Scripting with VBScript

Basic Data Types

Points

Points are transferred as an array of 2 values. Depending on the context the points contain float or of LONG values.

Sizes

Sizes are transferred as an array of 2 values. Depending on the context the sizes contain float or of LONG values.

Rectangles

Rectangles are transferred as an array of 4 values. Depending on the context the sizes contain float or of LONG values.

The first 2 values are the position of the rectangle, the following 2 are the width and the height. The width and the height always have to be larger than or equal to 0.

Ranges

A range is transferred as an array of 2 LONGs. The first one is the position of the range, the second one is its length.

Interfaces

IApplication

Used by *Application* object.

Properties

Property	Type	Description
ActiveDocument	IPLDocument *	The top document.
Count	LONG	(Read-only) The number of documents.
Item(index)	IPLDocument * LONG, Range [0;Count - 1]	(Read-only) The document with the given index.
Visible	Boolean	Visibility of PhotoLine

Methods

Method	Type	Description
GetIdentityMatrix	IPLMatrix *	Creates an identity matrix.
GetPerspectiveMatrix(rectangle, corners)	IPLMatrix * float[4] float[8]	Creates a transformation matrix, that maps a rectangle to a quadrilateral. Returns an error, if the corners don't form a valid perspective transformation.
GetRotationMatrix(degrees	IPLMatrix * float	Creates a rotation matrix. Optionally, it uses the given reference point as fix point.

Method	Type	Description
[, referencePoint])	float[2] (optional)	
GetScaleMatrix(scaleX, scaleY [, referencePoint])	IPLMatrix * float float float[2] (optional)	Creates a scaling matrix. Optionally, it uses the given reference point as fix point.
GetTranslationMatrix(offsetX, offsetY)	IPLMatrix * float float	Creates a translation matrix.
Index(document)	LONG IPLDocument *	Returns the index of the given document.
InputBox([title [, description [, defaultValue]]])	String String (optional) String (optional) String (optional)	Query user for a value with an input dialog. You can optionally set the title of the dialog, the description of the input field and a default value. If not set or empty, the title is "PhotoLine", the description is "Value:" and the default value is empty. If the user cancels the dialog, the return value will be an empty string.
Open(filename [, optionKey, optionValue]+)	IPLDocument * String String, variable	Opens a document file and returns the resulting document. OptionKey and optionValue are optional key-value pairs, that are dependent of the file format and are parameters for the document import. Additionally, there is the key "AddToRecentFiles" with a Boolean value. The default value is FALSE.
OpenAsPlaceholder(filename [, optionKey, optionValue]+)	IPLDocument * String String, variable	Opens a document file as placeholder and returns the resulting document. OptionKey and optionValue are optional key-value pairs, that are dependent of the file format and are parameters for the document import. Additionally, there is the key "EmbedFile" with a Boolean value. The default value is FALSE.
OpenDialog	String[]	Shows an Open dialog and allows the user to select files. Returns an array of filenames as strings.

IPLColor

Used by *Color* object.

If a property or method takes a IPLColor as parameter, you can usually use a float[] instead.

If the type is named "IPLColor * (variable, count)", the color will be created based on the number of elements in the float array:

- 1 element creates a CMGray color.

- 3 elements create a CMRGB color.

- 4 elements create a CMCMYK color.

If the type is named "IPLColor * (variable, RGB)", PhotoLine will create a CMRGB color. In this case the number of elements has to be 3 or 4 (with alpha).

Properties

Property	Type	Description
ApplyTransparency	Boolean	Only for document colors:

Property	Type	Description
		False: if applied, the colors inside the document keep their transparency. True: if applied, the color inside the document get the transparency from the document color.
Gradient	IPLDictionary *	If the color is not a gradient, this property is set to null/nothing.
Matrix	IPLMatrix *	If the color is a pattern or gradient, matrix is the transformation of the content. Otherwise it is null/nothing.
Model	ColorModel	The color model of the color.
Name	String	The name of the color.
SpotColor	Boolean	Only for document colors: Defines whether the color is a spot color. Spot colors must have a name.
Values	float[], default range [0;1]	The color values of the color. The number of elements depends on the model of the color. The last element is the color's alpha value. If this property is set, the alpha value is optionally and 1 (opaque) is used if it's missing.

IPLColorProfile

Implemented by the *ColorProfile* object.

Properties

Property	Type	Description
Data	Byte []	(Read-only) The data of the color profile.
Model	ColorModel	(Read-only) The color model of the color profile.
Name	String	(Read-only) The name of the color profile.
RenderingIntent	RenderingIntent	(Read-only) The rendering intent of the color profile.
Path	String	(Read-only) The file path of the color profile. May be empty.

Methods

Method	Type	Description
Init(profile, renderingIntent)	None variable RenderingIntent	Initialize a color profile. The color profile can be initialized by - another IPLColorProfile - a string defining the full pathname to a color profile - a string with the name of an installed color profile - a byte array containing the profile data
IsEqual(otherProfile)	Boolean IPLColorProfile *	Checks two color profiles for equality.

IPLCurve

Implemented by the *Curve* object.

If a property or method takes a IPLCurve as parameter, you can usually use a float[] instead. In that case, PhotoLine will create a CTSpline curve and the type is named "IPLCurve * (variable)" in this description.

Properties

Property	Type	Description
Points	float []	The curve points. A normal curve has at least 2 points, resulting in an array size of 4.
Type	CurveType	The curve type.

IPLDictionary

Implemented by the *Dictionary* object. It is an interface to describe data and contains key-value pairs. The key is always a string and the type of the value is variable.

Properties

Property	Type	Description
Count()	LONG	(Read-only) The number of keys in the dictionary.
Item(key)	variable String	Access the content of a key-value pair.
Key(index)	String LONG	(Read-only) The key at the given index.

Methods

Method	Type	Description
Add([dictionary] | [key, value]+)	None IPLDictionary * String, variable	Initialize a dictionary. The color profile can be initialized by - another IPLDictionary - a sequence of key-value pairs
Remove(key)	None String	Remove a key. If the key doesn't exist, nothing is done.

IPLDocument

Implemented by the *Document* object.

Properties

Property	Type	Description
ActiveLayer	IPLLayer *	The active layer of the active page. May be null/nothing.
ActivePage	IPLPage *	The active page.
Application	IApplication *	The application.
ColorProfile	IPLColorProfile *	The color profile of the document. May be null/nothing.
Colors	IPLColor *[]	An array of colors assigned to the document. Every color must have a unique name.
Count	LONG	(Read-only) The number of pages.
DocumentMode	Boolean	False: The document is in picture mode True: The document is in document mode
Item(index)	IPLPage * LONG, Range [0;Count - 1]	(Read-only) The document with the given index.
Path	String	The file path of the document.

Property	Type	Description
Resolution	float	The resolution of the document in dpi.
RootLayer	IPLLayer *	(Read-only)The root layer of the active page.
SelectedLayers	IPLLayerArray *	The selected layers of the active page. If there are entries, the first entry is always the active layer.
Size	LONG[2]	Size of the document in pixels.
TextStyles(paragraphStyles)	IPLDictionary *[] Boolean	An array of IPLDictionaries, that describe the text styles assigned to the document. If paragraph styles is True, it will return the paragraph styles, otherwise the character style.

Methods

Method	Type	Description
CanChangeDocumentMode	Boolean	Checks whether DocumentMode can be changed.
Close(option)	None CloseOption	Closes the document.
Copy([options] | [optionKey, optionValue]+)	None IPLDictionary * String, variable	Copy a merged image of the document to the clipboard. The available options are described here.
DeletePageContainer(position)	None LONG	Delete a page container at the given position.
DoOperation(operationName [, options] | [, optionKey, optionValue]+)	None String IPLDictionary * String, variable	Execute the operation with the given name. The options are optional and can either be set using an IPLDictionary or by using key-value pairs.
Duplicate()	IPLDocument *	Duplicates the document.
DuplicatePageContainer(srcPageOrPosition, destPageOrPosition)	IPLPage * IPLPage * or LONG IPLPage * or LONG	Duplicate a page container. The page container can be specified either as a position or an IPLPage. The destination position can also be either a position or a page container. In the second case the duplicated page container will be inserted after that. The result is the new page container.
Export(filename [, options] | [, optionKey, optionValue]+)	None String IPLDictionary * String, variable	Export the document with the given filename (including path). The export options are optional and can either be set using an IPLDictionary or by using key-value pairs. The available options are dependent of the file format.
Index(document)	LONG IPLPage *	Returns the index of the given document.
InsertPageContainer(position [, pageToInsert])	None LONG IPLPage *	Insert a new page container at a given position. Optionally you can insert an already existing page container. If missing, a new page container will be created.
Merge([page] | [layer] | [array of layers])	IPLImage * IPLPage * IPLLayer * IPLLayer * or IPLLayer *[]	Merges the document or optionally a page or a layer or an array of layers to a single image. The result is the newly created image or NULL, if nothing has been created.
MovePageContainer(srcPageOrPosition, destPageOrPosition)	None IPLPage * or LONG IPLPage * or LONG	Move to page container to another position. The page container can be specified either as a position or an IPLPage. The destination position can also be either a position or a page container. In the second case the source page container will be inserted after that.

Method	Type	Description
Paste	IPLLayerArray *	Paste the content of the clipboard as new layers. It returns the newly created layers. The new layers are inserted after the selected layers.
Save()	None	Save the document. If the document is untitled, an error is reported.
SaveAs(filename [, options] | [, optionKey, optionValue]+)	None String IPLDictionary * String, variable	Save the document with the given filename (including path). The export options are optional and can either be set using an IPLDictionary or by using key-value pairs. The available options are dependent of the file format.
SaveDialog([dialogTitle] [, defaultFilename])	String String	Show a save dialog for the document. You can set a dialog title and a default filename. If not set, default values are used. It returns the selected filename or an empty string, if the dialog was left by using "Cancel".

IPLImage

Implemented by the *Image* object. It is an interface, that offers access to all properties of a PhotoLine image layer. IPLImage inherits from IPLLayer.

Properties

Property	Type	Description
Alpha	Boolean	(Read-only) Returns whether the picture has an alpha channel.
BitsPerChannel	LONG	(Read-only) The bit depth of the picture (1, 8, 16 or 32).
Picture	IPLPicture *	The picture of the image.
PictureSize	LONG[2]	(Read-only) The size of the picture.
PictureType	PictureType	The type of the picture. This property is a combination of the values defined in PictureType. A possible value is (PTRGB | PT16Bit | PTAlpha).

Methods

Method	Type	Description
InitPicture(pictureType, size [, initValue])	None PictureType LONG[2] variable	Initialize the picture with a certain type and size. Optionally you can fill the newly created picture with - a single value (float) - a float[] with one entry per picture channel - an IPLColor

IPLLayer

Implemented by the *Layer*, *Image*, *Vector* and *Text* object. It is an interface, that offers access to all basic properties of a PhotoLine layer.

Properties

Property	Type	Description
Adjustment(	IPLDictionary *	The adjustment with the given index.

Property	Type	Description
index)	LONG, range [0; AdjustmentsCount - 1]	
Adjustments	IPLDictionary *[]	Array of dictionaries, that describe the adjustments attached to a layer. See Operations applicable to images and as adjustments.
AdjustmentsCount	LONG	(Read-only) The number of adjustments attached to the layer.
BlendMode	BlendMode	The blend mode/mix mode of the layer.
Bounds(coordinateSystem, boundsType)	float[4] CoordinateSystem BoundsType	(Read-only) The bounds of the layer as rectangle.
ChannelMask	LONG	The color channels of the document that are modified by the layer. This is a bit mask. In a RGB document 1 is red, 2 is green und 4 is blue. So a value of 6 means, that green and blue is modified.
Children	IPLLayerArray *	(Read-only) An array with the children of the layer.
Clipping	Boolean	True: The layer is either a clipping layer or a layer mask.
ClippingWidth	float	For clipping layers, layer masks and adjustment layer and a value > 0: an optional blur applied to mask, that is created by the layer.
ColorProfile	IPLColorProfile *	The color profile of the layer. This property is usually null/nothing.
Count	LONG	(Read-only) The number of children.
Document	IPLDocument *	(Read-only) The layer's document.
First	IPLLayer *	(Read-only) The first child of the layer or nothing.
Intensity	float, range [-2;2]	The layer's intensity (1 = 100%)
InvertClipping	Boolean	For clipping layers, layer masks and adjustment layer: The effect of the layer is inverted.
Item(index)	IPLLayer * LONG	(Read-only) The child with the given index.
Isolated	Boolean	True: The layer is drawn isolated (isn't affected by its background).
Last	IPLLayer *	(Read-only) The last child of the layer or nothing/null.
LayerMask	Boolean	If Clipping and LayerMask are true, the layer is a layer mask.
MatrixToPage	IPLMatrix *	The transformation of the layer relative to the page. This value is a concatenation of MatrixToParent of the layer with the MatrixToParent of its ancestors.
MatrixToParent	IPLMatrix *	The transformation of the layer relative to its parent.
ModifiesTransparency	Boolean	True: The layer modifies the transparency of its background.
Name	String	The layer's name.
Next	IPLLayer *	(Read-only) The layer's following layer (the layer above it) or nothing/null.
Origin(coordinateSystem,	float[2] CoordinateSystem	The position of the top left corner of a layer. This is only the top left corner in CSLayer.

Property	Type	Description
boundsType)	BoundsType	
Page	IPLPage *	(Read-only) The layer's page.
Parent	IPLLayer *	(Read-only) The layer's parent.
PixelAlignment	Alignment	The content of the layer is aligned to the document pixels. Only used with Layer, Image and Vector.
Position	LONG	The layer's index inside its parent.
Previous	IPLLayer *	(Read-only) The layer's preceding layer (the layer below it) or nothing/null.
Quality	Quality	The layer's rendering quality.
ReferencePoint(coordinateSystem)	float[2] CoordinateSystem	The layer's reference point. This point can be adjusted by the user, and can be used as fix point for rotations, scaling, ...
RelativeColors	Boolean	True: If the layer uses pattern colors, the patterns will by transformed just the same as the layer.
Root	IPLLayer *	(Read-only) The deepest ancestor of the layer or the layer itself.
Size(coordinateSystem, boundsType)	float[2] CoordinateSystem BoundsType	The size of a layer.
Type	LayerType	(Read-only) The layer's type.
Visible	Boolean	Visibility of layer.

Methods

Method	Type	Description
Copy([options] | [optionKey, optionValue]+)	None IPLDictionary * String, variable	Copy the layer or – if it is an image – optionally the selection to the clipboard. The available options are described here.
CreateDocument(options)	IPLDocument * LONG (LayerCreateDocumentFlags)	Duplicates the layer and creates a new document containing that duplicate.
Delete()	None	Delete the layer.
DoOperation(operationName [, options] | [, optionKey, optionValue]+)	IPLLayer String IPLDictionary * String, variable	Execute the operation with the given name. The options are optional and can either be set using an IPLDictionary or by using key-value pairs. If the operation creates a new layer, the result is that new layer. Otherwise, the layer for which DoOperation() was called is returned.
Duplicate(index | afterLayer)	IPLLayer * LONG IPLLayer *	Duplicate the layer and returns the duplicated layer. The duplicate is either inserted as child in the original's parent or behind another layer.
DuplicateVirtual(index | afterLayer)	IPLLayer * LONG IPLLayer *	Duplicate the layer and returns the duplicated layer. The duplicate is either inserted as child in the original's parent or behind another layer. This other layer has to be in the same document as the original layer.
Index(child)	LONG IPLLayer *	The index of the given child layer. Child has to be a child of the called layer.
Insert(	None	Insert one or more layers as child. The layers are either

Method	Type	Description
layer | layers, index | afterLayer)	IPLLayer * IPLLayerArray *(variable) LONG IPLLayer *	inserted at a position or after another layer.
InsertAdjustment(index, [dictionary] | [key, value]+)	None LONG, range [0; AdjustmentsCount] IPLDictionary * String, variable	Insert a new adjustment. The adjustment can either be a IPLDictionary or a sequence of key-value pairs.
Paste([index | afterLayer])	IPLLayerArray * LONG IPLLayer *	Paste the content of the clipboard as new layers. It returns the newly created layers. The new layers are inserted as child at a given index or behind another layer.
RemoveAdjustment(index)	None LONG, range [0; AdjustmentsCount - 1]	Remove an adjustment.
Save(filename [, options] | [, optionKey, optionValue]+)	None String IPLDictionary * String, variable	Save the layer with the given filename (including path). The export options are optional and can either be set using an IPLDictionary or by using key-value pairs. The available options are dependent of the file format.
ShowOperationDialog(operationName [, options] | [, optionKey, optionValue]+)	IPLDictionary * String IPLDictionary * String, variable	Show the dialog of the operation with the given name. The operation is not executed. The return value can be used for DoOperation later on. The options are optional and can either be set using an IPLDictionary or by using key-value pairs. Returns the user's settings or nothing/null, if the user clicked cancel.

IPLLayerArray

Implemented by the *LayerArray* object.

If a property or method takes a IPLLayerArray as parameter, you can often either use an IPLLayer or an array of IPLLayers instead. This is indicated by the type name "IPLLayerArray * (variable)".

Properties

Property	Type	Description
Count	LONG	(Read-only) The number of layers in the array.
Item(index)	variable LONG, range [0;Count - 1]	The layer at the given index.

Methods

Method	Type	Description
Copy([options] | [optionKey, optionValue]+)	None IPLDictionary * String, variable	Copy the layers in the array to the clipboard. The available options are described here.
Index(layer)	LONG IPLLayer *	The index of the given layer or -1, if it isn't in the array.
Insert(layer, position)	None IPLLayer * LONG (optional)	Insert layer in the array. If no position is given, the layer is appended.

Method	Type	Description
Remove(index | layer)	None LONG IPLLayer *	Remove the entry at a given index or a layer..
Sort()	None	Sort the layers in the array according to their position in the document.

IPLLineStyle

Implemented by the *LineStyle* object.

If a property or method takes a IPLLineStyle as parameter, you can usually use a float instead. In that case, PhotoLine will create a line style with the given float value used as width. This is indicated by the type name "IPLLineStyle * (variable)".

Properties

Property	Type	Description
AdjustDashLength	Boolean	True: The length of the dashes is adjusted, so that they fit to the corners of the path.
Alignment	LineStyleAlignment	The alignment of the line on a path.
ArrowLength	Float	The length of an optional arrow (1 is 100%)
ArrowWidth	Float	The width of an optional arrow (1 is 100%)
DashLengths	Float []	An array with the lengths of the line style dashes. May be empty.
DashPhase	Float	The starting position inside the dashes. Usually 0.
EndCap	LONG (LineStyleCap)	The appearance of the end of the line.
Join	LineStyleJoin	The join type.
MaximumWidth	Float, range [MinimumWidth;1]	(Read-only) If a width curve is set, this value can be used to stretch the width of the curve. Usually 1.
MinimumWidth	Float, range [0;MaximumWidth]	(Read-only)If a width curve is set, this value can be used to stretch the width of the curve. Usually 0.
StartCap	LONG (LineStyleCap)	The appearance of the start of the line.
Width	float	The line width. If smaller than 0, the line is invisible.
WithCurve	IPLCurve *	(Read-only)The shape of the line style or nothing/null.

Methods

Method	Type	Description
SetWidthCurve(curve, minWidth, maxWidth)	None IPLCurve * (variable) float (optional) float (optional)	Set the width for variable width line styles. MinWidth and maxWidth can be used for scaling the line width.

IPLMatrix

Implemented by the *Matrix* object. It controls the transformation of layers, points, sizes and rectangles.

If a property or method takes a IPLMatrix as parameter, you can usually use a float[] instead. This is indicated by the type name "IPLMatrix * (variable)". In that case, PhotoLine will create transformation matrix with the given values.

The number of elements has to be 6 or 9. If the values don't result in a valid transformation, PhotoLine will return an error.

Methods

Method	Type	Description
Concatenate(matrix [, otherMatrices]+)	IPLMatrix * IPLMatrix * (variable) IPLMatrix * (variable)	Concatenate the matrix with one or more other matrices.
Invert()	IPLMatrix *	Calculate the inverse of the matrix.
TransformPoints(points)	float[2 * n] float[2 * n]	Transform n points.
TransformRectangles(rects)	float[4 * n] float[4 * n]	Transform n rectangles.
TransformSizes(sizes)	float[2 * n] float[2 * n]	Transform n sizes.

IPLPage

Implemented by the *Page* object.

Properties

Property	Type	Description
ActiveLayer	IPLLayer *	The active layer of the page. May be null/nothing.
ColorProfile	IPLColorProfile *	The color profile of the page. May be null/nothing.
RootLayer	IPLLayer *	(Read-only)The root layer of the page.
SelectedLayers	IPLLayerArray *	The selected layers of the page. If there are entries, the first entry is always the active layer.
Size	LONG[2]	Size of the page in pixels.

Methods

Method	Type	Description
Copy([options] | [optionKey, optionValue]+)	None IPLDictionary * String, variable	Copy a merged image of the page to the clipboard. The available options are described here.
DoOperation(operationName [, options] | [, optionKey, optionValue]+)	None String IPLDictionary * String, variable	Execute the operation with the given name. The options are optional and can either be set using an IPLDictionary or by using key-value pairs.
Paste	IPLLayerArray *	Paste the content of the clipboard as new layers. It returns the newly created layers. The new layers are inserted after the selected layers.

IPLPicture

Implemented by the Picture object. It represents a pixel picture. It is the datatype contained in IPLImages.

Properties

Property	Type	Description
Alpha	Boolean	(Read-only) Returns whether the picture has an alpha channel.
BitsPerChannel	LONG	(Read-only) The bit depth of the picture (1, 8, 16 or 32).
Size	LONG[2]	The size of the picture.
Type	PictureType	The type of the picture. This property is a combination of the values defined in PictureType. A possible value is (PTRGB | PT16Bit | PTAlpha).

Methods

Method	Type	Description
Init(pictureType, size [, initValue])	None PictureType LONG[2] variable	Initialize the picture with a certain type and size. Optionally you can fill the newly created picture with - a single value (float) - a float[] with one entry per picture channel - an IPLColor

IPLText

Implemented by the *Text* object. It is an interface, that offers access to all properties of a PhotoLine text layer. IPLText inherits from IPLLayer.

Properties

Property	Type	Description
Attribute(index, attributeName [, range = 0])	Variable LONG String LONG[2], out	Access the text attribute with the given name at the given index. Returns the range of the attribute optionally.
Attributes(index [, range = 0])	IPLDictionary * LONG LONG[2], out	Access the text attributes at the given index. Returns the range of the attributes optionally.
AutoFlowIn	Boolean	True: The text layer can create a text flow to the previous page automatically.
AutoFlowOut	Boolean	True: The text layer can create a text flow to the next page automatically.
NextInFlow	IPLText *	The next text layer in the text flow. This property must not be set for text text layers with AutoFlowOut equal to true.
Text([index = 0] [, count = -1])	String LONG LONG	Text text in the given range. Accesses the text from index 0 with no index, and accesses to the end of the text with no count.
TextRange	LONG[2]	(Read-only) The range of the text in the text layer. If the text layer is not part of a text flow, this is the complete text.
TextLength	LONG	(Read-only) The length of the text in characters. If the text layer is part of a text flow, this is the length of the complete text.
VerticalAlignment	TextVerticalAlignment	The vertical alignment of the text.

Methods

Method	Type	Description
ParagraphRange(index)	LONG[2] LONG	Returns the paragraph range for the given index.
SetAttribute(range, attributeName, newValue)	None LONG[2] String variable	Changes the attribute with the given name. The type of the new value depends on the attribute. If applicable, you can set partial attribute and only the set values will be transferred to the text layer.
SetAttributes(range, newAttributes)	None LONG[2] IPLDictionary *	Changes the attributes in the given range.

IPLVector

Implemented by the *Vector* object. It is an interface, that offers access to all properties of a PhotoLine vector layer.
IPLVector inherits from IPLLayer.

Properties

Property	Type	Description
FillColor(index)	IPLColor * (variable, RGB)	Fill color in the attribute set with the given index.
LineColor(index)	IPLColor * (variable, RGB)	Line color in the attribute set with the given index.
LineStyle(index)	IPLLineStyle * (variable)	Line style in the attribute set with the given index.
VectorAttributes	IPLDictionary *[]	The fill and line attributes of the vector layer. Each dictionary contains an optional - fill color - line color - line style
VectorAttributesCount	LONG	The number of attributes.
VectorPath	IPLVectorPath *	The vector path of the layer.

Methods

Method	Type	Description
InsertPoints(index [, pointType, points]+)	None LONG LONG float[]	Insert a sequence of points. The index must not be inside a curve. The point type has to be VPTMoveTo, VPTLineTo or VPTCurveTo. If the point type is VPTMoveTo, the first point of the points array will be a MoveTo, the following ones will be LineTos. If point type is VPTCurveTo, the points array must have a multiple of 3 points, resulting in a multiple of 6 float values.
InsertVectorAttribute(index)	None LONG, range [0;VectorAttributesCount]	Insert a new empty attribute set.
RemovePoints(index,	None LONG	Remove a range of points. You must not delete partial curves.

Method	Type	Description
count)	LONG	
ReplacePoints(index, points)	None LONG float[2 * n]	Replace n points at the given index with the given points.
RemoveVectorAttribute(index)	None LONG, range [0;VectorAttributesCount-1]	Remove an attribute set.

IPLVectorPath

Implemented by the VectorPath object. It is the datatype contained in IPLVector.

Properties

Property	Type	Description
Count	LONG	(Read-only) The number of points in the path.
Points	Float[2 * Count]	(Read-only) The points of the vector path.
PointTypes	BYTE[COUNT] (VectorPointType)	(Read-only) For each point the corresponding type.

Methods

Method	Type	Description
Insert(index [, pointType, points]+)	None LONG LONG float[]	Insert a sequence of points into a path. The index must not be inside a curve. The point type has to be VPTMoveTo, VPTLineTo or VPTCurveTo. If the point type is VPTMoveTo, the first point of the points array will be a MoveTo, the following ones will be LineTos. If point type is VPTCurveTo, the points array must have a multiple of 3 points, resulting in a multiple of 6 float values.
Remove(index, count)	None LONG LONG	Remove a range of points. You must not delete partial curves.
Replace(index, points)	None LONG float[2 * n]	Replace n points at the given index with the given points.

Enumerations

Alignment

Value	Description
AlignDefault = 0	Use the inherited alignment.
AlignToPixels = 1	Align to the document pixels.
AlignDont = 2	Don't align.

BarcodeType

BarcodeType is used as parameter for the Barcode operation.

Value	Description
BarcodeI25 = 0	I25 barcode
BarcodeEAN8 = 1	EAN8 barcode
BarcodeEAN13 = 2	EAN13 barcode
BarcodeUPCA = 3	UPCA barcode
BarcodeUPCE = 4	UPCE barcode
BarcodeISBN = 5	ISBN code
BarcodeC39 = 6	C39 barcode
BarcodeEAN128 = 7	EAN128 barcode
BarcodeC93 = 8	C93 barcode
BarcodeC128 = 9	C128 barcode
BarcodeQR = 10	QR code
BarcodeC39E = 11	C39E barcode

BlendMode

Value	Description
BMNormal = 0	Normal
BMMultiply = 1	Multiply
BMDissolve = 2	Dissolve
BMScreen = 3	Screen
BMOverlay = 4	Overlay
BMSoftLight = 5	Soft light
BMHardLight = 6	Hard light
BMColorDodge = 7	Color dodge
BMColorBurn = 8	Color burn
BMDarken = 9	Darken
BMLighten = 10	Lighten
BMDifference = 11	Difference
BMExclusion = 12	Exclusion
BMLinearDodge = 13	Linear dodge/add
BMRemove = 14	Remove
BMLinearBurn = 15	Linear burn
BMHardMix = 16	Hard mix
BMLinearLight = 17	Linear light
BMVividLight = 18	Vivid light
BMPinLight = 19	Pin light
BMLighterColor = 20	Lighter color
BMDarkerColor = 21	Darker color

Value	Description
BMSubtract = 22	Subtract
BMDivide = 23	Divide
BMHue = 24	Hue
BMSaturation = 25	Saturation
BMColor = 26	Color
BMLuminance = 27	Luminance

BoundsType

BoundsType defines various types of bounds used in combination with IPLLayers.

Value	Description
BTGeometric = 0	The geometric bounds define the base coordinate system of a layer. With vector layers, these bounds include the vector path, but not the outline. With text, these bounds are the rectangle, that can contain the text.
BTLayout = 1	The layout bounds are used by PhotoLine for setting the position and size of a layer.
BTAlignment = 2	The alignment bounds are used by PhotoLine for aligning layer next to each other.
BTContent = 3	The content bounds are the area occupied by the content of a layer. With vector layers, the outline is included. With text, the content is only the area containing text.

CloseOption

CloseOption is used as parameter for IPLDocument::Close.

Value	Description
COSave = 0	If the document has unsaved changes, it will be saved before closing. If the document is untitled, an error will be reported.
CODontSave = 1	The document will be closed without saving.
COAsk = 2	If the document has unsaved changes, the user will be prompted for saving. If the user cancels closing, an error will be reported.

CoordinateSystem

The coordinate system defines, how sizes or positions are set or returned by an IPLLayer.

Value	Description
CSPage = 0	The page coordinate system. To convert values from CSLayer to CSPage, the values have to be multiplied with the layer's MatrixToPage.
CSGroup = 1	The coordinate system of a layer's parent. To convert values from CSLayer to CSGroup, the values have to be multiplied with the layer's MatrixToParent.
CSLayer = 2	The layer's native coordinate system.

ColorModel

The ColorModel is used for IPLColor.

Value	Description
CMGray = 0	Gray
CMRGB = 1	RGB
CMCMYK = 2	CMYK
CMHIS = 10	HIS
CMHSV = 11	HSV
CMLab = 12	Lab

ColorSpaceMode

The ColorSpaceMode can be used with gradients and some operations. It defines how pixels or color data are written.

Most operations allow only a subset of these modes.

Value	Description
CSMNative = 0	Data will be written using the native color space of the destination.
CSMHIS = 1	HIS
CSMHSV = 2	HSV
CSMLab = 3	Lab
CSMRGB = 4	RGB
CSMCYMK = 5	CMYK
CSMGray = 6	Grays

CurveType

The CurveType is used for IPLCurve.

Value	Description
CTBezier = 0	Interpolation by bezier curves.
CTSpline = 1	Interpolation by splines.
CTLagrange = 2	Use Lagrange interpolation.
CTLinear = 3	Use linear interpolation.

EXIFPreviewMode

Value	Description
EPMAlways = 0	Always create an EXIF preview.
EPMKeepExisting = 1	If there is already an EXIF preview, create an updated version.
EPMNever = 2	Never create an EXIF preview.

EXRCompression

Value	Description
ECNone = 0	No compression.
ECRLE = 1	Run length encoding.
ECZIP = 2	Deflate for single scanlines.
ECZIPBlock16 = 3	Deflate for scanline block with 16 lines.
ECPIZ = 4	Wavelet compression.
ECPXR24 = 5	24 bit deflate.
ECB44 = 6	Lossy compression.
ECB44A = 7	Extended B44.

GradientInterpolation

GradientInterpolation defines the interpolation of gradient colors.

Value	Description
GILinear = 0	Linear Interpolation.
GICubic = 1	Cubic Interpolation.

GradientSpread

GradientSpread defines the interpolation of gradient colors.

Value	Description
GIContinue = 0	The gradient will continue with the end color.
GIReflect = 1	The colors will repeat reflected at the gradient end.
GIRepeat = 2	The colors will repeat at the gradient end.

GradientType

Value	Description
GTLinear = 0	Linear gradient.
GTCircle = 1	Circular gradients.
GTRadial = 2	Radial gradient.
GTRadialFull = 3	Reflected radial gradient.

InterpolationMode

InterpolationMode defines several types of interpolations.

Value	Description
IMNextPixel = 0	No Interpolation.
IMBilinear = 1	Bilinear.
IMLanczos3 = 2	Lanczos 3.
IMLanczos8 = 4	Lanczos 8.
IMMitchellNetravali = 5	Mitchell-Netravali.

Value	Description
IMCatmullRom = 6	Catmull-Rom.
IMCubicSpline = 7	Cubic Spline.
IMLiquid = 8	Only available for the Scale operation in combination with images.

LayerCreateDocumentFlags

Value	Description
LCDInvisible = 1	The new document is not visible.

LayerType

Value	Description
LTImage = 1	An image layer.
LTVector = 2	A vector layer.
LTText = 4	A text layer.
LTGroup = 8	A group.
LTVirtualCopy = 16	The virtual copy of another layer.
LTPlaceholder = 32	A placeholder.

LineStyleAlignment

Value	Description
LSAAlignCenter = 0	The line lies on the vector path.
LSAAlignInside = 1	The line is inside the vector path.
LSAAlignOutside = 2	The line is outside the vector path.

LineStyleCap

Value	Description
LSCButtCap = 0	Butt cap.
LSCRoundCap = 1	Round cap.
LSCSquareCap = 2	Square cap.
LSCArrowCap = 128	The start/end is an arrow. Used in combination with the other values.

LineStyleJoin

Value	Description
LSJMiterJoin = 0	Miter join.
LSJRoundJoin = 1	Round join.
LSJBevelJoin = 2	Bevel join.

MakeSelectionMode

Modes for combining a new selection with the current one.

Value	Description
MSMSet = 0	The new selection replaces the current one.
MSMAdd = 1	The new selection is added.
MSMSub = 2	The new selection is subtracted..
MSMIntersect = 3	The resulting selection is the intersection of the current and the new one.

NoiseType

Value	Description
NTNone = 0	No noise.
NTTurbulence = 1	Turbulence.
NTFractalSum = 2	Sum of fractals.
NTNoise = 3	Simple noise.

ParagraphAlignment

Value	Description
PALeft = 0	Left aligned text.
PARight = 1	Right aligned text.
PACenter = 2	Centered text.
PAJustified = 3	Justified text.
PAJustifiedAll = 4	Justified text, including the last line of a paragraph.

ParagraphRegister

Value	Description
PRNone = 0	No line is aligned to the line register.
PRWholeParagraph = 1	All lines of the paragraph are aligned.
PRFirstLine = 2	Only the first line is aligned to the line register.

PDFColorMode

PDFColorMode controls, how color data inside a document is converted on export.

Value	Description
PDFDocument = 0	Color data is not converted.
PDFCMYK = 1	Color data is converted to CMYK. Gray data is not converted.
PDFGray = 2	Color dat is converted to gray.
PDFX1a = 3	Colors are converted conforming to PDF/X1a.
PDFX3 = 4	Colors are converted conforming to PDF/X3.

PDFCompressionMode

PDFCompressionMode controls the compression of colored images.

Value	Description
PDFFlate = 0	Flate compression (Zlib).
PDFJPEGHighQuality = 1	High quality JPEG (not allowed for TextCompressionMode).
PDFJPEGMediumQuality = 2	Medium quality JPEG (not allowed for TextCompressionMode).
PDFJPEGLowQuality = 3	Low quality JPEG (not allowed for TextCompressionMode).
PDFFlateFast = 4	Faster, but weaker flate compression.
PDFUncompressed = 5	No compression.

PDFFontEmbedding

Value	Description
PDFNoEmbedding = 0	Fonts are not embedded.
PDFEmbedOptionalVector = 1	If allowed, fonts are embedded. Otherwise they are converted to vector. This is the only allowed option for PDF/X1a and PDF/X3.
PDFEmbed = 2	Allowed fonts are embedded.
PDFConvertToVector = 3	All fonts are converted to vector.

PDFTransparencyMode

PDFTransparency controls, how transparency is converted on export.

Value	Description
PDFReplaceWithBackground = -1	Transparency is replaced by an opaque image.
PDFDitherBayer = 0	Dither transparency, thus creating bileveled transparency.
PDFDitherCoarse = 1	Dither transparency, thus creating bileveled transparency.
PDFDitherVertical = 2	Dither transparency, thus creating bileveled transparency.
PDFDitherHorizontal = 3	Dither transparency, thus creating bileveled transparency.
PDFDitherFine = 4	Dither transparency, thus creating bileveled transparency.
PDFDitherOrdered = 5	Dither transparency, thus creating bileveled transparency.
PDFDitherOrderedFat = 6	Dither transparency, thus creating bileveled transparency.
PDFDither45 = 7	Dither transparency, thus creating bileveled transparency.
PDFDitherThreshold = 8	Use a threshold for transparency, thus creating bileveled transparency.
PDFFullTransparency = 1000	If possible, export transparency unmodified. This option does not work in combination with the PDF color modes PDFX1a and PDFX3.

PictureType

The type of a pixel picture is defined by the values of PictureType. There are 3 sections in this enumeration:

- the color model

- the bit depth

- a flag whether there is an alpha channel

So examples for valid picture types are "PTGray + PT8Bit + PTAlpha" or "PTCMYK + PT16Bit".

Value	Description
PTGray = 0	Gray
PTRGB = 1	RGB
PTCMYK = 2	CMYK
PTBitmap = 3	1 bit image
PTLab = 10	Lab
PTMask = 255	Mask to get the color model from a picture type.
PT8Bit = 0	8 Bit image
PT32Bit = 4096	32 bit (float) image
PT16Bit = 8192	16 bit image
PTAlpha = 16384	Image with alpha

Quality

Value	Description
QualityDefault = 0	Use the inherited/default quality.
QualityAntialias = 1	Always use antialiasing/interpolation.
QualityNoAntialias = 2	Never use antialiasing/interpolation.

RenderingIntent

RenderingIntent declares the possible rendering intents for ICC color profiles.

Value	Description
RIAutomatic = -1	Use the default rendering intent of the color profile.
RIPerceptual = 0	Perceptive.
RIRelativeColorimetric = 1	Relative colorimetric.
RISaturation = 2	Saturation.
RIAbsoluteColorimetric = 4	Absolute Colorimetric.

ResizeMode

ResizeMode is used as parameter for the Resize operation.

Value	Description
ResizeNormal = 0	The resize parameter will define the new size.
ResizeResize = 1	The resize parameter will be added to the original size.
ResizeFormula = 2	The new size and the position of the original content will be expressed as formulas. "w" and "h" are the original size, "dw" and "dh" are the document size.

ResizeHorizontalAlignment

ResizeMode is used as parameter for the Resize operation.

Value	Description
ResizeLeft = 0	The original content will be aligned left inside the new area.
ResizeRight = 1	The original content will be aligned right inside the new area.
ResizeHorizontalCentered = 2	The original content will be centered inside the new area.
ResizeHorizontalAbsolute = 3	There is no automatic alignment.

ResizeVerticalAlignment

ResizeMode is used as parameter for the Resize operation.

Value	Description
ResizeTop = 0	The original content will be aligned top inside the new area.
ResizeBottom = 1	The original content will be aligned bottom inside the new area.
ResizeVerticalCentered = 2	The original content will be centered inside the new area.
ResizeVerticalAbsolute = 3	There is no automatic alignment.

ScaleMode

Scale is used as parameter for the Scale operation.

Value	Description
SMNormal = 0	The resulting size is set in pixels by the parameters "ValueX" and "ValueY". With documents, "DPI" can be used to set the resolution.
SMDPI = 1	"DPI" is used as new dpi value.
SMPercent = 2	"ValueX" and "ValueY" are the scaling values in percent (100: no scaling).
SMWidth = 3	"ValueX" is the new width.
SMHeight = 4	"ValueY" is the new height.
SMFit = 5	The object is scaled proportionally to fit in "ValueX" and "ValueY".
SMUnused = 6	Unused
SMFormula = 7	"FormulaX", "FormulaY" and "FormulaDPI" are formulas for the new width, height and dpi. "w", "h" and "d" can be used as constant for the original width, height and dpi.

SVGCompressionMode

Value	Description
SVGPNGFast = 0	Fast, but weaker PNG compression.
SVGPNGStrong = 1	Strong, but slower PNG compression.
SVGJPEGLow = 2	Low quality, small size JPEG.
SVGJPEGMid = 3	Medium quality, medium size JPEG.
SVGJPEGHigh = 4	High quality, large size JPEG.

TabType

Value	Description
TTLeft = 0	Left aligned tab.
TTRight = 1	Right aligned tab.
TTCentered = 2	Centered tab.
TTDecimal = 3	Tab with alignment to a decimal point character.

TextVerticalAlignment

TextVerticalAlignment defines the options for vertical alignment of text layer.

Value	Description
TVATop = 0	The text is at the top of the layer frame (default).
TVACenter = 1	The text is centered.
TVABottom = 2	The text is at the bottom.

VectorPointType

Each point in a IPLVectorPath has an associated type. This type is a combination of three parts:

- the point type

- if a point is part of a curve, the subindex of the point (from 0 to 2)

- a flag whether the point is selected

Value	Description
VPTMoveTo = 0	The start of a (sub) path, a "Move To".
VPTLineTo = 1	A line point.
VPTCurveTo = 2	A point, that is part of a curve. A curve always has 3 points: the control point 1 (index 0), the control point 2 (index 1) and the end point (index 2).
VPTTypeMask = 3	The and-mask to isolate the point type.
VPTIndexMask = 12	If a point is a curve point, VPTIndexMask and VPTIndexShift can be used to get index of the curve point: (type & VPTIndexMask)>>VPTIndexShift
VPTIndexShift = 2	See VPTIndexMask.
VPTSelected = 128	If set, the point is selected. With curve points you must not select the control points (the ones with the index 0 and 1).

Operations

Operations can be executed on documents, pages and layers. Every operation has a unique name and optional parameters. The parameters can either be set by using a single IPLDictionary or by key-value pairs.

If key-value pairs are used, the key is always a string and is the name of the parameter. The value is dependent of the type of parameter.

Operations applicable to all objects

All of the operations have the optional parameter "ShowDialog". If it is set to true, PhotoLine will show the dialog of the operation before executing it.

Action

This operation executes an action. You can either execute an action installed in PhotoLine by using its name or a saved action by passing its path or its binary data.

Parameter	Type	Description
Data	Byte [] or Base64 string or hex string	(Optional) The binary data of a saved action.
Name	String	(Optional) The name of the action.
Path	String	(Optional) The path of a saved action.

Barcode

This operation creates a barcode layer.

Parameter	Type	Description
Font	String	(Optional) The font used for the description of the barcode. Not used, if ShowText is FALSE. Default: "Arial"
Position	float[4]	The rectangle defining the position of the barcode.
ShowText	LONG (boolean)	(Optional) The barcode operation can display the value of the barcode below the actual barcode as text. Default: FALSE
Type	LONG (BarcodeType)	(Optional) The type of the barcode. Default: QR code (BarcodeQR)
Value	String	The content of the barcode. Must not be empty.

Resize

Resize can operate with documents, pages and images und changes the size of that object.

Parameter	Type	Description
Color	IPLColor * (variable, count)	(Optional) For documents the color that will be used for new areas. Default: white
FormulaX, FormulaY, FormulaW, FormulaH	String	If Mode is Formula (2): The new size of the object and the position of the original content in this new area as formula. The formulas can use the values "w" and "h" (original size) and "dw" and "dh" (document size).
Mode	LONG (ResizeMode)	(Optional) The resize mode: 0: Normal mode (the new size will be the value of Size). 1: Resize mode (the new size will be the original size plus the value of Size). 2: Formula mode (the size and position are defined by formulas). Default: 0
Horizontal, Vertical	LONG (ResizeHorizontalAlignment or ResizeVerticalAlignment)	(Optional) The alignment of the original content inside the new area. Default: 3

Parameter	Type	Description
		(ResizeHorizontalAbsolute/ResizeVerticalAbsolute:use Position)
Position	float[2]	(Optional) If Mode is 3, the position of the original content inside the new area. Default: (0; 0)
Size	float[2]	The new size.

Rotate

Parameter	Type	Description
Angle	float	The rotation angle in degrees (counter-clockwise). Either that or "Radiant" has to be set.
Crop	LONG (boolean)	(Optional) TRUE: The result will be cropped removing parts not filled by the original content-
Interpolation	LONG (InterpolationMode)	The interpolation mode used for scaling images.
Radians	float	The rotation angle in radians (counter-clockwise). Either that or "Angle" has to be set.

Scale

Parameter	Type	Description
Mode	LONG (ScaleMode)	The scale mode.
Interpolation	LONG (InterpolationMode)	The interpolation mode used for scaling images.
ValueX	float	For the modes SMNormal, SMPercent, SMWidth and SMFit the new width. SMPercent expects percent values, the rest pixels.
ValueY	float	For the modes SMNormal, SMPercent, SMHeight and SMFit the new height. SMPercent expects percent values, the rest pixels.
ValueDPI	float	For the modes SMNormal and SMDPI the new dpi value.
FormulaX FormulaY FormulaDPI	String	The new width/height/dpi as formula. "w", "h" and "d" can be used as the original width, height and dpi.

Operations applicable to images and as adjustments

AdaptiveSharpen

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1

Parameter	Type	Description
Radius	float, larger than 0.5	The filter radius.
Special	float, range [0;100]	The filter threshold.

AdaptiveSoften

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.
Special	float, range [0;100]	The filter threshold.

ChannelMixer

Parameter	Type	Description
Mode	LONG (boolean)	(Optional) False: normal, channel-wise mode. True: Brightness mode: Factor1 and Offset1 are the values for the new brightness. Default: False
Factor1 Factor2 Factor3 Factor4	Float[4], range[-2;+2]	The factors for each channel of the image, that will be used to produce the new channel value. Example with RGB and Factor1 values {0.7, 0.2, 0.1}: newR = oldR * 0.7 + oldG * 0.2 + oldB * 0.1
Offset1 Offset2 Offset3 Offset4	Float, range [-2;2]	(Optional) Offset that will be added the new channel value. Default: 0

ChromaticAberration

Parameter	Type	Description
BlueShift	float, range [-10;10]	(Optional) Scaling of the blue channel in pixels. Default: 0
Center	Float[2], range [0;1]	(Optional) Relative center of scaling. Default: (0.5;0.5)
RedShift	float, range [-10;10]	(Optional) Scaling of the red channel in pixels. Default: 0
WidthCompensation	LONG (boolean)	(Optional) True: Depending on the size of the image a different scaling factor may be applied to x and to y. Default: Falses

Clouds

Parameter	Type	Description
Color1	IPLColor * (variable, RGB)	(Optional) The first cloud color. Default: black.

Parameter	Type	Description
Color2	IPLColor * (variable, RGB)	(Optional) The second cloud color. Default: white.
Contrast	Float, range [0;1]	(Optional) Contrast of the created clouds. 0.5 is the neutral value. Default: 0.5
Intensity	Float, range [0;1]	(Optional) Intensity of the created clouds. 0.5 is the neutral value. Default: 0.25
NoiseAmplitudeStep	Float, range [1;4]	(Optional) Amplitude scaling of the additional noises for NTTurbulence and NTFractalSum. Default: 2
NoiseDetail	Float, range [1;256]	(Optional) Resolution of clouds, higher values create more fine grained clouds. Default: 4
NoiseDetailStep	Float, range [1;8]	(Optional) Detail scaling of the additional noises for NTTurbulence and NTFractalSum. Default: 2
NoiseScaleX	Float, range [1;64]	(Optional) Additional scaling of NoiseDetail in x direction. Default: 1
NoiseScaleY	Float, range [1;64]	(Optional) Additional scaling of NoiseDetail in y direction. Default: 1
NoiseSteps	LONG, range [1;6]	(Optional) Number of overlapped noise functions for NTTurbulence and NTFractalSum. Default: 4
NoiseType	LONG (NoiseType)	(Optional) Noise type, that is the base of the cloud creation. NTNone is not allowed. Default: NTFractalSum

ColorBalance

Parameter	Type	Description
Data	LONG[9], range [-100;100]	Three groups of three values. 0 to 2: cyan-red correction 3 to 5: magenta-green correction 6 to 8: yellow-blue correction. The first value corrects the shadows, the second the midtones and the third the highlights.
Preserve	LONG (boolean)	(Optional) Preserve luminosity. The default value is false.

ColorCorrection

Parameter	Type	Description
BlueYellow	float, range [-0.25;0.25]	(Optional) Shift along the blue-yellow axis. Default: 0
GreenRed	float, range [-0.25;0.25]	(Optional) Shift along the green-red axis. Default: 0
FixWhitePoint	LONG (boolean)	(Optional) True: Don't change bright areas, the effect on saturated colors is stronger. Default: True

ColorLookup

Parameter	Type	Description
Profile	IPLColorProfile * (variable) | String	Either a color profile or a file path to 3D look-up table.

ColorTemperature

Parameter	Type	Description
Temperature	LONG, range [2000;13000]	Color temperature.

Parameter	Type	Description
WorkMode	LONG (boolean)	(Optional) False: Set the given temperature. The color profile of the image defines the source color temperature. True: The given temperature is the source color temperature. The color profile defines the destination temperature. Default: False

ColorToTransparency

Parameter	Type	Description
Color	IPLColor * (variable, count)	The color, that will be made transparent.
SimpleMode	LONG (boolean)	(Optional) False: Convert a color range to transparent. True: Use a simple calculation. Default: True
HueSize	Float, range [0°;180°]	(Optional) If SimpleMode is false: The hue range, that will be made transparent. Default: 30°
BrightnessStrength	Float, range [0;2]	(Optional) If SimpleMode is false: The strength with which the transparency is influenced by the brightness. Default: 1
SaturationStrength	Float, range [0;2]	(Optional) If SimpleMode is false: The strength with which the transparency is influenced by the saturation. Default: 1
SaturationFilter	LONG (boolean)	(Optional) If SimpleMode is true: Controls whether the saturation influences the result. Default: True
LowLimit HighLimit	Float, range [0;1]	(Optional) Used range of the calculated transparency. Default: 0 and 1

CorrectHighlights

CorrectHighlights is an adjustment, that will only be created by the import of raw files. Over exposed raw files often create magenta highlights, and it is CorrectHighlights job to fix that.

Parameter	Type	Description
Limit	float, range [0;1]	(Optional) Brightness values above this limit will be corrected. Default: 1

Curves

Parameter	Type	Description
PictureType	LONG (PictureType)	(Optional) The picture type which the curves are defined for. If not set, it is assumed to be the picture type of the image.
Contrast	LONG, range [0;100]	(Optional) The contrast.
Brightness	LONG, range [0;100]	(Optional) The brightness.
Gamma	float, larger than 0	(Optional) The gamma value.
CurveMain	IPLCurve * (variable)	(Optional) The curve that controls the brightness.
Curve1 – Curve4	IPLCurve * (variable)	(Optional) The curves for the channels 1 to 4.

Custom

Parameter	Type	Description
Bias	LONG	(Optional) Offset added to the result of the filter kernel. Default: 0
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
Clamp	LONG (boolean)	(Optional) True: The end result of "filter kernel / divider + bias" is clamped to the range [0;255]. Default: False
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Data	Float[9] | float[25]	The values of a 3x3 or a 5x5 filter kernel.
Divider	LONG	(Optional) Divider for the result of the filter kernel. Default: 1
Intensity	Float, range [0;1]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1

Denoise

Parameter	Type	Description
IntensityIntensity	Float, range [0;1]	(Optional) Intensity of intensity noise reduction. Default: 1
ThresholdIntensity	Float, range [0;1]	(Optional) Threshold for intensity noise reduction. Default: 0.04
SizeColor	Float, range [0;20]	(Optional) Filter size for color noise reduction. If absent, no color noise reduction. Default: no value.
SizeIntensity	Float, range [0;20]	(Optional) Filter size for intensity noise reduction. If absent, no intensity noise reduction. Default: no value.

Dither

Parameter	Type	Description
Angle	Float, range [0°;360°]	(Optional) Angle of dither pattern. Default: 0°
RasterSize	Float, range [4;100]	(Optional) Size of dither pattern. Default: 32

Exposure

Parameter	Type	Description
Brightness	Float, range [-150;150]	(Optional) Brightness
Contrast	Float, range [-50;100]	(Optional) Contrast
Exposure	Float, range [-20;20]	(Optional) Exposure.
Gamma	Float, range [0;9]	(Optional) Gamma. Default: 1
Offset	Float, range [-0.5;0.5]	(Optional) Offset

FalseColor

Parameter	Type	Description
Gradient	IPLDictionary * (Gradient)	The gradient used.

GaussianBlur

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.
Special	float, range [0;100]	The filter threshold.

Grain

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.
Special	float, range [0;100]	The grain density.

GrayMixer

Parameter	Type	Description
Tint	LONG (boolean)	(Optional) True: The result will be tinted. Default: False
Color	IPLColor * (variable, RGB)	(Optional) If Tint is true, this is the color of the tint.
Factors	Float[8], range [-1;1]	The factors with which the single color ranges influence the result.

Highpass

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is

Parameter	Type	Description
		the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.

Histogram

Parameter	Type	Description
Auto	LONG (boolean)	(Optional) True: The parameters are preset with values calculated from the image (cannot be used with adjustments).
Gamma	float[]	float, larger than 0	(Optional) The gamma values to be set.
PicMin PicMax OutputMin OutputMax	float[]	float, range [0;1]	(Optional) [PicMin;PicMax] is the picture range, that will be mapped to [OutputMin;OutputMax].

If you use float arrays for Gamma, PicMin, PicMax, OutputMin and OutputMax, the values will be applied to the corresponding channel. With RGB pictures, float[3] will be used for the sum channel. With gray it is a bit weird: float[3] controls the gray channel. This way RGB values can be used for gray and viceversa.

With gray and RGB you can set a single float value instead of an array. With RGB this will control the sum channel, with gray it will control the single image channel.

HorizontalEdge

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	Float, 1.5 or 2.5	The filter radius.
Special	float, range [0;100]	The filter threshold.

HueEditor

Parameter	Type	Description
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMHIS, CSMHSV, CSMLab or CSMRGB. The default value is CSMHSV.

Parameter	Type	Description
Brightness	IPLCurve * (variable)	(Optional) The correction curve for brightness.
Hue	IPLCurve * (variable)	(Optional) The correction curve for hue.
Saturation	IPLCurve * (variable)	(Optional) The correction curve for saturation.

HueSaturation

Parameter	Type	Description
Colorize	LONG (boolean)	(Optional) True: Colorize mode. The default value is false.s
Hue	float	(Optional) Colorize: Hue in the range [0;1] Other: Hue change in the range [-0.5;0.5]
Saturation	float	(Optional) Colorize: Saturation in the range [0;1] Other: Saturation change in the range [-1;1]
Brightness	float	(Optional) Colorize: Intensity in the range [0;1] Other: Intensity change in the range [-1;1]
Ranges	float[7*n], 0 <= n <= 6	(Optional) If not in colorize mode, Ranges describes the modifications applied to up to 6 hue ranges. Every sequence has a length of 7 floats: The first 4 floats define the hue range, that will be modified. The other 3 are the hue, saturation and intensity change.

LightShadow

Parameter	Type	Description
Auto	LONG (boolean)	(Optional) True: The parameters are preset with values calculated from the image (cannot be used with adjustments).
LightGamma	Float, range [0;1]	(Optional) Gamma applied to the lights. Default: 1
LightIntensity	Float, range [0;1]	(Optional) Stretching of the lights. 0: No stretching. Default: 0
LightMin	Float, range [0;1]	(Optional) Amount of lights range, that will be clipped. 0: Nothing is clipped. Default: 0
ShadowGamma	Float, range [1;2]	(Optional) Gamma applied to the shadows. Default: 1
ShadowIntensity	Float, range [0;1]	(Optional) Stretching of the shadows. 0: No stretching. Default: 0
ShadowMin	Float, range [0;1]	(Optional) Amount of shadows range, that will be clipped. 0: Nothing is clipped. Default: 0

Median

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative

Parameter	Type	Description
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius. The diameter is rounded to the next odd integer value.

OptimizeHDR

OptimizeHDR maps the pixel values of a 32-bit-HDR image to a [0;1] range.

Parameter	Type	Description
Gamma	float, larger than 0	(Optional) The gamma value to be set.
Min Max	float	The pixel value range, that will be mapped to [0;1].

MatchColors

Parameter	Type	Description
DestinationAverage	float[3], range [0;1]	Average of the destination pixel values.
DestinationDeviation	float[3], range [0;1]	Deviation of the destination pixel values.
DestinationReadFromSelection	LONG (boolean)	(Optional) UI setting. True: Read destination pixel values from the selection only.
SourceAverage	float[3], range [0;1]	Average of the source pixels values.
SourceDeviation	float[3], range [0;1]	Deviation of the source pixel values.
SourceReadFromSelection	LONG (boolean)	(Optional) UI setting. True: Read source pixel values from the selection only.
Strength	float[3], range [0;2]	Strength of the adjustment, float[0] controls the brightness, float[1] and float[2] control the color.
UseMask	LONG (boolean)	(Optional) True: Edit only in the selection area. Default: Use the PhotoLine setting.

Maximum

Maximum is a square, channel-wise maximum filter.

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.

MaximumRound

MaximumRound is a round, channel-wise maximum filter. For large radii it is significantly slower than Maximum.

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.
Special	float, range [0;100]	The filter threshold.

Minimum

Minimum is a square, channel-wise minimum filter.

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.

MotionBlur

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.
Special	float, range [0;360[	(Optional) The filter direction. Default: 0

OutlineFilter

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius. The diameter is rounded to the next odd integer.
Special	float, range [0;100]	The filter threshold.

Perturbation

Parameter	Type	Description
Granularity	Float, range [0;1]	(Optional) 0: The result is smooth, 1: the result is extremely fine grained. Default: 0.8
Distance	Float, range [0;0.5]	(Optional) The range of the effect. Default: 0.2

Relief

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	Float, 1.5	2.5	The filter radius.
Special	Float, 0	45	90	135	180	225	270	315	(Optional) The filter direction. Default: 0

RemoveDirt

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000

Parameter	Type	Description
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius. The diameter is rounded to the next odd integer value.

RemoveDisturbance

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius. The diameter is rounded to the next odd integer value.
Special	float, range [0;100]	The filter threshold.

ReplaceColor

Parameter	Type	Description
DestinationColor	IPLColor * (variable, RGB)	The destination color.
DestinationTolerance	float, range [0;1]	The range of the destination color.
SourceColor	IPLColor * (variable, RGB)	The color, that should be replaced.
SourceTolerance	float, range [0;1]	The color range, that should be replaced.

Roughen

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.
Special	float, range [0;100]	The density of the noise.

SelectiveColorCorrection

SelectiveColorCorrection has 9 sets of parameters, one set for each available color range. Each set is an array of 4 floats and each of these floats corrects one color aspect:

- float[0]: cyan

- float[1]: magenta

- float[2]: yellow

- float[3]: black

Parameter	Type	Description
Red	float[4]	(Optional) Red
Yellow	float[4]	(Optional) Yellow
Green	float[4]	(Optional) Green
Cyan	float[4]	(Optional) Cyan
Blue	float[4]	(Optional) Blue
Magenta	float[4]	(Optional) Magenta
White	float[4]	(Optional) White
Gray	float[4]	(Optional) Gray
Black	float[4]	(Optional) Black

Sharpen

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.

Soften

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1

Parameter	Type	Description
Radius	float, larger than 0.5	The filter radius.

Sponge

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius. The diameter is rounded to the next odd integer value.

Threshold

Parameter	Type	Description
Threshold	float, range [0;1]	The threshold value.

UnsharpMasking

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.
Special	float, range [0;100]	The filter threshold.

VerticalEdge

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative

Parameter	Type	Description
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	Float, 1.5 or 2.5	The filter radius.
Special	float, range [0;100]	The filter threshold.

WhitePoint

Parameter	Type	Description
Auto	LONG (boolean)	(Optional) True: The parameters are preset with values calculated from the image (cannot be used with adjustments).
Color	IPLColor * (variable, RGB)	(Optional) The color that should become white. Default: white.
FixWhitePoint	LONG (boolean)	(Optional) If WhitePoint uses the gray point mode (Mode is 1), the FixWhitePoint controls whether white is a fix color, that shouldn't be modified. Default: False.
Gamma	float, range [0;5]	(Optional) Gamma value applied to the image. Default: 1
Limit	Float, range [0;1]	(Optional) If WhitePoint uses the gray point mode (Mode is 1), Limit controls the brightness value, that corresponds to the color. Default: 1
Mode	LONG	(Optional) 0: White point mode 1: Gray point mode Default: 0

Operations applicable to images

Descreen

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	Float, larger than 0.5	The filter radius.
Special	float, range [0;100]	The filter threshold.

VariableBlur

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values:

Parameter	Type	Description
		0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.
Special	float, range [0;1]	(Optional) Contrast value applied to depth values. Default: 0.5

WipeEffect

Parameter	Type	Description
Channels	LONG, bit mask	(Optional) A bit mask defining the channels to filter. 1 is the first color channel, 2 the second, … The last one is the alpha channel. There are two special values: 0x40000000: Filter every color channel 0x80000000: Filter alpha Default value: 0xc0000000
ColorMode	LONG (ColorSpaceMode)	(Optional) CSMNative, CSMHIs and CSMLab. Default: CSMNative
Intensity	Float, range [-5;5]	(Optional) Intensity, 1 corresponds to 100%. Do not use for adjustments. Default: 1
Radius	float, larger than 0.5	The filter radius.
Special	float, range [0;360]	(Optional) The filter direction in degrees. Default: 0

Operations Manipulating Selections

ClearSelection

This operation clears the selection. It doesn't edit any layer and doesn't have options.

FillSelection

This operation fills the selection of an image layer with a color. If there is no selection, the whole image is filled.

Parameter	Type	Description
Color	IPLColor * (variable, count)	(Optional) The color used to fill the selection. If missing, the foreground color will be used.

InvertSelection

This operation inverts the selection. If there is no selection, it creates an empty selection.

MakeSelection

This operation creates or modifies a selection based on the opacity of the layer it is applied to. If the operation is applied to a document or page, it will base the selection on the active layer.

Parameter	Type	Description
Mode	LONG (MakeSelectionMode)	(Optional) The creation mode.

Dictionaries

Dictionaries for File Exports

On saving/exporting an object, you can use customized export settings. All settings are optional. If no customized settings are used, the values set inside PhotoLine are used.

Additionally to the file format options, there are:

Key	Value	Description
AddToRecentFiles	LONG (Boolean)	(Optional) If true, the used filename will be added the list of recently used filed. Only used in combination with the Save and Save command of IPLDocument or with ShowDialog set to true. Default: false
ShowDialog	LONG (Boolean)	(Optional) True: The file dialog is shown. Default: false
Title	String	(Optional) A title for the file dialog.

The BMP Parameters

Key	Value	Description
ColorDepths	LONG	(Optional) 0: Automatic, depending on the content, 8, 15, 16, 24
Transparency	LONG (Boolean)	(Optional): True: Create transparent BMP.

The DDS Parameters

Key	Value	Description
SaveMipMaps	LONG (boolean)	(Optional) True: Save mipmaps.

The EXR Parameters

Key	Value	Description
Compression	LONG (EXRCompression)	(Optional) The compression used.

The GIF Parameters

Key	Value	Description
Interlace	LONG (boolean)	(Optional) True: Create an interlaced GIF.

The JPEG Parameters

Key	Value	Description
Compress	LONG, range [0;100]	(Optional) The resulting image quality.
Progress	LONG (boolean)	(Optional) TRUE: Create a progressive JPEG.
EXIFPreview	LONG (EXIFPreviewMode)	(Optional) Controls whether an EXIF preview image is created.
ColorSubsampling	LONG (boolean)	(Optional) False: Use standard subsampling. True: Use

Key	Value	Description
		subsampling for better color quality.

The JPEG 2000 Parameters

Key	Value	Description
Compress	LONG, range [0;100]	(Optional) The resulting image quality.
Lossless	LONG (boolean)	(Optional) True: The file is compressed lossless.

The JPEG XR Parameters

Key	Value	Description
Compress	LONG, range [0;100]	(Optional) The resulting image quality.
Lossless	LONG (boolean)	(Optional) True: The file is compressed lossless.

The PDF Parameters

Key	Value	Description
Bleed	float	(Optional) The bleed width in inch.
ClipImages	LONG (boolean)	(Optional) True: Clipped images are cropped.
ClipToBleed	LONG (boolean)	(Optional) True: A clipping rectangle is created, so that everything outside the bleed rect is clipped.
ColorMode	LONG (PDFColorMode)	(Optional) The treatment of colors.
FontEmbedding	LONG (PDFFontEmbedding)	(Optional) controls the embedding of fonts.
MaxResolution	LONG	(Optional) The maximum resolution of images. Images with a higher resolution will be scaled down. 0: Don't scale any images.
OnlyPrintable	LONG (boolean)	(Optional) Only printable layers are exported.
PictureCompression	LONG (PDFCompressionMode)	(Optional) The compression mode for colored images.
Preview	LONG (Boolean)	(Optional) If true, a small preview image is created.
TextCompression	LONG (PDFCompressionMode)	(Optional) The compression mode for text data.
TransparencyMode	LONG (PDFTransparencyMode)	(Optional) The treatment of transparency.

The PLD Parameters

Key	Value	Description
Compress	LONG	(Optional) 0: No compression, 3: best compression, 5: faster compression

The PNG Parameters

Key	Value	Description
Compress	LONG, range [0;9]	(Optional) 0: No compression, 9: best compression
Interlace	LONG (boolean)	(Optional) True: Create an interlaced PNG.

The SVG Parameters

Key	Value	Description
EmbedFonts	LONG	(Optional) 0: Keep text unchanged, 3: convert text to vector
ImageCompression	LONG (SVGCompressionMode)	(Optional) Image Compression

The TIFF Parameters

Key	Value	Description
Compression	LONG	(Optional) 1: No compression 3: CCITT/Fax3 4: CCITT/Fax4 32773: PackBits 32946: ZIP
SaveLayers	LONG (boolean)	False: The document will be reduced to a background layer before saving. True: Every layer of the document will be exported as separate image.

The WebP Parameters

Key	Value	Description
Compress	LONG, range [0;100]	(Optional) The resulting image quality.
Filter	LONG (boolean)	(Optional) True: A prefilter is applied.

Dictionaries for IPLColor

Gradient Dictionary

The Gradient dictionary defines the appearance of gradients. If a property or method expects an IPLColor, you can usually also use a Gradient dictionary instead.

Key	Value	Description
ColorMode	LONG (ColorSpaceMode)	(Optional) The colors of the gradient will be interpolated in this color model.
Colors	IPLColor *[] (variable, RGB)	An array of colors defining the color stops. If this property is set, you can use an array of float arrays instead. The float array data will be interpreted as RGB.
Gammas	Float[number of colors[- 1]], range]0;1[	(Optional) Gamma value used for interpolation of two colors. The number of float values must be equal to or 1 less than the number of colors.
Interpolation	LONG (GradientInterpolation)	(Optional)The interpolation used to interpolate the colors.
Name	String	(Optional)Name of the gradient.
Points	float[4]	(Optional)Start and end point of gradient in a unity coordinate system. If not set, [0 0.5 1 0.5] will be used.
Scale	float	(Optional) A circular gradient allows scaling the gradient perpendicular to the line defining it. Values smaller than 1 compress the gradient, values larger than 1 stretch it. Default: 1

Key	Value	Description
Spread	LONG (GradientSpread)	(Optional)Repeating behavior of the gradient. Default value is GSContinue.
Stops	float[number of colors], range [0;1]	The stop positions of the colors. The positions must be ascending.
Type	LONG (GradientType)	(Optional)The type of gradient. Default value is GTLinear.

Dictionaries for Text

Text Styles Dictionary

The text styles dictionary is used to define a character or a paragraph styles. Character styles must not and paragraph styles must have a paragraph attribute.

Key	Value	Description
Name	String	Name of the text styles. Text styles must have a unique name.
Parent	String	(Optional) The name of the parent style. Attributes which are equal to the parent style, are inherited. The parent style must exist.
Follow	String	(Optional, only paragraph styles)The name of the following style.
Attributes	IPLDictionary *	The text attributes of the style.

Text Attributes Dictionary

The text styles dictionary is used to define a character or a paragraph styles. Character styles must not and paragraph styles must have a paragraph attribute.

Key	Value	Description
Attachment	IPLDictionary *	(Optional)A text attribute defining an attachment like page number, document name, … Attachments my only be assigned to a single character with the hex value 0x02 which symbolizes attachments in the text.
AutoKerning	LONG	(Optional) != 0: use the font kerning as it is defined in the font. Default value: 1
Baseline	float	(Optional) Distance of the text to the baseline.
Color	IPLColor * (variable, count)	The color of the text.
Font	IPLDictionary *	The font dictionary defining the font.
Kerning	float	(Optional) Manual kerning: additional distance between two characters.
Ligatures	LONG	(Optional) != 0: use the ligatures as defined in the font. Default value: 1
Outline	IPLDictionary *	(Optional) Outline dictionary. If this key exists, the text will be outlined.
Paragraph	IPLDictionary *	The paragraph attribute. The paragraph attribute must not change inside a paragraph.
Superscript	LONG	(Optional) -1: subscript, 0: normal, 1: superscript
TextStyle	String	(Optional) The name of the character style assigned to

Key	Value	Description
		the text.
Underline	IPLDictionary *	(Optional) Underline dictionary.

Dictionaries Used in Text Attributes

The Attachment Dictionary

Text attributes containing an attachment attribute must only be assigned to a single character and this character must be 0x02 which symbolizes text attachments in the text.

Key	Value	Description
Type	String	There are the following types: - Date - PageNumber - PageCount - Document name

The other keys in this dictionary depend on the type.

Date

Key	Value	Description
Date	String	(Optional) The date has the format "day:month:year"
DayOffset	LONG	(Optional) The number of days added to the date.
Format	LONG	(Optional) != 0: Use long format. Default: Use short format.
MonthOffset	LONG	(Optional) The number of months added to the date.
YearOffset	LONG	(Optional) The number of years added to the date.

DocumentName

The document name doesn't have any additional keys.

PageCount

Key	Value	Description
Offset	LONG	(Optional) An offset added to the page count.

PageNumber

Key	Value	Description
Offset	LONG	(Optional) An offset added to the page number.

The Font Dictionary

Key	Value	Description
FamilyName	String	The name of the font family
PostScriptName	String	The postscript name of the font.
Scale	float	(Optional) A horizontal scaling of the font. 1 means no additional scaling, 0.5 halves the character width, 2 doubles it.
Size	float	The size of the font.
Style	LONG	(Optional) 0: No special style, 1: italic

Key	Value	Description
Weight	LONG, range]0;1000]	Weight of the font. 300 is light, 400 is normal/regular, 700 is bold.
Width	LONG, range]0;1000]	Width of the font. 300 is condensed, 500 is medium and 700 is expanded.

If PostScriptName is set and a font with that name exists, FamilyName, Style, Weight and Width don't have to be set, because they are implicitly defined by the properties of that font.

The Outline Dictionary

The Outline dictionary controls the appearance of outlined text.

Key	Value	Description
Color	IPLColor * (variable, count)	The color of the outline.
LineStyle	IPLLineStyle *	The line style of the outline.

The Paragraph Dictionary

The Outline dictionary controls the formatting of paragraphs. It must not change inside a paragraph.

Key	Value	Description
After	float	(Optional) An additional space after a paragraph. The default value is 0.
Alignment	LONG (ParagraphAlignment)	The alignment of the paragraph. The default value is PALeft (0).
Before	float	(Optional) An additional space before a paragraph. The default value is 0.
Connect	LONG (Boolean)	(Optional) True: The paragraph will be on the same page as the next paragraph. The default value is false.
FirstIndent	float	(Optional) The indention of the first line of the paragraph. This value must be larger than or equal to 0. The default value is 0.
FixLine	LONG (Boolean)	(Optionally) True: The Line key controls the distance between two baselines inside a paragraph. False: The Line key is an additional offset between two lines. The default value is false.
LeftIndent	float	(Optional) The indention of all lines of the paragraph except the first one. This value must be larger than or equal to 0. The default value is 0.
Line	float	(Optional) The line distance between two lines of the paragraph. FixLine controls its exact meaning. The default value is 0.
Register	LONG (ParagraphRegister)	(Optional) Controls whether the lines of the paragraph should be placed on the line register. The default value is PRNone (0).
RightIndent	float	(Optional) The right indention of the paragraph. If the value is larger than 0, it is relative to the left edge of text layer. In that case it has to be larger than the left indent. If the value is smaller than 0, it is relative to the right edge of the text layer. The default value is 0.
StickStart	LONG	(Optional) The number of following lines, that have to be on the same page as the first paragraph line. The default value is 0.

Key	Value	Description
StickEnd	LONG	(Optional) The number of preceding lines, that have to be on the same page as the last paragraph line. The default value is 0.
Style	String	(Optional) The name of the paragraph style assigned to the text.
Tabs	IPLDictionary *[]	(Optional) An array of tab dictionaries, that define the tabs for the paragraph.

If PostScriptName is set and a font with that name exists, FamilyName, Style, Weight and Width don't have to be set, because they are implicitly defined by the properties of that font. In other words: PostScriptName will override these settings.

The Tab Dictionary

The Tab dictionary defines the properties of a tab..

Key	Value	Description
Type	LONG (TabType)	(Optional) The tab type (left, right, …) The default value is TTLeft (0).
Position	float	The position of the tab. The position must be larger than or equal to 0.
Filler	String	(Optional) Fill character used for the tab space. Only the first character of the string is used. The default value is no fill character.
Decimal	String	(Optional) If the type is TTDecimal, the value defines the decimal character, that will be used for alignment. Only the first character of the string is used. The default value is ".".

The Underline Dictionary

The Outline dictionary controls the appearance of outlined text.

Key	Value	Description
Color	IPLColor * (variable, count)	(Optional) The color of the outline. If absent, the text color will be used.
LineStyle	IPLLineStyle *	(Optional) The line style of the outline. If absent, the default value for the font will be used as line width.
Position	float	(Optional) Offset factor for underline position. Positive values move the line up. Default: 0

Vector Attributes Dictionary

The vector attributes dictionary is used to define the appearance of a vector layer.

Key	Value	Description
FillColor	IPLColor * (variable, RGB)	(Optional) The fill color. If it is missing, the vector layer will not be filled.
LineColor	IPLColor * (variable, RGBs)	(Optional) The line color. If it is missing, the vector layer will not be stroked.
LineStyle	IPLLineStyle * (variable)	(Optional) The line style. If it is missing, the vector layer will not be stroked.

Other Dictionaries

Copy Options Dictionary

These are the available options for copying layer data to the clipboard (IPLDocument, IPLPage, IPLLayer, IPLLayerArray). Not all options are applicable in every situation.

Key	Value	Description
Clear	LONG (Boolean)	(Optional) True: Delete the copied parts after copying. True is only allowed, if • used with a single image layer (IPLImage) • used with a layer array (IPLLayerArray) and "Selection" is set to false. Default: false
Defringe	LONG (Boolean)	(Optional) True: Remove the color fringe at the edge of the copied selection. This is only applicable, if "Selection" is true and the command is applied to • a single image layer (IPLImage) • a page (IPLPage) or documents (IPLDocument) Default: false
Selection	LONG (Boolean)	(Optional) False: Copy the selected layers. True: Copy the contents of the selection, if there is any. If there is no selection, the selected layers will be copied. True is not allowed for a list of layers. Default: false

Adjustments

Adjustments are a subset of the available operations, that can be applied to images. The parameters of the adjustments are stored in IPLDictionaries. All adjustment dictionaries contain the key "Type" whose value is the name of the operation as string. The other keys and values are the parameters of the operation.

The available adjustments are listed in *Operations applicable to images and as adjustments*.