as integer, y as integer)	228
* 2.31.1 SetPictureMask(maskpic as picture, x as integer, y as integer)	229
* 2.31.1 SetPixel(x as integer, y as integer, newPixel as IMColorQ32MBS)	229
* 2.31.1 Shade(gray as boolean, azimuth as double, elevation as double) as IMImageQ32MBS	230
* 2.31.1 SharpenChannel(channel as integer, radius as double, sigma as double) as IMImageQ32MBS	230
* 2.31.1 Shave(x as integer, y as integer, width as integer, height as integer) as IMImageQ32MBS	231
* 2.31.1 Shear(Xshear as double, Yshear as double) as IMImageQ32MBS	238
* 2.31.1 Solarize(factor as double) as boolean	238
* 2.31.1 Splice(x as integer, y as integer, width as integer, height as integer) as IMImageQ32MBS	238
* 2.31.1 Spread(radius as double) as IMImageQ32MBS	239
* 2.31.1 Stegano(watermarkImage as IMImageQ32MBS) as IMImageQ32MBS	239
* 2.31.1 Stereo(otherImage as IMImageQ32MBS) as IMImageQ32MBS	239
* 2.31.1 Swirl(degrees as double) as IMImageQ32MBS	240
* 2.31.1 Thumbnail(width as integer, height as integer) as IMImageQ32MBS	240
* 2.31.1 TransformImage(CropGeometry as string, ImageGeometry as string) as boolean	240
* 2.31.1 TransformImages(CropGeometry as string, ImageGeometry as string) as boolean	241
* 2.31.1 TransformRGBImage(Colorspace as integer) as boolean	241
* 2.31.1 TransposeImage as IMImageQ32MBS	241
* 2.31.1 TransverseImage as IMImageQ32MBS	242
* 2.31.1 Trim as IMImageQ32MBS	242
* 2.31.1 UnsharpMaskChannel(channel as integer, radius as double, sigma as double, amount as double, threshold as double) as IMImageQ32MBS	243
* 2.31.1 Wave(amplitude as double, wavelength as double) as IMImageQ32MBS	243
* 2.31.1 WhiteThreshold(threshold as string) as boolean	244
* 2.31.1 WriteImage(info as IMImageInfoQ32MBS) as boolean	244
* 2.31.2 BackgroundColor as IMColorQ32MBS	244
* 2.31.2 Bias as double	245
* 2.31.2 BlurFactor as double	245
* 2.31.2 BorderColor as IMColorQ32MBS	245
* 2.31.2 Colors as Integer	245
* 2.31.2 ColorSpace as Integer	245
* 2.31.2 Compression as Integer	246
* 2.31.2 Depth as Integer	247
* 2.31.2 Directory as String	247
* 2.31.2 Endian as Integer	247

* 2.31.2 Filename as String	248
* 2.31.2 Filter as Integer	248
* 2.31.2 Fuzz as double	249
* 2.31.2 Gamma as double	249
* 2.31.2 Geometry as String	249
* 2.31.2 Gravity as Integer	249
* 2.31.2 Handle as Integer	250
* 2.31.2 Height as integer	250
* 2.31.2 Interlace as Integer	250
* 2.31.2 LastError as Integer	251
* 2.31.2 LastException as IMExceptionQ32MBS	251
* 2.31.2 Magick as String	251
* 2.31.2 Matte as Boolean	252
* 2.31.2 MatteColor as IMColorQ32MBS	252
* 2.31.2 Montage as String	252
* 2.31.2 Offset as Integer	252
* 2.31.2 Orientation as Integer	252
* 2.31.2 Quality as Integer	253
* 2.31.2 Release as Boolean	254
* 2.31.2 RenderingIntent as Integer	255
* 2.31.2 ResolutionUnits as Integer	255
* 2.31.2 ResolutionX as double	255
* 2.31.2 ResolutionY as double	256
* 2.31.2 Scene as Integer	256
* 2.31.2 StorageClass as Integer	256
* 2.31.2 Taint as Boolean	257
* 2.31.2 Width as integer	257
* 2.31.3 kBackgroundDispose = 2	257
* 2.31.3 kCoalesceLayer = 1	257
* 2.31.3 kCompareAnyLayer = 2	258
* 2.31.3 kCompareClearLayer = 3	258
* 2.31.3 kCompareOverlayLayer = 4	258
* 2.31.3 kCompositeLayer = & h0000000C	258
* 2.31.3 kDisposeLayer = 5	258
* 2.31.3 kFlattenLayer = & h0000000E	258
* 2.31.3 kMergeLayer = & h0000000D	259
* 2.31.3 kMosaicLayer = & h0000000F	259
* 2.31.3 kNoneDispose = 1	259
* 2.31.3 kOptimizeImageLayer = 7	259
* 2.31.3 kOptimizeLayer = 6	259
* 2.31.3 kOptimizePlusLayer = 8	259
* 2.31.3 kOptimizeTransLayer = 9	260

* 2.31.3 kPreviousDispose = 3	260
* 2.31.3 kRemoveDupsLayer = & h0000000A	260
* 2.31.3 kRemoveZeroLayer = & h0000000B	260
* 2.31.3 kUndefinedDispose = 0	260
* 2.31.3 kUndefinedLayer = 0	260
* 2.31.3 kUnrecognizedDispose = 0	261
– 2.1 class QImageQ8MBS	29
* 2.1.1 AdaptiveThreshold(width as integer, height as integer, offset as integer) as QImageQ8MBS	29
* 2.1.1 AddNoise(NoiseType as integer) as QImageQ8MBS	30
* 2.1.1 AffineTransformImage(matrix as QImageAffineMatrixQ8MBS) as QImageQ8MBS	30
* 2.1.1 AppendImageToList(img as QImageQ8MBS)	30
* 2.1.1 Average as QImageQ8MBS	31
* 2.1.1 BilevelChannel(channel as integer, threshold as double) as boolean	31
* 2.1.1 BlackThreshold(threshold as string) as boolean	32
* 2.1.1 BlobSize as integer	32
* 2.1.1 Blur(radius as double, sigma as double) as QImageQ8MBS	32
* 2.1.1 BlurImageChannel(channel as integer, radius as double, sigma as double) as QImageQ8MBS	33
* 2.1.1 BorderImage(x as integer, y as integer, width as integer, height as integer) as QImageQ8MBS	34
* 2.1.1 Charcoal(radius as double, sigma as double) as QImageQ8MBS	34
* 2.1.1 Chop(x as integer, y as integer, width as integer, height as integer) as QImageQ8MBS	34
* 2.1.1 ClipPath(path as string, inside as boolean) as boolean	34
* 2.1.1 Clone as QImageQ8MBS	35
* 2.1.1 CloneImageAttributes(image as QImageAttributeQ8MBS) as Boolean	35
* 2.1.1 CloneImageProfiles(SourceImage as QImageQ8MBS) as boolean	35
* 2.1.1 Close	35
* 2.1.1 CoalesceImages as QImageQ8MBS	36
* 2.1.1 Colorize(opacity as string, PenColorRed as integer, PenColorGreen as integer, PenColorBlue as integer, PenColorOpacity as integer) as QImageQ8MBS	36
* 2.1.1 Combine(channel as integer) as QImageQ8MBS	36
* 2.1.1 CompareImageLayers(ImageLayerMethod as integer) as QImageQ8MBS	37
* 2.1.1 Composite(ComposeOperator as integer, Image as QImageQ8MBS, x as integer, y as integer)	37
* 2.1.1 ConsolidateCMYKImages as QImageQ8MBS	38
* 2.1.1 CopyPicture as picture	38
* 2.1.1 CopyPicture(x as integer, y as integer, width as integer, height as integer) as picture	38
* 2.1.1 CopyPictureMask as picture	39
* 2.1.1 CopyPictureMask(x as integer, y as integer, width as integer, height as integer) as picture	39

* 2.1.1 CopyPixel(x as integer, y as integer) as IMColorQ8MBS	40
* 2.1.1 Crop(x as integer, y as integer, width as integer, height as integer) as IMImageQ8MBS	40
* 2.1.1 CropImageToTiles(CropGeometry as string) as IMImageQ8MBS	40
* 2.1.1 CycleColormap(displace as integer) as boolean	41
* 2.1.1 DecipherImage(passkey as string) as boolean	41
* 2.1.1 DeconstructImages as IMImageQ8MBS	41
* 2.1.1 DeleteImageAttribute(key as string) as Boolean	41
* 2.1.1 Despeckle() as IMImageQ8MBS	42
* 2.1.1 DestroyImage	42
* 2.1.1 DestroyImageAttributes	42
* 2.1.1 DestroyImageList	42
* 2.1.1 DestroyImageProfiles	43
* 2.1.1 Edge(radius as double) as IMImageQ8MBS	43
* 2.1.1 Emboss(radius as double, sigma as double) as IMImageQ8MBS	43
* 2.1.1 EncipherImage(passkey as string) as boolean	43
* 2.1.1 ExcerptImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ8MBS	44
* 2.1.1 ExtentImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ8MBS	44
* 2.1.1 FlattenImages as IMImageQ8MBS	44
* 2.1.1 Flip as IMImageQ8MBS	44
* 2.1.1 Flop as IMImageQ8MBS	45
* 2.1.1 FrameImage(x as integer, y as integer, width as integer, height as integer, innerBevel as integer, OuterBevel as integer) as IMImageQ8MBS	45
* 2.1.1 FxImage(expression as string) as IMImageQ8MBS	45
* 2.1.1 GaussianBlurChannel(channel as integer, radius as double, sigma as double) as IMImageQ8MBS	45
* 2.1.1 GetImageAttribute(key as string) as IMImageAttributeQ8MBS	46
* 2.1.1 GetImageClippingPathAttribute as IMImageAttributeQ8MBS	46
* 2.1.1 GetImageProfile(name as string) as string	47
* 2.1.1 GetNextImageAttribute as IMImageAttributeQ8MBS	47
* 2.1.1 GetNextImageProfile as string	47
* 2.1.1 HandleMemory as memoryblock	47
* 2.1.1 ImagesToBlob(info as IMImageInfoQ8MBS) as String	47
* 2.1.1 ImageToBlob(info as IMImageInfoQ8MBS) as String	48
* 2.1.1 Implode(factor as double) as IMImageQ8MBS	50
* 2.1.1 IsBlobExempt as boolean	50
* 2.1.1 IsBlobSeekable as boolean	50
* 2.1.1 IsBlobTemporary as boolean	50
* 2.1.1 Magnify as IMImageQ8MBS	50
* 2.1.1 MedianFilter(radius as double) as IMImageQ8MBS	51

* 2.1.1 MergeImageLayers(ImageLayerMethod as integer) as IImageQ8MBS	51
* 2.1.1 Minify as IImageQ8MBS	52
* 2.1.1 MosaicImages as IImageQ8MBS	52
* 2.1.1 MotionBlur(radius as double, sigma as double, angle as double) as IImageQ8MBS	52
* 2.1.1 NewImage(info as IImageInfoQ8MBS, width as integer, height as integer, background as IMMagickPixelPacketQ8MBS) as boolean	53
* 2.1.1 OilPaint(radius as double) as IImageQ8MBS	53
* 2.1.1 OptimizeImageLayers as IImageQ8MBS	54
* 2.1.1 OptimizeImageTransparency	54
* 2.1.1 OptimizePlusImageLayers as IImageQ8MBS	55
* 2.1.1 ProfileImage(name as string, ProfileData as string) as boolean	55
* 2.1.1 RadialBlur(angle as double) as IImageQ8MBS	55
* 2.1.1 RaiseImage(x as integer, y as integer, width as integer, height as integer, raise as boolean) as boolean	55
* 2.1.1 RandomThresholdChannel(channel as integer, thresholds as string) as boolean	56
* 2.1.1 ReduceNoise(radius as double) as IImageQ8MBS	57
* 2.1.1 RemoveDuplicateLayers	57
* 2.1.1 RemoveFirstImageFromList as IImageQ8MBS	57
* 2.1.1 RemoveImageProfile(name as string) as string	58
* 2.1.1 RemoveZeroDelayLayers	58
* 2.1.1 ResetImageAttributeIterator	58
* 2.1.1 ResetImageProfileIterator	59
* 2.1.1 Resize(width as integer, height as integer, FilterID as integer, blur as double) as IImageQ8MBS	59
* 2.1.1 RGBTransformImage(Colorspace as integer) as boolean	59
* 2.1.1 Roll(x as integer, y as integer) as IImageQ8MBS	60
* 2.1.1 Rotate(degrees as double) as IImageQ8MBS	61
* 2.1.1 Sample(width as integer, height as integer) as IImageQ8MBS	61
* 2.1.1 Scale(width as integer, height as integer) as IImageQ8MBS	61
* 2.1.1 SetImageAttribute(key as string, value as string) as boolean	62
* 2.1.1 SetImageColorspace(Colorspace as integer) as boolean	62
* 2.1.1 SetImageProfile(name as string, ProfileData as string) as boolean	62
* 2.1.1 SetPicture(pic as picture, x as integer, y as integer)	63
* 2.1.1 SetPictureMask(maskpic as picture, x as integer, y as integer)	63
* 2.1.1 SetPixel(x as integer, y as integer, newPixel as IMColorQ8MBS)	64
* 2.1.1 Shade(gray as boolean, azimuth as double, elevation as double) as IImageQ8MBS	65
* 2.1.1 SharpenChannel(channel as integer, radius as double, sigma as double) as IImageQ8MBS	65
* 2.1.1 Shave(x as integer, y as integer, width as integer, height as integer) as IImageQ8MBS	65
* 2.1.1 Shear(Xshear as double, Yshear as double) as IImageQ8MBS	67
* 2.1.1 Solarize(factor as double) as boolean	67

* 2.1.1 Splice(x as integer, y as integer, width as integer, height as integer) as IMImageQ8MBS	67
* 2.1.1 Spread(radius as double) as IMImageQ8MBS	68
* 2.1.1 Stegano(watermarkImage as IMImageQ8MBS) as IMImageQ8MBS	68
* 2.1.1 Stereo(otherImage as IMImageQ8MBS) as IMImageQ8MBS	68
* 2.1.1 Swirl(degrees as double) as IMImageQ8MBS	69
* 2.1.1 Thumbnail(width as integer, height as integer) as IMImageQ8MBS	69
* 2.1.1 TransformImage(CropGeometry as string, ImageGeometry as string) as boolean	69
* 2.1.1 TransformImages(CropGeometry as string, ImageGeometry as string) as boolean	70
* 2.1.1 TransformRGBImage(Colorspace as integer) as boolean	70
* 2.1.1 TransposeImage as IMImageQ8MBS	70
* 2.1.1 TransverseImage as IMImageQ8MBS	71
* 2.1.1 Trim as IMImageQ8MBS	71
* 2.1.1 UnsharpMaskChannel(channel as integer, radius as double, sigma as double, amount as double, threshold as double) as IMImageQ8MBS	72
* 2.1.1 Wave(amplitude as double, wavelength as double) as IMImageQ8MBS	72
* 2.1.1 WhiteThreshold(threshold as string) as boolean	73
* 2.1.1 WriteImage(info as IMImageInfoQ8MBS) as boolean	73
* 2.1.2 BackgroundColor as IMColorQ8MBS	73
* 2.1.2 Bias as double	74
* 2.1.2 BlurFactor as double	74
* 2.1.2 BorderColor as IMColorQ8MBS	74
* 2.1.2 Colors as Integer	74
* 2.1.2 ColorSpace as Integer	74
* 2.1.2 Compression as Integer	75
* 2.1.2 Depth as Integer	76
* 2.1.2 Directory as String	76
* 2.1.2 Endian as Integer	76
* 2.1.2 Filename as String	77
* 2.1.2 Filter as Integer	77
* 2.1.2 Fuzz as double	78
* 2.1.2 Gamma as double	78
* 2.1.2 Geometry as String	78
* 2.1.2 Gravity as Integer	78
* 2.1.2 Handle as Integer	79
* 2.1.2 Height as integer	79
* 2.1.2 Interlace as Integer	79
* 2.1.2 LastError as Integer	80
* 2.1.2 LastException as IMExceptionQ8MBS	80
* 2.1.2 Magick as String	80
* 2.1.2 Matte as Boolean	81
* 2.1.2 MatteColor as IMColorQ8MBS	81

* 2.1.2 Montage as String	81
* 2.1.2 Offset as Integer	81
* 2.1.2 Orientation as Integer	81
* 2.1.2 Quality as Integer	82
* 2.1.2 Release as Boolean	83
* 2.1.2 RenderingIntent as Integer	84
* 2.1.2 ResolutionUnits as Integer	84
* 2.1.2 ResolutionX as double	84
* 2.1.2 ResolutionY as double	85
* 2.1.2 Scene as Integer	85
* 2.1.2 StorageClass as Integer	85
* 2.1.2 Taint as Boolean	86
* 2.1.2 Width as integer	86
* 2.1.3 kBackgroundDispose = 2	86
* 2.1.3 kCoalesceLayer = 1	86
* 2.1.3 kCompareAnyLayer = 2	87
* 2.1.3 kCompareClearLayer = 3	87
* 2.1.3 kCompareOverlayLayer = 4	87
* 2.1.3 kCompositeLayer = & h0000000C	87
* 2.1.3 kDisposeLayer = 5	87
* 2.1.3 kFlattenLayer = & h0000000E	87
* 2.1.3 kMergeLayer = & h0000000D	88
* 2.1.3 kMosaicLayer = & h0000000F	88
* 2.1.3 kNoneDispose = 1	88
* 2.1.3 kOptimizeImageLayer = 7	88
* 2.1.3 kOptimizeLayer = 6	88
* 2.1.3 kOptimizePlusLayer = 8	88
* 2.1.3 kOptimizeTransLayer = 9	89
* 2.1.3 kPreviousDispose = 3	89
* 2.1.3 kRemoveDupsLayer = & h0000000A	89
* 2.1.3 kRemoveZeroLayer = & h0000000B	89
* 2.1.3 kUndefinedDispose = 0	89
* 2.1.3 kUndefinedLayer = 0	89
* 2.1.3 kUnrecognizedDispose = 0	90
– 2.3 class IMMagickInfoQ16MBS	94
* 2.3.1 Close	94
* 2.3.2 Adjoin as Boolean	95
* 2.3.2 BlobSupport as Boolean	95
* 2.3.2 Description as String	95
* 2.3.2 EndianSupport as Boolean	96
* 2.3.2 Handle as Integer	96

* 2.3.2 ModuleName as String	96
* 2.3.2 Name as String	97
* 2.3.2 Note as String	97
* 2.3.2 Raw as Boolean	97
* 2.3.2 SeekableStream as Boolean	97
* 2.3.2 Stealth as Boolean	98
* 2.3.2 ThreadSupport as Boolean	98
* 2.3.2 Version as String	98
– 2.2 class IMMagickInfoQ32MBS	90
* 2.2.1 Close	90
* 2.2.2 Adjoin as Boolean	90
* 2.2.2 BlobSupport as Boolean	91
* 2.2.2 Description as String	91
* 2.2.2 EndianSupport as Boolean	91
* 2.2.2 Handle as Integer	92
* 2.2.2 ModuleName as String	92
* 2.2.2 Name as String	92
* 2.2.2 Note as String	92
* 2.2.2 Raw as Boolean	93
* 2.2.2 SeekableStream as Boolean	93
* 2.2.2 Stealth as Boolean	93
* 2.2.2 ThreadSupport as Boolean	94
* 2.2.2 Version as String	94
– 2.4 class IMMagickInfoListQ8MBS	99
* 2.4.1 Item(index as integer) as IMMagickInfoQ8MBS	99
* 2.4.2 Count as Integer	99
* 2.4.2 Handle as Integer	99
– 2.6 class IMMagickInfoListQ32MBS	101
* 2.6.1 Item(index as integer) as IMMagickInfoQ32MBS	101
* 2.6.2 Count as Integer	102
* 2.6.2 Handle as Integer	102
– 2.5 class IMMagickInfoListQ16MBS	100
* 2.5.1 Item(index as integer) as IMMagickInfoQ16MBS	100
* 2.5.2 Count as Integer	100
* 2.5.2 Handle as Integer	101
– 2.8 class IMMagickPixelPacketQ32MBS	106
* 2.8.1 HandleMemory as memoryblock	106
* 2.8.2 Blue as Single	106
* 2.8.2 ColorSpace as Integer	107
* 2.8.2 Depth as Integer	107

* 2.8.2 Fuzz as double	108
* 2.8.2 Green as Single	108
* 2.8.2 Handle as Integer	108
* 2.8.2 Index as Single	108
* 2.8.2 Matte as Boolean	109
* 2.8.2 Opacity as Single	109
* 2.8.2 Red as Single	109
– 2.7 class IMMagickPixelPacketQ8MBS	102
* 2.7.1 HandleMemory as memoryblock	103
* 2.7.2 Blue as Single	103
* 2.7.2 ColorSpace as Integer	103
* 2.7.2 Depth as Integer	103
* 2.7.2 Fuzz as double	104
* 2.7.2 Green as Single	104
* 2.7.2 Handle as Integer	105
* 2.7.2 Index as Single	105
* 2.7.2 Matte as Boolean	105
* 2.7.2 Opacity as Single	105
* 2.7.2 Red as Single	106
– 2.13 class IMMagickPixelPacketQ16MBS	115
* 2.13.1 HandleMemory as memoryblock	115
* 2.13.2 Blue as Single	115
* 2.13.2 ColorSpace as Integer	115
* 2.13.2 Depth as Integer	116
* 2.13.2 Fuzz as double	116
* 2.13.2 Green as Single	117
* 2.13.2 Handle as Integer	117
* 2.13.2 Index as Single	117
* 2.13.2 Matte as Boolean	117
* 2.13.2 Opacity as Single	118
* 2.13.2 Red as Single	118
– 2.12 class IMMagickInfoQ8MBS	110
* 2.12.1 Close	111
* 2.12.2 Adjoin as Boolean	111
* 2.12.2 BlobSupport as Boolean	111
* 2.12.2 Description as String	112
* 2.12.2 EndianSupport as Boolean	112
* 2.12.2 Handle as Integer	112
* 2.12.2 ModuleName as String	112
* 2.12.2 Name as String	113
* 2.12.2 Note as String	113

* 2.12.2 Raw as Boolean	113
* 2.12.2 SeekableStream as Boolean	114
* 2.12.2 Stealth as Boolean	114
* 2.12.2 ThreadSupport as Boolean	114
* 2.12.2 Version as String	114
– 2.15 class IMImageInfoQ16MBS	119
* 2.15.1 Clone as IMImageInfoQ16MBS	119
* 2.15.1 Close	120
* 2.15.1 DestroyImageInfo	120
* 2.15.1 HandleMemory as memoryblock	120
* 2.15.2 Adjoin as Boolean	120
* 2.15.2 Affirm as Boolean	121
* 2.15.2 Antialias as Boolean	121
* 2.15.2 Authenticate as String	121
* 2.15.2 BackgroundColor as IMColorQ16MBS	121
* 2.15.2 BorderColor as IMColorQ16MBS	121
* 2.15.2 Channel as Integer	122
* 2.15.2 Colors as Integer	122
* 2.15.2 ColorSpace as Integer	122
* 2.15.2 Compression as Integer	124
* 2.15.2 Density as String	125
* 2.15.2 Depth as Integer	125
* 2.15.2 Dither as Boolean	125
* 2.15.2 Endian as Integer	126
* 2.15.2 Extract as String	126
* 2.15.2 Filename as String	126
* 2.15.2 Font as String	126
* 2.15.2 Group as Integer	127
* 2.15.2 Handle as Integer	127
* 2.15.2 HeaderOnly as Boolean	127
* 2.15.2 Interlace as Integer	127
* 2.15.2 Magick as String	128
* 2.15.2 MatteColor as IMColorQ16MBS	129
* 2.15.2 Monochrome as Boolean	129
* 2.15.2 Orientation as Integer	129
* 2.15.2 Page as String	129
* 2.15.2 PointSize as Double	130
* 2.15.2 Preview as Integer	130
* 2.15.2 Quality as Integer	130
* 2.15.2 Release as Boolean	130
* 2.15.2 ResolutionUnits as Integer	131

* 2.15.2 SamplingFactor as String	132
* 2.15.2 Scene as Integer	132
* 2.15.2 SceneCount as Integer	132
* 2.15.2 Scenes as String	132
* 2.15.2 ServerName as String	132
* 2.15.2 Size as String	133
* 2.15.2 Temporary as Boolean	133
* 2.15.2 Texture as String	133
* 2.15.2 Type as Integer	133
* 2.15.2 Verbose as Boolean	134
* 2.15.2 View as String	134
– 2.14 class IMImageAttributeQ8MBS	118
* 2.14.1 Compression as Boolean	118
* 2.14.1 Key as String	119
* 2.14.1 Value as String	119
– 2.19 class IMImageAttributeQ32MBS	139
* 2.19.1 Compression as Boolean	139
* 2.19.1 Key as String	139
* 2.19.1 Value as String	140
– 2.16 class IMImageAffineMatrixQ32MBS	134
* 2.16.1 Constructor	135
* 2.16.2 RX as Double	135
* 2.16.2 RY as Double	135
* 2.16.2 SX as Double	135
* 2.16.2 SY as Double	135
* 2.16.2 TX as Double	136
* 2.16.2 TY as Double	136
– 2.17 class IMImageAffineMatrixQ16MBS	136
* 2.17.1 Constructor	136
* 2.17.2 RX as Double	137
* 2.17.2 RY as Double	137
* 2.17.2 SX as Double	137
* 2.17.2 SY as Double	137
* 2.17.2 TX as Double	137
* 2.17.2 TY as Double	138
– 2.18 class IMImageAttributeQ16MBS	138
* 2.18.1 Compression as Boolean	138
* 2.18.1 Key as String	138
* 2.18.1 Value as String	139
– 2.20 class IMImageAffineMatrixQ8MBS	140

* 2.20.1 Constructor	140
* 2.20.2 RX as Double	140
* 2.20.2 RY as Double	141
* 2.20.2 SX as Double	141
* 2.20.2 SY as Double	141
* 2.20.2 TX as Double	141
* 2.20.2 TY as Double	141
– 2.21 class QImageInfoQ32MBS	142
* 2.21.1 Clone as QImageInfoQ32MBS	142
* 2.21.1 Close	142
* 2.21.1 DestroyImageInfo	142
* 2.21.1 HandleMemory as memoryblock	143
* 2.21.2 Adjoin as Boolean	143
* 2.21.2 Affirm as Boolean	143
* 2.21.2 Antialias as Boolean	143
* 2.21.2 Authenticate as String	144
* 2.21.2 BackgroundColor as QColorQ32MBS	144
* 2.21.2 BorderColor as QColorQ32MBS	144
* 2.21.2 Channel as Integer	144
* 2.21.2 Colors as Integer	145
* 2.21.2 ColorSpace as Integer	145
* 2.21.2 Compression as Integer	147
* 2.21.2 Density as String	148
* 2.21.2 Depth as Integer	148
* 2.21.2 Dither as Boolean	148
* 2.21.2 Endian as Integer	148
* 2.21.2 Extract as String	149
* 2.21.2 Filename as String	149
* 2.21.2 Font as String	149
* 2.21.2 Group as Integer	149
* 2.21.2 Handle as Integer	150
* 2.21.2 HeaderOnly as Boolean	150
* 2.21.2 Interlace as Integer	150
* 2.21.2 Magick as String	151
* 2.21.2 MatteColor as QColorQ32MBS	151
* 2.21.2 Monochrome as Boolean	151
* 2.21.2 Orientation as Integer	152
* 2.21.2 Page as String	152
* 2.21.2 PointSize as Double	152
* 2.21.2 Preview as Integer	152
* 2.21.2 Quality as Integer	153

* 2.21.2 Release as Boolean	154
* 2.21.2 ResolutionUnits as Integer	154
* 2.21.2 SamplingFactor as String	154
* 2.21.2 Scene as Integer	154
* 2.21.2 SceneCount as Integer	155
* 2.21.2 Scenes as String	155
* 2.21.2 ServerName as String	155
* 2.21.2 Size as String	155
* 2.21.2 Temporary as Boolean	156
* 2.21.2 Texture as String	156
* 2.21.2 Type as Integer	156
* 2.21.2 Verbose as Boolean	157
* 2.21.2 View as String	157
– 2.24 class ImageMagickQ32MBS	174
* 2.24.1 Copyright as String	174
* 2.24.1 Features as String	175
* 2.24.1 HomeURL as String	175
* 2.24.1 InitializeMagick(path as string = ””)	175
* 2.24.1 IsMagickInstantiated as boolean	176
* 2.24.1 LoadErrorString as string	176
* 2.24.1 LoadLibrary(path as string) as boolean	176
* 2.24.1 LoadLibraryFile(path as folderitem) as boolean	177
* 2.24.1 MagickInfoList as IMMagickInfoListQ32MBS	178
* 2.24.1 MagickToMime(name as string) as string	179
* 2.24.1 NewImageInfo as IMImageInfoQ32MBS	179
* 2.24.1 NewImageList as IMImageQ32MBS	179
* 2.24.1 PackageName as String	179
* 2.24.1 QuantumDepth as String	180
* 2.24.1 QuantumDepthLibrary as Integer	180
* 2.24.1 QuantumRange as String	180
* 2.24.1 ReadImage(info as IMImageInfoQ32MBS) as IMImageQ32MBS	180
* 2.24.1 ReadImageFromString(info as IMImageInfoQ32MBS, data as string) as IMImageQ32MBS	181
* 2.24.1 ReadImageHeaderFromString(info as IMImageInfoQ32MBS, data as string) as IMImageQ32MBS	181
* 2.24.1 ReleaseDate as String	181
* 2.24.1 SetCurrentDirectory(path as folderitem) as boolean	181
* 2.24.1 Version as String	182
* 2.24.2 LastError as Integer	182
* 2.24.2 LastException as IMExceptionQ32MBS	182
– 2.22 class ImageMagickQ16MBS	157

* 2.22.1 Copyright as String	158
* 2.22.1 Features as String	158
* 2.22.1 HomeURL as String	158
* 2.22.1 InitializeMagick(path as string = "")	158
* 2.22.1 IsMagickInstantiated as boolean	159
* 2.22.1 LoadErrorString as string	159
* 2.22.1 LoadLibrary(path as string) as boolean	159
* 2.22.1 LoadLibraryFile(path as folderitem) as boolean	160
* 2.22.1 MagickInfoList as IMMagickInfoListQ16MBS	161
* 2.22.1 MagickToMime(name as string) as string	162
* 2.22.1 NewImageInfo as IMImageInfoQ16MBS	162
* 2.22.1 NewImageList as IMImageQ16MBS	162
* 2.22.1 PackageName as String	162
* 2.22.1 QuantumDepth as String	163
* 2.22.1 QuantumDepthLibrary as Integer	163
* 2.22.1 QuantumRange as String	163
* 2.22.1 ReadImage(info as IMImageInfoQ16MBS) as IMImageQ16MBS	163
* 2.22.1 ReadImageFromString(info as IMImageInfoQ16MBS, data as string) as IMImageQ16MBS	164
* 2.22.1 ReadImageHeaderFromString(info as IMImageInfoQ16MBS, data as string) as IMImageQ16MBS	164
* 2.22.1 ReleaseDate as String	164
* 2.22.1 SetCurrentDirectory(path as folderitem) as boolean	164
* 2.22.1 Version as String	165
* 2.22.2 LastError as Integer	165
* 2.22.2 LastException as IMExceptionQ16MBS	165
– 2.23 class ImageMagickQ8MBS	165
* 2.23.1 Copyright as String	166
* 2.23.1 Features as String	166
* 2.23.1 HomeURL as String	166
* 2.23.1 InitializeMagick(path as string = "")	167
* 2.23.1 IsMagickInstantiated as boolean	167
* 2.23.1 LoadErrorString as string	167
* 2.23.1 LoadLibrary(path as string) as boolean	168
* 2.23.1 LoadLibraryFile(path as folderitem) as boolean	169
* 2.23.1 MagickInfoList as IMMagickInfoListQ8MBS	170
* 2.23.1 MagickToMime(name as string) as string	170
* 2.23.1 NewImageInfo as IMImageInfoQ8MBS	170
* 2.23.1 NewImageList as IMImageQ8MBS	171
* 2.23.1 PackageName as String	171
* 2.23.1 QuantumDepth as String	171
* 2.23.1 QuantumDepthLibrary as Integer	171

* 2.23.1 QuantumRange as String	171
* 2.23.1 ReadImage(info as IImageInfoQ8MBS) as IImageQ8MBS	172
* 2.23.1 ReadImageFromString(info as IImageInfoQ8MBS, data as string) as IImageQ8MBS	172
* 2.23.1 ReadImageHeaderFromString(info as IImageInfoQ8MBS, data as string) as IImageQ8MBS	172
* 2.23.1 ReleaseDate as String	173
* 2.23.1 SetCurrentDirectory(path as folderitem) as boolean	173
* 2.23.1 Version as String	173
* 2.23.2 LastError as Integer	173
* 2.23.2 LastException as IExceptionQ8MBS	173
– 2.26 class IColorQ8MBS	184
* 2.26.1 Constructor	185
* 2.26.1 Constructor(c as color)	185
* 2.26.1 Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0)	185
* 2.26.2 Blue as UInt32	186
* 2.26.2 ColorValue as Color	186
* 2.26.2 Green as UInt32	186
* 2.26.2 Opacity as UInt32	187
* 2.26.2 Red as UInt32	187
– 2.25 class IExceptionQ16MBS	182
* 2.25.1 Close	183
* 2.25.2 Description as String	183
* 2.25.2 Reason as String	183
* 2.25.2 Severity as Integer	184
* 2.25.2 Signature as Integer	184
– 2.29 class IColorQ32MBS	191
* 2.29.1 Constructor	192
* 2.29.1 Constructor(c as color)	192
* 2.29.1 Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0)	192
* 2.29.2 Blue as UInt32	192
* 2.29.2 ColorValue as Color	193
* 2.29.2 Green as UInt32	193
* 2.29.2 Opacity as UInt32	193
* 2.29.2 Red as UInt32	193
– 2.27 class IExceptionQ8MBS	187
* 2.27.1 Close	188
* 2.27.2 Description as String	188
* 2.27.2 Reason as String	188

* 2.27.2 Severity as Integer	188
* 2.27.2 Signature as Integer	189
– 2.28 class IMExceptionQ32MBS	189
* 2.28.1 Close	190
* 2.28.2 Description as String	190
* 2.28.2 Reason as String	190
* 2.28.2 Severity as Integer	190
* 2.28.2 Signature as Integer	191
– 2.30 class IMColorQ16MBS	194
* 2.30.1 Constructor	194
* 2.30.1 Constructor(c as color)	195
* 2.30.1 Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0)	195
* 2.30.2 Blue as UInt32	195
* 2.30.2 ColorValue as Color	195
* 2.30.2 Green as UInt32	196
* 2.30.2 Opacity as UInt32	196
* 2.30.2 Red as UInt32	196
– 2.32 class IMImageQ16MBS	261
* 2.32.1 AdaptiveThreshold(width as integer, height as integer, offset as integer) as IMImageQ16MBS	261
* 2.32.1 AddNoise(NoiseType as integer) as IMImageQ16MBS	262
* 2.32.1 AffineTransformImage(matrix as IMImageAffineMatrixQ16MBS) as IMImageQ16MBS	262
* 2.32.1 AppendImageToList(img as IMImageQ16MBS)	262
* 2.32.1 Average as IMImageQ16MBS	262
* 2.32.1 BilevelChannel(channel as integer, threshold as double) as boolean	263
* 2.32.1 BlackThreshold(threshold as string) as boolean	264
* 2.32.1 BlobSize as integer	264
* 2.32.1 Blur(radius as double, sigma as double) as IMImageQ16MBS	264
* 2.32.1 BlurImageChannel(channel as integer, radius as double, sigma as double) as IMImageQ16MBS	264
* 2.32.1 BorderImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ16MBS	265
* 2.32.1 Charcoal(radius as double, sigma as double) as IMImageQ16MBS	265
* 2.32.1 Chop(x as integer, y as integer, width as integer, height as integer) as IMImageQ16MBS	266
* 2.32.1 ClipPath(path as string, inside as boolean) as boolean	266
* 2.32.1 Clone as IMImageQ16MBS	266
* 2.32.1 CloneImageAttributes(image as IMImageAttributeQ16MBS) as Boolean	267
* 2.32.1 CloneImageProfiles(SourceImage as IMImageQ16MBS) as boolean	267
* 2.32.1 Close	267

* 2.32.1 CoalesceImages as QImageQ16MBS	267
* 2.32.1 Colorize(opacity as string, PenColorRed as integer, PenColorGreen as integer, PenColorBlue as integer, PenColorOpacity as integer) as QImageQ16MBS	268
* 2.32.1 Combine(channel as integer) as QImageQ16MBS	268
* 2.32.1 CompareImageLayers(ImageLayerMethod as integer) as QImageQ16MBS	268
* 2.32.1 Composite(ComposeOperator as integer, Image as QImageQ16MBS, x as integer, y as integer)	269
* 2.32.1 ConsolidateCMYKImages as QImageQ16MBS	269
* 2.32.1 CopyPicture as picture	270
* 2.32.1 CopyPicture(x as integer, y as integer, width as integer, height as integer) as picture	270
* 2.32.1 CopyPictureMask as picture	271
* 2.32.1 CopyPictureMask(x as integer, y as integer, width as integer, height as integer) as picture	271
* 2.32.1 CopyPixel(x as integer, y as integer) as QColorQ16MBS	272
* 2.32.1 Crop(x as integer, y as integer, width as integer, height as integer) as QImageQ16MBS	272
* 2.32.1 CropImageToTiles(CropGeometry as string) as QImageQ16MBS	272
* 2.32.1 CycleColormap(displace as integer) as boolean	272
* 2.32.1 DecipherImage(passkey as string) as boolean	273
* 2.32.1 DeconstructImages as QImageQ16MBS	273
* 2.32.1 DeleteImageAttribute(key as string) as Boolean	273
* 2.32.1 Despeckle() as QImageQ16MBS	273
* 2.32.1 DestroyImage	274
* 2.32.1 DestroyImageAttributes	274
* 2.32.1 DestroyImageList	274
* 2.32.1 DestroyImageProfiles	274
* 2.32.1 Edge(radius as double) as QImageQ16MBS	274
* 2.32.1 Emboss(radius as double, sigma as double) as QImageQ16MBS	275
* 2.32.1 EncipherImage(passkey as string) as boolean	275
* 2.32.1 ExcerptImage(x as integer, y as integer, width as integer, height as integer) as QImageQ16MBS	275
* 2.32.1 ExtentImage(x as integer, y as integer, width as integer, height as integer) as QImageQ16MBS	276
* 2.32.1 FlattenImages as QImageQ16MBS	276
* 2.32.1 Flip as QImageQ16MBS	276
* 2.32.1 Flop as QImageQ16MBS	276
* 2.32.1 FrameImage(x as integer, y as integer, width as integer, height as integer, innerBevel as integer, OuterBevel as integer) as QImageQ16MBS	277
* 2.32.1 FxImage(expression as string) as QImageQ16MBS	277
* 2.32.1 GaussianBlurChannel(channel as integer, radius as double, sigma as double) as QImageQ16MBS	277
* 2.32.1 GetImageAttribute(key as string) as QImageAttributeQ16MBS	278

* 2.32.1 GetImageClippingPathAttribute as IImageAttributeQ16MBS	278
* 2.32.1 GetImageProfile(name as string) as string	278
* 2.32.1 GetNextImageAttribute as IImageAttributeQ16MBS	278
* 2.32.1 GetNextImageProfile as string	279
* 2.32.1 HandleMemory as memoryblock	279
* 2.32.1 ImagesToBlob(info as IImageInfoQ16MBS) as String	279
* 2.32.1 ImageToBlob(info as IImageInfoQ16MBS) as String	279
* 2.32.1 Implode(factor as double) as IImageQ16MBS	281
* 2.32.1 IsBlobExempt as boolean	281
* 2.32.1 IsBlobSeekable as boolean	282
* 2.32.1 IsBlobTemporary as boolean	282
* 2.32.1 Magnify as IImageQ16MBS	282
* 2.32.1 MedianFilter(radius as double) as IImageQ16MBS	282
* 2.32.1 MergeImageLayers(ImageLayerMethod as integer) as IImageQ16MBS	283
* 2.32.1 Minify as IImageQ16MBS	283
* 2.32.1 MosaicImages as IImageQ16MBS	284
* 2.32.1 MotionBlur(radius as double, sigma as double, angle as double) as IImageQ16MBS	284
* 2.32.1 NewImage(info as IImageInfoQ16MBS, width as integer, height as integer, background as IMMagickPixelPacketQ16MBS) as boolean	284
* 2.32.1 OilPaint(radius as double) as IImageQ16MBS	285
* 2.32.1 OptimizeImageLayers as IImageQ16MBS	285
* 2.32.1 OptimizeImageTransparency	286
* 2.32.1 OptimizePlusImageLayers as IImageQ16MBS	286
* 2.32.1 ProfileImage(name as string, ProfileData as string) as boolean	286
* 2.32.1 RadialBlur(angle as double) as IImageQ16MBS	287
* 2.32.1 RaiseImage(x as integer, y as integer, width as integer, height as integer, raise as boolean) as boolean	287
* 2.32.1 RandomThresholdChannel(channel as integer, thresholds as string) as boolean	287
* 2.32.1 ReduceNoise(radius as double) as IImageQ16MBS	288
* 2.32.1 RemoveDuplicateLayers	288
* 2.32.1 RemoveFirstImageFromList as IImageQ16MBS	289
* 2.32.1 RemoveImageProfile(name as string) as string	289
* 2.32.1 RemoveZeroDelayLayers	289
* 2.32.1 ResetImageAttributeIterator	290
* 2.32.1 ResetImageProfileIterator	290
* 2.32.1 Resize(width as integer, height as integer, FilterID as integer, blur as double) as IImageQ16MBS	290
* 2.32.1 RGBTransformImage(Colorspace as integer) as boolean	290
* 2.32.1 Roll(x as integer, y as integer) as IImageQ16MBS	292
* 2.32.1 Rotate(degrees as double) as IImageQ16MBS	292
* 2.32.1 Sample(width as integer, height as integer) as IImageQ16MBS	292

* 2.32.1 Scale(width as integer, height as integer) as IImageQ16MBS	293
* 2.32.1 SetImageAttribute(key as string, value as string) as boolean	293
* 2.32.1 SetImageColorspace(Colorspace as integer) as boolean	294
* 2.32.1 SetImageProfile(name as string, ProfileData as string) as boolean	294
* 2.32.1 SetPicture(pic as picture, x as integer, y as integer)	294
* 2.32.1 SetPictureMask(maskpic as picture, x as integer, y as integer)	295
* 2.32.1 SetPixel(x as integer, y as integer, newPixel as IMColorQ16MBS)	295
* 2.32.1 Shade(gray as boolean, azimuth as double, elevation as double) as IImageQ16MBS	296
* 2.32.1 SharpenChannel(channel as integer, radius as double, sigma as double) as IImageQ16MBS	296
* 2.32.1 Shave(x as integer, y as integer, width as integer, height as integer) as IImageQ16MBS	297
* 2.32.1 Shear(Xshear as double, Yshear as double) as IImageQ16MBS	298
* 2.32.1 Solarize(factor as double) as boolean	298
* 2.32.1 Splice(x as integer, y as integer, width as integer, height as integer) as IImageQ16MBS	298
* 2.32.1 Spread(radius as double) as IImageQ16MBS	299
* 2.32.1 Stegano(watermarkImage as IImageQ16MBS) as IImageQ16MBS	299
* 2.32.1 Stereo(otherImage as IImageQ16MBS) as IImageQ16MBS	299
* 2.32.1 Swirl(degrees as double) as IImageQ16MBS	300
* 2.32.1 Thumbnail(width as integer, height as integer) as IImageQ16MBS	300
* 2.32.1 TransformImage(CropGeometry as string, ImageGeometry as string) as boolean	300
* 2.32.1 TransformImages(CropGeometry as string, ImageGeometry as string) as boolean	301
* 2.32.1 TransformRGBImage(Colorspace as integer) as boolean	301
* 2.32.1 TransposeImage as IImageQ16MBS	301
* 2.32.1 TransverseImage as IImageQ16MBS	302
* 2.32.1 Trim as IImageQ16MBS	302
* 2.32.1 UnsharpMaskChannel(channel as integer, radius as double, sigma as double, amount as double, threshold as double) as IImageQ16MBS	303
* 2.32.1 Wave(amplitude as double, wavelength as double) as IImageQ16MBS	303
* 2.32.1 WhiteThreshold(threshold as string) as boolean	304
* 2.32.1 WriteImage(info as IImageInfoQ16MBS) as boolean	304
* 2.32.2 BackgroundColor as IMColorQ16MBS	304
* 2.32.2 Bias as double	305
* 2.32.2 BlurFactor as double	305
* 2.32.2 BorderColor as IMColorQ16MBS	305
* 2.32.2 Colors as Integer	305
* 2.32.2 ColorSpace as Integer	305
* 2.32.2 Compression as Integer	306
* 2.32.2 Depth as Integer	307
* 2.32.2 Directory as String	307

* 2.32.2 Endian as Integer	307
* 2.32.2 Filename as String	308
* 2.32.2 Filter as Integer	308
* 2.32.2 Fuzz as double	309
* 2.32.2 Gamma as double	309
* 2.32.2 Geometry as String	309
* 2.32.2 Gravity as Integer	309
* 2.32.2 Handle as Integer	310
* 2.32.2 Height as integer	310
* 2.32.2 Interlace as Integer	310
* 2.32.2 LastError as Integer	311
* 2.32.2 LastException as IMExceptionQ16MBS	311
* 2.32.2 Magick as String	311
* 2.32.2 Matte as Boolean	312
* 2.32.2 MatteColor as IMColorQ16MBS	312
* 2.32.2 Montage as String	312
* 2.32.2 Offset as Integer	312
* 2.32.2 Orientation as Integer	312
* 2.32.2 Quality as Integer	313
* 2.32.2 Release as Boolean	314
* 2.32.2 RenderingIntent as Integer	315
* 2.32.2 ResolutionUnits as Integer	315
* 2.32.2 ResolutionX as double	315
* 2.32.2 ResolutionY as double	316
* 2.32.2 Scene as Integer	316
* 2.32.2 StorageClass as Integer	316
* 2.32.2 Taint as Boolean	317
* 2.32.2 Width as integer	317
* 2.32.3 kBackgroundDispose = 2	317
* 2.32.3 kCoalesceLayer = 1	317
* 2.32.3 kCompareAnyLayer = 2	318
* 2.32.3 kCompareClearLayer = 3	318
* 2.32.3 kCompareOverlayLayer = 4	318
* 2.32.3 kCompositeLayer = & h0000000C	318
* 2.32.3 kDisposeLayer = 5	318
* 2.32.3 kFlattenLayer = & h0000000E	318
* 2.32.3 kMergeLayer = & h0000000D	319
* 2.32.3 kMosaicLayer = & h0000000F	319
* 2.32.3 kNoneDispose = 1	319
* 2.32.3 kOptimizeImageLayer = 7	319
* 2.32.3 kOptimizeLayer = 6	319
* 2.32.3 kOptimizePlusLayer = 8	319

* 2.32.3 kOptimizeTransLayer = 9	320
* 2.32.3 kPreviousDispose = 3	320
* 2.32.3 kRemoveDupsLayer = & h0000000A	320
* 2.32.3 kRemoveZeroLayer = & h0000000B	320
* 2.32.3 kUndefinedDispose = 0	320
* 2.32.3 kUndefinedLayer = 0	320
* 2.32.3 kUnrecognizedDispose = 0	321
– 2.33 class QImageInfoQ8MBS	321
* 2.33.1 Clone as QImageInfoQ8MBS	321
* 2.33.1 Close	321
* 2.33.1 DestroyImageInfo	322
* 2.33.1 HandleMemory as memoryblock	322
* 2.33.2 Adjoin as Boolean	322
* 2.33.2 Affirm as Boolean	322
* 2.33.2 Antialias as Boolean	323
* 2.33.2 Authenticate as String	323
* 2.33.2 BackgroundColor as QColorQ8MBS	323
* 2.33.2 BorderColor as QColorQ8MBS	323
* 2.33.2 Channel as Integer	323
* 2.33.2 Colors as Integer	324
* 2.33.2 ColorSpace as Integer	324
* 2.33.2 Compression as Integer	326
* 2.33.2 Density as String	327
* 2.33.2 Depth as Integer	327
* 2.33.2 Dither as Boolean	327
* 2.33.2 Endian as Integer	328
* 2.33.2 Extract as String	328
* 2.33.2 Filename as String	328
* 2.33.2 Font as String	328
* 2.33.2 Group as Integer	329
* 2.33.2 Handle as Integer	329
* 2.33.2 HeaderOnly as Boolean	329
* 2.33.2 Interlace as Integer	329
* 2.33.2 Magick as String	330
* 2.33.2 MatteColor as QColorQ8MBS	331
* 2.33.2 Monochrome as Boolean	331
* 2.33.2 Orientation as Integer	331
* 2.33.2 Page as String	331
* 2.33.2 PointSize as Double	332
* 2.33.2 Preview as Integer	332
* 2.33.2 Quality as Integer	332

* 2.33.2 Release as Boolean	332
* 2.33.2 ResolutionUnits as Integer	333
* 2.33.2 SamplingFactor as String	334
* 2.33.2 Scene as Integer	334
* 2.33.2 SceneCount as Integer	334
* 2.33.2 Scenes as String	334
* 2.33.2 ServerName as String	334
* 2.33.2 Size as String	335
* 2.33.2 Temporary as Boolean	335
* 2.33.2 Texture as String	335
* 2.33.2 Type as Integer	335
* 2.33.2 Verbose as Boolean	336
* 2.33.2 View as String	336

Chapter 2

Image Magick

2.1 class IMImageQ8MBS

class IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for an Image Magick Image in memory.

Notes:

Can exist with or without pixel data.

For more details please check the ImageMagick documentation.

2.1.1 Methods

AdaptiveThreshold(width as integer, height as integer, offset as integer) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** AdaptiveThreshold selects an individual threshold for each pixel based on the range of intensity values in its local neighborhood.

Notes:

This allows for thresholding of an image whose global intensity histogram doesn't contain distinctive peaks. Sets the last exception property.

width: The width of the local neighborhood.
 height: The height of the local neighborhood.
 offset: The mean offset.

For more details please check the ImageMagick documentation.

AddNoise(NoiseType as integer) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds random noise to the image.

Notes:

Constants

UndefinedNoise	=0
UniformNoise	=1
GaussianNoise	=2
MultiplicativeGaussianNoise	=3
ImpulseNoise	=4
LaplacianNoise	=5
PoissonNoise	=6

For more details please check the ImageMagick documentation.
 Sets the last exception property.

AffineTransformImage(matrix as IMImageAffineMatrixQ8MBS) as IMImageQ8MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Transforms an image as dictated by the affine matrix.

AppendImageToList(img as IMImageQ8MBS)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds an image to the image list.

Notes: For more details please check the ImageMagick documentation.

Average as IMImageQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Average() method takes a set of images and averages them together.

Notes:

Each image in the set must have the same width and height. Average() returns a single image with each corresponding pixel component of each image averaged. On failure, a nil image is returned and exception describes the reason for the failure.

Sets the last exception property.

For more details please check the ImageMagick documentation.

BilevelChannel(channel as integer, threshold as double) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Changes the value of individual pixels based on the intensity of each pixel channel.

Notes:

The result is a high-contrast image.

channel: The channel type.

threshold: define the threshold values.

Constants for channel:

For more details please check the ImageMagick documentation.

```

const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel      = & h0001
const CyanChannel      = & h0001
const GreenChannel     = & h0002
const MagentaChannel   = & h0002
const BlueChannel      = & h0004
const YellowChannel    = & h0004
const AlphaChannel     = & h0008
const OpacityChannel   = & h0008
const BlackChannel     = & h0020
const IndexChannel     = & h0020
const AllChannels      = & h7ffffff

```

BlackThreshold(threshold as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** BlackThreshold is like Threshold but forces all pixels below the threshold into black while leaving all pixels above the threshold unchanged.

Notes:

No exceptions are generated.

threshold: Define the threshold value. (ASCII string)

For more details please check the ImageMagick documentation.

BlobSize as integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The expected size for this image written to a file.

Notes: For more details please check the ImageMagick documentation.

Blur(radius as double, sigma as double) as IMImageQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blurs an image.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, the radius should be larger than sigma. Use a radius of 0 and BlurImage selects a suitable radius for you.

radius: The radius of the Gaussian, in pixels, not counting the center pixel.
 sigma: The standard deviation of the Gaussian, in pixels.

For more details please check the ImageMagick documentation.

BlurImageChannel(channel as integer, radius as double, sigma as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blurs an image.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, the radius should be larger than sigma. Use a radius of 0 and BlurImageChannel selects a suitable radius for you.

channel: The channel type.

radius: The radius of the Gaussian, in pixels, not counting the center pixel.

sigma: The standard deviation of the Gaussian, in pixels.

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel     = & h0001
const CyanChannel     = & h0001
const GreenChannel    = & h0002
const MagentaChannel  = & h0002
const BlueChannel     = & h0004
const YellowChannel   = & h0004
const AlphaChannel    = & h0008
const OpacityChannel  = & h0008
const BlackChannel    = & h0020
const IndexChannel    = & h0020
const AllChannels     = & h7ffffff
```

For more details please check the ImageMagick documentation.

BorderImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ8MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Surrounds the image with a border of the color defined by the bordercolor member of the image.

Notes: The width and height of the border are defined by the corresponding parameters.

Charcoal(radius as double, sigma as double) as IMImageQ8MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Charcoal creates a new image that is a copy of an existing one with the edge highlighted.

Notes:

radius: the radius of the pixel neighborhood.

sigma: The standard deviation of the Gaussian, in pixels.

Returns nil on any error.

Sets the last exception property.

Chop(x as integer, y as integer, width as integer, height as integer) as IMImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Chop removes a region of an image and collapses the image to occupy the removed portion.

Notes:

Returns nil on any error.

Sets the last exception property.

ClipPath(path as string, inside as boolean) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the image clip mask based any clipping path information if it exists.

Notes:

Returns true on success and false on any error.

pathname: name of clipping path resource. If name is preceded by # , use clipping path numbered by name.
 inside: if true, later operations take effect inside clipping path. Otherwise later operations take effect outside clipping path.

Clone as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a copy of this image object.

Notes: For more details please check the ImageMagick documentation.

CloneImageAttributes(image as IMImageAttributeQ8MBS) as Boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** CloneImageAttributes() clones one or more image attributes.

Notes: Returns false on any error.

CloneImageProfiles(SourceImage as IMImageQ8MBS) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clones one or more image profiles.

Notes: Returns false on any error and true on success.

Close

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.

Notes:

There is no need to call this method except you want to free all resources used by this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

CoalesceImages as IMImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** CoalesceImages composites a set of images while respecting any page offsets and disposal methods.

Notes:

GIF, MIFF, and MNG animation sequences typically start with an image background and each subsequent image varies in size and offset. CoalesceImages() returns a new sequence where each image in the sequence is the same size as the first and composited with the next image in the sequence.

Returns nil on any error.

Sets the last exception property.

Colorize(opacity as string, PenColorRed as integer, PenColorGreen as integer, PenColorBlue as integer, PenColorOpacity as integer) as IMImageQ8MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method ColorizeImage creates a new image that is a copy of an existing one with the image pixels colorized.

Notes:

The colorization is controlled with the pen color and the opacity levels.

opacity: A character string indicating the level of opacity as a percentage (0-100).

PenColorRed, PenColorGreen, PenColorBlue and PenColorOpacity define the pen color used.

Returns nil on any error.

Sets the last exception property.

Combine(channel as integer) as IMImageQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Combines one or more images into a single image.

Notes:

The grayscale value of the pixels of each image in the sequence is assigned in order to the specified channels of the combined image. The typical ordering would be image 1 =>Red, 2 =>Green, 3 =>Blue, etc.

The lastexception property is set.

CompareImageLayers(ImageLayerMethod as integer) as IMImageQ8MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** CompareImageLayers() compares each image with the next in a sequence and returns the minimum bounding region of all the pixel differences (of the mageLayerMethod specified) it discovers.

Notes:

Images do NOT have to be the same size, though it is best that all the images are 'coalesced' (images are all the same size, on a flattened canvas, so as to represent exactly how an specific frame should look).

No GIF dispose methods are applied, so GIF animations must be coalesced before applying this image operator to find differences to them.

ImageLayerMethod:

the layers type to compare images with. Must be one of... CompareAnyLayer, CompareClearLayer, CompareOverlayLayer.

Can raise an exception.

Composite(ComposeOperator as integer, Image as IMImageQ8MBS, x as integer, y as integer)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the second image composited onto the first at the specified offsets.

Notes:

compose: Specifies an image composite operator.

Image: The second image.

x: An integer that specifies the column offset of the composited image.

y: An integer that specifies the row offset of the composited image.

No error code and exception!

ConsolidateCMYKImages as IImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Consolidates a sequence of CMYK images.

Notes:

Returns nil on any error.
Sets the last exception property.

CopyPicture as picture

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IImageQ8MBS // your image
Canvas1.Backdrop=image.CopyPicture
```

Notes:

Sets the last exception property.
Returns nil on any error.
This method works only for bitmap images.
See also:

- 2.1.1 CopyPicture(x as integer, y as integer, width as integer, height as integer) as picture 38

CopyPicture(x as integer, y as integer, width as integer, height as integer) as picture

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies a portion of the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IImageQ8MBS // your image
Canvas1.Backdrop=image.CopyPicture(0,0,image.Width,image.Height)
```

Notes:

Sets the last exception property.
 Returns nil on any error.
 This method works only for bitmap images.
 x and y are zero based.
 See also:

- 2.1.1 CopyPicture as picture

38

CopyPictureMask as picture

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the mask of the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IMImageQ8MBS // your image
Canvas1.Backdrop=image.CopyPictureMask
```

Notes:

Sets the last exception property.
 Returns nil on any error.
 This method works only for bitmap images.
 See also:

- 2.1.1 CopyPictureMask(x as integer, y as integer, width as integer, height as integer) as picture 39

CopyPictureMask(x as integer, y as integer, width as integer, height as integer) as picture

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies a portion of the mask of the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IMImageQ8MBS // your image
Canvas1.Backdrop=image.CopyPictureMask(0,0,image.Width,image.Height)
```

Notes:

Sets the last exception property.
 Returns nil on any error.
 This method works only for bitmap images.
 x and y are zero based.
 See also:

- 2.1.1 CopyPictureMask as picture

39

CopyPixel(x as integer, y as integer) as IMColorQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies a pixel.

Notes:

Returns nil on any error.
 This method works only for bitmap images.
 x and y are zero based.

Crop(x as integer, y as integer, width as integer, height as integer) as IMImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Crop extracts a region of the image starting at the offset defined by geometry.

Notes:

Returns nil on any error.
 Sets the last exception property.

CropImageToTiles(CropGeometry as string) as IMImageQ8MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Crops a single image, into a possible list of tiles.

Notes: This may include a single sub-region of the image. This basically applies all the normal geometry flags for Crop.

CycleColormap(displace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Displaces an image's colormap by a given number of positions.

Notes:

If you cycle the colormap a number of times you can produce a psychedelic effect.

Returns true on success.

displace: displace the colormap this amount.

DecipherImage(passkey as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts cipher pixels to plain pixels.

Notes:

Passkey: decipher cipher pixels with this passphrase.

Returns true on success.

DeconstructImages as IMImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** DeconstructImages() compares each image with the next in a sequence and returns the minimum bounding region of all differences from the first image.

Notes:

Returns nil on any error.

Sets the last exception property.

DeleteImageAttribute(key as string) as Boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** DeleteImageAttribute() deletes an attribute from the image.

Notes: Returns false on any error.

Despeckle() as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reduces the speckle noise in an image while perserving the edges of the original image.

Notes:

Sets the last exception property.

For more details please check the ImageMagick documentation.

DestroyImage

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Releases the memory used for this image and sets handle to 0.

Notes:

For more details please check the ImageMagick documentation.

The destructor will call this for you if release=true.

DestroyImageAttributes

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Deallocates memory associated with the image attribute list.

DestroyImageList

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Destroys the image list and sets the handle to 0.

Notes:

For more details please check the ImageMagick documentation.

The destructor will call this for you if release=true.

DestroyImageProfiles

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Releases memory associated with an image profile map.

Edge(radius as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds edges in an image.

Notes:

Radius defines the radius of the convolution filter. Use a radius of 0 and Edge selects a suitable radius for you.

Sets the last exception property.

For more details please check the ImageMagick documentation.

Emboss(radius as double, sigma as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a grayscale image with a three-dimensional effect.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, radius should be larger than sigma. Use a radius of 0 and Emboss selects a suitable radius for you.

Sets the last exception property.

For more details please check the ImageMagick documentation.

EncipherImage(passkey as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts pixels to cipher-pixels.

Notes:

passkey: encipher pixels with this passphrase.
Returns true on success.

ExcerptImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ8MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a excerpt of the image as defined by the geometry.

Notes: Define the region of the image to extend with x, y, width, and height.

ExtentImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ8MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Extends the image as defined by the geometry, gravity, and image background color.

Notes:

Define the region of the image to extend with x, y, width, and height.

Set the (x,y) offset of the geometry to move the original image relative to the extended image.

FlattenImages as IMImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Flatten composites all images from the current image pointer to the end of the image list and returns a single flattened image.

Notes:

Returns nil on any error.

Sets the last exception property.

Flip as IMImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Flip creates a vertical mirror image by reflecting the pixels around the central x-axis.

Notes:

Returns nil on any error.
Sets the last exception property.

Flop as IMImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Flop creates a horizontal mirror image by reflecting the pixels around the central y-axis.

Notes:

Returns nil on any error.
Sets the last exception property.

FrameImage(x as integer, y as integer, width as integer, height as integer, innerBevel as integer, OuterBevel as integer) as IMImageQ8MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a simulated three-dimensional border around the image.

Notes: The color of the border is defined by the MatteColor of image. Width and height specify the border width of the vertical and horizontal sides of the frame. innerBevel and OuterBevel indicate the width of the inner and outer shadows of the frame.

FxImage(expression as string) as IMImageQ8MBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** FxImage() applies a mathematical expression to the specified image.

Notes: Can raise an exception.

GaussianBlurChannel(channel as integer, radius as double, sigma as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blurs an image.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, the radius should be larger than sigma. Use a radius of 0 and GaussianBlur selects a suitable radius for you.

Sets the last exception property.

radius: the radius of the Gaussian, in pixels, not counting the center pixel.

channel: The channel type.

sigma: the standard deviation of the Gaussian, in pixels.

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel      = & h0001
const CyanChannel      = & h0001
const GreenChannel     = & h0002
const MagentaChannel   = & h0002
const BlueChannel      = & h0004
const YellowChannel    = & h0004
const AlphaChannel     = & h0008
const OpacityChannel   = & h0008
const BlackChannel     = & h0020
const IndexChannel     = & h0020
const AllChannels      = & h7fffffff
```

For more details please check the ImageMagick documentation.

GetImageAttribute(key as string) as IMImageAttributeQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** GetImageAttribute searches the list of image attributes and returns a reference to the attribute if it exists otherwise nil.

GetImageClippingPathAttribute as IMImageAttributeQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** GetImageClippingPathAttribute searches the list of image attributes and returns a reference to a clipping path if it exists otherwise nil.

GetImageProfile(name as string) as string

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets a profile associated with an image by name.

Notes: Returns "" on any error.

GetNextImageAttribute as IImageAttributeQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** GetNextImageAttribute() gets the next image attribute.

Notes: Returns nil on any error.

GetNextImageProfile as string

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets the next profile name for an image.

Notes: Returns "" on any error.

HandleMemory as memoryblock

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The content of the whole Image structure copied into a memoryblock.

Notes: Returns nil on any error.

ImagesToBlob(info as IImageInfoQ8MBS) as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** ImagesToBlob implements direct to memory image formats.

Notes:

It returns the image sequence as a string. The magick member of the ImageInfo structure determines the format of the returned blob (GIF, JPEG, PNG, etc.)

Note, some image formats do not permit multiple images to the same image stream (e.g. JPEG). in this instance, just the first image of the sequence is returned as a blob.

Sets the last exception property and returns "" on any error.
For more details please check the ImageMagick documentation.

ImageToBlob(info as IMImageInfoQ8MBS) as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** ImagesToBlob implements direct to memory image formats.

Example:

```
dim im as ImageMagickQ8MBS // global
```

```
Function IMPictureToString(p as picture, magick as string, quality as integer) As string
```

```
dim image as new IMImageQ8MBS
```

```
dim imageinfo as IMImageInfoQ8MBS
```

```
dim s,data as string
```

```
dim impp as new IMMagickPixelPacketQ8MBS
```

```
// empty string for nil picture
```

```
if p = nil then
```

```
Return ""
```

```
end if
```

```
// create a new picture info
```

```
imageinfo = im.NewImageInfo
```

```
imageinfo.ColorSpace=1
```

```
// only color space is needed. 1 for RGB.
```

```
// background color of image
```

```
impp.red = 0
```

```
impp.Green = 0
```

```
impp.Blue = 0
```

```
// creates a new image object
```

```
if not image.NewImage(imageinfo,p.Width,p.Height,impp) then
```

```
Return ""
```

```
end if
```

```
// copy RB picture into IM Image at position 0/0
```

```
image.ColorSpace = 1
```

```
image.SetPicture(p,0,0)

// set compression data
imageinfo.Magick = magick
imageinfo.Quality = quality

// and rendering intent: 2=PerceptualIntent
image.RenderingIntent = 2

// create image data
data = image.ImageToBlob(imageinfo)

// release memory
image.DestroyImage
imageinfo.DestroyImageInfo

// return result
Return data

Exception
// in case of an exception return nothing
Return ""

End Function
```

Notes:

It returns the image sequence as a string. The magick member of the ImageInfo structure determines the format of the returned blob (GIF, JPEG, PNG, etc.)

Note, some image formats do not permit multiple images to the same image stream (e.g. JPEG). in this instance, just the first image of the sequence is returned as a blob.

Sets the last exception property and returns "" on any error.
For more details please check the ImageMagick documentation.

Implode(factor as double) as IMImageQ8MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method `ImplodeImage` creates a new image that is a copy of an existing one with the image pixels "implode" by the specified percentage.

Notes:

factor: A double value that defines the extent of the implosion.

Returns nil on any error.

Sets the last exception property.

IsBlobExempt as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the blob is exempt.

Notes: For more details please check the ImageMagick documentation.

IsBlobSeekable as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the blob is seekable.

Notes: For more details please check the ImageMagick documentation.

IsBlobTemporary as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the blob is temporary.

Notes: For more details please check the ImageMagick documentation.

Magnify as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A convenience method that scales an image proportionally to twice its size.

Notes:

Sets the last exception property.
For more details please check the ImageMagick documentation.

MedianFilter(radius as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Applies a digital filter that improves the quality of a noisy image.

Notes:

Each pixel is replaced by the median in a set of neighboring pixels as defined by radius.
Sets the last exception property.

For more details please check the ImageMagick documentation.

MergeImageLayers(ImageLayerMethod as integer) as IMImageQ8MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** MergeImageLayers() composes all the image layers from the current given image onward to produce a single image of the merged layers.

Notes:

The initial canvas's size depends on the given ImageLayerMethod, and is initialized using the first images background color. The images are then composited onto that image in sequence using the given composition that has been assigned to each individual image.

ImageLayerMethod:

the method of selecting the size of the initial canvas.

MergeLayer: Merge all layers onto a canvas just large enough to hold all the actual images. The virtual canvas of the first image is preserved but otherwise ignored.

FlattenLayer: Use the virtual canvas size of first image. Images which fall outside this canvas is clipped. This can be used to 'fill out' a given virtual canvas.

MosaicLayer: Start with the virtual canvas of the first image, enlarging left and right edges to contain all images. Images with negative offsets will be clipped.

Can raise an exception.

Minify as `IMImageQ8MBS`

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A convenience method that scales an image proportionally to half its size.

Notes:

Sets the last exception property.

For more details please check the ImageMagick documentation.

MosaicImages as `IMImageQ8MBS`

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** MosaicImages inlays an image sequence to form a single coherent picture.

Notes:

It returns a single image with each image in the sequence composited at the location defined by the page member of the image structure.

Returns nil on any error.

Sets the last exception property.

MotionBlur(radius as double, sigma as double, angle as double) as `IMImageQ8MBS`

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Simulates motion blur.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma). For reasonable results, radius should be larger than sigma. Use a radius of 0 and MotionBlur selects a suitable radius for you. Angle gives the angle of the blurring motion.

Sets the last exception property.

For more details please check the ImageMagick documentation.

NewImage(info as IMImageInfoQ8MBS, width as integer, height as integer, background as IMMagickPixelPacketQ8MBS) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new image.

Example:

```
dim im as ImageMagickQ8MBS // global
dim p as picture
dim imageinfo as IMImageInfoQ8MBS
dim image as IMImageQ8MBS
dim b as new IMMagickPixelPacketQ8MBS
b.Blue=65535
b.ColorSpace=1 // RGB
b.Depth=16

imageinfo = im.NewImageInfo
imageinfo.Depth=16
imageinfo.ColorSpace=1

//this should read any image IM understands
image = new IMImageQ8MBS
if image.NewImage(imageinfo,500,500,b) then
p=NewPicture(300,300,32)
p.Graphics.ForeColor=Rgb(255,0,0)
p.Graphics.FillOval 0,0,300,300
image.SetPicture p,0,0
else
MsgBox "failed"
end if
```

Notes: Returns false on failure and true on success.

OilPaint(radius as double) as IMImageQ8MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method OilPaintImage creates a new image that is a copy of an existing one with each pixel component replaced with the color

of greatest frequency in a circular neighborhood.

Notes:

radius parameter: radius of the circular neighborhood.

Returns nil on any error.

Sets the last exception property.

OptimizeImageLayers as IMImageQ8MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** OptimizeImageLayers() compares each image the GIF disposed forms of the previous image in the sequence.

Notes:

From this it attempts to select the smallest cropped image to replace each frame, while preserving the results of the GIF animation.

Can raise an exception.

OptimizeImageTransparency

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** OptimizeImageTransparency() takes a frame optimized GIF animation, and compares the overlaid pixels against the disposal image resulting from all the previous frames in the animation.

Notes:

Any pixel that does not change the disposal image (and thus does not effect the outcome of an overlay) is made transparent.

WARNING: This modifies the current images directly, rather than generate a new image sequence.

Can raise an exception.

OptimizePlusImageLayers as IMImageQ8MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** OptimizeImagePlusLayers() is exactly as OptimizeImageLayers(), but may also add or even remove extra frames in the animation, if it improves the total number of pixels in the resulting GIF animation.

Notes: Can raise an exception.

ProfileImage(name as string, ProfileData as string) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds or removes a ICC, IPTC, or generic profile from an image.

Notes:

If the ProfileData is "", it is removed from the image otherwise added. Use a name of '*' and a ProfileData of "" to remove all profiles from the image.

Returns false on any error and true on success.

RadialBlur(angle as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** RadialBlur applies a radial blur to the image.

Notes:

angle: The angle of the radial blur.

Sets the last exception property.

For more details please check the ImageMagick documentation.

RaiseImage(x as integer, y as integer, width as integer, height as integer, raise as boolean) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a simulated three-dimensional button-like effect by lightening and darkening the edges of the image.

Notes:

Width and height define the width of the vertical and horizontal edge of the effect.
 raise: A value other than zero creates a 3-D raise effect, otherwise it has a lowered effect.

RandomThresholdChannel(channel as integer, thresholds as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Canges the value of individual pixels based on the intensity of each pixel compared to a random threshold.

Notes:

The result is a low-contrast, two color image.

channel: The channel or channels to be thresholded.

thresholds: a geometry string containing low,high thresholds. If the string contains 2x2, 3x3, or 4x4, an ordered dither of order 2, 3, or 4 is performed instead. (ASCII string)

Sets the last exception property.

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel      = & h0001
const CyanChannel      = & h0001
const GreenChannel     = & h0002
const MagentaChannel   = & h0002
const BlueChannel      = & h0004
const YellowChannel    = & h0004
const AlphaChannel     = & h0008
const OpacityChannel   = & h0008
const BlackChannel     = & h0020
const IndexChannel     = & h0020
const AllChannels      = & h7ffffff
```

For more details please check the ImageMagick documentation.

ReduceNoise(radius as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Smooths the contours of an image while still preserving edge information.

Notes:

The algorithm works by replacing each pixel with its neighbor closest in value. A neighbor is defined by radius. Use a radius of 0 and ReduceNoise selects a suitable radius for you.

For more details please check the ImageMagick documentation.

RemoveDuplicateLayers

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes any image that is exactly the same as the next image in the given image list.

Notes:

Image size and virtual canvas offset must also match, though not the virtual canvas size itself.

No check is made with regards to image disposal setting, though it is the dispose setting of later image that is kept. Also any time delays are also added together. As such coalesced image animations should still produce the same result, though with duplicate frames merged into a single frame.

RemoveFirstImageFromList as IMImageQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the first image from the image list and returns the image.

Notes:

Returns nil on any error.

For more details please check the ImageMagick documentation.

RemoveImageProfile(name as string) as string

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes a profile from the image-map by its name.

RemoveZeroDelayLayers

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes any image that as a zero delay time.

Notes:

Such images generally represent intermediate or partial updates in GIF animations used for file optimization. They are not ment to be displayed to users of the animation. Viewable images in an animation should have a time delay of 3 or more centi-seconds (hundredths of a second).

However if all the frames have a zero time delay, then either the animation is as yet incomplete, or it is not a GIF animation. This is a non-sensible situation, so no image will be removed and a 'Zero Time Animation' warning (exception) given.

No warning will be given if no image was removed because all images had an appropriate non-zero time delay set.

Due to the special requirements of GIF disposal handling, GIF animations should be coalesced first, before calling this function, though that is not a requirement.

ResetImageAttributeIterator

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** ResetImageAttributeIterator() resets the image attributes iterator.

Notes: Use it in conjunction with GetNextImageAttribute() to iterate over all the values associated with an image.

ResetImageProfileIterator

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Resets the image profile iterator.

Notes: Use it in conjunction with `GetNextImageProfile()` to iterate over all the profiles associated with an image.

Resize(width as integer, height as integer, FilterID as integer, blur as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Scales an image to the desired dimensions.

Notes:

Constants for the FilterID:

```
const PointFilter      =1
const BoxFilter       =2
const TriangleFilter  =3
const HermiteFilter   =4
const HanningFilter   =5
const HammingFilter   =6
const BlackmanFilter  =7
const GaussianFilter  =8
const QuadraticFilter =9
const CubicFilter     =10
const CatromFilter    =11
const MitchellFilter  =12
const LanczosFilter   =13
const BesselFilter    =14
const SincFilter      =15
```

Most of the filters are FIR (finite impulse response), however, Bessel, Gaussian, and Sinc are IIR (infinite impulse response). Bessel and Sinc are windowed (brought down to zero) with the Blackman filter.

Sets the last exception property.

RGBTransformImage(Colorspace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method `RGBTransformImage` converts the reference image from RGB to an alternate colorspace.

Notes:

The transformation matrices are not the standard ones: the weights are rescaled to normalized the range of the transformed values to be [0..MaxRGB] .

colorspace: An integer value that indicates which colorspace to transform the image.

Returns false on any error and true on success.

constants:

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

Roll(x as integer, y as integer) as IMImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Roll offsets an image as defined by x and y.

Notes:

Returns nil on any error.

Sets the last exception property.

Rotate(degrees as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Rotation of an image.

Notes:

Method RotateImage creates a new image that is a rotated copy of an existing one. Positive angles rotate counter-clockwise (right-hand rule), while negative angles rotate clockwise. Rotated images are usually larger than the originals and have 'empty' triangular corners. X axis. Empty triangles left over from shearing the image are filled with the color specified by the image background_ color. RotateImage allocates the memory necessary for the new Image structure and returns a pointer to the new image.

Method RotateImage is based on the paper "A Fast Algorithm for General Raster Rotatation" by Alan W. Paeth. RotateImage is adapted from a similar method based on the Paeth paper written by Michael Halle of the Spatial Imaging Group, MIT Media Lab.

degrees: Specifies the number of degrees to rotate the image.

Sets the lastexception property.

Returns nil on low memory.

For more details please check the ImageMagick documentation.

Sample(width as integer, height as integer) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Scales an image to the desired dimensions with pixel sampling.

Notes:

Unlike other scaling methods, this method does not introduce any additional color into the scaled image.

For more details please check the ImageMagick documentation.

Sets the last exception property.

Scale(width as integer, height as integer) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Changes the size of an image to the given dimensions.

Example:

```
dim image as IMImageQ8MBS // your image
image=Image.Scale(100,80)
```

Notes:

This method was designed by Bob Friesenhahn as a low cost thumbnail generator.

columns: The number of columns in the scaled image.

rows: The number of rows in the scaled image.

Sets the last exception property.

For more details please check the ImageMagick documentation.

SetImageAttribute(key as string, value as string) as boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** SetImageAttribute searches the list of image attributes and replaces the attribute value.

Notes: If it is not found in the list, the attribute name and value is added to the list. If the attribute exists in the list, the value is concatenated to the attribute. SetImageAttribute returns True if the attribute is successfully concatenated or added to the list, otherwise False. If the value is "", the matching key is deleted from the list.

SetImageColorspace(Colorspace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the colorspace member of the Image structure.

Notes: Returns false on any error and true on success.

SetImageProfile(name as string, ProfileData as string) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a named profile to the image.

Notes:

If a profile with the same name already exists, it is replaced. This method differs from the ProfileImage() method in that it does not apply CMS color profiles.

name: The profile name.

profiledata: The binary data of the profile.

Returns false on any error and true on success.

SetPicture(pic as picture, x as integer, y as integer)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the pixels from a given Realbasic picture into the Image Magick Image at the given location.

Example:

```
dim image as IMImageQ8MBS // your image
dim p as picture

p=NewPicture(32,32,32)
p.Graphics.ForeColor=rgb(0,255,0)
p.Graphics.FillRect 0,0,32,32

image.SetPicture(p,30,30)
```

Notes:

Sets the last exception property.

The method will do nothing on bad bounds.

This method works only for bitmap images.

x and y are zero based.

SetPictureMask(maskpic as picture, x as integer, y as integer)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the pixels from a given Realbasic picture into the mask of the Image Magick Image at the given location.

Example:

```

dim i as QImageQ8MBS // your image
dim p as picture

p=NewPicture(32,32,32)
p.Graphics.ForeColor=rgb(0,255,0)
p.Graphics.FillRect 0,0,32,32

i.SetPictureMask(p,30,30)

```

Notes:

Sets the last exception property.
 The method will do nothing on bad bounds.
 This method works only for bitmap images.
 x and y are zero based.
 You may need to set matte=True after this.

SetPixel(x as integer, y as integer, newPixel as IMColorQ8MBS)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets a pixel value.

Example:

```

dim image as QImageQ8MBS // your image
dim co as IMColorQ8MBS

co=new IMColorQ8MBS
co.blue=65535 // max value
image.SetPixel 50,50,co // Makes Pixel 50/50 blue

```

Notes:

The method will fail silently if the values are out of bounds or the image is not a bitmap image.
 This method works only for bitmap images.
 x and y are zero based.

Shade(gray as boolean, azimuth as double, elevation as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shines a distant light on an image to create a three-dimensional effect.

Notes:

You control the positioning of the light with azimuth and elevation; azimuth is measured in degrees off the x axis and elevation is measured in pixels above the Z axis.

Sets the last exception property.

For more details please check the ImageMagick documentation.

SharpenChannel(channel as integer, radius as double, sigma as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sharpens one or more image channels.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, radius should be larger than sigma. Use a radius of 0 and Sharpen selects a suitable radius for you.

channel: The channel type.

radius: The radius of the Gaussian, in pixels, not counting the center pixel.

sigma: The standard deviation of the Laplacian, in pixels.

Constants for channel:

Sets the last exception property.

For more details please check the ImageMagick documentation.

Shave(x as integer, y as integer, width as integer, height as integer) as IMImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shave shaves pixels from the image edges.

Notes:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel     = & h0001
const CyanChannel     = & h0001
const GreenChannel    = & h0002
const MagentaChannel  = & h0002
const BlueChannel     = & h0004
const YellowChannel   = & h0004
const AlphaChannel    = & h0008
const OpacityChannel  = & h0008
const BlackChannel    = & h0020
const IndexChannel    = & h0020
const AllChannels     = & h7fffffff
```

It allocates the memory necessary for the new Image structure and returns a pointer to the new image.

Returns nil on any error.

Sets the last exception property.

Shear(Xshear as double, Yshear as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method ShearImage creates a new image that is a shear_ image copy of an existing one.

Notes:

Shearing slides one edge of an image along the X or Y axis, creating a parallelogram. An X direction shear slides an edge along the X axis, while a Y direction shear slides an edge along the Y axis. The amount of the shear is controlled by a shear angle. For X direction shears, x_ shear is measured relative to the Y axis, and similarly, for Y direction shears y_ shear is measured relative to the X axis. Empty triangles left over from shearing the image are filled with the color defined by the pixel at location (0,0). ShearImage allocates the memory necessary for the new Image structure and returns a pointer to the new image.

Xshear and Yshear specify the number of degrees to shear the image.

Sets the last exception property.

For more details please check the ImageMagick documentation.

Solarize(factor as double) as boolean

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method SolarizeImage produces a 'solarization' effect seen when exposing a photographic film to light during the development process.

Notes:

factor: An double value that defines the extent of the solarization.

Returns nil on any error.

Sets the last exception property.

Splice(x as integer, y as integer, width as integer, height as integer) as IMImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Splice splices a solid color into the image as defined by the geometry.

Notes:

Returns nil on any error.

Sets the last exception property.

Spread(radius as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This is a special effects method that randomly displaces each pixel in a block defined by the radius parameter.

Notes:

radius: Choose a random pixel in a neighborhood of this extent.
Sets the last exception property.

For more details please check the ImageMagick documentation.

Stegano(watermarkImage as IMImageQ8MBS) as IMImageQ8MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method SteganoImage hides a digital watermark within the image.

Notes:

Returns nil on any error.
Sets the last exception property.

Stereo(otherImage as IMImageQ8MBS) as IMImageQ8MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method StereoImage combines two images and produces a single image that is the composite of a left and right image of a stereo pair.

Notes:

The left image is converted to gray scale and written to the red channel of the stereo image. The right image is converted to gray scale and written to the blue channel of the stereo image. View the composite image with red-blue glasses to create a stereo effect.

left image = self
right image = otherImage parameter

Returns nil on any error.
Sets the last exception property.

Swirl(degrees as double) as QImageQ8MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method SwirlImage creates a new image that is a copy of an existing one with the image pixels "swirl" at a specified angle.

Notes:

degrees: An double value that defines the tightness of the swirling.

Returns nil on any error.

Sets the last exception property.

Thumbnail(width as integer, height as integer) as QImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Changes the size of an image to the given dimensions.

Notes:

Sets the last exception property.

This method was designed by Bob Friesenhahn as a low cost thumbnail generator.

For more details please check the ImageMagick documentation.

TransformImage(CropGeometry as string, ImageGeometry as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransformImage() is a convenience method that behaves like ResizeImage() or CropImage() but accepts scaling and/or cropping information as a region geometry specification. If the operation fails, the original image handle is left as is.

Notes:

This should only be used for single images.

CropGeometry: A crop geometry string. This geometry defines a subregion of the image to crop.

ImageGeometry: An image geometry string. This geometry defines the final size of the image.

Returns true on success.

TransformImages(CropGeometry as string, ImageGeometry as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransformImages() calls TransformImage() on each image of a sequence.

Notes:

TransformImage() is a convenience method that behaves like ResizeImage() or CropImage() but accepts scaling and/or cropping information as a region geometry specification. If the operation fails, the original image handle is left as is.

CropGeometry: A crop geometry string. This geometry defines a subregion of the image to crop.

ImageGeometry: An image geometry string. This geometry defines the final size of the image.

Returns true on success.

TransformRGBImage(Colorspace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method TransformRGBImage converts the reference image from an alternate colorspace.

Notes:

The transformation matrices are not the standard ones: the weights are rescaled to normalized the range of the transformed values to be [0..MaxRGB] .

colorspace: An integer value that indicates the colorspace the image is currently in. On return the image is in the RGB color space.

Returns false on any error and true on success.

constants:

TransposeImage as IMImageQ8MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransposeImage() creates a horizontal mirror image by reflecting the pixels around the central y-axis while rotating them by

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVCOLORSPACE	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

90 degrees.

TransverseImage as IMImageQ8MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransverseImage() creates a vertical mirror image by reflecting the pixels around the central x-axis while rotating them by 270 degrees.

Trim as IMImageQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Trim trims pixels from the image edges.

Notes:

It allocates the memory necessary for the new Image structure and returns a pointer to the new image.

Returns nil on any error.

Sets the last exception property.

UnsharpMaskChannel(channel as integer, radius as double, sigma as double, amount as double, threshold as double) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sharpens one or more image channels.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma). For reasonable results, radius should be larger than sigma. Use a radius of 0 and UnsharpMask selects a suitable radius for you.

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel     = & h0001
const CyanChannel     = & h0001
const GreenChannel    = & h0002
const MagentaChannel  = & h0002
const BlueChannel     = & h0004
const YellowChannel   = & h0004
const AlphaChannel    = & h0008
const OpacityChannel  = & h0008
const BlackChannel    = & h0020
const IndexChannel    = & h0020
const AllChannels     = & h7ffffff
```

Sets the last exception property.

For more details please check the ImageMagick documentation.

Wave(amplitude as double, wavelength as double) as IMImageQ8MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method Wave creates a new image that is a copy of an existing one with the image pixels altered along a sine wave.

Notes:

Parameters are double values that indicates the amplitude and wavelength of the sine wave.

Returns nil on any error.

Sets the last exception property.

WhiteThreshold(threshold as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** WhiteThreshold is like Threshold but forces all pixels above the threshold into white while leaving all pixels below the threshold unchanged.

Notes:

No exceptions are generated.

threshold: Define the threshold value. (ASCII string)

For more details please check the ImageMagick documentation.

WriteImage(info as IMImageInfoQ8MBS) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method WriteImage writes an image to a file as defined by image.filename.

Notes:

You can specify a particular image format by prefixing the file with the image type and a colon (i.e. ps:image) or specify the image type as the filename suffix (i.e. image.ps). The image may be modified to adapt it to the requirements of the image format. For example, DirectClass images must be color-reduced to PseudoClass if the format is GIF.

WriteImage returns True if the image is written. False is returned if there is a memory shortage or if the image file fails to write.

2.1.2 Properties

BackgroundColor as IMColorQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image background color.

Notes: (Read and Write property)

Bias as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

BlurFactor as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blur factor to apply to the image when zooming. Default is 1.0 (no blur).

Notes: (Read and Write property)

BorderColor as IMColorQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image border color.

Notes: (Read and Write property)

Colors as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The desired number of colors.

Notes:

Used by `Quantize()`.

(Read and Write property)

ColorSpace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image pixel interpretation.

Notes:

If the colorspace is RGB the pixels are red, green, blue. If matte is true, then red, green, blue, and index. If it is CMYK, the pixels are cyan, yellow, magenta, black. Otherwise the colorspace is ignored.

constants:

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

(Read and Write property)

Compression as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image compression type.

Notes:

useful constants:

const UndefinedCompression	= 0
const NoCompression	= 1
const BZipCompression	= 2
const FaxCompression	= 3
const Group4Compression	= 4
const JPEGCompression	= 5
const LosslessJPEGCompression	= 6
const LZWCompression	= 7
const RLECompression	= 8
const ZipCompression	= 9

The default is the compression type of the specified image file.
For more details please check the ImageMagick documentation.

(Read and Write property)

Depth as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image depth (8 or 16).

Notes:

QuantumLeap must be defined before a depth of 16 is valid.

(Read and Write property)

Directory as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Tile names from within an image montage.

Notes:

Only valid after calling MontageImages() or reading a MIFF file which contains a directory.

(Read and Write property)

Endian as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The endian setting to use.

Notes:

constants:

UndefinedEndian	0	
LSBEndian	1	(Windows)
MSBEndian	2	(Mac)

e.g. tiff files support different endian settings.

(Read and Write property)

Filename as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The file path/name.
Notes:

The string must be in the encoding of the library and is limited to 4000 bytes.
For more details please check the ImageMagick documentation.
(Read and Write property)

Filter as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Filter to use when resizing image.
Notes:

Constants:

const PointFilter	=1
const BoxFilter	=2
const TriangleFilter	=3
const HermiteFilter	=4
const HanningFilter	=5
const HammingFilter	=6
const BlackmanFilter	=7
const GaussianFilter	=8
const QuadraticFilter	=9
const CubicFilter	=10
const CatromFilter	=11
const MitchellFilter	=12
const LanczosFilter	=13
const BesselFilter	=14
const SincFilter	=15

The reduction filter employed has a sipngicant effect on the time required to resize an image and the resulting quality. The default filter is Lanczos which has been shown to produce high quality results when reducing most images.
(Read and Write property)

Fuzz as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Colors within this distance are considered equal.

Notes:

A number of algorithms search for a target color. By default the color must be exact. Use this to match colors that are close to the target color in RGB space.

(Read and Write property)

Gamma as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gamma level of the image.

Notes:

The same color image displayed on two different workstations may look different due to differences in the display monitor. Use gamma correction to adjust for this color difference.

(Read and Write property)

Geometry as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Preferred size of the image when encoding.

Notes: (Read and Write property)

Gravity as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to an Image structure.

For more details please check the ImageMagick documentation.

(Read and Write property)

Height as integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The height of the image in pixels.

Notes:

For more details please check the ImageMagick documentation.

(Read and Write property)

Interlace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The type of interlacing scheme (default NoInterlace).

Notes:

This option is used to specify the type of interlacing scheme for raw image formats such as RGB or YUV. NoInterlace means do not interlace, LineInterlace uses scanline interlacing, and PlaneInterlace uses plane interlacing. PartitionInterlace is like PlaneInterlace except the different planes are saved to individual files (e.g. image.R, image.G, and image.B). Use LineInterlace or PlaneInterlace to create an interlaced GIF or progressive JPEG image.

constants:

(Read and Write property)

UndefinedInterlace	0	Unset value.
NoInterlace	1	Don't interlace image (RBRGBRGRGBRGRGBRGRB...)
LineInterlace	2	Use scanline interlacing (RRR...GGG...BBB...RRR...GGG...BBB...)
PlaneInterlace	3	Use plane interlacing (RRRRRR...GGGGGG...BBBBBB...)
PartitionInterlace	4	Similar to plane interlacing except that the different planes are saved to individual files (e.g. image.R, image.G, and image.B)

LastError as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last error code reported.

Notes:

If an exception is raised and it is not a warning exception, this exception code is saved in this property.
(Read and Write property)

LastException as IMExceptionQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last exception thrown by the Image Magick library.

Notes:

You should check this value after every call to the library, process the error and set the property to nil.

For more details please check the ImageMagick documentation.
(Read and Write property)

Magick as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image encoding format (e.g. "GIF").

Notes: (Read and Write property)

Matte as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether an alpha channel is used/present.

Notes:

Set to true to enable masks.
(Read and Write property)

MatteColor as IMColorQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image matte (transparent) color.

Notes: (Read and Write property)

Montage as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Tile size and offset within an image montage. Only valid for montage images.

Notes: (Read and Write property)

Offset as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Number of initial bytes to skip over when reading raw image.

Notes: (Read and Write property)

Orientation as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The image orientation.

Notes:

constants:

```

const UndefinedOrientation = 0
const TopLeftOrientation  = 1
const TopRightOrientation = 2
const BottomRightOrientation = 3
const BottomLeftOrientation = 4
const LeftTopOrientation   = 5
const RightTopOrientation  = 6
const RightBottomOrientation = 7
const LeftBottomOrientation = 8

```

For more details please check the ImageMagick documentation.
(Read and Write property)

Quality as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** JPEG/MIFF/PNG compression level.

Example:

```
dim im as ImageMagickQ8MBS // global
```

```
Function TestJPEG(f as folderitem) As picture
```

```
// Reads an image, compresses in memory to JPEG, decompresses using JPEGlib and returns the image
```

```
// if quality setting works, you see it in the result.
```

```
// no error checking included!
```

```
// needs: im as ImageMagickQ8MBS ready initialized
```

```
dim image as IMImageQ8MBS
```

```
dim imageinfo as IMImageInfoQ8MBS
```

```
dim s,blob as string
```

```
dim p as Picture
```

```
dim i as integer
```

```
if f = nil then
```

```
Return nil
```

```
end if
```

```
imageinfo = im.NewImageInfo
```

```
# if TargetWin32 then //do not use shellpath, if spaces, IM doesn't like escaped paths
```

```
imageinfo.Filename = f.AbsolutePath
```

```
# else
```

```
imageinfo.Filename = f.UnixpathMBS
```

```

# endif

//this should read any image IM understands
image = im.ReadImage(imageinfo)
//check for error
if im.lastexception <>nil and im.LastException.Severity >= 400 then
s = "LastError: " + Format(im.LastError, "-0") + " - Severity: " + str(im.LastException.Severity) + EndOfLine + im.LastException.Message
MsgBox s
Return nil
elseif image = nil then
MsgBox "image=nil"
Return nil
end if

// Now lets convert to jpeg
imageinfo.Filename = "image.jpg"
imageinfo.Quality = 10 // 100 is max
blob = image.ImageToBlob(imageinfo)

// It may fail
if blob.lenb = 0 then
Return nil
end if
p = JPEGStringToPictureMBS(blob,true)

image.DestroyImage
imageinfo.DestroyImageInfo

Return p
Exception
Return nil
End Function

```

Notes:

Default value is 75.
(Read and Write property)

Release as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** If true, the destructor will release the handle.

Notes: (Read and Write property)

RenderingIntent as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The rendering intent to use.

Notes:

constants:

UndefinedIntent	0
SaturationIntent	1
PerceptualIntent	2
AbsoluteIntent	3
RelativeIntent	4

(Read and Write property)

ResolutionUnits as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Units of image resolution.

Notes:

constants:

UndefinedResolution	0	Unset value.
PixelsPerInchResolution	1	Density specifications are specified in units of pixels per inch (english units).
PixelsPerCentimeterResolution	2	Density specifications are specified in units of pixels per centimeter (metric units).

(Read and Write property)

ResolutionX as double

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The horizontal resolution of the image.

Notes:

The unit for resolution must be specified.
 For more details please check the ImageMagick documentation.
 (Read and Write property)

ResolutionY as double

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The vertical resolution of the image.

Notes:

The unit for resolution must be specified.
 For more details please check the ImageMagick documentation.
 (Read and Write property)

Scene as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

StorageClass as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image storage class.

Notes:

If DirectClass then the image packets contain valid RGB or CMYK colors. If PseudoClass then the image has a colormap referenced by pixel's index member.

constants:

UndefinedClass	0	Unset value.
DirectClass	1	Image is composed of pixels which represent literal color values.
PseudoClass	2	Image is composed of pixels which specify an index in a color palette.

(Read and Write property)

Taint as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Set to True if the image pixels have been modified.

Notes: (Read and Write property)

Width as integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The width of the image in pixels.

Notes:

For more details please check the ImageMagick documentation.

(Read and Write property)

2.1.3 Constants

kBackgroundDispose = 2

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kCoalesceLayer = 1

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompareAnyLayer = 2

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompareClearLayer = 3

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompareOverlayLayer = 4

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompositeLayer = & h0000000C

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kDisposeLayer = 5

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kFlattenLayer = & h0000000E

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kMergeLayer = & h0000000D

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kMosaicLayer = & h0000000F

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kNoneDispose = 1

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kOptimizeImageLayer = 7

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kOptimizeLayer = 6

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kOptimizePlusLayer = 8

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kOptimizeTransLayer = 9

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kPreviousDispose = 3

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kRemoveDupsLayer = & h0000000A

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kRemoveZeroLayer = & h0000000B

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kUndefinedDispose = 0

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kUndefinedLayer = 0

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kUnrecognizedDispose = 0

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

2.2 class IMMagickInfoQ32MBS

class IMMagickInfoQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for information about a file import/export format Image Magick can handle.

Notes: For more details please check the ImageMagick documentation.

2.2.1 Methods

Close

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.

Notes:

There is no need to call this method except you want to free all resources used by this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

2.2.2 Properties

Adjoin as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if this file format supports multi-frame images.

Notes:

For more details please check the ImageMagick documentation.

Returns false for an invalid MagickInfo (handle=0).

(Read only property)

BlobSupport as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the encoder and decoder for this format supports operating on arbitrary BLOBs (rather than only disk files).

Notes:

As currently disc read/write does not work with the 5.1 plugins, we really need that to use the classes. Returns false for an invalid MagickInfo (handle=0).

For more details please check the ImageMagick documentation.
(Read only property)

Description as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Long form image format description (e.g. "CompuServe graphics interchange format").

Notes:

For more details please check the ImageMagick documentation.
Returns "" for an invalid MagickInfo (handle=0).
(Read only property)

EndianSupport as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether endian support is available.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to a MagickInfo structure.

For more details please check the ImageMagick documentation.

(Read and Write property)

ModuleName as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Name of module (e.g. "GIF") which registered this format.

Notes:

Value is "" if format is not registered by a module.

For more details please check the ImageMagick documentation.

Returns "" for an invalid MagickInfo (handle=0).

(Read only property)

Name as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Magick string (e.g. "GIF") which identifies this format.

Notes:

For more details please check the ImageMagick documentation.

Returns "" for an invalid MagickInfo (handle=0).

(Read only property)

Note as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Additional notes for this format.

Notes:

e.g. compilation parameters or copyright notices.
Returns "" for an invalid MagickInfo (handle=0).

For more details please check the ImageMagick documentation.
(Read only property)

Raw as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if Image format does not contain size (must be specified in ImageInfo).

Notes:

Returns false for an invalid MagickInfo (handle=0).
For more details please check the ImageMagick documentation.
(Read only property)

SeekableStream as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if the magick supports a seekable stream.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

Stealth as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unknown.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

ThreadSupport as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if threading is supported.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

Version as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Version string.

Notes:

For more details please check the ImageMagick documentation.
Returns "" for an invalid MagickInfo (handle=0).
(Read only property)

2.3 class IMMagickInfoQ16MBS**class IMMagickInfoQ16MBS**

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for information about a file import/export format Image Magick can handle.

Notes: For more details please check the ImageMagick documentation.

2.3.1 Methods**Close**

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.

Notes:

There is no need to call this method except you want to free all resources used by this object now without

waiting for Realbasic to do it for you.
(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

2.3.2 Properties

Adjoin as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if this file format supports multi-frame images.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

BlobSupport as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the encoder and decoder for this format supports operating on arbitrary BLOBs (rather than only disk files).

Notes:

As currently disc read/write does not work with the 5.1 plugins, we really need that to use the classes.
Returns false for an invalid MagickInfo (handle=0).

For more details please check the ImageMagick documentation.
(Read only property)

Description as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Long form image format description (e.g. "CompuServe graphics interchange format").

Notes:

For more details please check the ImageMagick documentation.
Returns "" for an invalid MagickInfo (handle=0).

(Read only property)

EndianSupport as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether endian support is available.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to a MagickInfo structure.
For more details please check the ImageMagick documentation.
(Read and Write property)

ModuleName as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Name of module (e.g. "GIF") which registered this format.

Notes:

Value is "" if format is not registered by a module.
For more details please check the ImageMagick documentation.
Returns "" for an invalid MagickInfo (handle=0).
(Read only property)

Name as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Magick string (e.g. "GIF") which identifies this format.

Notes:

For more details please check the ImageMagick documentation.

Returns "" for an invalid MagickInfo (handle=0).

(Read only property)

Note as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Additional notes for this format.

Notes:

e.g. compilation parameters or copyright notices.

Returns "" for an invalid MagickInfo (handle=0).

For more details please check the ImageMagick documentation.

(Read only property)

Raw as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if Image format does not contain size (must be specified in ImageInfo).

Notes:

Returns false for an invalid MagickInfo (handle=0).

For more details please check the ImageMagick documentation.

(Read only property)

SeekableStream as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if the magick supports a seekable stream.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

Stealth as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unknown.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

ThreadSupport as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if threading is supported.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

Version as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Version string.

Notes:

For more details please check the ImageMagick documentation.
Returns "" for an invalid MagickInfo (handle=0).
(Read only property)

2.4 class IMMagickInfoListQ8MBS

class IMMagickInfoListQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class with the list of the image formats supported in Image Magick.

Notes: For more details please check the ImageMagick documentation.

2.4.1 Methods

Item(index as integer) as IMMagickInfoQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The items inside this list.

Notes:

Index goes from 0 to count-1.

Returns nil on invalid index.

2.4.2 Properties

Count as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items.

Notes:

Index goes from 0 to count-1.

(Read only property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to a MagickInfo list.
For more details please check the ImageMagick documentation.
(Read only property)

2.5 class IMMagickInfoListQ16MBS

class IMMagickInfoListQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class with the list of the image formats supported in Image Magick.

Notes: For more details please check the ImageMagick documentation.

2.5.1 Methods

Item(index as integer) as IMMagickInfoQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The items inside this list.

Notes:

Index goes from 0 to count-1.
Returns nil on invalid index.

2.5.2 Properties

Count as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items.

Notes:

Index goes from 0 to count-1.
(Read only property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to a MagickInfo list.

For more details please check the ImageMagick documentation.

(Read only property)

2.6 class IMMagickInfoListQ32MBS

class IMMagickInfoListQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class with the list of the image formats supported in Image Magick.

Notes: For more details please check the ImageMagick documentation.

2.6.1 Methods

Item(index as integer) as IMMagickInfoQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The items inside this list.

Notes:

Index goes from 0 to count-1.

Returns nil on invalid index.

2.6.2 Properties

Count as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of items.

Notes:

Index goes from 0 to count-1.
(Read only property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to a MagickInfo list.
For more details please check the ImageMagick documentation.
(Read only property)

2.7 class IMMagickPixelPacketQ8MBS

class IMMagickPixelPacketQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class to describe a picture background.

Notes: Needed for IImageQ8MBS.NewImage function.

2.7.1 Methods

HandleMemory as memoryblock

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The content of the whole ImageInfo structure copied into a memoryblock.

Notes: Returns nil on any error.

2.7.2 Properties

Blue as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The blue color value.

Notes: (Read and Write property)

ColorSpace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image pixel interpretation.

Notes:

If the colorspace is RGB the pixels are red, green, blue. If matte is true, then red, green, blue, and index. If it is CMYK, the pixels are cyan, yellow, magenta, black. Otherwise the colorspace is ignored.

constants:

(Read and Write property)

Depth as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image depth (8 or 16).

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

Notes: (Read and Write property)

Fuzz as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Colors within this distance are considered equal.

Notes:

A number of algorithms search for a target color. By default the color must be exact. Use this to match colors that are close to the target color in RGB space.

(Read and Write property)

Green as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The green color value.

Notes: (Read and Write property)

Handle as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to an MagickPixelPacket structure.
For more details please check the ImageMagick documentation.
(Read and Write property)

Index as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The index color value.

Notes:

Only for indexed color spaces.
(Read and Write property)

Matte as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether an alpha channel is used/present.

Notes:

Set to true to enable masks.
(Read and Write property)

Opacity as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The opacity part of the color value.

Notes: (Read and Write property)

Red as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The red color value.
Notes: (Read and Write property)

2.8 class IMMagickPixelPacketQ32MBS

class IMMagickPixelPacketQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class to describe a picture background.
Notes: Needed for IMImageQ32MBS.NewImage function.

2.8.1 Methods

HandleMemory as memoryblock

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The content of the whole ImageInfo structure copied into a memoryblock.
Notes: Returns nil on any error.

2.8.2 Properties

Blue as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The blue color value.
Notes: (Read and Write property)

ColorSpace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image pixel interpretation.

Notes:

If the colorspace is RGB the pixels are red, green, blue. If matte is true, then red, green, blue, and index. If it is CMYK, the pixels are cyan, yellow, magenta, black. Otherwise the colorspace is ignored.

constants:

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

(Read and Write property)

Depth as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image depth (8 or 16).

Notes: (Read and Write property)

Fuzz as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Colors within this distance are considered equal.

Notes:

A number of algorithms search for a target color. By default the color must be exact. Use this to match colors that are close to the target color in RGB space.

(Read and Write property)

Green as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The green color value.

Notes: (Read and Write property)

Handle as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to an MagickPixelPacket structure.

For more details please check the ImageMagick documentation.

(Read and Write property)

Index as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The index color value.

Notes:

Only for indexed color spaces.

(Read and Write property)

Matte as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether an alpha channel is used/present.

Notes:

Set to true to enable masks.
(Read and Write property)

Opacity as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The opacity part of the color value.

Notes: (Read and Write property)

Red as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The red color value.

Notes: (Read and Write property)

2.9 class IMMissingFunctionExceptionQ8MBS

class IMMissingFunctionExceptionQ8MBS

Plugin Version: 5.2 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for an exception in Image Magick.

Notes:

This exception is raised on every IM function if the library function behind is not available.
(this can be a plugin bug or a bad compiled library or simply a too old library.)
Subclass of the RuntimeException class.

2.10 class IMMissingFunctionExceptionQ32MBS

class IMMissingFunctionExceptionQ32MBS

Plugin Version: 5.2 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for an exception in Image Magick.

Notes:

This exception is raised on every IM function if the library function behind is not available. (this can be a plugin bug or a bad compiled library or simply a too old library.)
Subclass of the RuntimeException class.

2.11 class IMMissingFunctionExceptionQ16MBS

class IMMissingFunctionExceptionQ16MBS

Plugin Version: 5.2 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for an exception in Image Magick.

Notes:

This exception is raised on every IM function if the library function behind is not available. (this can be a plugin bug or a bad compiled library or simply a too old library.)
Subclass of the RuntimeException class.

2.12 class IMMagickInfoQ8MBS

class IMMagickInfoQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for information about a file import/export format Image Magick can handle.

Notes: For more details please check the ImageMagick documentation.

2.12.1 Methods

Close

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.
Notes:

There is no need to call this method except you want to free all resources used by this object now without waiting for Realbasic to do it for you.
(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

2.12.2 Properties

Adjoin as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if this file format supports multi-frame images.
Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

BlobSupport as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if the encoder and decoder for this format supports operating on arbitrary BLOBs (rather than only disk files).
Notes:

As currently disc read/write does not work with the 5.1 plugins, we really need that to use the classes.
Returns false for an invalid MagickInfo (handle=0).

For more details please check the ImageMagick documentation.
(Read only property)

Description as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Long form image format description (e.g. "CompuServe graphics interchange format").

Notes:

For more details please check the ImageMagick documentation.

Returns "" for an invalid MagickInfo (handle=0).

(Read only property)

EndianSupport as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether endian support is available.

Notes:

For more details please check the ImageMagick documentation.

Returns false for an invalid MagickInfo (handle=0).

(Read only property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to a MagickInfo structure.

For more details please check the ImageMagick documentation.

(Read and Write property)

ModuleName as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Name of module (e.g. "GIF") which registered this format.

Notes:

Value is "" if format is not registered by a module.

For more details please check the ImageMagick documentation.
Returns "" for an invalid MagickInfo (handle=0).
(Read only property)

Name as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Magick string (e.g. "GIF") which identifies this format.
Notes:

For more details please check the ImageMagick documentation.
Returns "" for an invalid MagickInfo (handle=0).
(Read only property)

Note as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Additional notes for this format.
Notes:

e.g. compilation parameters or copyright notices.
Returns "" for an invalid MagickInfo (handle=0).

For more details please check the ImageMagick documentation.
(Read only property)

Raw as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if Image format does not contain size (must be specified in ImageInfo).
Notes:

Returns false for an invalid MagickInfo (handle=0).
For more details please check the ImageMagick documentation.
(Read only property)

SeekableStream as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns True if the magick supports a seekable stream.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

Stealth as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unknown.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

ThreadSupport as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if threading is supported.

Notes:

For more details please check the ImageMagick documentation.
Returns false for an invalid MagickInfo (handle=0).
(Read only property)

Version as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Version string.

Notes:

For more details please check the ImageMagick documentation.
Returns "" for an invalid MagickInfo (handle=0).

(Read only property)

2.13 class IMMagickPixelPacketQ16MBS

class IMMagickPixelPacketQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class to describe a picture background.

Notes: Needed for IMImageQ16MBS.NewImage function.

2.13.1 Methods

HandleMemory as memoryblock

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The content of the whole ImageInfo structure copied into a memoryblock.

Notes: Returns nil on any error.

2.13.2 Properties

Blue as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The blue color value.

Notes: (Read and Write property)

ColorSpace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image pixel interpretation.

Notes:

If the colorspace is RGB the pixels are red, green, blue. If matte is true, then red, green, blue, and index. If it is CMYK, the pixels are cyan, yellow, magenta, black. Otherwise the colorspace is ignored.

constants:

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVCColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

(Read and Write property)

Depth as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image depth (8 or 16).

Notes: (Read and Write property)

Fuzz as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Colors within this distance are considered equal.

Notes:

A number of algorithms search for a target color. By default the color must be exact. Use this to match colors that are close to the target color in RGB space.

(Read and Write property)

Green as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The green color value.

Notes: (Read and Write property)

Handle as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to an MagickPixelPacket structure.
For more details please check the ImageMagick documentation.
(Read and Write property)

Index as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The index color value.

Notes:

Only for indexed color spaces.
(Read and Write property)

Matte as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether an alpha channel is used/present.

Notes:

Set to true to enable masks.
(Read and Write property)

Opacity as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The opacity part of the color value.

Notes: (Read and Write property)

Red as Single

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The red color value.

Notes: (Read and Write property)

2.14 class IMImageAttributeQ8MBS

class IMImageAttributeQ8MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for an image attribute.

Notes:

Used only for reading the attributes.

Do not keep references over long times as memory of key/value pairs may be released.

2.14.1 Properties

Compression as Boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether compression is used.

Notes: (Read only property)

Key as String

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The key of this attribute.

Notes:

String is in binary text encoding.
(Read only property)

Value as String

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of this attribute.

Notes:

String is in binary text encoding.
(Read only property)

2.15 class IMImageInfoQ16MBS

class IMImageInfoQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for information about an image.

Notes: For more details please check the ImageMagick documentation.

2.15.1 Methods

Clone as IMImageInfoQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clones the Image-Info object.

Notes: For more details please check the ImageMagick documentation.

Close

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.

Notes:

There is no need to call this method except you want to free all resources used by this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

DestroyImageInfo

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Destroys the image info and sets the handle to 0.

Notes:

For more details please check the ImageMagick documentation.

The destructor will call this for you if release=true.

HandleMemory as memoryblock

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The content of the whole ImageInfo structure copied into a memoryblock.

Notes: Returns nil on any error.

2.15.2 Properties**Adjoin as Boolean**

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Join images into a single multi-image file.

Notes:

For more details please check the ImageMagick documentation.

(Read and Write property)

Affirm as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unknown.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Antialias as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Control antialiasing of rendered Postscript and Postscript or TrueType fonts.

Notes:

Enabled by default.
For more details please check the ImageMagick documentation.
(Read and Write property)

Authenticate as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

BackgroundColor as IMColorQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image background color.

Notes: (Read and Write property)

BorderColor as IMColorQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image border color.

Notes: (Read and Write property)

Channel as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The channel to use.
Notes:

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel      = & h0001
const CyanChannel      = & h0001
const GreenChannel     = & h0002
const MagentaChannel  = & h0002
const BlueChannel      = & h0004
const YellowChannel    = & h0004
const AlphaChannel     = & h0008
const OpacityChannel  = & h0008
const BlackChannel     = & h0020
const IndexChannel     = & h0020
const AllChannels     = & h7ffffff
```

(Read and Write property)

Colors as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

ColorSpace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image pixel interpretation.

Example:

```
dim im as ImageMagickQ16MBS // global
```

```
Function IMPictureToString(p as picture, magick as string, quality as integer) As string
```

```
dim image as new IMImageQ16MBS
```

```
dim imageinfo as IMImageInfoQ16MBS
```

```

dim s,data as string
dim impp as new IMMagickPixelPacketQ16MBS

// empty string for nil picture
if p = nil then
Return ""
end if

// create a new picture info

imageinfo = im.NewImageInfo
imageinfo.ColorSpace=1
// only color space is needed. 1 for RGB.

// background color of image
impp.red = 0
impp.Green = 0
impp.Blue = 0

// creates a new image object
if not image.NewImage(imageinfo,p.Width,p.Height,impp) then
Return ""
end if

// copy RB picture into IM Image at position 0/0
image.ColorSpace = 1
image.SetPicture(p,0,0)

// set compression data
imageinfo.Magick = magick
imageinfo.Quality = quality

// and rendering intent: 2=PerceptualIntent
image.RenderingIntent = 2

// create image data
data = image.ImageToBlob(imageinfo)

// release memory
image.DestroyImage
imageinfo.DestroyImageInfo

// return result
Return data

Exception
// in case of an exception return nothing
Return ""

```

End Function

Notes:

If the colorspace is RGB the pixels are red, green, blue. If matte is true, then red, green, blue, and index. If it is CMYK, the pixels are cyan, yellow, magenta, black. Otherwise the colorspace is ignored.

constants:

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVCColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

(Read and Write property)

Compression as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image compression type.

Notes:

useful constants:

```
const UndefinedCompression    = 0
const NoCompression          = 1
const BZipCompression         = 2
const FaxCompression         = 3
const Group4Compression      = 4
const JPEGCompression        = 5
const LosslessJPEGCompression = 6
const LZWCompression         = 7
const RLECompression         = 8
const ZipCompression         = 9
```

The default is the compression type of the specified image file.
For more details please check the ImageMagick documentation.
(Read and Write property)

Density as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Vertical and horizontal resolution in pixels of the image.

Notes:

This option specifies an image density when decoding a Postscript or Portable Document page.
(Read and Write property)

Depth as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image depth (8 or 16).

Notes:

QuantumLeap must be defined before a depth of 16 is valid.
(Read and Write property)

Dither as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Endian as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The endian setting to use.

Notes:

constants:

UndefinedEndian	0	
LSBEndian	1	(Windows)
MSBEndian	2	(Mac)

e.g. tiff files support different endian settings.
(Read and Write property)

Extract as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Filename as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The file path/name.

Notes:

The string must be in the encoding of the library and is limited to 4000 bytes.

For more details please check the ImageMagick documentation.

(Read and Write property)

Font as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Text rendering font.

Notes:

If the font is a fully qualified X server font name, the font is obtained from an X server. To use a TrueType font, precede the TrueType filename with an @. Otherwise, specify a Postscript font name (e.g. "helvetica"). (Read and Write property)

Group as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to a ImageInfo structure.

For more details please check the ImageMagick documentation.

(Read and Write property)

HeaderOnly as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if only the header was read from the image data.

Notes: (Read and Write property)

Interlace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The type of interlacing scheme (default NoInterlace).

Notes:

This option is used to specify the type of interlacing scheme for raw image formats such as RGB or YUV. NoInterlace means do not interlace, LineInterlace uses scanline interlacing, and PlaneInterlace uses plane interlacing. PartitionInterlace is like PlaneInterlace except the different planes are saved to individual files

(e.g. `image.R`, `image.G`, and `image.B`). Use `LineInterlace` or `PlaneInterlace` to create an interlaced GIF or progressive JPEG image.

constants:

<code>UndefinedInterlace</code>	0	Unset value.
<code>NoInterlace</code>	1	Don't interlace image (RGBRGBRGBRGBRGB...)
<code>LineInterlace</code>	2	Use scanline interlacing (RRR...GGG...BBB...RRR...GGG...BBB...)
<code>PlaneInterlace</code>	3	Use plane interlacing (RRRRRR...GGGGG...BBBBB...)
<code>PartitionInterlace</code>	4	Similar to plane interlacing except that the different planes are saved to individual files (e.g. <code>image.R</code> , <code>image.G</code> , and <code>image.B</code>)

(Read and Write property)

Magick as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image encoding format (e.g. "GIF").

Example:

```
dim imageinfo as IMImageInfoQ16MBS
dim blob as string
dim image as IMImageQ16MBS
```

```
// Now lets convert to tiff
imageinfo.Filename = "image"
imageinfo.Magick="JPEG"
imageinfo.Quality = 10 //since we are displaying, lets use highest quality, lowest compression
blob = image.ImageToBlob(imageinfo)
```

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

MatteColor as IMColorQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image matte (transparent) color.

Notes: (Read and Write property)

Monochrome as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Transform the image to black and white.

Notes: (Read and Write property)

Orientation as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The image orientation.

Notes:

constants:

```
const UndefinedOrientation    = 0
const TopLeftOrientation     = 1
const TopRightOrientation    = 2
const BottomRightOrientation = 3
const BottomLeftOrientation  = 4
const LeftTopOrientation     = 5
const RightTopOrientation    = 6
const RightBottomOrientation = 7
const LeftBottomOrientation  = 8
```

For more details please check the ImageMagick documentation.
(Read and Write property)

Page as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Equivalent size of Postscript page.

Notes: (Read and Write property)

PointSize as Double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Text rendering font point size.

Notes: (Read and Write property)

Preview as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image manipulation preview option.

Notes:

Used by 'display'.

constants:

(Read and Write property)

Quality as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** JPEG/MIFF/PNG compression level.

Notes:

Default value is 75.

(Read and Write property)

Release as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** If true, the destructor will release the handle.

UndefinedPreview	0
RotatePreview	1
ShearPreview	2
RollPreview	3
HuePreview	4
SaturationPreview	5
BrightnessPreview	6
GammaPreview	7
SpiffPreview	8
DullPreview	9
GrayscalePreview	10
QuantizePreview	11
DespecklePreview	12
ReduceNoisePreview	13
AddNoisePreview	14
SharpenPreview	15
BlurPreview	16
ThresholdPreview	17
EdgeDetectPreview	18
SpreadPreview	19
SolarizePreview	20
ShadePreview	21
RaisePreview	22
SegmentPreview	23
SwirlPreview	24
ImplodePreview	25
WavePreview	26
OilPaintPreview	27
CharcoalDrawingPreview	28
JPEGPreview	29

Notes: (Read and Write property)

ResolutionUnits as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Units of image resolution.

Notes:

constants:

(Read and Write property)

UndefinedResolution	0	Unset value.
PixelsPerInchResolution	1	Density specifications are specified in units of pixels per inch (english units).
PixelsPerCentimeterResolution	2	Density specifications are specified in units of pixels per centimeter (metric units).

SamplingFactor as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Scene as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

SceneCount as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Scenes as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

ServerName as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** X11 display to display to.

Notes:

obtain fonts from, or to capture image from.
(Read and Write property)

Size as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Width and height of a raw image (an image which does not support width and height information).

Notes:

Size may also be used to affect the image size read from a multi-resolution format (e.g. Photo CD, JBIG, or JPEG).

(Read and Write property)

Temporary as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unknown.

Notes:

For more details please check the ImageMagick documentation.

(Read and Write property)

Texture as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image filename to use as background texture.

Notes: (Read and Write property)

Type as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Image type.

Notes:

constants:

UndefinedType	0
BilevelType	1
GrayscaleType	2
GrayscaleMatteType	3
PaletteType	4
PaletteMatteType	5
TrueColorType	6
TrueColorMatteType	7
ColorSeparationType	8
ColorSeparationMatteType	9
OptimizeType	10

(Read and Write property)

Verbose as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Print detailed information about the image if True.

Notes: (Read and Write property)

View as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** FlashPix viewing parameters.

Notes: (Read and Write property)

2.16 class IMImageAffineMatrixQ32MBS

class IMImageAffineMatrixQ32MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for an Image Magick affine transformation matrix.

2.16.1 Methods

Constructor

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates an identity matrix.

2.16.2 Properties

RX as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The rotate x value.
Notes: (Read and Write property)

RY as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The rotate y value.
Notes: (Read and Write property)

SX as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The scale x value.
Notes: (Read and Write property)

SY as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The scale y value.
Notes: (Read and Write property)

TX as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The translate y value.

Notes: (Read and Write property)

TY as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The translate y value.

Notes: (Read and Write property)

2.17 class `IMImageAffineMatrixQ16MBS`

class `IMImageAffineMatrixQ16MBS`

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for an Image Magick affine transformation matrix.

2.17.1 Methods

Constructor

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates an identity matrix.

2.17.2 Properties

RX as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The rotate x value.
Notes: (Read and Write property)

RY as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The rotate y value.
Notes: (Read and Write property)

SX as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The scale x value.
Notes: (Read and Write property)

SY as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The scale y value.
Notes: (Read and Write property)

TX as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The translate y value.
Notes: (Read and Write property)

TY as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The translate y value.

Notes: (Read and Write property)

2.18 class IMImageAttributeQ16MBS

class IMImageAttributeQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for an image attribute.

Notes:

Used only for reading the attributes.

Do not keep references over long times as memory of key/value pairs may be released.

2.18.1 Properties

Compression as Boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether compression is used.

Notes: (Read only property)

Key as String

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The key of this attribute.

Notes:

String is in binary text encoding.

(Read only property)

Value as String

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of this attribute.

Notes:

String is in binary text encoding.
(Read only property)

2.19 class IMImageAttributeQ32MBS

class IMImageAttributeQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for an image attribute.

Notes:

Used only for reading the attributes.
Do not keep references over long times as memory of key/value pairs may be released.

2.19.1 Properties

Compression as Boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether compression is used.

Notes: (Read only property)

Key as String

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The key of this attribute.

Notes:

String is in binary text encoding.

(Read only property)

Value as String

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The value of this attribute.

Notes:

String is in binary text encoding.

(Read only property)

2.20 class IMImageAffineMatrixQ8MBS

class IMImageAffineMatrixQ8MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for an Image Magick affine transformation matrix.

2.20.1 Methods

Constructor

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates an identity matrix.

2.20.2 Properties

RX as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The rotate x value.
Notes: (Read and Write property)

RY as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The rotate y value.
Notes: (Read and Write property)

SX as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The scale x value.
Notes: (Read and Write property)

SY as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The scale y value.
Notes: (Read and Write property)

TX as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The translate x value.
Notes: (Read and Write property)

TY as Double

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The translate y value.
Notes: (Read and Write property)

2.21 class IMImageInfoQ32MBS

class IMImageInfoQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for information about an image.

Notes: For more details please check the ImageMagick documentation.

2.21.1 Methods

Clone as IMImageInfoQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clones the Image-Info object.

Notes: For more details please check the ImageMagick documentation.

Close

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.
Notes:

There is no need to call this method except you want to free all resources used by this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

DestroyImageInfo

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Destroys the image info and sets the handle to 0.

Notes:

For more details please check the ImageMagick documentation.

The destructor will call this for you if release=true.

HandleMemory as memoryblock

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The content of the whole ImageInfo structure copied into a memoryblock.

Notes: Returns nil on any error.

2.21.2 Properties

Adjoin as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Join images into a single multi-image file.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Affirm as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unknown.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Antialias as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Control antialiasing of rendered Postscript and Postscript or TrueType fonts.

Notes:

Enabled by default.

For more details please check the ImageMagick documentation.
(Read and Write property)

Authenticate as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

BackgroundColor as IMColorQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image background color.

Notes: (Read and Write property)

BorderColor as IMColorQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image border color.

Notes: (Read and Write property)

Channel as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The channel to use.

Notes:

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel     = & h0001
const CyanChannel     = & h0001
const GreenChannel    = & h0002
const MagentaChannel  = & h0002
const BlueChannel     = & h0004
const YellowChannel   = & h0004
const AlphaChannel    = & h0008
const OpacityChannel  = & h0008
const BlackChannel    = & h0020
const IndexChannel    = & h0020
const AllChannels     = & h7fffffff
```

(Read and Write property)

Colors as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

ColorSpace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image pixel interpretation.

Example:

```
dim im as ImageMagickQ32MBS // global
```

```
Function IMPictureToString(p as picture, magick as string, quality as integer) As string
```

```
dim image as new IMImageQ32MBS
```

```
dim imageinfo as IMImageInfoQ32MBS
```

```
dim s,data as string
```

```
dim impp as new IMMagickPixelPacketQ32MBS
```

```
// empty string for nil picture
```

```
if p = nil then
```

```
Return ""
```

```
end if
```

```
// create a new picture info
```

```
imageinfo = im.NewImageInfo
```

```
imageinfo.ColorSpace=1
```

```
// only color space is needed. 1 for RGB.
```

```
// background color of image
```

```
impp.red = 0
```

```
impp.Green = 0
```

```
impp.Blue = 0
```

```
// creates a new image object
```

```
if not image.NewImage(imageinfo,p.Width,p.Height,impp) then
```

```
Return ""
```

```
end if
```

```
// copy RB picture into IM Image at position 0/0
image.ColorSpace = 1
image.SetPicture(p,0,0)

// set compression data
imageinfo.Magick = magick
imageinfo.Quality = quality

// and rendering intent: 2=PerceptualIntent
image.RenderingIntent = 2

// create image data
data = image.ImageToBlob(imageinfo)

// release memory
image.DestroyImage
imageinfo.DestroyImageInfo

// return result
Return data

Exception
// in case of an exception return nothing
Return ""

End Function
```

Notes:

If the colorspace is RGB the pixels are red, green, blue. If matte is true, then red, green, blue, and index. If it is CMYK, the pixels are cyan, yellow, magenta, black. Otherwise the colorspace is ignored.

constants:

(Read and Write property)

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVCColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

Compression as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image compression type.

Notes:

useful constants:

const UndefinedCompression	= 0
const NoCompression	= 1
const BZipCompression	= 2
const FaxCompression	= 3
const Group4Compression	= 4
const JPEGCompression	= 5
const LosslessJPEGCompression	= 6
const LZWCompression	= 7
const RLECompression	= 8
const ZipCompression	= 9

The default is the compression type of the specified image file.
For more details please check the ImageMagick documentation.
(Read and Write property)

Density as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Vertical and horizontal resolution in pixels of the image.

Notes:

This option specifies an image density when decoding a Postscript or Portable Document page.
(Read and Write property)

Depth as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image depth (8 or 16).

Notes:

QuantumLeap must be defined before a depth of 16 is valid.
(Read and Write property)

Dither as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Endian as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The endian setting to use.

Notes:

constants:

UndefinedEndian	0	
LSBEndian	1	(Windows)
MSBEndian	2	(Mac)

e.g. tiff files support different endian settings.
(Read and Write property)

Extract as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.
Notes: (Read and Write property)

Filename as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The file path/name.
Notes:

The string must be in the encoding of the library and is limited to 4000 bytes.
For more details please check the ImageMagick documentation.
(Read and Write property)

Font as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Text rendering font.
Notes:

If the font is a fully qualified X server font name, the font is obtained from an X server. To use a TrueType font, precede the TrueType filename with an @. Otherwise, specify a Postscript font name (e.g. "helvetica").
(Read and Write property)

Group as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.
Notes: (Read and Write property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to a ImageInfo structure.

For more details please check the ImageMagick documentation.

(Read and Write property)

HeaderOnly as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if only the header was read from the image data.

Notes: (Read and Write property)

Interlace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The type of interlacing scheme (default NoInterlace).

Notes:

This option is used to specify the type of interlacing scheme for raw image formats such as RGB or YUV. NoInterlace means do not interlace, LineInterlace uses scanline interlacing, and PlaneInterlace uses plane interlacing. PartitionInterlace is like PlaneInterlace except the different planes are saved to individual files (e.g. image.R, image.G, and image.B). Use LineInterlace or PlaneInterlace to create an interlaced GIF or progressive JPEG image.

constants:

UndefinedInterlace	0	Unset value.
NoInterlace	1	Don't interlace image (RBRGRBRGRBRGRBRGRB...)
LineInterlace	2	Use scanline interlacing (RRR...GGG...BBB...RRR...GGG...BBB...)
PlaneInterlace	3	Use plane interlacing (RRRRRR...GGGGG...BBBBB...)
PartitionInterlace	4	Similar to plane interlacing except that the different planes are saved to individual files (e.g. image.R, image.G, and image.B)

(Read and Write property)

Magick as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image encoding format (e.g. "GIF").

Example:

```
dim imageinfo as IMImageInfoQ32MBS
dim blob as string
dim image as IMImageQ32MBS
```

```
// Now lets convert to tiff
imageinfo.Filename = "image"
imageinfo.Magick="JPEG"
imageinfo.Quality = 10 //since we are displaying, lets use highest quality, lowest compression
blob = image.ImageToBlob(imageinfo)
```

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

MatteColor as IMColorQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image matte (transparent) color.

Notes: (Read and Write property)

Monochrome as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Transform the image to black and white.

Notes: (Read and Write property)

Orientation as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The image orientation.

Notes:

constants:

```
const UndefinedOrientation    = 0
const TopLeftOrientation      = 1
const TopRightOrientation     = 2
const BottomRightOrientation  = 3
const BottomLeftOrientation   = 4
const LeftTopOrientation      = 5
const RightTopOrientation     = 6
const RightBottomOrientation  = 7
const LeftBottomOrientation   = 8
```

For more details please check the ImageMagick documentation.
(Read and Write property)

Page as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Equivalent size of Postscript page.

Notes: (Read and Write property)

PointSize as Double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Text rendering font point size.

Notes: (Read and Write property)

Preview as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image manipulation preview option.

Notes:

Used by 'display'.

constants:

UndefinedPreview	0
RotatePreview	1
ShearPreview	2
RollPreview	3
HuePreview	4
SaturationPreview	5
BrightnessPreview	6
GammaPreview	7
SpiffPreview	8
DullPreview	9
GrayscalePreview	10
QuantizePreview	11
DespecklePreview	12
ReduceNoisePreview	13
AddNoisePreview	14
SharpenPreview	15
BlurPreview	16
ThresholdPreview	17
EdgeDetectPreview	18
SpreadPreview	19
SolarizePreview	20
ShadePreview	21
RaisePreview	22
SegmentPreview	23
SwirlPreview	24
ImplodePreview	25
WavePreview	26
OilPaintPreview	27
CharcoalDrawingPreview	28
JPEGPreview	29

(Read and Write property)

Quality as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** JPEG/MIFF/PNG compression level.

Notes:

Default value is 75.
(Read and Write property)

Release as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** If true, the destructor will release the handle.

Notes: (Read and Write property)

ResolutionUnits as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Units of image resolution.

Notes:

constants:

UndefinedResolution	0	Unset value.
PixelsPerInchResolution	1	Density specifications are specified in units of pixels per inch (english units).
PixelsPerCentimeterResolution	2	Density specifications are specified in units of pixels per centimeter (metric units).

(Read and Write property)

SamplingFactor as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Scene as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

SceneCount as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Scenes as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

ServerName as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** X11 display to display to.

Notes:

obtain fonts from, or to capture image from.
(Read and Write property)

Size as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Width and height of a raw image (an image which does not support width and height information).

Notes:

Size may also be used to affect the image size read from a multi-resolution format (e.g. Photo CD, JBIG, or JPEG).
(Read and Write property)

Temporary as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unknown.
Notes:

For more details please check the ImageMagick documentation.
 (Read and Write property)

Texture as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image filename to use as background texture.
Notes: (Read and Write property)

Type as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Image type.
Notes:

constants:

UndefinedType	0
BilevelType	1
GrayscaleType	2
GrayscaleMatteType	3
PaletteType	4
PaletteMatteType	5
TrueColorType	6
TrueColorMatteType	7
ColorSeparationType	8
ColorSeparationMatteType	9
OptimizeType	10

(Read and Write property)

Verbose as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Print detailed information about the image if True.

Notes: (Read and Write property)

View as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** FlashPix viewing parameters.

Notes: (Read and Write property)

2.22 class ImageMagickQ16MBS

class ImageMagickQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for global functions from the Image Magick library

Notes:

Before using this class you need to load the ImageMagick dylib or dll.

Not all functions from the library are available through the plugin. If you need something special, please ask.

For Mac OS X you need the ImageMagick dylib/bundle and for Windows the normal ImageMagick installation with the DLL.

For more details please check the ImageMagick documentation.

The plugin implements three versions of this ImageMagick classes. One with Q8 for 8 bit quantum depth, one with Q16 for 16 bit depth and Q32 for 32 bit depth.

2.22.1 Methods

Copyright as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The copyright notice for this format.

Notes: For more details please check the ImageMagick documentation.

Features as String

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the ImageMagick features.

Notes: For example whether library is compiled with OpenMP for faster performance.

HomeURL as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the home url of the library.

Notes: For more details please check the ImageMagick documentation.

InitializeMagick(path as string = "")

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes the ImageMagick environment.

Example:

```
dim i as new ImageMagickQ16MBS
i.InitializeMagick("")
```

Notes:

Path: The execution path of the current ImageMagick client.

For more details please check the ImageMagick documentation.

IsMagickInstantiated as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the ImageMagick environment is currently instantiated.

Notes:

In other words: True if InitializeMagick has been called before.

For more details please check the ImageMagick documentation.

LoadErrorString as string

Plugin Version: 9.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last error message from loading the image magick library.

LoadLibrary(path as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Loads the dll/bundle on the give path.

Example:

```
dim i as new ImageMagickQ16MBS

if TargetLinux then
if i.LoadLibrary("libMagick.so.6") then
//MsgBox "library loaded."
else
MsgBox "library failed."
end if
elseif TargetWin32 then
if i.LoadLibrary("CORE_RL_magick_.dll") then
//MsgBox "library loaded."
else
MsgBox "library failed."
end if
```

```

else
// Mac OS X
if i.LoadLibraryFile(GetFolderItem("ImageMagick.bundle")) then
//MsgBox "library loaded."
else
MsgBox "The library failed to load."
end if
end if

```

Notes:

In case the loading fails the library may be linked to some other libraries (e.g. X11) and you need to install them to get it working.

On Windows you can just pass the name of the library and the system will search it on the paths in the environment variable "PATH" (or the Windows folder).

On Linux, pass the path or name of the library and the system will search for it.

For more details please check the ImageMagick documentation.

With plugin version 6.1 the Mac OS X part accepts a path to a dylib file, too.
 Changed to a shared method in plugin version 10.4.

LoadLibraryFile(path as folderitem) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Loads the dll/bundle on the give path.

Example:

```

dim i as new ImageMagickQ16MBS

if TargetLinux then
if i.LoadLibrary("libMagick.so.6") then
//MsgBox "library loaded."
else
MsgBox "library failed."

```

```
end if
elseif TargetWin32 then
if i.LoadLibrary("CORE_RL_magick_.dll") then
//MsgBox "library loaded."
else
MsgBox "library failed."
end if
else
// Mac OS X
if i.LoadLibraryFile(GetFolderItem("ImageMagick.bundle")) then
//MsgBox "library loaded."
else
MsgBox "The library failed to load."
end if
end if
```

Notes:

In case the loading fails the library may be linked to some other libraries (e.g. X11) and you need to install them to get it working.

This is the preferred way for Mac OS X as paths may not be unique.

For more details please check the ImageMagick documentation.

With plugin version 6.1 the Mac OS X part accepts a folderitem for a dylib file, too.
Changed to a shared method in plugin version 10.4.

MagickInfoList as IMMagickInfoListQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the list of known image formats.

Notes:

Sets the last exception property.

For more details please check the ImageMagick documentation.

MagickToMime(name as string) as string

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the officially registered (or de facto) MIME media-type corresponding to a magick string.

Notes:

If there is no registered media-type, then the string "image/x-magick" (all lower case) is returned.

For more details please check the ImageMagick documentation.

NewImageInfo as IMImageInfoQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new ImageInfo object.

Notes:

Returns nil on low memory.

For more details please check the ImageMagick documentation.

NewImageList as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new empty image list.

Notes: For more details please check the ImageMagick documentation.

PackageName as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The package name of the library.

Notes: For more details please check the ImageMagick documentation.

QuantumDepth as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Quantum Depth of the library.

Notes: For more details please check the ImageMagick documentation.

QuantumDepthLibrary as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The quantum depth used to compile the library.

Notes: QuantumDepthLibrary and QuantumDepthPlugin must be equal for the plugin to work correctly. Currently it is compiled for 16bit support.

QuantumRange as String

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The quantum range used by this library.

Notes: Should be a string like "Q16".

ReadImage(info as IMImageInfoQ16MBS) as IMImageQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads an image from a file.

Notes:

Sets the last exception property.

Returns nil on any error.

You need to pass in an info object to describe the image.

For more details please check the ImageMagick documentation.

ReadImageFromString(info as IMImageInfoQ16MBS, data as string) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads an image from a string.

Notes:

Sets the last exception property.

Returns nil on any error.

You need to pass in an info object to describe the image.

For more details please check the ImageMagick documentation.

ReadImageHeaderFromString(info as IMImageInfoQ16MBS, data as string) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the image header.

Notes:

Same as ReadImageFromString except the pixel data is not read.

Sets the last exception property.

For more details please check the ImageMagick documentation.

ReleaseDate as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The release date of the library.

Notes: For more details please check the ImageMagick documentation.

SetCurrentDirectory(path as folderitem) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the current working directory.

Notes: This is needed for most installations to point to the folder with the libraries in order for LoadLibrary to find the dependencies.

Version as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The version of the library.

Notes: For more details please check the ImageMagick documentation.

2.22.2 Properties

LastError as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last error code reported.

Notes:

If an exception is raised and it is not a warning exception, this exception code is saved in this property.
(Read and Write property)

LastException as IMExceptionQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last exception thrown by the Image Magick library.

Notes:

You should check this value after every call to the library, process the error and set the property to nil.

For more details please check the ImageMagick documentation.
(Read and Write property)

2.23 class ImageMagickQ8MBS

class ImageMagickQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for global functions from the Image Magick library

Notes:

Before using this class you need to load the ImageMagick dylib or dll.

Not all functions from the library are available through the plugin. If you need something special, please ask.

For Mac OS X you need the ImageMagick dylib/bundle and for Windows the normal ImageMagick installation with the DLL.

For more details please check the ImageMagick documentation.

The plugin implements three versions of this ImageMagick classes. One with Q8 for 8 bit quantum depth, one with Q16 for 16 bit depth and Q32 for 32 bit depth.

2.23.1 Methods

Copyright as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The copyright notice for this format.

Notes: For more details please check the ImageMagick documentation.

Features as String

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the ImageMagick features.

Notes: For example whether library is compiled with OpenMP for faster performance.

HomeURL as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the home url of the library.

Notes: For more details please check the ImageMagick documentation.

InitializeMagick(path as string = "")

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes the ImageMagick environment.

Example:

```
dim i as new ImageMagickQ8MBS
i.InitializeMagick("")
```

Notes:

Path: The execution path of the current ImageMagick client.

For more details please check the ImageMagick documentation.

IsMagickInstantiated as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the ImageMagick environment is currently instantiated.

Notes:

In other words: True if InitializeMagick has been called before.

For more details please check the ImageMagick documentation.

LoadErrorString as string

Plugin Version: 9.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last error message from loading the image magick library.

LoadLibrary(path as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Loads the dll/bundle on the give path.

Example:

```

dim i as new ImageMagickQ8MBS

if TargetLinux then
if i.LoadLibrary("libMagick.so.6") then
//MsgBox "library loaded."
else
MsgBox "library failed."
end if
elseif TargetWin32 then
if i.LoadLibrary("CORE_RL_magick_.dll") then
//MsgBox "library loaded."
else
MsgBox "library failed."
end if
else
// Mac OS X
if i.LoadLibraryFile(GetFolderItem("ImageMagick.bundle")) then
//MsgBox "library loaded."
else
MsgBox "The library failed to load."
end if
end if

```

Notes:

In case the loading fails the library may be linked to some other libraries (e.g. X11) and you need to install them to get it working.

On Windows you can just pass the name of the library and the system will search it on the paths in the environment variable "PATH" (or the Windows folder).

On Linux, pass the path or name of the library and the system will search for it.

For more details please check the ImageMagick documentation.

With plugin version 6.1 the Mac OS X part accepts a path to a dylib file, too.
 Changed to a shared method in plugin version 10.4.

LoadLibraryFile(path as folderitem) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Loads the dll/bundle on the give path.

Example:

```
dim i as new ImageMagickQ8MBS

if TargetLinux then
if i.LoadLibrary("libMagick.so.6") then
//MsgBox "library loaded."
else
MsgBox "library failed."
end if
elseif TargetWin32 then
if i.LoadLibrary("CORE_RL_magick_.dll") then
//MsgBox "library loaded."
else
MsgBox "library failed."
end if
else
// Mac OS X
if i.LoadLibraryFile(GetFolderItem("ImageMagick.bundle")) then
//MsgBox "library loaded."
else
MsgBox "The library failed to load."
end if
end if
```

Notes:

In case the loading fails the library may be linked to some other libraries (e.g. X11) and you need to install them to get it working.

This is the preferred way for Mac OS X as paths may not be unique.

For more details please check the ImageMagick documentation.

With plugin version 6.1 the Mac OS X part accepts a folderitem for a dylib file, too.
Changed to a shared method in plugin version 10.4.

MagickInfoList as IMMagickInfoListQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the list of known image formats.

Notes:

Sets the last exception property.
For more details please check the ImageMagick documentation.

MagickToMime(name as string) as string

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the officially registered (or de facto) MIME media-type corresponding to a magick string.

Notes:

If there is no registered media-type, then the string "image/x-magick" (all lower case) is returned.

For more details please check the ImageMagick documentation.

NewImageInfo as IMImageInfoQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new ImageInfo object.

Notes:

Returns nil on low memory.
For more details please check the ImageMagick documentation.

NewImageList as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new empty image list.

Notes: For more details please check the ImageMagick documentation.

PackageName as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The package name of the library.

Notes: For more details please check the ImageMagick documentation.

QuantumDepth as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Quantum Depth of the library.

Notes: For more details please check the ImageMagick documentation.

QuantumDepthLibrary as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The quantum depth used to compile the library.

Notes: QuantumDepthLibrary and QuantumDepthPlugin must be equal for the plugin to work correctly. Currently it is compiled for 16bit support.

QuantumRange as String

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The quantum range used by this library.

Notes: Should be a string like "Q16".

ReadImage(info as IMImageInfoQ8MBS) as IMImageQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads an image from a file.

Notes:

Sets the last exception property.

Returns nil on any error.

You need to pass in an info object to describe the image.

For more details please check the ImageMagick documentation.

ReadImageFromString(info as IMImageInfoQ8MBS, data as string) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads an image from a string.

Notes:

Sets the last exception property.

Returns nil on any error.

You need to pass in an info object to describe the image.

For more details please check the ImageMagick documentation.

ReadImageHeaderFromString(info as IMImageInfoQ8MBS, data as string) as IMImageQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the image header.

Notes:

Same as ReadImageFromString except the pixel data is not read.

Sets the last exception property.

For more details please check the ImageMagick documentation.

ReleaseDate as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The release date of the library.

Notes: For more details please check the ImageMagick documentation.

SetCurrentDirectory(path as folderitem) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the current working directory.

Notes: This is needed for most installations to point to the folder with the libraries in order for LoadLibrary to find the dependencies.

Version as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The version of the library.

Notes: For more details please check the ImageMagick documentation.

2.23.2 Properties

LastError as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last error code reported.

Notes:

If an exception is raised and it is not a warning exception, this exception code is saved in this property. (Read and Write property)

LastException as IMExceptionQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last exception thrown by the Image Magick library.

Notes:

You should check this value after every call to the library, process the error and set the property to nil.

For more details please check the ImageMagick documentation.
(Read and Write property)

2.24 class ImageMagickQ32MBS

class ImageMagickQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for global functions from the Image Magick library

Notes:

Before using this class you need to load the ImageMagick dylib or dll.

Not all functions from the library are available through the plugin. If you need something special, please ask.

For Mac OS X you need the ImageMagick dylib/bundle and for Windows the normal ImageMagick installation with the DLL.

For more details please check the ImageMagick documentation.

The plugin implements three versions of this ImageMagick classes. One with Q8 for 8 bit quantum depth, one with Q16 for 16 bit depth and Q32 for 32 bit depth.

2.24.1 Methods

Copyright as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The copyright notice for this format.

Notes: For more details please check the ImageMagick documentation.

Features as String

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the ImageMagick features.

Notes: For example whether library is compiled with OpenMP for faster performance.

HomeURL as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the home url of the library.

Notes: For more details please check the ImageMagick documentation.

InitializeMagick(path as string = "")

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Initializes the ImageMagick environment.

Example:

```
dim i as new ImageMagickQ32MBS
i.InitializeMagick("")
```

Notes:

Path: The execution path of the current ImageMagick client.

For more details please check the ImageMagick documentation.

IsMagickInstantiated as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the ImageMagick environment is currently instantiated.

Notes:

In other words: True if InitializeMagick has been called before.

For more details please check the ImageMagick documentation.

LoadErrorString as string

Plugin Version: 9.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last error message from loading the image magick library.

LoadLibrary(path as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Loads the dll/bundle on the give path.

Example:

```
dim i as new ImageMagickQ32MBS

if TargetLinux then
if i.LoadLibrary("libMagick.so.6") then
//MsgBox "library loaded."
else
MsgBox "library failed."
end if
elseif TargetWin32 then
if i.LoadLibrary("CORE_RL_magick_.dll") then
//MsgBox "library loaded."
else
MsgBox "library failed."
end if
else
// Mac OS X
if i.LoadLibraryFile(GetFolderItem("ImageMagick.bundle")) then
//MsgBox "library loaded."
```

```

else
MsgBox "The library failed to load."
end if
end if

```

Notes:

In case the loading fails the library may be linked to some other libraries (e.g. X11) and you need to install them to get it working.

On Windows you can just pass the name of the library and the system will search it on the paths in the environment variable "PATH" (or the Windows folder).

On Linux, pass the path or name of the library and the system will search for it.

For more details please check the ImageMagick documentation.

With plugin version 6.1 the Mac OS X part accepts a path to a dylib file, too.
 Changed to a shared method in plugin version 10.4.

LoadLibraryFile(path as folderitem) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Loads the dll/bundle on the give path.

Example:

```

dim i as new ImageMagickQ32MBS

if TargetLinux then
if i.LoadLibrary("libMagick.so.6") then
//MsgBox "library loaded."
else
MsgBox "library failed."
end if
elseif TargetWin32 then
if i.LoadLibrary("CORE_RL_magick_.dll") then
//MsgBox "library loaded."

```

```
else
MsgBox "library failed."
end if
else
// Mac OS X
if i.LoadLibraryFile(GetFolderItem("ImageMagick.bundle")) then
//MsgBox "library loaded."
else
MsgBox "The library failed to load."
end if
end if
```

Notes:

In case the loading fails the library may be linked to some other libraries (e.g. X11) and you need to install them to get it working.

This is the preferred way for Mac OS X as paths may not be unique.

For more details please check the ImageMagick documentation.

With plugin version 6.1 the Mac OS X part accepts a folderitem for a dylib file, too.
Changed to a shared method in plugin version 10.4.

MagickInfoList as IMMagickInfoListQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the list of known image formats.

Notes:

Sets the last exception property.

For more details please check the ImageMagick documentation.

MagickToMime(name as string) as string

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the officially registered (or de facto) MIME media-type corresponding to a magick string.

Notes:

If there is no registered media-type, then the string "image/x-magick" (all lower case) is returned.

For more details please check the ImageMagick documentation.

NewImageInfo as IImageInfoQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new ImageInfo object.

Notes:

Returns nil on low memory.

For more details please check the ImageMagick documentation.

NewImageList as IImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new empty image list.

Notes: For more details please check the ImageMagick documentation.

PackageName as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The package name of the library.

Notes: For more details please check the ImageMagick documentation.

QuantumDepth as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Quantum Depth of the library.

Notes: For more details please check the ImageMagick documentation.

QuantumDepthLibrary as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The quantum depth used to compile the library.

Notes: QuantumDepthLibrary and QuantumDepthPlugin must be equal for the plugin to work correctly. Currently it is compiled for 16bit support.

QuantumRange as String

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The quantum range used by this library.

Notes: Should be a string like "Q16".

ReadImage(info as IMImageInfoQ32MBS) as IMImageQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads an image from a file.

Notes:

Sets the last exception property.

Returns nil on any error.

You need to pass in an info object to describe the image.

For more details please check the ImageMagick documentation.

ReadImageFromString(info as IMImageInfoQ32MBS, data as string) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads an image from a string.

Notes:

Sets the last exception property.

Returns nil on any error.

You need to pass in an info object to describe the image.

For more details please check the ImageMagick documentation.

ReadImageHeaderFromString(info as IMImageInfoQ32MBS, data as string) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the image header.

Notes:

Same as ReadImageFromString except the pixel data is not read.

Sets the last exception property.

For more details please check the ImageMagick documentation.

ReleaseDate as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The release date of the library.

Notes: For more details please check the ImageMagick documentation.

SetCurrentDirectory(path as folderitem) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the current working directory.

Notes: This is needed for most installations to point to the folder with the libraries in order for LoadLibrary to find the dependencies.

Version as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The version of the library.

Notes: For more details please check the ImageMagick documentation.

2.24.2 Properties

LastError as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last error code reported.

Notes:

If an exception is raised and it is not a warning exception, this exception code is saved in this property.
(Read and Write property)

LastException as IMExceptionQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last exception thrown by the Image Magick library.

Notes:

You should check this value after every call to the library, process the error and set the property to nil.

For more details please check the ImageMagick documentation.
(Read and Write property)

2.25 class IMExceptionQ16MBS

class IMExceptionQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for information about an Image Magick Exception.

Notes:

Some functions can throw an exception and you find this exception object after calling the function inside the class. For Example after calling `IMImageQ16MBS.resize`, the `IMImageQ16MBS.LastException` property will be nil for no exception or just contain the exception from the resize operation.

For more details please check the ImageMagick documentation.

2.25.1 Methods

Close

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.

Notes:

There is no need to call this method except you want to free all resources used by this object now without waiting for Realbasic to do it for you.
(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

2.25.2 Properties

Description as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The description of the exception.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Reason as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The reason of the exception.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Severity as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The exception type.
Notes:

some usefull constants:

For more details please check the ImageMagick documentation.
(Read and Write property)

Signature as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The signature of the exception.
Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

2.26 class IMColorQ8MBS

class IMColorQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Image Magick class to represent a color.

Example:

```
dim i as IMImageInfoQ8MBS
dim c as IMColorQ8MBS
```

```
c=i.BackgroundColor
c.red=65535 // full red
```

i.BackgroundColor=c

Notes:

As you see above the IMColorQ8MBS object does not reference the original values, but contains a copy, so you must assign the modified color back to store it.
(Same as on the Realbasic Color class)

2.26.1 Methods

Constructor

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a color object where all four properties are zero.

See also:

- 2.26.1 Constructor(c as color) 185
- 2.26.1 Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0) 185

Constructor(c as color)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new color value and fills it with the given Real Studio color.

See also:

- 2.26.1 Constructor 185
- 2.26.1 Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0) 185

Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new color object with the given values.

See also:

- 2.26.1 Constructor 185
- 2.26.1 Constructor(c as color) 185

2.26.2 Properties

Blue as UInt32

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The blue component.
Notes:

Value from 0 to 65535.
 (Read and Write property)

ColorValue as Color

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The color as a Real Studio color.
Notes:

Please note that Real Studio colors are 8 bit. So for Q16 and Q32 classes the colors are scaled up or down. This reads/writes the red, green and blue property, but not the opacity property.
 (Read and Write property)

Green as UInt32

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The green component.
Notes:

Value from 0 to 65535.
 (Read and Write property)

Opacity as UInt32

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The opacity part of the color.

Notes:

Value from 0 to 65535.
(Read and Write property)

Red as UInt32

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The red component.

Notes:

Value from 0 to 65535.
(Read and Write property)

2.27 class IMExceptionQ8MBS

class IMExceptionQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for information about an Image Magick Exception.

Notes:

Some functions can throw an exception and you find this exception object after calling the function inside the class. For Example after calling `IMImageQ8MBS.resize`, the `IMImageQ8MBS.LastException` property will be nil for no exception or just contain the exception from the resize operation.

For more details please check the ImageMagick documentation.

2.27.1 Methods

Close

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.
Notes:

There is no need to call this method except you want to free all resources used by this object now without waiting for Realbasic to do it for you.
(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

2.27.2 Properties

Description as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The description of the exception.
Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Reason as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The reason of the exception.
Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Severity as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The exception type.
Notes:

some usefull constants:

For more details please check the ImageMagick documentation.
(Read and Write property)

Signature as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The signature of the exception.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

2.28 class IMExceptionQ32MBS

class IMExceptionQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The class for information about an Image Magick Exception.

Notes:

Some functions can throw an exception and you find this exception object after calling the function inside the class. For Example after calling `IMImageQ32MBS.resize`, the `IMImageQ32MBS.LastException` property will be nil for no exception or just contain the exception from the resize operation.

For more details please check the ImageMagick documentation.

2.28.1 Methods

Close

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.
Notes:

There is no need to call this method except you want to free all resources used by this object now without waiting for Realbasic to do it for you.
(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

2.28.2 Properties

Description as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The description of the exception.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Reason as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The reason of the exception.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Severity as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The exception type.
Notes:

some usefull constants:

For more details please check the ImageMagick documentation.
(Read and Write property)

Signature as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The signature of the exception.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

2.29 class IMColorQ32MBS

class IMColorQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Image Magick class to represent a color.

Example:

```
dim i as IImageInfoQ32MBS
dim c as IMColorQ32MBS
```

```
c=i.BackgroundColor
c.red=65535 // full red
i.BackgroundColor=c
```

Notes:

As you see above the IMColorQ32MBS object does not reference the original values, but contains a copy, so you must assign the modified color back to store it.
(Same as on the Realbasic Color class)

2.29.1 Methods

Constructor

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a color object where all four properties are zero.

See also:

- 2.29.1 Constructor(c as color) 192
- 2.29.1 Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0) 192

Constructor(c as color)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new color value and fills it with the given Real Studio color.

See also:

- 2.29.1 Constructor 192
- 2.29.1 Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0) 192

Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new color object with the given values.

See also:

- 2.29.1 Constructor 192
- 2.29.1 Constructor(c as color) 192

2.29.2 Properties

Blue as UInt32

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The blue component.
Notes:

Value from 0 to 65535.
(Read and Write property)

ColorValue as Color

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The color as a Real Studio color.

Notes:

Please note that Real Studio colors are 8 bit. So for Q16 and Q32 classes the colors are scaled up or down. This reads/writes the red, green and blue property, but not the opacity property.
(Read and Write property)

Green as UInt32

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The green component.

Notes:

Value from 0 to 65535.
(Read and Write property)

Opacity as UInt32

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The opacity part of the color.

Notes:

Value from 0 to 65535.
(Read and Write property)

Red as UInt32

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The red component.
Notes:

Value from 0 to 65535.
(Read and Write property)

2.30 class IMColorQ16MBS

class IMColorQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Image Magick class to represent a color.

Example:

```
dim i as IImageInfoQ16MBS
dim c as IMColorQ16MBS
```

```
c=i.BackgroundColor
c.red=65535 // full red
i.BackgroundColor=c
```

Notes:

As you see above the IMColorQ16MBS object does not reference the original values, but contains a copy, so you must assign the modified color back to store it.
(Same as on the Realbasic Color class)

2.30.1 Methods

Constructor

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a color object where all four properties are zero.

See also:

- 2.30.1 Constructor(c as color) 195
- 2.30.1 Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0) 195

Constructor(c as color)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new color value and fills it with the given Real Studio color.

See also:

- 2.30.1 Constructor 194
- 2.30.1 Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0) 195

Constructor(red as UInt32, green as UInt32, blue as UInt32, Opacity as UInt32 = 0)

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new color object with the given values.

See also:

- 2.30.1 Constructor 194
- 2.30.1 Constructor(c as color) 195

2.30.2 Properties**Blue as UInt32**

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The blue component.
Notes:

Value from 0 to 65535.
(Read and Write property)

ColorValue as Color

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The color as a Real Studio color.

Notes:

Please note that Real Studio colors are 8 bit. So for Q16 and Q32 classes the colors are scaled up or down. This reads/writes the red, green and blue property, but not the opacity property.

(Read and Write property)

Green as UInt32

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The green component.

Notes:

Value from 0 to 65535.

(Read and Write property)

Opacity as UInt32

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The opacity part of the color.

Notes:

Value from 0 to 65535.

(Read and Write property)

Red as UInt32

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The red component.

Notes:

Value from 0 to 65535.

(Read and Write property)

2.31 class IMImageQ32MBS

class IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for an Image Magick Image in memory.

Notes:

Can exist with or without pixel data.

For more details please check the ImageMagick documentation.

2.31.1 Methods

AdaptiveThreshold(width as integer, height as integer, offset as integer) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** AdaptiveThreshold selects an individual threshold for each pixel based on the range of intensity values in its local neighborhood.

Notes:

This allows for thresholding of an image whose global intensity histogram doesn't contain distinctive peaks. Sets the last exception property.

width: The width of the local neighborhood.

height: The height of the local neighborhood.

offset: The mean offset.

For more details please check the ImageMagick documentation.

AddNoise(NoiseType as integer) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds random noise to the image.

Notes:

Constants

For more details please check the ImageMagick documentation.
Sets the last exception property.

AffineTransformImage(matrix as IMImageAffineTransformQ32MBS) as IMImageQ32MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Transforms an image as dictated by the affine matrix.

AppendImageToList(img as IMImageQ32MBS)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds an image to the image list.

Notes: For more details please check the ImageMagick documentation.

Average as IMImageQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Average() method takes a set of images and averages them together.

Notes:

Each image in the set must have the same width and height. Average() returns a single image with each corresponding pixel component of each image averaged. On failure, a nil image is returned and exception describes the reason for the failure.

Sets the last exception property.

For more details please check the ImageMagick documentation.

BilevelChannel(channel as integer, threshold as double) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Changes the value of individual pixels based on the intensity of each pixel channel.

Notes:

The result is a high-contrast image.

channel: The channel type.

threshold: define the threshold values.

Constants for channel:

For more details please check the ImageMagick documentation.

BlackThreshold(threshold as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** BlackThreshold is like Threshold but forces all pixels below the threshold into black while leaving all pixels above the threshold unchanged.

Notes:

No exceptions are generated.

threshold: Define the threshold value. (ASCII string)

For more details please check the ImageMagick documentation.

BlobSize as integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The expected size for this image written to a file.

Notes: For more details please check the ImageMagick documentation.

Blur(radius as double, sigma as double) as IMImageQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blurs an image.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For

reasonable results, the radius should be larger than sigma. Use a radius of 0 and `BlurImage` selects a suitable radius for you.

radius: The radius of the Gaussian, in pixels, not counting the center pixel.

sigma: The standard deviation of the Gaussian, in pixels.

For more details please check the ImageMagick documentation.

BlurImageChannel(channel as integer, radius as double, sigma as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blurs an image.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, the radius should be larger than sigma. Use a radius of 0 and `BlurImageChannel` selects a suitable radius for you.

channel: The channel type.

radius: The radius of the Gaussian, in pixels, not counting the center pixel.

sigma: The standard deviation of the Gaussian, in pixels.

Constants for channel:

For more details please check the ImageMagick documentation.

BorderImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ32MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Surrounds the image with a border of the color defined by the `bordercolor` member of the image.

Notes: The width and height of the border are defined by the corresponding parameters.

Charcoal(radius as double, sigma as double) as IMImageQ32MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Charcoal creates a new image that is a copy of an existing one with the edge highlighted.

Notes:

radius: the radius of the pixel neighborhood.

sigma: The standard deviation of the Gaussian, in pixels.

Returns nil on any error.

Sets the last exception property.

Chop(x as integer, y as integer, width as integer, height as integer) as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Chop removes a region of an image and collapses the image to occupy the removed portion.

Notes:

Returns nil on any error.

Sets the last exception property.

ClipPath(path as string, inside as boolean) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the image clip mask based any clipping path information if it exists.

Notes:

Returns true on success and false on any error.

Clone as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a copy of this image object.

Notes: For more details please check the ImageMagick documentation.

CloneImageAttributes(image as IMImageAttributeQ32MBS) as Boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** CloneImageAttributes() clones one or more image attributes.

Notes: Returns false on any error.

CloneImageProfiles(SourceImage as IMImageQ32MBS) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clones one or more image profiles.

Notes: Returns false on any error and true on success.

Close

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.

Notes:

There is no need to call this method except you want to free all resources used by this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

CoalesceImages as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** CoalesceImages composites a set of images while respecting any page offsets and disposal methods.

Notes:

GIF, MIFF, and MNG animation sequences typically start with an image background and each subsequent image varies in size and offset. CoalesceImages() returns a new sequence where each image in the sequence is the same size as the first and composited with the next image in the sequence.

Returns nil on any error.

Sets the last exception property.

Colorize(opacity as string, PenColorRed as integer, PenColorGreen as integer, PenColorBlue as integer, PenColorOpacity as integer) as IMImageQ32MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method ColorizeImage creates a new image that is a copy of an existing one with the image pixels colorized.

Notes:

The colorization is controlled with the pen color and the opacity levels.

opacity: A character string indicating the level of opacity as a percentage (0-100).

PenColorRed, PenColorGreen, PenColorBlue and PenColorOpacity define the pen color used.

Returns nil on any error.

Sets the last exception property.

Combine(channel as integer) as IMImageQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Combines one or more images into a single image.

Notes:

The grayscale value of the pixels of each image in the sequence is assigned in order to the specified channels of the combined image. The typical ordering would be image 1 =>Red, 2 =>Green, 3 =>Blue, etc.

The lastexception property is set.

CompareImageLayers(ImageLayerMethod as integer) as IMImageQ32MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** CompareImageLayers() compares each image with the next in a sequence and returns the minimum bounding region of all the pixel differences (of the mageLayerMethod specified) it discovers.

Notes:

Images do NOT have to be the same size, though it is best that all the images are 'coalesced' (images are all the same size, on a flattened canvas, so as to represent exactly how an specific frame should look).

No GIF dispose methods are applied, so GIF animations must be coalesced before applying this image operator to find differences to them.

ImageLayerMethod:

the layers type to compare images with. Must be one of... CompareAnyLayer, CompareClearLayer, CompareOverlayLayer.

Can raise an exception.

Composite(ComposeOperator as integer, Image as IMImageQ32MBS, x as integer, y as integer)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the second image composited onto the first at the specified offsets.

Notes:

compose: Specifies an image composite operator.

Image: The second image.

x: An integer that specifies the column offset of the composited image.

y: An integer that specifies the row offset of the composited image.

No error code and exception!

ConsolidateCMYKImages as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Consolidates a sequence of CMYK images.

Notes:

Returns nil on any error.

Sets the last exception property.

CopyPicture as picture

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IMImageQ32MBS // your image
Canvas1.Backdrop=image.CopyPicture
```

Notes:

Sets the last exception property.

Returns nil on any error.

This method works only for bitmap images.

See also:

- 2.31.1 CopyPicture(x as integer, y as integer, width as integer, height as integer) as picture 205

CopyPicture(x as integer, y as integer, width as integer, height as integer) as picture

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies a portion of the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IMImageQ32MBS // your image
Canvas1.Backdrop=image.CopyPicture(0,0,image.Width,image.Height)
```

Notes:

Sets the last exception property.

Returns nil on any error.

This method works only for bitmap images.

x and y are zero based.

See also:

- 2.31.1 CopyPicture as picture 205

CopyPictureMask as picture

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the mask of the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IMImageQ32MBS // your image
Canvas1.Backdrop=image.CopyPictureMask
```

Notes:

Sets the last exception property.

Returns nil on any error.

This method works only for bitmap images.

See also:

- 2.31.1 CopyPictureMask(x as integer, y as integer, width as integer, height as integer) as picture 206

CopyPictureMask(x as integer, y as integer, width as integer, height as integer) as picture

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies a portion of the mask of the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IMImageQ32MBS // your image
Canvas1.Backdrop=image.CopyPictureMask(0,0,image.Width,image.Height)
```

Notes:

Sets the last exception property.

Returns nil on any error.

This method works only for bitmap images.

x and y are zero based.

See also:

- 2.31.1 CopyPictureMask as picture

CopyPixel(x as integer, y as integer) as IMColorQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies a pixel.

Notes:

Returns nil on any error.

This method works only for bitmap images.

x and y are zero based.

Crop(x as integer, y as integer, width as integer, height as integer) as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Crop extracts a region of the image starting at the offset defined by geometry.

Notes:

Returns nil on any error.

Sets the last exception property.

CropImageToTiles(CropGeometry as string) as IMImageQ32MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Crops a single image, into a possible list of tiles.

Notes: This may include a single sub-region of the image. This basically applies all the normal geometry flags for Crop.

CycleColormap(displace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Displaces an image's colormap by a given number of positions.

Notes:

If you cycle the colormap a number of times you can produce a psychedelic effect.

Returns true on success.

displace: displace the colormap this amount.

DecipherImage(passkey as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts cipher pixels to plain pixels.

Notes:

Passkey: decipher cipher pixels with this passphrase.
Returns true on success.

DeconstructImages as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** DeconstructImages() compares each image with the next in a sequence and returns the minimum bounding region of all differences from the first image.

Notes:

Returns nil on any error.
Sets the last exception property.

DeleteImageAttribute(key as string) as Boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** DeleteImageAttribute() deletes an attribute from the image.

Notes: Returns false on any error.

Despeckle() as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reduces the speckle noise in an image while perserving the edges of the original image.

Notes:

Sets the last exception property.
For more details please check the ImageMagick documentation.

DestroyImage

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Releases the memory used for this image and sets handle to 0.

Notes:

For more details please check the ImageMagick documentation.
The destructor will call this for you if release=true.

DestroyImageAttributes

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Deallocates memory associated with the image attribute list.

DestroyImageList

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Destroys the image list and sets the handle to 0.

Notes:

For more details please check the ImageMagick documentation.
The destructor will call this for you if release=true.

DestroyImageProfiles

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Releases memory associated with an image profile map.

Edge(radius as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds edges in an image.

Notes:

Radius defines the radius of the convolution filter. Use a radius of 0 and Edge selects a suitable radius for you.

Sets the last exception property.

For more details please check the ImageMagick documentation.

Emboss(radius as double, sigma as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a grayscale image with a three-dimensional effect.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, radius should be larger than sigma. Use a radius of 0 and Emboss selects a suitable radius for you.

Sets the last exception property.

For more details please check the ImageMagick documentation.

EncipherImage(passkey as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts pixels to cipher-pixels.

Notes:

passkey: encipher pixels with this passphrase.

Returns true on success.

ExcerptImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ32MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a excerpt of the image as defined by the geometry.

Notes: Define the region of the image to extend with x, y, width, and height.

ExtentImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ32MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Extends the image as defined by the geometry, gravity, and image background color.

Notes:

Define the region of the image to extend with x, y, width, and height.

Set the (x,y) offset of the geometry to move the original image relative to the extended image.

FlattenImages as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Flatten composites all images from the current image pointer to the end of the image list and returns a single flattened image.

Notes:

Returns nil on any error.

Sets the last exception property.

Flip as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Flip creates a vertical mirror image by reflecting the pixels around the central x-axis.

Notes:

Returns nil on any error.

Sets the last exception property.

Flop as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Flop creates a horizontal mirror image by reflecting the pixels around the central y-axis.

Notes:

Returns nil on any error.

Sets the last exception property.

FrameImage(x as integer, y as integer, width as integer, height as integer, innerBevel as integer, OuterBevel as integer) as **IMImageQ32MBS**

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a simulated three-dimensional border around the image.

Notes: The color of the border is defined by the `MatteColor` of image. Width and height specify the border width of the vertical and horizontal sides of the frame. `innerBevel` and `OuterBevel` indicate the width of the inner and outer shadows of the frame.

FxImage(expression as string) as **IMImageQ32MBS**

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** `FxImage()` applies a mathematical expression to the specified image.

Notes: Can raise an exception.

GaussianBlurChannel(channel as integer, radius as double, sigma as double) as **IMImageQ32MBS**

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blurs an image.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (`sigma`) . For reasonable results, the radius should be larger than `sigma`. Use a radius of 0 and `GaussianBlur` selects a suitable radius for you.

Sets the last exception property.

radius: the radius of the Gaussian, in pixels, not counting the center pixel.

channel: The channel type.

sigma: the standard deviation of the Gaussian, in pixels.

Constants for channel:

For more details please check the ImageMagick documentation.

GetImageAttribute(key as string) as IMImageAttributeQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** GetImageAttribute searches the list of image attributes and returns a reference to the attribute if it exists otherwise nil.

GetImageClippingPathAttribute as IMImageAttributeQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** GetImageClippingPathAttribute searches the list of image attributes and returns a reference to a clipping path if it exists otherwise nil.

GetImageProfile(name as string) as string

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets a profile associated with an image by name.

Notes: Returns "" on any error.

GetNextImageAttribute as IMImageAttributeQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** GetNextImageAttribute() gets the next image attribute.

Notes: Returns nil on any error.

GetNextImageProfile as string

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets the next profile name for an image.

Notes: Returns "" on any error.

HandleMemory as memoryblock

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The content of the whole Image structure copied into a memoryblock.

Notes: Returns nil on any error.

ImagesToBlob(info as IMImageInfoQ32MBS) as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** ImagesToBlob implements direct to memory image formats.

Notes:

It returns the image sequence as a string. The magick member of the ImageInfo structure determines the format of the returned blob (GIF, JPEG, PNG, etc.)

Note, some image formats do not permit multiple images to the same image stream (e.g. JPEG). in this instance, just the first image of the sequence is returned as a blob.

Sets the last exception property and returns "" on any error.
For more details please check the ImageMagick documentation.

ImageToBlob(info as IMImageInfoQ32MBS) as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** ImagesToBlob implements direct to memory image formats.

Example:

```
dim im as ImageMagickQ32MBS // global
```

```
Function IMPictureToString(p as picture, magick as string, quality as integer) As string
```

```
dim image as new IMImageQ32MBS
```

```
dim imageinfo as IMImageInfoQ32MBS
```

```
dim s,data as string
```

```
dim impp as new IMMagickPixelPacketQ32MBS
```

```
// empty string for nil picture
```

```
if p = nil then
```

```
Return ""
```

```
end if

// create a new picture info

imageinfo = im.NewImageInfo
imageinfo.ColorSpace=1
// only color space is needed. 1 for RGB.

// background color of image
impp.red = 0
impp.Green = 0
impp.Blue = 0

// creates a new image object
if not image.NewImage(imageinfo,p.Width,p.Height,impp) then
Return ""
end if

// copy RB picture into IM Image at position 0/0
image.ColorSpace = 1
image.SetPicture(p,0,0)

// set compression data
imageinfo.Magick = magick
imageinfo.Quality = quality

// and rendering intent: 2=PerceptualIntent
image.RenderingIntent = 2

// create image data
data = image.ImageToBlob(imageinfo)

// release memory
image.DestroyImage
imageinfo.DestroyImageInfo

// return result
Return data

Exception
// in case of an exception return nothing
Return ""

End Function
```

Notes:

It returns the image sequence as a string. The magick member of the ImageInfo structure determines the format of the returned blob (GIF, JPEG, PNG, etc.)

Note, some image formats do not permit multiple images to the same image stream (e.g. JPEG). in this instance, just the first image of the sequence is returned as a blob.

Sets the last exception property and returns "" on any error.
For more details please check the ImageMagick documentation.

Implode(factor as double) as IMImageQ32MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method ImplodeImage creates a new image that is a copy of an existing one with the image pixels "implode" by the specified percentage.

Notes:

factor: A double value that defines the extent of the implosion.

Returns nil on any error.
Sets the last exception property.

IsBlobExempt as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the blob is exempt.

Notes: For more details please check the ImageMagick documentation.

IsBlobSeekable as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the blob is seekable.

Notes: For more details please check the ImageMagick documentation.

IsBlobTemporary as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the blob is temporary.

Notes: For more details please check the ImageMagick documentation.

Magnify as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A convenience method that scales an image proportionally to twice its size.

Notes:

Sets the last exception property.

For more details please check the ImageMagick documentation.

MedianFilter(radius as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Applies a digital filter that improves the quality of a noisy image.

Notes:

Each pixel is replaced by the median in a set of neighboring pixels as defined by radius.

Sets the last exception property.

For more details please check the ImageMagick documentation.

MergeImageLayers(ImageLayerMethod as integer) as IMImageQ32MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** MergeImageLayers() composes all the image layers from the current given image onward to produce a single image of the merged layers.

Notes:

The initial canvas's size depends on the given ImageLayerMethod, and is initialized using the first images background color. The images are then composited onto that image in sequence using the given

composition that has been assigned to each individual image.

ImageLayerMethod:
the method of selecting the size of the initial canvas.

MergeLayer: Merge all layers onto a canvas just large enough to hold all the actual images. The virtual canvas of the first image is preserved but otherwise ignored.

FlattenLayer: Use the virtual canvas size of first image. Images which fall outside this canvas is clipped. This can be used to 'fill out' a given virtual canvas.

MosaicLayer: Start with the virtual canvas of the first image, enlarging left and right edges to contain all images. Images with negative offsets will be clipped.

Can raise an exception.

Minify as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A convenience method that scales an image proportionally to half its size.

Notes:

Sets the last exception property.
For more details please check the ImageMagick documentation.

MosaicImages as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** MosaicImages inlays an image sequence to form a single coherent picture.

Notes:

It returns a single image with each image in the sequence composited at the location defined by the page member of the image structure.
Returns nil on any error.
Sets the last exception property.

MotionBlur(radius as double, sigma as double, angle as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Simulates motion blur.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma). For reasonable results, radius should be larger than sigma. Use a radius of 0 and MotionBlur selects a suitable radius for you. Angle gives the angle of the blurring motion.
Sets the last exception property.

For more details please check the ImageMagick documentation.

NewImage(info as IMImageInfoQ32MBS, width as integer, height as integer, background as IMMackPixelPacketQ32MBS) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new image.

Example:

```
dim im as ImageMagickQ32MBS // global
dim p as picture
dim imageinfo as IMImageInfoQ32MBS
dim image as IMImageQ32MBS
dim b as new IMMackPixelPacketQ32MBS
b.Blue=65535
b.ColorSpace=1 // RGB
b.Depth=16

imageinfo = im.NewImageInfo
imageinfo.Depth=16
imageinfo.ColorSpace=1

//this should read any image IM understands
image = new IMImageQ32MBS
if image.NewImage(imageinfo,500,500,b) then
p=NewPicture(300,300,32)
p.Graphics.ForeColor=Rgb(255,0,0)
p.Graphics.FillOval 0,0,300,300
```

```
image.SetPicture p,0,0
else
MsgBox "failed"
end if
```

Notes: Returns false on failure and true on success.

OilPaint(radius as double) as IImageQ32MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method OilPaintImage creates a new image that is a copy of an existing one with each pixel component replaced with the color of greatest frequency in a circular neighborhood.

Notes:

radius parameter: radius of the circular neighborhood.
Returns nil on any error.
Sets the last exception property.

OptimizeImageLayers as IImageQ32MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** OptimizeImageLayers() compares each image the GIF disposed forms of the previous image in the sequence.

Notes:

From this it attempts to select the smallest cropped image to replace each frame, while preserving the results of the GIF animation.

Can raise an exception.

OptimizeImageTransparency

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** OptimizeImageTransparency() takes a frame optimized GIF animation, and compares the overlayed pixels against the disposal image resulting from all the previous frames in the animation.

Notes:

Any pixel that does not change the disposal image (and thus does not effect the outcome of an overlay) is made transparent.

WARNING: This modifies the current images directly, rather than generate a new image sequence.

Can raise an exception.

OptimizePlusImageLayers as IImageQ32MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** OptimizeImagePlusLayers() is exactly as OptimizeImageLayers(), but may also add or even remove extra frames in the animation, if it improves the total number of pixels in the resulting GIF animation.

Notes: Can raise an exception.

ProfileImage(name as string, ProfileData as string) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds or removes a ICC, IPTC, or generic profile from an image.

Notes:

If the ProfileData is "", it is removed from the image otherwise added. Use a name of '*' and a ProfileData of "" to remove all profiles from the image.

Returns false on any error and true on success.

RadialBlur(angle as double) as IImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** RadialBlur applies a radial blur to the image.

Notes:

angle: The angle of the radial blur.

Sets the last exception property.
For more details please check the ImageMagick documentation.

RaiseImage(x as integer, y as integer, width as integer, height as integer, raise as boolean) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a simulated three-dimensional button-like effect by lightening and darkening the edges of the image.

Notes:

Width and height define the width of the vertical and horizontal edge of the effect.
raise: A value other than zero creates a 3-D raise effect, otherwise it has a lowered effect.

RandomThresholdChannel(channel as integer, thresholds as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Changes the value of individual pixels based on the intensity of each pixel compared to a random threshold.

Notes:

The result is a low-contrast, two color image.

channel: The channel or channels to be thresholded.

thresholds: a geometry string containing low,high thresholds. If the string contains 2x2, 3x3, or 4x4, an ordered dither of order 2, 3, or 4 is performed instead. (ASCII string)

Sets the last exception property.

Constants for channel:

For more details please check the ImageMagick documentation.

ReduceNoise(radius as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Smooths the contours of an image while still preserving edge information.

Notes:

The algorithm works by replacing each pixel with its neighbor closest in value. A neighbor is defined by radius. Use a radius of 0 and ReduceNoise selects a suitable radius for you.

For more details please check the ImageMagick documentation.

RemoveDuplicateLayers

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes any image that is exactly the same as the next image in the given image list.

Notes:

Image size and virtual canvas offset must also match, though not the virtual canvas size itself.

No check is made with regards to image disposal setting, though it is the dispose setting of later image that is kept. Also any time delays are also added together. As such coalesced image animations should still produce the same result, though with duplicate frames merged into a single frame.

RemoveFirstImageFromList as IMImageQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the first image from the image list and returns the image.

Notes:

Returns nil on any error.

For more details please check the ImageMagick documentation.

RemoveImageProfile(name as string) as string

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes a profile from the image-map by its name.

RemoveZeroDelayLayers

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes any image that as a zero delay time.

Notes:

Such images generally represent intermediate or partial updates in GIF animations used for file optimization. They are not ment to be displayed to users of the animation. Viewable images in an animation should have a time delay of 3 or more centi-seconds (hundredths of a second).

However if all the frames have a zero time delay, then either the animation is as yet incomplete, or it is not a GIF animation. This is a non-sensible situation, so no image will be removed and a 'Zero Time Animation' warning (exception) given.

No warning will be given if no image was removed because all images had an appropriate non-zero time delay set.

Due to the special requirements of GIF disposal handling, GIF animations should be coalesced first, before calling this function, though that is not a requirement.

ResetImageAttributeIterator

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** ResetImageAttributeIterator() resets the image attributes iterator.

Notes: Use it in conjunction with GetNextImageAttribute() to iterate over all the values associated with an image.

ResetImageProfileIterator

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Resets the image profile iterator.

Notes: Use it in conjunction with `GetNextImageProfile()` to iterate over all the profiles associated with an image.

Resize(width as integer, height as integer, FilterID as integer, blur as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Scales an image to the desired dimensions.

Notes:

Constants for the FilterID:

Most of the filters are FIR (finite impulse response), however, Bessel, Gaussian, and Sinc are IIR (infinite impulse response). Bessel and Sinc are windowed (brought down to zero) with the Blackman filter. Sets the last exception property.

RGBTransformImage(Colorspace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method `RGBTransformImage` converts the reference image from RGB to an alternate colorspace.

Notes:

The transformation matrices are not the standard ones: the weights are rescaled to normalized the range of the transformed values to be [0..MaxRGB] .

colorspace: An integer value that indicates which colorspace to transform the image.

Returns false on any error and true on success.

constants:

Roll(x as integer, y as integer) as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Roll offsets an image as defined by x and y.

Notes:

Returns nil on any error.

Sets the last exception property.

Rotate(degrees as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Rotation of an image.

Notes:

Method RotateImage creates a new image that is a rotated copy of an existing one. Positive angles rotate counter-clockwise (right-hand rule), while negative angles rotate clockwise. Rotated images are usually larger than the originals and have 'empty' triangular corners. X axis. Empty triangles left over from shearing the image are filled with the color specified by the image background_ color. RotateImage allocates the memory necessary for the new Image structure and returns a pointer to the new image.

Method RotateImage is based on the paper "A Fast Algorithm for General Raster Rotation" by Alan W. Paeth. RotateImage is adapted from a similar method based on the Paeth paper written by Michael Halle of the Spatial Imaging Group, MIT Media Lab.

degrees: Specifies the number of degrees to rotate the image.

Sets the lastexception property.

Returns nil on low memory.

For more details please check the ImageMagick documentation.

Sample(width as integer, height as integer) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Scales an image to the desired dimensions with pixel sampling.

Notes:

Unlike other scaling methods, this method does not introduce any additional color into the scaled image. For more details please check the ImageMagick documentation.
Sets the last exception property.

Scale(width as integer, height as integer) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Changes the size of an image to the given dimensions.

Example:

```
dim image as IMImageQ32MBS // your image
image=Image.Scale(100,80)
```

Notes:

This method was designed by Bob Friesenhahn as a low cost thumbnail generator.

columns: The number of columns in the scaled image.

rows: The number of rows in the scaled image.

Sets the last exception property.

For more details please check the ImageMagick documentation.

SetImageAttribute(key as string, value as string) as boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** SetImageAttribute searches the list of image attributes and replaces the attribute value.

Notes: If it is not found in the list, the attribute name and value is added to the list. If the attribute exists in the list, the value is concatenated to the attribute. SetImageAttribute returns True if the attribute is successfully concatenated or added to the list, otherwise False. If the value is "", the matching key is deleted from the list.

SetImageColorspace(Colorspace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the colorspace member of the Image structure.

Notes: Returns false on any error and true on success.

SetImageProfile(name as string, ProfileData as string) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a named profile to the image.

Notes:

If a profile with the same name already exists, it is replaced. This method differs from the ProfileImage() method in that it does not apply CMS color profiles.

name: The profile name.

profiledata: The binary data of the profile.

Returns false on any error and true on success.

SetPicture(pic as picture, x as integer, y as integer)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the pixels from a given Realbasic picture into the Image Magick Image at the given location.

Example:

```
dim image as IMImageQ32MBS // your image
dim p as picture
```

```
p=NewPicture(32,32,32)
p.Graphics.ForeColor=rgb(0,255,0)
p.Graphics.FillRect 0,0,32,32
```

```
image.SetPicture(p,30,30)
```

Notes:

Sets the last exception property.
 The method will do nothing on bad bounds.
 This method works only for bitmap images.
 x and y are zero based.

SetPictureMask(maskpic as picture, x as integer, y as integer)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the pixels from a given Realbasic picture into the mask of the Image Magick Image at the given location.

Example:

```
dim i as IMImageQ32MBS // your image
dim p as picture

p=NewPicture(32,32,32)
p.Graphics.ForeColor=rgb(0,255,0)
p.Graphics.FillRect 0,0,32,32

i.SetPictureMask(p,30,30)
```

Notes:

Sets the last exception property.
 The method will do nothing on bad bounds.
 This method works only for bitmap images.
 x and y are zero based.
 You may need to set matte=True after this.

SetPixel(x as integer, y as integer, newPixel as IMColorQ32MBS)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets a pixel value.

Example:

```
dim image as IMImageQ32MBS // your image
dim co as IMColorQ32MBS

co=new IMColorQ32MBS
co.blue=65535 // max value
image.SetPixel 50,50,co // Makes Pixel 50/50 blue
```

Notes:

The method will fail silently if the values are out of bounds or the image is not a bitmap image.
This method works only for bitmap images.
x and y are zero based.

Shade(gray as boolean, azimuth as double, elevation as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shines a distant light on an image to create a three-dimensional effect.

Notes:

You control the positioning of the light with azimuth and elevation; azimuth is measured in degrees off the x axis and elevation is measured in pixels above the Z axis.
Sets the last exception property.

For more details please check the ImageMagick documentation.

SharpenChannel(channel as integer, radius as double, sigma as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sharpens one or more image channels.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, radius should be larger than sigma. Use a radius of 0 and Sharpen selects a suitable radius for you.

channel: The channel type.

radius: The radius of the Gaussian, in pixels, not counting the center pixel.

sigma: The standard deviation of the Laplacian, in pixels.

Constants for channel:

Sets the last exception property.
For more details please check the ImageMagick documentation.

Shave(x as integer, y as integer, width as integer, height as integer) as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shave shaves pixels from the image edges.

Notes:

It allocates the memory necessary for the new Image structure and returns a pointer to the new image.
Returns nil on any error.
Sets the last exception property.

const UndefinedException	= 0
const WarningException	= 300
const ResourceLimitWarning	= 300
const TypeWarning	= 305
const OptionWarning	= 310
const DelegateWarning	= 315
const MissingDelegateWarning	= 320
const CorruptImageWarning	= 325
const FileOpenWarning	= 330
const BlobWarning	= 335
const StreamWarning	= 340
const CacheWarning	= 345
const CoderWarning	= 350
const ModuleWarning	= 355
const DrawWarning	= 360
const ImageWarning	= 365
const XServerWarning	= 380
const MonitorWarning	= 385
const RegistryWarning	= 390
const ConfigureWarning	= 395
const ErrorException	= 400
const ResourceLimitError	= 400
const TypeError	= 405
const OptionError	= 410
const DelegateError	= 415
const MissingDelegateError	= 420
const CorruptImageError	= 425
const FileOpenError	= 430
const BlobError	= 435
const StreamError	= 440
const CacheError	= 445
const CoderError	= 450
const ModuleError	= 455
const DrawError	= 460
const ImageError	= 465
const XServerError	= 480
const MonitorError	= 485
const RegistryError	= 490
const ConfigureError	= 495
const FatalErrorException	= 700
const ResourceLimitFatalError	= 700
const TypeFatalError	= 705
const OptionFatalError	= 710
const DelegateFatalError	= 715
const MissingDelegateFatalError	= 720
const CorruptImageFatalError	= 725
const FileOpenFatalError	= 730
const BlobFatalError	= 735
const StreamFatalError	= 740
const CacheFatalError	= 745
const CoderFatalError	= 750
const ModuleFatalError	= 755
const DrawFatalError	= 760
const ImageFatalError	= 765
const XServerFatalError	= 780
const MonitorFatalError	= 785
const RegistryFatalError	= 790
const ConfigureFatalError	= 795

const UndefinedException	= 0
const WarningException	= 300
const ResourceLimitWarning	= 300
const TypeWarning	= 305
const OptionWarning	= 310
const DelegateWarning	= 315
const MissingDelegateWarning	= 320
const CorruptImageWarning	= 325
const FileOpenWarning	= 330
const BlobWarning	= 335
const StreamWarning	= 340
const CacheWarning	= 345
const CoderWarning	= 350
const ModuleWarning	= 355
const DrawWarning	= 360
const ImageWarning	= 365
const XServerError	= 380
const MonitorWarning	= 385
const RegistryWarning	= 390
const ConfigureWarning	= 395
const ErrorException	= 400
const ResourceLimitError	= 400
const TypeError	= 405
const OptionError	= 410
const DelegateError	= 415
const MissingDelegateError	= 420
const CorruptImageError	= 425
const FileOpenError	= 430
const BlobError	= 435
const StreamError	= 440
const CacheError	= 445
const CoderError	= 450
const ModuleError	= 455
const DrawError	= 460
const ImageError	= 465
const XServerError	= 480
const MonitorError	= 485
const RegistryError	= 490
const ConfigureError	= 495
const FatalErrorException	= 700
const ResourceLimitFatalError	= 700
const TypeFatalError	= 705
const OptionFatalError	= 710
const DelegateFatalError	= 715
const MissingDelegateFatalError	= 720
const CorruptImageFatalError	= 725
const FileOpenFatalError	= 730
const BlobFatalError	= 735
const StreamFatalError	= 740
const CacheFatalError	= 745
const CoderFatalError	= 750
const ModuleFatalError	= 755
const DrawFatalError	= 760
const ImageFatalError	= 765
const XServerFatalError	= 780
const MonitorFatalError	= 785
const RegistryFatalError	= 790
const ConfigureFatalError	= 795

const UndefinedException	= 0
const WarningException	= 300
const ResourceLimitWarning	= 300
const TypeWarning	= 305
const OptionWarning	= 310
const DelegateWarning	= 315
const MissingDelegateWarning	= 320
const CorruptImageWarning	= 325
const FileOpenWarning	= 330
const BlobWarning	= 335
const StreamWarning	= 340
const CacheWarning	= 345
const CoderWarning	= 350
const ModuleWarning	= 355
const DrawWarning	= 360
const ImageWarning	= 365
const XServerWarning	= 380
const MonitorWarning	= 385
const RegistryWarning	= 390
const ConfigureWarning	= 395
const ErrorException	= 400
const ResourceLimitError	= 400
const TypeError	= 405
const OptionError	= 410
const DelegateError	= 415
const MissingDelegateError	= 420
const CorruptImageError	= 425
const FileOpenError	= 430
const BlobError	= 435
const StreamError	= 440
const CacheError	= 445
const CoderError	= 450
const ModuleError	= 455
const DrawError	= 460
const ImageError	= 465
const XServerError	= 480
const MonitorError	= 485
const RegistryError	= 490
const ConfigureError	= 495
const FatalErrorException	= 700
const ResourceLimitFatalError	= 700
const TypeFatalError	= 705
const OptionFatalError	= 710
const DelegateFatalError	= 715
const MissingDelegateFatalError	= 720
const CorruptImageFatalError	= 725
const FileOpenFatalError	= 730
const BlobFatalError	= 735
const StreamFatalError	= 740
const CacheFatalError	= 745
const CoderFatalError	= 750
const ModuleFatalError	= 755
const DrawFatalError	= 760
const ImageFatalError	= 765
const XServerFatalError	= 780
const MonitorFatalError	= 785
const RegistryFatalError	= 790
const ConfigureFatalError	= 795

```

UndefinedNoise      =0
UniformNoise        =1
GaussianNoise       =2
MultiplicativeGaussianNoise =3
ImpulseNoise        =4
LaplacianNoise      =5
PoissonNoise        =6

```

```

const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel      = & h0001
const CyanChannel      = & h0001
const GreenChannel     = & h0002
const MagentaChannel   = & h0002
const BlueChannel      = & h0004
const YellowChannel    = & h0004
const AlphaChannel     = & h0008
const OpacityChannel   = & h0008
const BlackChannel     = & h0020
const IndexChannel     = & h0020
const AllChannels      = & h7fffffff

```

```

const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel      = & h0001
const CyanChannel      = & h0001
const GreenChannel     = & h0002
const MagentaChannel   = & h0002
const BlueChannel      = & h0004
const YellowChannel    = & h0004
const AlphaChannel     = & h0008
const OpacityChannel   = & h0008
const BlackChannel     = & h0020
const IndexChannel     = & h0020
const AllChannels      = & h7fffffff

```

pathname: name of clipping path resource. If name is preceded by # , use clipping path numbered by name.

inside: if true, later operations take effect inside clipping path. Otherwise later operations take effect outside clipping path.

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel      = & h0001
const CyanChannel      = & h0001
const GreenChannel     = & h0002
const MagentaChannel   = & h0002
const BlueChannel      = & h0004
const YellowChannel    = & h0004
const AlphaChannel     = & h0008
const OpacityChannel   = & h0008
const BlackChannel     = & h0020
const IndexChannel     = & h0020
const AllChannels      = & h7ffffff
```

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel      = & h0001
const CyanChannel      = & h0001
const GreenChannel     = & h0002
const MagentaChannel   = & h0002
const BlueChannel      = & h0004
const YellowChannel    = & h0004
const AlphaChannel     = & h0008
const OpacityChannel   = & h0008
const BlackChannel     = & h0020
const IndexChannel     = & h0020
const AllChannels      = & h7ffffff
```

```
const PointFilter      =1
const BoxFilter        =2
const TriangleFilter   =3
const HermiteFilter    =4
const HanningFilter    =5
const HammingFilter    =6
const BlackmanFilter   =7
const GaussianFilter   =8
const QuadraticFilter  =9
const CubicFilter      =10
const CatromFilter     =11
const MitchellFilter   =12
const LanczosFilter    =13
const BesselFilter     =14
const SincFilter       =15
```

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

const UndefinedChannel	= 0
const RedChannel	= & h0001
const GrayChannel	= & h0001
const CyanChannel	= & h0001
const GreenChannel	= & h0002
const MagentaChannel	= & h0002
const BlueChannel	= & h0004
const YellowChannel	= & h0004
const AlphaChannel	= & h0008
const OpacityChannel	= & h0008
const BlackChannel	= & h0020
const IndexChannel	= & h0020
const AllChannels	= & h7ffffff

Shear(Xshear as double, Yshear as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method ShearImage creates a new image that is a shear_ image copy of an existing one.

Notes:

Shearing slides one edge of an image along the X or Y axis, creating a parallelogram. An X direction shear slides an edge along the X axis, while a Y direction shear slides an edge along the Y axis. The amount of the shear is controlled by a shear angle. For X direction shears, x_ shear is measured relative to the Y axis, and similarly, for Y direction shears y_ shear is measured relative to the X axis. Empty triangles left over from shearing the image are filled with the color defined by the pixel at location (0,0). ShearImage allocates the memory necessary for the new Image structure and returns a pointer to the new image.

Xshear and Yshear specify the number of degrees to shear the image.

Sets the last exception property.

For more details please check the ImageMagick documentation.

Solarize(factor as double) as boolean

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method SolarizeImage produces a 'solarization' effect seen when exposing a photographic film to light during the development process.

Notes:

factor: An double value that defines the extent of the solarization.

Returns nil on any error.

Sets the last exception property.

Splice(x as integer, y as integer, width as integer, height as integer) as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Splice splices a solid color into the image as defined by the geometry.

Notes:

Returns nil on any error.

Sets the last exception property.

Spread(radius as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This is a special effects method that randomly displaces each pixel in a block defined by the radius parameter.

Notes:

radius: Choose a random pixel in a neighborhood of this extent.
Sets the last exception property.

For more details please check the ImageMagick documentation.

Stegano(watermarkImage as IMImageQ32MBS) as IMImageQ32MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method SteganoImage hides a digital watermark within the image.

Notes:

Returns nil on any error.
Sets the last exception property.

Stereo(otherImage as IMImageQ32MBS) as IMImageQ32MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method StereoImage combines two images and produces a single image that is the composite of a left and right image of a stereo pair.

Notes:

The left image is converted to gray scale and written to the red channel of the stereo image. The right image is converted to gray scale and written to the blue channel of the stereo image. View the composite image with red-blue glasses to create a stereo effect.

left image = self
right image = otherImage parameter

Returns nil on any error.
Sets the last exception property.

Swirl(degrees as double) as IMImageQ32MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method SwirlImage creates a new image that is a copy of an existing one with the image pixels "swirl" at a specified angle.

Notes:

degrees: An double value that defines the tightness of the swirling.

Returns nil on any error.

Sets the last exception property.

Thumbnail(width as integer, height as integer) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Changes the size of an image to the given dimensions.

Notes:

Sets the last exception property.

This method was designed by Bob Friesenhahn as a low cost thumbnail generator.

For more details please check the ImageMagick documentation.

TransformImage(CropGeometry as string, ImageGeometry as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransformImage() is a convenience method that behaves like ResizeImage() or CropImage() but accepts scaling and/or cropping information as a region geometry specification. If the operation fails, the original image handle is left as is.

Notes:

This should only be used for single images.

CropGeometry: A crop geometry string. This geometry defines a subregion of the image to crop.

ImageGeometry: An image geometry string. This geometry defines the final size of the image.

Returns true on success.

TransformImages(CropGeometry as string, ImageGeometry as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransformImages() calls TransformImage() on each image of a sequence.

Notes:

TransformImage() is a convenience method that behaves like ResizeImage() or CropImage() but accepts scaling and/or cropping information as a region geometry specification. If the operation fails, the original image handle is left as is.

CropGeometry: A crop geometry string. This geometry defines a subregion of the image to crop.

ImageGeometry: An image geometry string. This geometry defines the final size of the image.

Returns true on success.

TransformRGBImage(Colorspace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method TransformRGBImage converts the reference image from an alternate colorspace.

Notes:

The transformation matrices are not the standard ones: the weights are rescaled to normalized the range of the transformed values to be [0..MaxRGB] .

colorspace: An integer value that indicates the colorspace the image is currently in. On return the image is in the RGB color space.

Returns false on any error and true on success.

constants:

TransposeImage as IMImageQ32MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransposeImage() creates a horizontal mirror image by reflecting the pixels around the central y-axis while rotating them by

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVCColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

90 degrees.

TransverseImage as IMImageQ32MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransverseImage() creates a vertical mirror image by reflecting the pixels around the central x-axis while rotating them by 270 degrees.

Trim as IMImageQ32MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Trim trims pixels from the image edges.

Notes:

It allocates the memory necessary for the new Image structure and returns a pointer to the new image.

Returns nil on any error.

Sets the last exception property.

UnsharpMaskChannel(channel as integer, radius as double, sigma as double, amount as double, threshold as double) as IMImageQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sharpens one or more image channels.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma). For reasonable results, radius should be larger than sigma. Use a radius of 0 and UnsharpMask selects a suitable radius for you.

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel     = & h0001
const CyanChannel     = & h0001
const GreenChannel    = & h0002
const MagentaChannel  = & h0002
const BlueChannel     = & h0004
const YellowChannel   = & h0004
const AlphaChannel    = & h0008
const OpacityChannel  = & h0008
const BlackChannel    = & h0020
const IndexChannel    = & h0020
const AllChannels     = & h7fffffff
```

Sets the last exception property.

For more details please check the ImageMagick documentation.

Wave(amplitude as double, wavelength as double) as IMImageQ32MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method Wave creates a new image that is a copy of an existing one with the image pixels altered along a sine wave.

Notes:

Parameters are double values that indicates the amplitude and wavelength of the sine wave.

Returns nil on any error.

Sets the last exception property.

WhiteThreshold(threshold as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** WhiteThreshold is like Threshold but forces all pixels above the threshold into white while leaving all pixels below the threshold unchanged.

Notes:

No exceptions are generated.

threshold: Define the threshold value. (ASCII string)

For more details please check the ImageMagick documentation.

WriteImage(info as IMImageInfoQ32MBS) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method WriteImage writes an image to a file as defined by image.filename.

Notes:

You can specify a particular image format by prefixing the file with the image type and a colon (i.e. ps:image) or specify the image type as the filename suffix (i.e. image.ps). The image may be modified to adapt it to the requirements of the image format. For example, DirectClass images must be color-reduced to PseudoClass if the format is GIF.

WriteImage returns True if the image is written. False is returned if there is a memory shortage or if the image file fails to write.

2.31.2 Properties

BackgroundColor as IMColorQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image background color.

Notes: (Read and Write property)

Bias as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

BlurFactor as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blur factor to apply to the image when zooming. Default is 1.0 (no blur).

Notes: (Read and Write property)

BorderColor as IMColorQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image border color.

Notes: (Read and Write property)

Colors as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The desired number of colors.

Notes:

Used by Quantize().

(Read and Write property)

ColorSpace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image pixel interpretation.

Notes:

If the colorspace is RGB the pixels are red, green, blue. If matte is true, then red, green, blue, and index. If it is CMYK, the pixels are cyan, yellow, magenta, black. Otherwise the colorspace is ignored.

constants:

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

(Read and Write property)

Compression as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image compression type.

Notes:

useful constants:

const UndefinedCompression	= 0
const NoCompression	= 1
const BZipCompression	= 2
const FaxCompression	= 3
const Group4Compression	= 4
const JPEGCompression	= 5
const LosslessJPEGCompression	= 6
const LZWCompression	= 7
const RLECompression	= 8
const ZipCompression	= 9

The default is the compression type of the specified image file.
For more details please check the ImageMagick documentation.

(Read and Write property)

Depth as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image depth (8 or 16).

Notes:

QuantumLeap must be defined before a depth of 16 is valid.

(Read and Write property)

Directory as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Tile names from within an image montage.

Notes:

Only valid after calling MontageImages() or reading a MIFF file which contains a directory.

(Read and Write property)

Endian as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The endian setting to use.

Notes:

constants:

UndefinedEndian	0	
LSBEndian	1	(Windows)
MSBEndian	2	(Mac)

e.g. tiff files support different endian settings.

(Read and Write property)

Filename as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The file path/name.
Notes:

The string must be in the encoding of the library and is limited to 4000 bytes.
For more details please check the ImageMagick documentation.
(Read and Write property)

Filter as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Filter to use when resizing image.
Notes:

Constants:

const PointFilter	=1
const BoxFilter	=2
const TriangleFilter	=3
const HermiteFilter	=4
const HanningFilter	=5
const HammingFilter	=6
const BlackmanFilter	=7
const GaussianFilter	=8
const QuadraticFilter	=9
const CubicFilter	=10
const CatromFilter	=11
const MitchellFilter	=12
const LanczosFilter	=13
const BesselFilter	=14
const SincFilter	=15

The reduction filter employed has a significant effect on the time required to resize an image and the resulting quality. The default filter is Lanczos which has been shown to produce high quality results when reducing most images.
(Read and Write property)

Fuzz as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Colors within this distance are considered equal.

Notes:

A number of algorithms search for a target color. By default the color must be exact. Use this to match colors that are close to the target color in RGB space.

(Read and Write property)

Gamma as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gamma level of the image.

Notes:

The same color image displayed on two different workstations may look different due to differences in the display monitor. Use gamma correction to adjust for this color difference.

(Read and Write property)

Geometry as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Preferred size of the image when encoding.

Notes: (Read and Write property)

Gravity as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to an Image structure.

For more details please check the ImageMagick documentation.

(Read and Write property)

Height as integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The height of the image in pixels.

Notes:

For more details please check the ImageMagick documentation.

(Read and Write property)

Interlace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The type of interlacing scheme (default NoInterlace).

Notes:

This option is used to specify the type of interlacing scheme for raw image formats such as RGB or YUV. NoInterlace means do not interlace, LineInterlace uses scanline interlacing, and PlaneInterlace uses plane interlacing. PartitionInterlace is like PlaneInterlace except the different planes are saved to individual files (e.g. image.R, image.G, and image.B). Use LineInterlace or PlaneInterlace to create an interlaced GIF or progressive JPEG image.

constants:

(Read and Write property)

UndefinedInterlace	0	Unset value.
NoInterlace	1	Don't interlace image (RBRGRBRGRBRGRBRGRB...)
LineInterlace	2	Use scanline interlacing (RRR...GGG...BBB...RRR...GGG...BBB...)
PlaneInterlace	3	Use plane interlacing (RRRRRR...GGGGG...BBBBBB...)
PartitionInterlace	4	Similar to plane interlacing except that the different planes are saved to individual files (e.g. image.R, image.G, and image.B)

LastError as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last error code reported.

Notes:

If an exception is raised and it is not a warning exception, this exception code is saved in this property.
(Read and Write property)

LastException as IMExceptionQ32MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last exception thrown by the Image Magick library.

Notes:

You should check this value after every call to the library, process the error and set the property to nil.

For more details please check the ImageMagick documentation.
(Read and Write property)

Magick as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image encoding format (e.g. "GIF").

Notes: (Read and Write property)

Matte as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether an alpha channel is used/present.

Notes:

Set to true to enable masks.
(Read and Write property)

MatteColor as IMColorQ32MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image matte (transparent) color.

Notes: (Read and Write property)

Montage as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Tile size and offset within an image montage. Only valid for montage images.

Notes: (Read and Write property)

Offset as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Number of initial bytes to skip over when reading raw image.

Notes: (Read and Write property)

Orientation as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The image orientation.

Notes:

constants:

```

const UndefinedOrientation    = 0
const TopLeftOrientation     = 1
const TopRightOrientation    = 2
const BottomRightOrientation = 3
const BottomLeftOrientation  = 4
const LeftTopOrientation     = 5
const RightTopOrientation    = 6
const RightBottomOrientation = 7
const LeftBottomOrientation  = 8

```

For more details please check the ImageMagick documentation.
(Read and Write property)

Quality as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** JPEG/MIFF/PNG compression level.

Example:

```
dim im as ImageMagickQ32MBS // global
```

```
Function TestJPEG(f as folderitem) As picture
```

```
// Reads an image, compresses in memory to JPEG, decompresses using JPEGlib and returns the image
```

```
// if quality setting works, you see it in the result.
```

```
// no error checking included!
```

```
// needs: im as ImageMagickQ32MBS ready initialized
```

```
dim image as IMImageQ32MBS
```

```
dim imageinfo as IMImageInfoQ32MBS
```

```
dim s,blob as string
```

```
dim p as Picture
```

```
dim i as integer
```

```
if f = nil then
```

```
Return nil
```

```
end if
```

```
imageinfo = im.NewImageInfo
```

```
# if TargetWin32 then //do not use shellpath, if spaces, IM doesn't like escaped paths
```

```
imageinfo.Filename = f.AbsolutePath
```

```
# else
```

```
imageinfo.Filename = f.UnixpathMBS
```

```

# endif

//this should read any image IM understands
image = im.ReadImage(imageinfo)
//check for error
if im.lastexception <>nil and im.LastException.Severity >= 400 then
s = "LastError: " + Format(im.LastError,"-0") + " - Severity: " + str(im.LastException.Severity) + EndOfLine + im.LastException.Message
MsgBox s
Return nil
elseif image = nil then
MsgBox "image=nil"
Return nil
end if

// Now lets convert to jpeg
imageinfo.Filename = "image.jpg"
imageinfo.Quality = 10 // 100 is max
blob = image.ImageToBlob(imageinfo)

// It may fail
if blob.lenb = 0 then
Return nil
end if
p = JPEGStringToPictureMBS(blob,true)

image.DestroyImage
imageinfo.DestroyImageInfo

Return p
Exception
Return nil
End Function

```

Notes:

Default value is 75.
(Read and Write property)

Release as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** If true, the destructor will release the handle.

Notes: (Read and Write property)

RenderingIntent as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The rendering intent to use.

Notes:

constants:

UndefinedIntent	0
SaturationIntent	1
PerceptualIntent	2
AbsoluteIntent	3
RelativeIntent	4

(Read and Write property)

ResolutionUnits as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Units of image resolution.

Notes:

constants:

UndefinedResolution	0	Unset value.
PixelsPerInchResolution	1	Density specifications are specified in units of pixels per inch (english units).
PixelsPerCentimeterResolution	2	Density specifications are specified in units of pixels per centimeter (metric units).

(Read and Write property)

ResolutionX as double

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The horizontal resolution of the image.

Notes:

The unit for resolution must be specified.
 For more details please check the ImageMagick documentation.
 (Read and Write property)

ResolutionY as double

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The vertical resolution of the image.

Notes:

The unit for resolution must be specified.
 For more details please check the ImageMagick documentation.
 (Read and Write property)

Scene as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

StorageClass as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image storage class.

Notes:

If DirectClass then the image packets contain valid RGB or CMYK colors. If PseudoClass then the image has a colormap referenced by pixel's index member.

constants:

UndefinedClass	0	Unset value.
DirectClass	1	Image is composed of pixels which represent literal color values.
PseudoClass	2	Image is composed of pixels which specify an index in a color palette.

(Read and Write property)

Taint as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Set to True if the image pixels have been modified.

Notes: (Read and Write property)

Width as integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The width of the image in pixels.

Notes:

For more details please check the ImageMagick documentation.

(Read and Write property)

2.31.3 Constants

kBackgroundDispose = 2

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kCoalesceLayer = 1

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompareAnyLayer = 2

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompareClearLayer = 3

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompareOverlayLayer = 4

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompositeLayer = & h0000000C

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kDisposeLayer = 5

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kFlattenLayer = & h0000000E

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kMergeLayer = & h0000000D

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kMosaicLayer = & h0000000F

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kNoneDispose = 1

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kOptimizeImageLayer = 7

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kOptimizeLayer = 6

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kOptimizePlusLayer = 8

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kOptimizeTransLayer = 9

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kPreviousDispose = 3

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kRemoveDupsLayer = & h0000000A

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kRemoveZeroLayer = & h0000000B

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kUndefinedDispose = 0

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kUndefinedLayer = 0

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kUnrecognizedDispose = 0

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

2.32 class IMImageQ16MBS

class IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for an Image Magick Image in memory.

Notes:

Can exist with or without pixel data.

For more details please check the ImageMagick documentation.

2.32.1 Methods

AdaptiveThreshold(width as integer, height as integer, offset as integer) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** AdaptiveThreshold selects an individual threshold for each pixel based on the range of intensity values in its local neighborhood.

Notes:

This allows for thresholding of an image whose global intensity histogram doesn't contain distinctive peaks. Sets the last exception property.

width: The width of the local neighborhood.

height: The height of the local neighborhood.

offset: The mean offset.

For more details please check the ImageMagick documentation.

AddNoise(NoiseType as integer) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds random noise to the image.

Notes:

Constants

UndefinedNoise	=0
UniformNoise	=1
GaussianNoise	=2
MultiplicativeGaussianNoise	=3
ImpulseNoise	=4
LaplacianNoise	=5
PoissonNoise	=6

For more details please check the ImageMagick documentation.
Sets the last exception property.

AffineTransformImage(matrix as IMImageAffineMatrixQ16MBS) as IMImageQ16MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Transforms an image as dictated by the affine matrix.

AppendImageToList(img as IMImageQ16MBS)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds an image to the image list.

Notes: For more details please check the ImageMagick documentation.

Average as IMImageQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Average() method takes a set of images and averages them together.

Notes:

Each image in the set must have the same width and height. Average() returns a single image with each

corresponding pixel component of each image averaged. On failure, a nil image is returned and exception describes the reason for the failure.

Sets the last exception property.

For more details please check the ImageMagick documentation.

BilevelChannel(channel as integer, threshold as double) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Changes the value of individual pixels based on the intensity of each pixel channel.

Notes:

The result is a high-contrast image.

channel: The channel type.

threshold: define the threshold values.

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel      = & h0001
const CyanChannel      = & h0001
const GreenChannel     = & h0002
const MagentaChannel   = & h0002
const BlueChannel      = & h0004
const YellowChannel    = & h0004
const AlphaChannel     = & h0008
const OpacityChannel   = & h0008
const BlackChannel     = & h0020
const IndexChannel     = & h0020
const AllChannels      = & h7ffffff
```

For more details please check the ImageMagick documentation.

BlackThreshold(threshold as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** BlackThreshold is like Threshold but forces all pixels below the threshold into black while leaving all pixels above the threshold unchanged.

Notes:

No exceptions are generated.

threshold: Define the threshold value. (ASCII string)

For more details please check the ImageMagick documentation.

BlobSize as integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The expected size for this image written to a file.

Notes: For more details please check the ImageMagick documentation.

Blur(radius as double, sigma as double) as IMImageQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blurs an image.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, the radius should be larger than sigma. Use a radius of 0 and BlurImage selects a suitable radius for you.

radius: The radius of the Gaussian, in pixels, not counting the center pixel.

sigma: The standard deviation of the Gaussian, in pixels.

For more details please check the ImageMagick documentation.

BlurImageChannel(channel as integer, radius as double, sigma as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blurs an image.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, the radius should be larger than sigma. Use a radius of 0 and BlurImageChannel selects a suitable radius for you.

channel: The channel type.

radius: The radius of the Gaussian, in pixels, not counting the center pixel.

sigma: The standard deviation of the Gaussian, in pixels.

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel     = & h0001
const CyanChannel     = & h0001
const GreenChannel    = & h0002
const MagentaChannel  = & h0002
const BlueChannel     = & h0004
const YellowChannel   = & h0004
const AlphaChannel    = & h0008
const OpacityChannel  = & h0008
const BlackChannel    = & h0020
const IndexChannel    = & h0020
const AllChannels     = & h7ffffff
```

For more details please check the ImageMagick documentation.

BorderImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ16MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Surrounds the image with a border of the color defined by the bordercolor member of the image.

Notes: The width and height of the border are defined by the corresponding parameters.

Charcoal(radius as double, sigma as double) as IMImageQ16MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Charcoal creates a new image that is a copy of an existing one with the edge highlighted.

Notes:

radius: the radius of the pixel neighborhood.

sigma: The standard deviation of the Gaussian, in pixels.

Returns nil on any error.
Sets the last exception property.

Chop(x as integer, y as integer, width as integer, height as integer) as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Chop removes a region of an image and collapses the image to occupy the removed portion.

Notes:

Returns nil on any error.
Sets the last exception property.

ClipPath(path as string, inside as boolean) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the image clip mask based any clipping path information if it exists.

Notes:

- pathname: name of clipping path resource. If name is preceded by # , use clipping path numbered by name.
- inside: if true, later operations take effect inside clipping path. Otherwise later operations take effect outside clipping path.

Returns true on success and false on any error.

Clone as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a copy of this image object.

Notes: For more details please check the ImageMagick documentation.

CloneImageAttributes(image as IImageAttributeQ16MBS) as Boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** CloneImageAttributes() clones one or more image attributes.

Notes: Returns false on any error.

CloneImageProfiles(SourceImage as IImageQ16MBS) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clones one or more image profiles.

Notes: Returns false on any error and true on success.

Close

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.

Notes:

There is no need to call this method except you want to free all resources used by this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

CoalesceImages as IImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** CoalesceImages composites a set of images while respecting any page offsets and disposal methods.

Notes:

GIF, MIFF, and MNG animation sequences typically start with an image background and each subsequent image varies in size and offset. CoalesceImages() returns a new sequence where each image in the sequence is the same size as the first and composited with the next image in the sequence.

Returns nil on any error.

Sets the last exception property.

Colorize(opacity as string, PenColorRed as integer, PenColorGreen as integer, PenColorBlue as integer, PenColorOpacity as integer) as IMImageQ16MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method ColorizeImage creates a new image that is a copy of an existing one with the image pixels colorized.

Notes:

The colorization is controlled with the pen color and the opacity levels.

opacity: A character string indicating the level of opacity as a percentage (0-100).

PenColorRed, PenColorGreen, PenColorBlue and PenColorOpacity define the pen color used.

Returns nil on any error.

Sets the last exception property.

Combine(channel as integer) as IMImageQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Combines one or more images into a single image.

Notes:

The grayscale value of the pixels of each image in the sequence is assigned in order to the specified channels of the combined image. The typical ordering would be image 1 =>Red, 2 =>Green, 3 =>Blue, etc.

The lastexception property is set.

CompareImageLayers(ImageLayerMethod as integer) as IMImageQ16MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** CompareImageLayers() compares each image with the next in a sequence and returns the minimum bounding region of all the pixel differences (of the mageLayerMethod specified) it discovers.

Notes:

Images do NOT have to be the same size, though it is best that all the images are 'coalesced' (images are all the same size, on a flattened canvas, so as to represent exactly how an specific frame should look).

No GIF dispose methods are applied, so GIF animations must be coalesced before applying this image operator to find differences to them.

ImageLayerMethod:

the layers type to compare images with. Must be one of... CompareAnyLayer, CompareClearLayer, CompareOverlayLayer.

Can raise an exception.

Composite(ComposeOperator as integer, Image as IMImageQ16MBS, x as integer, y as integer)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the second image composited onto the first at the specified offsets.

Notes:

compose: Specifies an image composite operator.

Image: The second image.

x: An integer that specifies the column offset of the composited image.

y: An integer that specifies the row offset of the composited image.

No error code and exception!

ConsolidateCMYKImages as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Consolidates a sequence of CMYK images.

Notes:

Returns nil on any error.

Sets the last exception property.

CopyPicture as picture

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IMImageQ16MBS // your image
Canvas1.Backdrop=image.CopyPicture
```

Notes:

Sets the last exception property.

Returns nil on any error.

This method works only for bitmap images.

See also:

- 2.32.1 CopyPicture(x as integer, y as integer, width as integer, height as integer) as picture 270

CopyPicture(x as integer, y as integer, width as integer, height as integer) as picture

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies a portion of the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IMImageQ16MBS // your image
Canvas1.Backdrop=image.CopyPicture(0,0,image.Width,image.Height)
```

Notes:

Sets the last exception property.

Returns nil on any error.

This method works only for bitmap images.

x and y are zero based.

See also:

- 2.32.1 CopyPicture as picture 270

CopyPictureMask as picture

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the mask of the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IMImageQ16MBS // your image
Canvas1.Backdrop=image.CopyPictureMask
```

Notes:

Sets the last exception property.

Returns nil on any error.

This method works only for bitmap images.

See also:

- 2.32.1 CopyPictureMask(x as integer, y as integer, width as integer, height as integer) as picture 271

CopyPictureMask(x as integer, y as integer, width as integer, height as integer) as picture

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies a portion of the mask of the Image Magick Image and returns a Realbasic picture.

Example:

```
dim image as IMImageQ16MBS // your image
Canvas1.Backdrop=image.CopyPictureMask(0,0,image.Width,image.Height)
```

Notes:

Sets the last exception property.

Returns nil on any error.

This method works only for bitmap images.

x and y are zero based.

See also:

- 2.32.1 CopyPictureMask as picture

CopyPixel(x as integer, y as integer) as IMColorQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies a pixel.

Notes:

Returns nil on any error.

This method works only for bitmap images.

x and y are zero based.

Crop(x as integer, y as integer, width as integer, height as integer) as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Crop extracts a region of the image starting at the offset defined by geometry.

Notes:

Returns nil on any error.

Sets the last exception property.

CropImageToTiles(CropGeometry as string) as IMImageQ16MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Crops a single image, into a possible list of tiles.

Notes: This may include a single sub-region of the image. This basically applies all the normal geometry flags for Crop.

CycleColormap(displace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Displaces an image's colormap by a given number of positions.

Notes:

If you cycle the colormap a number of times you can produce a psychedelic effect.

Returns true on success.

displace: displace the colormap this amount.

DecipherImage(passkey as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts cipher pixels to plain pixels.

Notes:

Passkey: decipher cipher pixels with this passphrase.
Returns true on success.

DeconstructImages as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** DeconstructImages() compares each image with the next in a sequence and returns the minimum bounding region of all differences from the first image.

Notes:

Returns nil on any error.
Sets the last exception property.

DeleteImageAttribute(key as string) as Boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** DeleteImageAttribute() deletes an attribute from the image.

Notes: Returns false on any error.

Despeckle() as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reduces the speckle noise in an image while perserving the edges of the original image.

Notes:

Sets the last exception property.
For more details please check the ImageMagick documentation.

DestroyImage

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Releases the memory used for this image and sets handle to 0.

Notes:

For more details please check the ImageMagick documentation.
The destructor will call this for you if release=true.

DestroyImageAttributes

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Deallocates memory associated with the image attribute list.

DestroyImageList

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Destroys the image list and sets the handle to 0.

Notes:

For more details please check the ImageMagick documentation.
The destructor will call this for you if release=true.

DestroyImageProfiles

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Releases memory associated with an image profile map.

Edge(radius as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Finds edges in an image.

Notes:

Radius defines the radius of the convolution filter. Use a radius of 0 and Edge selects a suitable radius for you.

Sets the last exception property.

For more details please check the ImageMagick documentation.

Emboss(radius as double, sigma as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a grayscale image with a three-dimensional effect.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, radius should be larger than sigma. Use a radius of 0 and Emboss selects a suitable radius for you.

Sets the last exception property.

For more details please check the ImageMagick documentation.

EncipherImage(passkey as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Converts pixels to cipher-pixels.

Notes:

passkey: encipher pixels with this passphrase.

Returns true on success.

ExcerptImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ16MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a excerpt of the image as defined by the geometry.

Notes: Define the region of the image to extend with x, y, width, and height.

ExtentImage(x as integer, y as integer, width as integer, height as integer) as IMImageQ16MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Extends the image as defined by the geometry, gravity, and image background color.

Notes:

Define the region of the image to extend with x, y, width, and height.

Set the (x,y) offset of the geometry to move the original image relative to the extended image.

FlattenImages as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Flatten composites all images from the current image pointer to the end of the image list and returns a single flattened image.

Notes:

Returns nil on any error.

Sets the last exception property.

Flip as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Flip creates a vertical mirror image by reflecting the pixels around the central x-axis.

Notes:

Returns nil on any error.

Sets the last exception property.

Flop as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Flop creates a horizontal mirror image by reflecting the pixels around the central y-axis.

Notes:

Returns nil on any error.

Sets the last exception property.

FrameImage(x as integer, y as integer, width as integer, height as integer, innerBevel as integer, OuterBevel as integer) as IMImageQ16MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a simulated three-dimensional border around the image.

Notes: The color of the border is defined by the MatteColor of image. Width and height specify the border width of the vertical and horizontal sides of the frame. innerBevel and OuterBevel indicate the width of the inner and outer shadows of the frame.

FxImage(expression as string) as IMImageQ16MBS

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** FxImage() applies a mathematical expression to the specified image.

Notes: Can raise an exception.

GaussianBlurChannel(channel as integer, radius as double, sigma as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blurs an image.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, the radius should be larger than sigma. Use a radius of 0 and GaussianBlur selects a suitable radius for you.

Sets the last exception property.

radius: the radius of the Gaussian, in pixels, not counting the center pixel.

channel: The channel type.

sigma: the standard deviation of the Gaussian, in pixels.

Constants for channel:

For more details please check the ImageMagick documentation.

```

const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel      = & h0001
const CyanChannel      = & h0001
const GreenChannel     = & h0002
const MagentaChannel   = & h0002
const BlueChannel      = & h0004
const YellowChannel    = & h0004
const AlphaChannel     = & h0008
const OpacityChannel   = & h0008
const BlackChannel     = & h0020
const IndexChannel     = & h0020
const AllChannels      = & h7fffffff

```

GetImageAttribute(key as string) as IMImageAttributeQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** GetImageAttribute searches the list of image attributes and returns a reference to the attribute if it exists otherwise nil.

GetImageClippingPathAttribute as IMImageAttributeQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** GetImageClippingPathAttribute searches the list of image attributes and returns a reference to a clipping path if it exists otherwise nil.

GetImageProfile(name as string) as string

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets a profile associated with an image by name.

Notes: Returns "" on any error.

GetNextImageAttribute as IMImageAttributeQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** GetNextImageAttribute() gets the next image attribute.

Notes: Returns nil on any error.

GetNextImageProfile as string

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gets the next profile name for an image.

Notes: Returns "" on any error.

HandleMemory as memoryblock

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The content of the whole Image structure copied into a memoryblock.

Notes: Returns nil on any error.

ImagesToBlob(info as IMImageInfoQ16MBS) as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** ImagesToBlob implements direct to memory image formats.

Notes:

It returns the image sequence as a string. The magick member of the ImageInfo structure determines the format of the returned blob (GIF, JPEG, PNG, etc.)

Note, some image formats do not permit multiple images to the same image stream (e.g. JPEG). in this instance, just the first image of the sequence is returned as a blob.

Sets the last exception property and returns "" on any error.

For more details please check the ImageMagick documentation.

ImageToBlob(info as IMImageInfoQ16MBS) as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** ImagesToBlob implements direct to memory image formats.

Example:

```
dim im as ImageMagickQ16MBS // global
```

```
Function IMPictureToString(p as picture, magick as string, quality as integer) As string
dim image as new IMImageQ16MBS
dim imageinfo as IMImageInfoQ16MBS
dim s,data as string
dim impp as new IMMagickPixelPacketQ16MBS

// empty string for nil picture
if p = nil then
Return ""
end if

// create a new picture info

imageinfo = im.NewImageInfo
imageinfo.ColorSpace=1
// only color space is needed. 1 for RGB.

// background color of image
impp.red = 0
impp.Green = 0
impp.Blue = 0

// creates a new image object
if not image.NewImage(imageinfo,p.Width,p.Height,impp) then
Return ""
end if

// copy RB picture into IM Image at position 0/0
image.ColorSpace = 1
image.SetPicture(p,0,0)

// set compression data
imageinfo.Magick = magick
imageinfo.Quality = quality

// and rendering intent: 2=PerceptualIntent
image.RenderingIntent = 2

// create image data
data = image.ImageToBlob(imageinfo)

// release memory
image.DestroyImage
imageinfo.DestroyImageInfo

// return result
Return data
```

```

Exception
// in case of an exception return nothing
Return ""

```

End Function

Notes:

It returns the image sequence as a string. The magick member of the ImageInfo structure determines the format of the returned blob (GIF, JPEG, PNG, etc.)

Note, some image formats do not permit multiple images to the same image stream (e.g. JPEG). in this instance, just the first image of the sequence is returned as a blob.

Sets the last exception property and returns "" on any error.
For more details please check the ImageMagick documentation.

Implode(factor as double) as IMImageQ16MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method ImplodeImage creates a new image that is a copy of an existing one with the image pixels "implode" by the specified percentage.

Notes:

factor: A double value that defines the extent of the implosion.

Returns nil on any error.
Sets the last exception property.

IsBlobExempt as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the blob is exempt.

Notes: For more details please check the ImageMagick documentation.

IsBlobSeekable as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the blob is seekable.

Notes: For more details please check the ImageMagick documentation.

IsBlobTemporary as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns true if the blob is temporary.

Notes: For more details please check the ImageMagick documentation.

Magnify as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A convenience method that scales an image proportionally to twice its size.

Notes:

Sets the last exception property.

For more details please check the ImageMagick documentation.

MedianFilter(radius as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Applies a digital filter that improves the quality of a noisy image.

Notes:

Each pixel is replaced by the median in a set of neighboring pixels as defined by radius.

Sets the last exception property.

For more details please check the ImageMagick documentation.

MergeImageLayers(ImageLayerMethod as integer) as IMImageQ16MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** MergeImageLayers() composes all the image layers from the current given image onward to produce a single image of the merged layers.

Notes:

The initial canvas's size depends on the given ImageLayerMethod, and is initialized using the first images images background color. The images are then composited onto that image in sequence using the given composition that has been assigned to each individual image.

ImageLayerMethod:

the method of selecting the size of the initial canvas.

MergeLayer: Merge all layers onto a canvas just large enough to hold all the actual images. The virtual canvas of the first image is preserved but otherwise ignored.

FlattenLayer: Use the virtual canvas size of first image. Images which fall outside this canvas is clipped. This can be used to 'fill out' a given virtual canvas.

MosaicLayer: Start with the virtual canvas of the first image, enlarging left and right edges to contain all images. Images with negative offsets will be clipped.

Can raise an exception.

Minify as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A convenience method that scales an image proportionally to half its size.

Notes:

Sets the last exception property.

For more details please check the ImageMagick documentation.

MosaicImages as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** MosaicImages inlays an image sequence to form a single coherent picture.

Notes:

It returns a single image with each image in the sequence composited at the location defined by the page member of the image structure.

Returns nil on any error.

Sets the last exception property.

MotionBlur(radius as double, sigma as double, angle as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Simulates motion blur.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma). For reasonable results, radius should be larger than sigma. Use a radius of 0 and MotionBlur selects a suitable radius for you. Angle gives the angle of the blurring motion.

Sets the last exception property.

For more details please check the ImageMagick documentation.

NewImage(info as IMImageInfoQ16MBS, width as integer, height as integer, background as IMMackPixelPacketQ16MBS) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new image.

Example:

```
dim im as ImageMagickQ16MBS // global
dim p as picture
dim imageinfo as IMImageInfoQ16MBS
dim image as IMImageQ16MBS
dim b as new IMMackPixelPacketQ16MBS
b.Blue=65535
b.ColorSpace=1 // RGB
b.Depth=16
```

```

imageinfo = im.NewImageInfo
imageinfo.Depth=16
imageinfo.ColorSpace=1

//this should read any image IM understands
image = new IMImageQ16MBS
if image.NewImage(imageinfo,500,500,b) then
p=NewPicture(300,300,32)
p.Graphics.ForeColor=Rgb(255,0,0)
p.Graphics.FillOval 0,0,300,300
image.SetPicture p,0,0
else
MsgBox "failed"
end if

```

Notes: Returns false on failure and true on success.

OilPaint(radius as double) as IMImageQ16MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method OilPaintImage creates a new image that is a copy of an existing one with each pixel component replaced with the color of greatest frequency in a circular neighborhood.

Notes:

radius parameter: radius of the circular neighborhood.

Returns nil on any error.

Sets the last exception property.

OptimizeImageLayers as IMImageQ16MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** OptimizeImageLayers() compares each image the GIF disposed forms of the previous image in the sequence.

Notes:

From this it attempts to select the smallest cropped image to replace each frame, while preserving the results of the GIF animation.

Can raise an exception.

OptimizeImageTransparency

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** `OptimizeImageTransparency()` takes a frame optimized GIF animation, and compares the overlaid pixels against the disposal image resulting from all the previous frames in the animation.

Notes:

Any pixel that does not change the disposal image (and thus does not effect the outcome of an overlay) is made transparent.

WARNING: This modifies the current images directly, rather than generate a new image sequence.

Can raise an exception.

OptimizePlusImageLayers as IImageQ16MBS

Plugin Version: 8.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** `OptimizeImagePlusLayers()` is exactly as `OptimizeImageLayers()`, but may also add or even remove extra frames in the animation, if it improves the total number of pixels in the resulting GIF animation.

Notes: Can raise an exception.

ProfileImage(name as string, ProfileData as string) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds or removes a ICC, IPTC, or generic profile from an image.

Notes:

If the ProfileData is "", it is removed from the image otherwise added. Use a name of '*' and a ProfileData of "" to remove all profiles from the image.

Returns false on any error and true on success.

RadialBlur(angle as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** RadialBlur applies a radial blur to the image.

Notes:

angle: The angle of the radial blur.

Sets the last exception property.

For more details please check the ImageMagick documentation.

RaiseImage(x as integer, y as integer, width as integer, height as integer, raise as boolean) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a simulated three-dimensional button-like effect by lightening and darkening the edges of the image.

Notes:

Width and height define the width of the vertical and horizontal edge of the effect.

raise: A value other than zero creates a 3-D raise effect, otherwise it has a lowered effect.

RandomThresholdChannel(channel as integer, thresholds as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Changes the value of individual pixels based on the intensity of each pixel compared to a random threshold.

Notes:

The result is a low-contrast, two color image.

channel: The channel or channels to be thresholded.

thresholds: a geometry string containing low,high thresholds. If the string contains 2x2, 3x3, or 4x4, an ordered dither of order 2, 3, or 4 is performed instead. (ASCII string)

Sets the last exception property.

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel     = & h0001
const CyanChannel     = & h0001
const GreenChannel    = & h0002
const MagentaChannel  = & h0002
const BlueChannel     = & h0004
const YellowChannel   = & h0004
const AlphaChannel    = & h0008
const OpacityChannel  = & h0008
const BlackChannel    = & h0020
const IndexChannel    = & h0020
const AllChannels     = & h7ffffff
```

For more details please check the ImageMagick documentation.

ReduceNoise(radius as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Smooths the contours of an image while still preserving edge information.

Notes:

The algorithm works by replacing each pixel with its neighbor closest in value. A neighbor is defined by radius. Use a radius of 0 and ReduceNoise selects a suitable radius for you.

For more details please check the ImageMagick documentation.

RemoveDuplicateLayers

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes any image that is exactly the same as the next image in the given image list.

Notes:

Image size and virtual canvas offset must also match, though not the virtual canvas size itself.

No check is made with regards to image disposal setting, though it is the dispose setting of later image that is kept. Also any time delays are also added together. As such coalesced image animations should still

produce the same result, though with duplicate frames merged into a single frame.

RemoveFirstImageFromList as IMImageQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes the first image from the image list and returns the image.

Notes:

Returns nil on any error.

For more details please check the ImageMagick documentation.

RemoveImageProfile(name as string) as string

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes a profile from the image-map by its name.

RemoveZeroDelayLayers

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Removes any image that as a zero delay time.

Notes:

Such images generally represent intermediate or partial updates in GIF animations used for file optimization. They are not ment to be displayed to users of the animation. Viewable images in an animation should have a time delay of 3 or more centi-seconds (hundredths of a second).

However if all the frames have a zero time delay, then either the animation is as yet incomplete, or it is not a GIF animation. This is a non-sensible situation, so no image will be removed and a 'Zero Time Animation' warning (exception) given.

No warning will be given if no image was removed because all images had an appropriate non-zero time delay set.

Due to the special requirements of GIF disposal handling, GIF animations should be coalesced first, before calling this function, though that is not a requirement.

ResetImageAttributeIterator

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** ResetImageAttributeIterator() resets the image attributes iterator.

Notes: Use it in conjunction with GetNextImageAttribute() to iterate over all the values associated with an image.

ResetImageProfileIterator

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Resets the image profile iterator.

Notes: Use it in conjunction with GetNextImageProfile() to iterate over all the profiles associated with an image.

Resize(width as integer, height as integer, FilterID as integer, blur as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Scales an image to the desired dimensions.

Notes:

Constants for the FilterID:

Most of the filters are FIR (finite impulse response), however, Bessel, Gaussian, and Sinc are IIR (infinite impulse response). Bessel and Sinc are windowed (brought down to zero) with the Blackman filter.

Sets the last exception property.

RGBTransformImage(Colorspace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method RGBTransformImage converts the reference image from RGB to an alternate colorspace.

Notes:

```

const PointFilter      =1
const BoxFilter        =2
const TriangleFilter   =3
const HermiteFilter    =4
const HanningFilter    =5
const HammingFilter    =6
const BlackmanFilter   =7
const GaussianFilter   =8
const QuadraticFilter  =9
const CubicFilter      =10
const CatromFilter     =11
const MitchellFilter   =12
const LanczosFilter    =13
const BesselFilter     =14
const SincFilter       =15

```

The transformation matrices are not the standard ones: the weights are rescaled to normalized the range of the transformed values to be [0..MaxRGB] .

colorspace: An integer value that indicates which colorspace to transform the image.

Returns false on any error and true on success.

constants:

```

UndefinedColorspace   0
RGBColorspace         1
GRAYColorspace        2
TransparentColorspace 3
OHTAColorspace        4
LABColorspace         5
XYZColorspace         6
YCbCrColorspace       7
YCCColorspace         8
YIQColorspace         9
YPbPrColorspace      10
YUVColorspace        11
CMYKColorspace       12
sRGBColorspace       13
HSBColorspace        14
HSLColorspace        15
HWBColorspace        16

```

Roll(x as integer, y as integer) as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Roll offsets an image as defined by x and y.

Notes:

Returns nil on any error.
Sets the last exception property.

Rotate(degrees as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Rotation of an image.

Notes:

Method RotateImage creates a new image that is a rotated copy of an existing one. Positive angles rotate counter-clockwise (right-hand rule), while negative angles rotate clockwise. Rotated images are usually larger than the originals and have 'empty' triangular corners. X axis. Empty triangles left over from shearing the image are filled with the color specified by the image background_ color. RotateImage allocates the memory necessary for the new Image structure and returns a pointer to the new image.

Method RotateImage is based on the paper "A Fast Algorithm for General Raster Rotation" by Alan W. Paeth. RotateImage is adapted from a similar method based on the Paeth paper written by Michael Halle of the Spatial Imaging Group, MIT Media Lab.

degrees: Specifies the number of degrees to rotate the image.

Sets the lastexception property.
Returns nil on low memory.
For more details please check the ImageMagick documentation.

Sample(width as integer, height as integer) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Scales an image to the desired dimensions with pixel sampling.

Notes:

Unlike other scaling methods, this method does not introduce any additional color into the scaled image. For more details please check the ImageMagick documentation.
Sets the last exception property.

Scale(width as integer, height as integer) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Changes the size of an image to the given dimensions.

Example:

```
dim image as IMImageQ16MBS // your image
image=Image.Scale(100,80)
```

Notes:

This method was designed by Bob Friesenhahn as a low cost thumbnail generator.

columns: The number of columns in the scaled image.

rows: The number of rows in the scaled image.

Sets the last exception property.

For more details please check the ImageMagick documentation.

SetImageAttribute(key as string, value as string) as boolean

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** SetImageAttribute searches the list of image attributes and replaces the attribute value.

Notes: If it is not found in the list, the attribute name and value is added to the list. If the attribute exists in the list, the value is concatenated to the attribute. SetImageAttribute returns True if the attribute is successfully concatenated or added to the list, otherwise False. If the value is "", the matching key is deleted from the list.

SetImageColorspace(Colorspace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the colorspace member of the Image structure.

Notes: Returns false on any error and true on success.

SetImageProfile(name as string, ProfileData as string) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a named profile to the image.

Notes:

If a profile with the same name already exists, it is replaced. This method differs from the ProfileImage() method in that it does not apply CMS color profiles.

name: The profile name.

profiledata: The binary data of the profile.

Returns false on any error and true on success.

SetPicture(pic as picture, x as integer, y as integer)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the pixels from a given Realbasic picture into the Image Magick Image at the given location.

Example:

```
dim image as IMImageQ16MBS // your image
dim p as picture
```

```
p=NewPicture(32,32,32)
p.Graphics.ForeColor=rgb(0,255,0)
p.Graphics.FillRect 0,0,32,32
```

```
image.SetPicture(p,30,30)
```

Notes:

Sets the last exception property.
 The method will do nothing on bad bounds.
 This method works only for bitmap images.
 x and y are zero based.

SetPictureMask(maskpic as picture, x as integer, y as integer)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Copies the pixels from a given Realbasic picture into the mask of the Image Magick Image at the given location.

Example:

```
dim i as IMImageQ16MBS // your image
dim p as picture

p=NewPicture(32,32,32)
p.Graphics.ForeColor=rgb(0,255,0)
p.Graphics.FillRect 0,0,32,32

i.SetPictureMask(p,30,30)
```

Notes:

Sets the last exception property.
 The method will do nothing on bad bounds.
 This method works only for bitmap images.
 x and y are zero based.
 You may need to set matte=True after this.

SetPixel(x as integer, y as integer, newPixel as IMColorQ16MBS)

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets a pixel value.

Example:

```
dim image as IMImageQ16MBS // your image
dim co as IMColorQ16MBS

co=new IMColorQ16MBS
co.blue=65535 // max value
image.SetPixel 50,50,co // Makes Pixel 50/50 blue
```

Notes:

The method will fail silently if the values are out of bounds or the image is not a bitmap image.
This method works only for bitmap images.
x and y are zero based.

Shade(gray as boolean, azimuth as double, elevation as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shines a distant light on an image to create a three-dimensional effect.

Notes:

You control the positioning of the light with azimuth and elevation; azimuth is measured in degrees off the x axis and elevation is measured in pixels above the Z axis.
Sets the last exception property.

For more details please check the ImageMagick documentation.

SharpenChannel(channel as integer, radius as double, sigma as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sharpens one or more image channels.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma) . For reasonable results, radius should be larger than sigma. Use a radius of 0 and Sharpen selects a suitable radius for you.

channel: The channel type.

radius: The radius of the Gaussian, in pixels, not counting the center pixel.

sigma: The standard deviation of the Laplacian, in pixels.

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel      = & h0001
const CyanChannel      = & h0001
const GreenChannel     = & h0002
const MagentaChannel   = & h0002
const BlueChannel      = & h0004
const YellowChannel    = & h0004
const AlphaChannel     = & h0008
const OpacityChannel   = & h0008
const BlackChannel     = & h0020
const IndexChannel     = & h0020
const AllChannels      = & h7fffffff
```

Sets the last exception property.

For more details please check the ImageMagick documentation.

Shave(x as integer, y as integer, width as integer, height as integer) as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Shave shaves pixels from the image edges.

Notes:

It allocates the memory necessary for the new Image structure and returns a pointer to the new image.

Returns nil on any error.

Sets the last exception property.

Shear(Xshear as double, Yshear as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method ShearImage creates a new image that is a shear_ image copy of an existing one.

Notes:

Shearing slides one edge of an image along the X or Y axis, creating a parallelogram. An X direction shear slides an edge along the X axis, while a Y direction shear slides an edge along the Y axis. The amount of the shear is controlled by a shear angle. For X direction shears, x_ shear is measured relative to the Y axis, and similarly, for Y direction shears y_ shear is measured relative to the X axis. Empty triangles left over from shearing the image are filled with the color defined by the pixel at location (0,0). ShearImage allocates the memory necessary for the new Image structure and returns a pointer to the new image.

Xshear and Yshear specify the number of degrees to shear the image.

Sets the last exception property.

For more details please check the ImageMagick documentation.

Solarize(factor as double) as boolean

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method SolarizeImage produces a 'solarization' effect seen when exposing a photographic film to light during the development process.

Notes:

factor: An double value that defines the extent of the solarization.

Returns nil on any error.

Sets the last exception property.

Splice(x as integer, y as integer, width as integer, height as integer) as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Splice splices a solid color into the image as defined by the geometry.

Notes:

Returns nil on any error.

Sets the last exception property.

Spread(radius as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This is a special effects method that randomly displaces each pixel in a block defined by the radius parameter.

Notes:

radius: Choose a random pixel in a neighborhood of this extent.
Sets the last exception property.

For more details please check the ImageMagick documentation.

Stegano(watermarkImage as IMImageQ16MBS) as IMImageQ16MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method SteganoImage hides a digital watermark within the image.

Notes:

Returns nil on any error.
Sets the last exception property.

Stereo(otherImage as IMImageQ16MBS) as IMImageQ16MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method StereoImage combines two images and produces a single image that is the composite of a left and right image of a stereo pair.

Notes:

The left image is converted to gray scale and written to the red channel of the stereo image. The right image is converted to gray scale and written to the blue channel of the stereo image. View the composite image with red-blue glasses to create a stereo effect.

left image = self
right image = otherImage parameter

Returns nil on any error.
Sets the last exception property.

Swirl(degrees as double) as IMImageQ16MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method SwirlImage creates a new image that is a copy of an existing one with the image pixels "swirl" at a specified angle.

Notes:

degrees: An double value that defines the tightness of the swirling.

Returns nil on any error.

Sets the last exception property.

Thumbnail(width as integer, height as integer) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Changes the size of an image to the given dimensions.

Notes:

Sets the last exception property.

This method was designed by Bob Friesenhahn as a low cost thumbnail generator.

For more details please check the ImageMagick documentation.

TransformImage(CropGeometry as string, ImageGeometry as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransformImage() is a convenience method that behaves like ResizeImage() or CropImage() but accepts scaling and/or cropping information as a region geometry specification. If the operation fails, the original image handle is left as is.

Notes:

This should only be used for single images.

CropGeometry: A crop geometry string. This geometry defines a subregion of the image to crop.

ImageGeometry: An image geometry string. This geometry defines the final size of the image.

Returns true on success.

TransformImages(CropGeometry as string, ImageGeometry as string) as boolean

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransformImages() calls TransformImage() on each image of a sequence.

Notes:

TransformImage() is a convenience method that behaves like ResizeImage() or CropImage() but accepts scaling and/or cropping information as a region geometry specification. If the operation fails, the original image handle is left as is.

CropGeometry: A crop geometry string. This geometry defines a subregion of the image to crop.

ImageGeometry: An image geometry string. This geometry defines the final size of the image.

Returns true on success.

TransformRGBImage(Colorspace as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method TransformRGBImage converts the reference image from an alternate colorspace.

Notes:

The transformation matrices are not the standard ones: the weights are rescaled to normalized the range of the transformed values to be [0..MaxRGB] .

colorspace: An integer value that indicates the colorspace the image is currently in. On return the image is in the RGB color space.

Returns false on any error and true on success.

constants:

TransposeImage as IMImageQ16MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransposeImage() creates a horizontal mirror image by reflecting the pixels around the central y-axis while rotating them by

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVCColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

90 degrees.

TransverseImage as IMImageQ16MBS

Plugin Version: 11.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** TransverseImage() creates a vertical mirror image by reflecting the pixels around the central x-axis while rotating them by 270 degrees.

Trim as IMImageQ16MBS

Plugin Version: 6.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Trim trims pixels from the image edges.

Notes:

It allocates the memory necessary for the new Image structure and returns a pointer to the new image.

Returns nil on any error.

Sets the last exception property.

UnsharpMaskChannel(channel as integer, radius as double, sigma as double, amount as double, threshold as double) as IMImageQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sharpens one or more image channels.

Notes:

We convolve the image with a Gaussian operator of the given radius and standard deviation (sigma). For reasonable results, radius should be larger than sigma. Use a radius of 0 and UnsharpMask selects a suitable radius for you.

Constants for channel:

```
const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel     = & h0001
const CyanChannel     = & h0001
const GreenChannel    = & h0002
const MagentaChannel  = & h0002
const BlueChannel     = & h0004
const YellowChannel   = & h0004
const AlphaChannel    = & h0008
const OpacityChannel  = & h0008
const BlackChannel    = & h0020
const IndexChannel    = & h0020
const AllChannels     = & h7fffffff
```

Sets the last exception property.

For more details please check the ImageMagick documentation.

Wave(amplitude as double, wavelength as double) as IMImageQ16MBS

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method Wave creates a new image that is a copy of an existing one with the image pixels altered along a sine wave.

Notes:

Parameters are double values that indicates the amplitude and wavelength of the sine wave.

Returns nil on any error.

Sets the last exception property.

WhiteThreshold(threshold as string) as boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** WhiteThreshold is like Threshold but forces all pixels above the threshold into white while leaving all pixels below the threshold unchanged.

Notes:

No exceptions are generated.

threshold: Define the threshold value. (ASCII string)

For more details please check the ImageMagick documentation.

WriteImage(info as IMImageInfoQ16MBS) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Method WriteImage writes an image to a file as defined by image.filename.

Notes:

You can specify a particular image format by prefixing the file with the image type and a colon (i.e. ps:image) or specify the image type as the filename suffix (i.e. image.ps). The image may be modified to adapt it to the requirements of the image format. For example, DirectClass images must be color-reduced to PseudoClass if the format is GIF.

WriteImage returns True if the image is written. False is returned if there is a memory shortage or if the image file fails to write.

2.32.2 Properties

BackgroundColor as IMColorQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image background color.

Notes: (Read and Write property)

Bias as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

BlurFactor as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Blur factor to apply to the image when zooming. Default is 1.0 (no blur).

Notes: (Read and Write property)

BorderColor as IMColorQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image border color.

Notes: (Read and Write property)

Colors as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The desired number of colors.

Notes:

Used by Quantize().

(Read and Write property)

ColorSpace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image pixel interpretation.

Notes:

If the colorspace is RGB the pixels are red, green, blue. If matte is true, then red, green, blue, and index. If it is CMYK, the pixels are cyan, yellow, magenta, black. Otherwise the colorspace is ignored.

constants:

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

(Read and Write property)

Compression as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image compression type.

Notes:

useful constants:

const UndefinedCompression	= 0
const NoCompression	= 1
const BZipCompression	= 2
const FaxCompression	= 3
const Group4Compression	= 4
const JPEGCompression	= 5
const LosslessJPEGCompression	= 6
const LZWCompression	= 7
const RLECompression	= 8
const ZipCompression	= 9

The default is the compression type of the specified image file.
For more details please check the ImageMagick documentation.

(Read and Write property)

Depth as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image depth (8 or 16).

Notes:

QuantumLeap must be defined before a depth of 16 is valid.

(Read and Write property)

Directory as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Tile names from within an image montage.

Notes:

Only valid after calling MontageImages() or reading a MIFF file which contains a directory.

(Read and Write property)

Endian as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The endian setting to use.

Notes:

constants:

UndefinedEndian	0	
LSBEndian	1	(Windows)
MSBEndian	2	(Mac)

e.g. tiff files support different endian settings.

(Read and Write property)

Filename as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The file path/name.
Notes:

The string must be in the encoding of the library and is limited to 4000 bytes.
For more details please check the ImageMagick documentation.
(Read and Write property)

Filter as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Filter to use when resizing image.
Notes:

Constants:

const PointFilter	=1
const BoxFilter	=2
const TriangleFilter	=3
const HermiteFilter	=4
const HanningFilter	=5
const HammingFilter	=6
const BlackmanFilter	=7
const GaussianFilter	=8
const QuadraticFilter	=9
const CubicFilter	=10
const CatromFilter	=11
const MitchellFilter	=12
const LanczosFilter	=13
const BesselFilter	=14
const SincFilter	=15

The reduction filter employed has a significant effect on the time required to resize an image and the resulting quality. The default filter is Lanczos which has been shown to produce high quality results when reducing most images.
(Read and Write property)

Fuzz as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Colors within this distance are considered equal.

Notes:

A number of algorithms search for a target color. By default the color must be exact. Use this to match colors that are close to the target color in RGB space.

(Read and Write property)

Gamma as double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Gamma level of the image.

Notes:

The same color image displayed on two different workstations may look different due to differences in the display monitor. Use gamma correction to adjust for this color difference.

(Read and Write property)

Geometry as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Preferred size of the image when encoding.

Notes: (Read and Write property)

Gravity as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to an Image structure.
For more details please check the ImageMagick documentation.
(Read and Write property)

Height as integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The height of the image in pixels.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Interlace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The type of interlacing scheme (default NoInterlace).

Notes:

This option is used to specify the type of interlacing scheme for raw image formats such as RGB or YUV. NoInterlace means do not interlace, LineInterlace uses scanline interlacing, and PlaneInterlace uses plane interlacing. PartitionInterlace is like PlaneInterlace except the different planes are saved to individual files (e.g. image.R, image.G, and image.B). Use LineInterlace or PlaneInterlace to create an interlaced GIF or progressive JPEG image.

constants:

(Read and Write property)

UndefinedInterlace	0	Unset value.
NoInterlace	1	Don't interlace image (RBRGBRGRGBRGRGBRGRB...)
LineInterlace	2	Use scanline interlacing (RRR...GGG...BBB...RRR...GGG...BBB...)
PlaneInterlace	3	Use plane interlacing (RRRRRR...GGGGG...BBBBBB...)
PartitionInterlace	4	Similar to plane interlacing except that the different planes are saved to individual files (e.g. image.R, image.G, and image.B)

LastError as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last error code reported.

Notes:

If an exception is raised and it is not a warning exception, this exception code is saved in this property.
(Read and Write property)

LastException as IMExceptionQ16MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The last exception thrown by the Image Magick library.

Notes:

You should check this value after every call to the library, process the error and set the property to nil.

For more details please check the ImageMagick documentation.
(Read and Write property)

Magick as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image encoding format (e.g. "GIF").

Notes: (Read and Write property)

Matte as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether an alpha channel is used/present.

Notes:

Set to true to enable masks.
(Read and Write property)

MatteColor as IMColorQ16MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image matte (transparent) color.

Notes: (Read and Write property)

Montage as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Tile size and offset within an image montage. Only valid for montage images.

Notes: (Read and Write property)

Offset as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Number of initial bytes to skip over when reading raw image.

Notes: (Read and Write property)

Orientation as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The image orientation.

Notes:

constants:

```

const UndefinedOrientation    = 0
const TopLeftOrientation     = 1
const TopRightOrientation    = 2
const BottomRightOrientation = 3
const BottomLeftOrientation  = 4
const LeftTopOrientation     = 5
const RightTopOrientation    = 6
const RightBottomOrientation = 7
const LeftBottomOrientation  = 8

```

For more details please check the ImageMagick documentation.
(Read and Write property)

Quality as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** JPEG/MIFF/PNG compression level.

Example:

```
dim im as ImageMagickQ16MBS // global
```

```
Function TestJPEG(f as folderitem) As picture
```

```
// Reads an image, compresses in memory to JPEG, decompresses using JPEGlib and returns the image
```

```
// if quality setting works, you see it in the result.
```

```
// no error checking included!
```

```
// needs: im as ImageMagickQ16MBS ready initialized
```

```
dim image as IMImageQ16MBS
```

```
dim imageinfo as IMImageInfoQ16MBS
```

```
dim s,blob as string
```

```
dim p as Picture
```

```
dim i as integer
```

```
if f = nil then
```

```
Return nil
```

```
end if
```

```
imageinfo = im.NewImageInfo
```

```
# if TargetWin32 then //do not use shellpath, if spaces, IM doesn't like escaped paths
```

```
imageinfo.Filename = f.AbsolutePath
```

```
# else
```

```
imageinfo.Filename = f.UnixpathMBS
```

```

# endif

//this should read any image IM understands
image = im.ReadImage(imageinfo)
//check for error
if im.lastexception <>nil and im.LastException.Severity >= 400 then
s = "LastError: " + Format(im.LastError,"-0") + " - Severity: " + str(im.LastException.Severity) + EndOfLine + im.LastException.Message
MsgBox s
Return nil
elseif image = nil then
MsgBox "image=nil"
Return nil
end if

// Now lets convert to jpeg
imageinfo.Filename = "image.jpg"
imageinfo.Quality = 10 // 100 is max
blob = image.ImageToBlob(imageinfo)

// It may fail
if blob.lenb = 0 then
Return nil
end if
p = JPEGStringToPictureMBS(blob,true)

image.DestroyImage
imageinfo.DestroyImageInfo

Return p
Exception
Return nil
End Function

```

Notes:

Default value is 75.
(Read and Write property)

Release as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** If true, the destructor will release the handle.

Notes: (Read and Write property)

RenderingIntent as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The rendering intent to use.

Notes:

constants:

UndefinedIntent	0
SaturationIntent	1
PerceptualIntent	2
AbsoluteIntent	3
RelativeIntent	4

(Read and Write property)

ResolutionUnits as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Units of image resolution.

Notes:

constants:

UndefinedResolution	0	Unset value.
PixelsPerInchResolution	1	Density specifications are specified in units of pixels per inch (english units).
PixelsPerCentimeterResolution	2	Density specifications are specified in units of pixels per centimeter (metric units).

(Read and Write property)

ResolutionX as double

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The horizontal resolution of the image.

Notes:

The unit for resolution must be specified.
 For more details please check the ImageMagick documentation.
 (Read and Write property)

ResolutionY as double

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The vertical resolution of the image.

Notes:

The unit for resolution must be specified.
 For more details please check the ImageMagick documentation.
 (Read and Write property)

Scene as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

StorageClass as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image storage class.

Notes:

If DirectClass then the image packets contain valid RGB or CMYK colors. If PseudoClass then the image has a colormap referenced by pixel's index member.

constants:

UndefinedClass	0	Unset value.
DirectClass	1	Image is composed of pixels which represent literal color values.
PseudoClass	2	Image is composed of pixels which specify an index in a color palette.

(Read and Write property)

Taint as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Set to True if the image pixels have been modified.

Notes: (Read and Write property)

Width as integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The width of the image in pixels.

Notes:

For more details please check the ImageMagick documentation.

(Read and Write property)

2.32.3 Constants

kBackgroundDispose = 2

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kCoalesceLayer = 1

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompareAnyLayer = 2

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompareClearLayer = 3

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompareOverlayLayer = 4

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kCompositeLayer = & h0000000C

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kDisposeLayer = 5

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kFlattenLayer = & h0000000E

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kMergeLayer = & h0000000D

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kMosaicLayer = & h0000000F

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kNoneDispose = 1

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kOptimizeImageLayer = 7

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kOptimizeLayer = 6

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kOptimizePlusLayer = 8

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kOptimizeTransLayer = 9

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kPreviousDispose = 3

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kRemoveDupsLayer = & h0000000A

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kRemoveZeroLayer = & h0000000B

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kUndefinedDispose = 0

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

kUndefinedLayer = 0

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer method constants.

kUnrecognizedDispose = 0

Plugin Version: 8.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** One of the Image layer Dispose Types.

2.33 class IMImageInfoQ8MBS

class IMImageInfoQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A class for information about an image.

Notes: For more details please check the ImageMagick documentation.

2.33.1 Methods

Clone as IMImageInfoQ8MBS

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Clones the Image-Info object.

Notes: For more details please check the ImageMagick documentation.

Close

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The destructor.
Notes:

There is no need to call this method except you want to free all resources used by this object now without waiting for Realbasic to do it for you.

(e.g. some Realbasic versions crash on Windows if there are plugin objects not closed.)

DestroyImageInfo

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Destroys the image info and sets the handle to 0.

Notes:

For more details please check the ImageMagick documentation.
The destructor will call this for you if release=true.

HandleMemory as memoryblock

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The content of the whole ImageInfo structure copied into a memoryblock.

Notes: Returns nil on any error.

2.33.2 Properties

Adjoin as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Join images into a single multi-image file.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Affirm as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unknown.

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

Antialias as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Control antialiasing of rendered Postscript and Postscript or TrueType fonts.

Notes:

Enabled by default.

For more details please check the ImageMagick documentation.

(Read and Write property)

Authenticate as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

BackgroundColor as IMColorQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image background color.

Notes: (Read and Write property)

BorderColor as IMColorQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image border color.

Notes: (Read and Write property)

Channel as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The channel to use.

Notes:

Constants for channel:

```

const UndefinedChannel = 0
const RedChannel       = & h0001
const GrayChannel     = & h0001
const CyanChannel     = & h0001
const GreenChannel    = & h0002
const MagentaChannel  = & h0002
const BlueChannel     = & h0004
const YellowChannel   = & h0004
const AlphaChannel    = & h0008
const OpacityChannel  = & h0008
const BlackChannel    = & h0020
const IndexChannel    = & h0020
const AllChannels     = & h7fffffff

```

(Read and Write property)

Colors as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

ColorSpace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image pixel interpretation.

Example:

```
dim im as ImageMagickQ8MBS // global
```

```
Function IMPictureToString(p as picture, magick as string, quality as integer) As string
```

```
dim image as new IImageQ8MBS
```

```
dim imageinfo as IImageInfoQ8MBS
```

```
dim s,data as string
```

```
dim impp as new IMMagickPixelPacketQ8MBS
```

```
// empty string for nil picture
```

```
if p = nil then
```

```
Return ""
```

```
end if
```

```

// create a new picture info

imageinfo = im.NewImageInfo
imageinfo.ColorSpace=1
// only color space is needed. 1 for RGB.

// background color of image
impp.red = 0
impp.Green = 0
impp.Blue = 0

// creates a new image object
if not image.NewImage(imageinfo,p.Width,p.Height,impp) then
Return ""
end if

// copy RB picture into IM Image at position 0/0
image.ColorSpace = 1
image.SetPicture(p,0,0)

// set compression data
imageinfo.Magick = magick
imageinfo.Quality = quality

// and rendering intent: 2=PerceptualIntent
image.RenderingIntent = 2

// create image data
data = image.ImageToBlob(imageinfo)

// release memory
image.DestroyImage
imageinfo.DestroyImageInfo

// return result
Return data

Exception
// in case of an exception return nothing
Return ""

End Function

```

Notes:

If the colorspace is RGB the pixels are red, green, blue. If matte is true, then red, green, blue, and index. If it is CMYK, the pixels are cyan, yellow, magenta, black. Otherwise the colorspace is ignored.

constants:

UndefinedColorspace	0
RGBColorspace	1
GRAYColorspace	2
TransparentColorspace	3
OHTAColorspace	4
LABColorspace	5
XYZColorspace	6
YCbCrColorspace	7
YCCColorspace	8
YIQColorspace	9
YPbPrColorspace	10
YUVCColorspace	11
CMYKColorspace	12
sRGBColorspace	13
HSBColorspace	14
HSLColorspace	15
HWBColorspace	16

(Read and Write property)

Compression as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image compression type.

Notes:

useful constants:

The default is the compression type of the specified image file.
For more details please check the ImageMagick documentation.
(Read and Write property)

```
const UndefinedCompression    = 0
const NoCompression           = 1
const BZipCompression         = 2
const FaxCompression          = 3
const Group4Compression       = 4
const JPEGCompression         = 5
const LosslessJPEGCompression = 6
const LZWCompression          = 7
const RLECompression          = 8
const ZipCompression          = 9
```

Density as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Vertical and horizontal resolution in pixels of the image.

Notes:

This option specifies an image density when decoding a Postscript or Portable Document page.
(Read and Write property)

Depth as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image depth (8 or 16).

Notes:

QuantumLeap must be defined before a depth of 16 is valid.
(Read and Write property)

Dither as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Endian as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The endian setting to use.

Notes:

constants:

UndefinedEndian	0	
LSBEndian	1	(Windows)
MSBEndian	2	(Mac)

e.g. tiff files support different endian settings.
(Read and Write property)

Extract as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Filename as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The file path/name.

Notes:

The string must be in the encoding of the library and is limited to 4000 bytes.

For more details please check the ImageMagick documentation.

(Read and Write property)

Font as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Text rendering font.

Notes:

If the font is a fully qualified X server font name, the font is obtained from an X server. To use a TrueType font, precede the TrueType filename with an @. Otherwise, specify a Postscript font name (e.g. "helvetica").

(Read and Write property)

Group as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Handle as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The handle used internally by the plugin.

Notes:

A pointer to a ImageInfo structure.

For more details please check the ImageMagick documentation.

(Read and Write property)

HeaderOnly as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** True if only the header was read from the image data.

Notes: (Read and Write property)

Interlace as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The type of interlacing scheme (default NoInterlace).

Notes:

This option is used to specify the type of interlacing scheme for raw image formats such as RGB or YUV. NoInterlace means do not interlace, LineInterlace uses scanline interlacing, and PlaneInterlace uses plane interlacing. PartitionInterlace is like PlaneInterlace except the different planes are saved to individual files (e.g. image.R, image.G, and image.B). Use LineInterlace or PlaneInterlace to create an interlaced GIF or

progressive JPEG image.

constants:

UndefinedInterlace	0	Unset value.
NoInterlace	1	Don't interlace image (RGBRGBRGBRGBRGB...)
LineInterlace	2	Use scanline interlacing (RRR...GGG...BBB...RRR...GGG...BBB...)
PlaneInterlace	3	Use plane interlacing (RRRRRR...GGGGGG...BBBBBB...)
PartitionInterlace	4	Similar to plane interlacing except that the different planes are saved to individual files (e.g. image.R, image.G, and image.B)

(Read and Write property)

Magick as String

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image encoding format (e.g. "GIF").

Example:

```
dim imageinfo as IMImageInfoQ8MBS
dim blob as string
dim image as IMImageQ8MBS

// Now lets convert to tiff
imageinfo.Filename = "image"
imageinfo.Magick="JPEG"
imageinfo.Quality = 10 //since we are displaying, lets use highest quality, lowest compression
blob = image.ImageToBlob(imageinfo)
```

Notes:

For more details please check the ImageMagick documentation.
(Read and Write property)

MatteColor as IMColorQ8MBS

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image matte (transparent) color.

Notes: (Read and Write property)

Monochrome as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Transform the image to black and white.

Notes: (Read and Write property)

Orientation as Integer

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The image orientation.

Notes:

constants:

```
const UndefinedOrientation    = 0
const TopLeftOrientation     = 1
const TopRightOrientation    = 2
const BottomRightOrientation = 3
const BottomLeftOrientation  = 4
const LeftTopOrientation     = 5
const RightTopOrientation    = 6
const RightBottomOrientation = 7
const LeftBottomOrientation  = 8
```

For more details please check the ImageMagick documentation.
(Read and Write property)

Page as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Equivalent size of Postscript page.

Notes: (Read and Write property)

PointSize as Double

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Text rendering font point size.

Notes: (Read and Write property)

Preview as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image manipulation preview option.

Notes:

Used by 'display'.

constants:

(Read and Write property)

Quality as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** JPEG/MIFF/PNG compression level.

Notes:

Default value is 75.

(Read and Write property)

Release as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** If true, the destructor will release the handle.

UndefinedPreview	0
RotatePreview	1
ShearPreview	2
RollPreview	3
HuePreview	4
SaturationPreview	5
BrightnessPreview	6
GammaPreview	7
SpiffPreview	8
DullPreview	9
GrayscalePreview	10
QuantizePreview	11
DespecklePreview	12
ReduceNoisePreview	13
AddNoisePreview	14
SharpenPreview	15
BlurPreview	16
ThresholdPreview	17
EdgeDetectPreview	18
SpreadPreview	19
SolarizePreview	20
ShadePreview	21
RaisePreview	22
SegmentPreview	23
SwirlPreview	24
ImplodePreview	25
WavePreview	26
OilPaintPreview	27
CharcoalDrawingPreview	28
JPEGPreview	29

Notes: (Read and Write property)

ResolutionUnits as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Units of image resolution.

Notes:

constants:

(Read and Write property)

UndefinedResolution	0	Unset value.
PixelsPerInchResolution	1	Density specifications are specified in units of pixels per inch (english units).
PixelsPerCentimeterResolution	2	Density specifications are specified in units of pixels per centimeter (metric units).

SamplingFactor as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Scene as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

SceneCount as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

Scenes as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An undocumented property.

Notes: (Read and Write property)

ServerName as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** X11 display to display to.

Notes:

obtain fonts from, or to capture image from.
(Read and Write property)

Size as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Width and height of a raw image (an image which does not support width and height information).

Notes:

Size may also be used to affect the image size read from a multi-resolution format (e.g. Photo CD, JBIG, or JPEG).

(Read and Write property)

Temporary as Boolean

Plugin Version: 5.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Unknown.

Notes:

For more details please check the ImageMagick documentation.

(Read and Write property)

Texture as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Image filename to use as background texture.

Notes: (Read and Write property)

Type as Integer

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The Image type.

Notes:

constants:

UndefinedType	0
BilevelType	1
GrayscaleType	2
GrayscaleMatteType	3
PaletteType	4
PaletteMatteType	5
TrueColorType	6
TrueColorMatteType	7
ColorSeparationType	8
ColorSeparationMatteType	9
OptimizeType	10

(Read and Write property)

Verbose as Boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Print detailed information about the image if True.

Notes: (Read and Write property)

View as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** FlashPix viewing parameters.

Notes: (Read and Write property)

Chapter 3

List of all classes

• ImageMagickQ16MBS	157
• ImageMagickQ32MBS	174
• ImageMagickQ8MBS	165
• IMColorQ16MBS	194
• IMColorQ32MBS	191
• IMColorQ8MBS	184
• IMExceptionQ16MBS	182
• IMExceptionQ32MBS	189
• IMExceptionQ8MBS	187
• IMImageAffineMatrixQ16MBS	136
• IMImageAffineMatrixQ32MBS	134
• IMImageAffineMatrixQ8MBS	140
• IMImageAttributeQ16MBS	138
• IMImageAttributeQ32MBS	139
• IMImageAttributeQ8MBS	118
• IMImageInfoQ16MBS	119
• IMImageInfoQ32MBS	142
• IMImageInfoQ8MBS	321

• IMImageQ16MBS	261
• IMImageQ32MBS	197
• IMImageQ8MBS	29
• IMMagickInfoListQ16MBS	100
• IMMagickInfoListQ32MBS	101
• IMMagickInfoListQ8MBS	99
• IMMagickInfoQ16MBS	94
• IMMagickInfoQ32MBS	90
• IMMagickInfoQ8MBS	110
• IMMagickPixelPacketQ16MBS	115
• IMMagickPixelPacketQ32MBS	106
• IMMagickPixelPacketQ8MBS	102
• IMMissingFunctionExceptionQ16MBS	110
• IMMissingFunctionExceptionQ32MBS	110
• IMMissingFunctionExceptionQ8MBS	109