

MBS Real Studio Tiff 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 Tiff Plugin

0.2 Content

- 1 List of all topics 3
- 2 All items in this plugin 9
- 4 List of all classes 75
- 5 List of all global methods 77

Chapter 1

List of Topics

• 2 Pictures Import and Export	9
– 2.1 Globals	9
* 2.1 Split1BitFileMBS(f as folderitem, fc as folderitem, fm as folderitem, fy as folderitem, fk as folderitem, width as integer, height as integer, CallbackTarget as object, CacheSizeRead as integer, CacheSizeWrite as integer) as integer	9
* 2.1 Split1BitFileMBS(f as folderitem, fc as folderitem, fm as folderitem, fy as folderitem, fk as folderitem, width as integer, height as integer, CallbackTarget as object, CacheSizeRead as integer, CacheSizeWrite as integer, ReadLines as integer, WriteLines as integer) as integer	10
• 3 TIFF	13
– 3.2 class TiffPictureMBS	22
* 3.2.1 AddCustomTag(Tag as integer, FieldReadCount as integer, FieldWriteCount as integer, FieldType as integer, FieldBit as integer, OkToChange as integer, PassCount as integer, FieldName as string) as boolean	22
* 3.2.1 AddImage as boolean	23
* 3.2.1 close	24
* 3.2.1 CombinePictureWithMask as picture	24
* 3.2.1 Create(file as folderitem) as boolean	24
* 3.2.1 Create(file as folderitem, endian as integer) as boolean	25
* 3.2.1 CreateString(Size as integer) as boolean	26
* 3.2.1 CreateString(Size as integer, Mode as string) as boolean	27
* 3.2.1 Flush	27
* 3.2.1 FlushData	27
* 3.2.1 GetColorMap(byref red as memoryblock, byref green as memoryblock, byref blue as memoryblock) as boolean	27

* 3.2.1 GetColorProfile as string	28
* 3.2.1 GetData(Tag as integer) as string	28
* 3.2.1 GetField(Tag as integer, mem as memoryblock) as boolean	28
* 3.2.1 GetFieldByte(Tag as integer, byref value as integer) as boolean	29
* 3.2.1 GetFieldCount(Tag as integer, byref count as integer, mem as memoryblock) as boolean	29
* 3.2.1 GetFieldDefaultedByte(Tag as integer, byref value as integer) as boolean	30
* 3.2.1 GetFieldDefaultedDouble(Tag as integer, byref value as double) as boolean	30
* 3.2.1 GetFieldDefaultedInteger(Tag as integer, byref value as integer) as boolean	31
* 3.2.1 GetFieldDefaultedShort(Tag as integer, byref value as integer) as boolean	31
* 3.2.1 GetFieldDefaultedSingle(Tag as integer, byref value as Single) as boolean	32
* 3.2.1 GetFieldDefaultedString(Tag as integer, byref value as String) as boolean	32
* 3.2.1 GetFieldDouble(Tag as integer, byref value as double) as boolean	33
* 3.2.1 GetFieldInteger(Tag as integer, byref value as integer) as boolean	33
* 3.2.1 GetFieldMemory(Tag as integer, byref ItemCount as integer) as memoryblock	33
* 3.2.1 GetFieldShort(Tag as integer, byref value as integer) as boolean	34
* 3.2.1 GetFieldSingle(Tag as integer, byref value as Single) as boolean	34
* 3.2.1 GetFieldString(Tag as integer, byref value as string) as boolean	34
* 3.2.1 GetXMP as string	35
* 3.2.1 ImageCount as integer	36
* 3.2.1 ImageIndex as integer	36
* 3.2.1 IsLastImage as boolean	36
* 3.2.1 MirrorVertical(output as TiffPictureMBS) as boolean	36
* 3.2.1 NextImage as boolean	37
* 3.2.1 NextImage(HeaderOnly as boolean) as boolean	37
* 3.2.1 Open(file as folderitem) as boolean	37
* 3.2.1 Open(file as folderitem, Mode as string) as boolean	38
* 3.2.1 OpenString(data as string) as boolean	38
* 3.2.1 OpenString(data as string, Mode as string) as boolean	39
* 3.2.1 ReadBW as boolean	39
* 3.2.1 ReadBW(left as integer, top as integer, width as integer, height as integer) as boolean	40
* 3.2.1 ReadPreviewBW as boolean	41
* 3.2.1 ReadPreviewBW(left as integer, top as integer, width as integer, height as integer) as boolean	42
* 3.2.1 ReadPreviewRGB(ReduceFactor as integer) as boolean	42
* 3.2.1 ReadRGB as boolean	43
* 3.2.1 ReadRGB(byref ErrorMessage as string) as memoryblock	43
* 3.2.1 ReadRGBMemoryBegin(byref ErrorMessage as string) as boolean	44
* 3.2.1 ReadRGBMemoryEnd	44
* 3.2.1 ReadRGBMemoryStep(x as integer, y as integer, width as integer, height as integer) as memoryblock	44

* 3.2.1 ReadWithLUT(ColorLookupTable() as color) as boolean	45
* 3.2.1 ReadWithLUT(ColorLookupTable() as color, left as integer, top as integer, width as integer, height as integer) as boolean	46
* 3.2.1 RewriteDirectory as boolean	46
* 3.2.1 SaveImage as boolean	46
* 3.2.1 Scanline(index as integer, sample as integer = 0) as memoryblock	47
* 3.2.1 Scanline(mem as memoryblock, index as integer, sample as integer = 0) as boolean	47
* 3.2.1 Scanlines(index as integer, count as integer, sample as integer = 0) as memoryblock	47
* 3.2.1 SetColorMap(red as memoryblock, green as memoryblock, blue as memoryblock) as boolean	48
* 3.2.1 SetColorProfile(ProfileData as String) as boolean	48
* 3.2.1 SetData(Tag as integer, data as string) as boolean	48
* 3.2.1 SetFieldByte(Tag as integer, value as integer) as boolean	48
* 3.2.1 SetFieldDouble(Tag as integer, value as double) as boolean	49
* 3.2.1 SetFieldInteger(Tag as integer, value as integer) as boolean	49
* 3.2.1 SetFieldMemory(Tag as integer, ItemCount as integer, data as memoryblock) as boolean	50
* 3.2.1 SetFieldShort(Tag as integer, value as integer) as boolean	50
* 3.2.1 SetFieldSingle(Tag as integer, value as Single) as boolean	50
* 3.2.1 SetFieldString(Tag as integer, value as string) as boolean	51
* 3.2.1 SetImageIndex(index as integer) as boolean	52
* 3.2.1 SetImageIndex(index as integer, HeaderOnly as boolean) as boolean	53
* 3.2.1 SetXMP(ProfileData as String) as boolean	53
* 3.2.1 WriteBW as boolean	53
* 3.2.1 WriteGray as boolean	55
* 3.2.1 WriteRGB as boolean	56
* 3.2.2 BitsPerSample as integer	57
* 3.2.2 BytesPerRow as Integer	58
* 3.2.2 Compression as Integer	58
* 3.2.2 Copyright as String	58
* 3.2.2 DateTime as String	58
* 3.2.2 DocumentName as String	59
* 3.2.2 ExtraSamples as MemoryBlock	59
* 3.2.2 FillOrder as Integer	59
* 3.2.2 height as integer	60
* 3.2.2 HorizontalPosition as Single	60
* 3.2.2 HorizontalResolution as Single	60
* 3.2.2 HostComputer as String	61
* 3.2.2 ImageDescription as String	61
* 3.2.2 InputBuffer as String	61
* 3.2.2 Make as String	61
* 3.2.2 mask as picture	62

* 3.2.2 Model as String	62
* 3.2.2 Orientation as integer	62
* 3.2.2 OutputBuffer as String	64
* 3.2.2 PageName as String	64
* 3.2.2 Photometric as integer	64
* 3.2.2 pict as picture	66
* 3.2.2 PlanarConfig as Integer	67
* 3.2.2 ResolutionUnit as Integer	67
* 3.2.2 RowsPerStrip as Integer	67
* 3.2.2 SamplesPerPixel as integer	67
* 3.2.2 Software as String	68
* 3.2.2 Version as Integer	68
* 3.2.2 VersionString as String	68
* 3.2.2 VerticalPosition as Single	69
* 3.2.2 VerticalResolution as Single	69
* 3.2.2 width as integer	69
* 3.2.2 YieldTicks as Integer	70
* 3.2.3 Error(libModule as string, message as string)	70
* 3.2.3 Progress(line as integer, total as integer)	70
* 3.2.3 Warning(libModule as string, message as string)	71
– 3.1 Globals	13
* 3.1 CombineBitCMYKtoCMYKMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, Files() as FolderItem, scale as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, CacheSizeRead as integer) as integer	13
* 3.1 CombineBitCMYKtoRGBMBS(CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, Files() as FolderItem, scale as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, byref output as picture, CacheSizeRead as integer) as integer	13
* 3.1 CombineTiff1BitCMYKtoTiffMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, TiffData() as TiffPictureMBS, scale as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, ditherMode as integer = 0) as integer	15
* 3.1 CombineTiff1BitCMYKtoTiffMBS(dest as TiffPictureMBS, TiffData as TiffPictureMBS, scalex as double, scaley as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, ditherMode as integer = 0) as integer	15
* 3.1 CombineTiff1BitCMYKtoTiffSWMBS(dest as TiffPictureMBS, a as UInt32, TiffData as TiffPictureMBS, b as UInt32, c as UInt32, d as UInt32) as integer	16
* 3.1 CombineTiff8BitCMYKtoTiffMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, TiffData() as TiffPictureMBS, scale as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, ditherMode as integer = 0) as integer	16
* 3.1 CombineTiffCMYKtoCMYKMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, TiffData() as TiffPictureMBS) as integer	16

- * 3.1 CombineTiffCMYKtoRGBMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, TiffData() as TiffPictureMBS) as integer 19
- * 3.1 TIFFStringToPictureMBS(data as string) as picture 21
- * 3.1 TIFFStringToTiffPictureMBS(data as string) as TiffPictureMBS 21

Chapter 2

Pictures Import and Export

2.1 Globals

Split1BitFileMBS(f as folderitem, fc as folderitem, fm as folderitem, fy as folderitem, fk as folderitem, width as integer, height as integer, CallbackTarget as object, CacheSizeRead as integer, CacheSizeWrite as integer) as integer

Plugin Version: 6.1 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Splits a one bit CMYK file into different files.

Notes:

Source file has 1 bit for each channel.

Error code is returned which is 0 for no error.

CacheSize can be set to a value greater than 0. And it may make the process faster or slower depending on what you do.

CallbackTarget can be nil or must be an object with a method with the following declaration: "Progress(RowIndex as integer, RowCount as integer)"

Errorcodes:

See also:

- 2.1 Split1BitFileMBS(f as folderitem, fc as folderitem, fm as folderitem, fy as folderitem, fk as folderitem,

- 1 not used
- 2 Failed to open f
- 3 Failed to open fc
- 4 Failed to open fm
- 5 Failed to open fy
- 6 Failed to open fk
- 7 Width<1
- 8 Height<1
- 9 Allocating read buffer failed
- 10 Allocating write buffer failed for c
- 11 Allocating write buffer failed for m
- 12 Allocating write buffer failed for y
- 13 Allocating write buffer failed for k
- 14 Read failed
- 15 Write failed for c
- 16 Write failed for m
- 17 Write failed for y
- 18 Write failed for k

width as integer, height as integer, CallbackTarget as object, CacheSizeRead as integer, CacheSizeWrite as integer, ReadLines as integer, WriteLines as integer) as integer 10

Split1BitFileMBS(f as folderitem, fc as folderitem, fm as folderitem, fy as folderitem, fk as folderitem, width as integer, height as integer, CallbackTarget as object, CacheSizeRead as integer, CacheSizeWrite as integer, ReadLines as integer, WriteLines as integer) as integer

Plugin Version: 6.3 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Splits a one bit CMYK file into different files.

Notes:

Source file has 1 bit for each channel.

Error code is returned which is 0 for no error.

CacheSize can be set to a value greater than 0. And it may make the process faster or slower depending on what you do.

CallbackTarget can be nil or must be an object with a method with the following declaration: "Progress(RowIndex as integer, RowCount as integer)"

ReadLines and WriteLines define how many rows to read in one I/O operation.

Errorcodes:

- 1 not used
- 2 Failed to open f
- 3 Failed to open fc
- 4 Failed to open fm
- 5 Failed to open fy
- 6 Failed to open fk
- 7 Width<1
- 8 Height<1
- 9 Allocating read buffer failed
- 10 Allocating write buffer failed for c
- 11 Allocating write buffer failed for m
- 12 Allocating write buffer failed for y
- 13 Allocating write buffer failed for k
- 14 Read failed
- 15 Write failed for c
- 16 Write failed for m
- 17 Write failed for y
- 18 Write failed for k

See also:

- 2.1 Split1BitFileMBS(f as folderitem, fc as folderitem, fm as folderitem, fy as folderitem, fk as folderitem, width as integer, height as integer, CallbackTarget as object, CacheSizeRead as integer, CacheSizeWrite as integer) as integer

Chapter 3

TIFF

3.1 Globals

CombineBitCMYKtoCMYKMBS(dest as **TiffPictureMBS**, **CyanChannel()** as integer, **MagentaChannel()** as integer, **YellowChannel()** as integer, **BlackChannel()** as integer, **Files()** as **FolderItem**, scale as double, width as integer, height as integer, **X1** as integer, **Y1** as integer, **X2** as integer, **Y2** as integer, **CacheSizeRead** as integer) as integer

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Combines 1bit raw image files to one CMYK tiff.

Notes:

All arrays have the same size specifying for each 1bit grayscale source image the colors to be used in the final image.

Result image is written to the output picture which is created. Error code is returned which is 0 for no error. CacheSize can be set to a value greater than 0. And it may make the process faster or slower depending on what you do.

CombineBitCMYKtoRGBMBS(**CyanChannel()** as integer, **MagentaChannel()** as integer, **YellowChannel()** as integer, **BlackChannel()** as integer, **Files()** as **FolderItem**, scale as double, width as integer, height as integer, **X1** as integer, **Y1** as integer, **X2** as integer, **Y2** as integer, byref output as picture, **CacheSizeRead** as integer) as integer

Plugin Version: 6.0 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Combines 1bit raw image files to one 8bit RGB tiff.

Example:

```
dim t(4) as FolderItem
dim dC(4), dM(4), dY(4), dK(4) As Double
dim nC(4), nM(4), nY(4), nK(4) as integer
dim i as integer
dim p as Picture
```

```
//Cyan
dK(0)=0.0
dC(0)=1.0
dM(0)=0.0
dY(0)=0.0
//Magenta
dK(1)=0.00
dC(1)=0.0
dM(1)=1.0
dY(1)=0.0
//Yellow
dK(2)=0.00
dC(2)=0.0
dM(2)=0.0
dY(2)=1.0
//Black
dK(3)=1.00
dC(3)=0.0
dM(3)=0.0
dY(3)=0.0
//Pantone, Sonderfarbe, S0
dK(4)=0.00
dC(4)=0.60
dM(4)=0.35
dY(4)=0.15
```

```
//Bilder
t(0)=GetFolderItem("test.Cyan.bit")
t(1)=GetFolderItem("test.Magenta.bit")
t(2)=GetFolderItem("test.Yellow.bit")
t(3)=GetFolderItem("test.Black.bit")
t(4)=GetFolderItem("test.S0.bit")
```

```
for i=0 to 4
nC(i)=dC(i)*1000.0
nM(i)=dM(i)*1000.0
nY(i)=dY(i)*1000.0
nK(i)=dK(i)*1000.0
next
```

```
// Scale 1/n, 1/3, 1/2, 1, 2, 3, n
```

```
Title=str(CombineBitCMYKtoRGBMBS(nC,nM,nY,nK, t, 1, 545,567,0,0,545,567,p,0))
Backdrop=p
Width=p.Width
Height=p.Height
```

Notes:

All arrays have the same size specifying for each 1bit grayscale source image the colors to be used in the final image.

Result image is written to the output picture which is created. Error code is returned which is 0 for no error. CacheSize can be set to a value greater than 0. And it may make the process faster or slower depending on what you do.

CombineTiff1BitCMYKtoTiffMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, TiffData() as TiffPictureMBS, scale as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, ditherMode as integer = 0) as integer

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Combines 1bit tiff image files to one CMYK tiff.

Notes:

All arrays have the same size specifying for each 1bit grayscale source image the colors to be used in the final image.

Result image is written to the output picture which is created. Error code is returned which is 0 for no error. Compression can be set before data is written to current tiff which is for output. The tiff object for output must be perfectly setup before using this function.

CombineTiff1BitCMYKtoTiffMBS(dest as TiffPictureMBS, TiffData as TiffPictureMBS, scalex as double, scaley as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, ditherMode as integer = 0) as integer

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Combines a one bit gray tiff image to one 1 bit gray tiff.

Notes:

Basicly only scales the data.

Result image is written to the current tiffpicture. Error code is returned which is 0 for no error. You need to set yourself all needed tiffpicture parameters for the output image and you must create it before. The plugin sets width and height for the output tiff.

CombineTiff1BitCMYKtoTiffSWMBS(dest as TiffPictureMBS, a as UInt32, TiffData as TiffPictureMBS, b as UInt32, c as UInt32, d as UInt32) as integer

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Combines a one bit gray tiff image to one 1 bit gray tiff.

Notes: Internal function.

CombineTiff8BitCMYKtoTiffMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, TiffData() as TiffPictureMBS, scale as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, ditherMode as integer = 0) as integer

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Combines 8bit tiff image files to one CMYK tiff.

Notes:

All arrays have the same size specifying for each 8bit grayscale source image the colors to be used in the final image.

Result image is written to the output picture which is created. Error code is returned which is 0 for no error. Compression can be set before data is written to current tiff which is for output. The tiff object for output must be perfectly setup before using this function.

CombineTiffCMYKtoCMYKMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, TiffData() as TiffPictureMBS) as integer

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Combines Grayscale tiff images to one CMYK tiff.

Example:

```
const PLANARCONFIG_ CONTIG=1
const PHOTOMETRIC_ RGB=2
const FILLORDER_ MSB2LSB=1
const PHOTOMETRIC_ SEPARATED=5
```

```
dim u,v,ot,t(4) as TiffPictureMBS
dim f as FolderItem
dim pnach as Picture
dim dC(4), dM(4), dY(4), dK(4) As Double
dim nC(4), nM(4), nY(4), nK(4) as integer
dim i as integer
```

```
//Cyan
dK(0)=0.0
dC(0)=1.0
dM(0)=0.0
dY(0)=0.0
//Magenta
dK(1)=0.00
dC(1)=0.0
dM(1)=1.0
dY(1)=0.0
//Yellow
dK(2)=0.00
dC(2)=0.0
dM(2)=0.0
dY(2)=1.0
//Black
dK(3)=1.00
dC(3)=0.0
dM(3)=0.0
dY(3)=0.0
//Pantone, S0
dK(4)=0.00
dC(4)=0.60
dM(4)=0.35
dY(4)=0.15
```

```
//Bilder
f=GetFolderItem("test.Cyan.tif")
t(0)=f.OpenAsTiffMBS

f=GetFolderItem("test.Magenta.tif")
t(1)=f.OpenAsTiffMBS

f=GetFolderItem("test.Yellow.tif")
t(2)=f.OpenAsTiffMBS

f=GetFolderItem("test.Black.tif")
t(3)=f.OpenAsTiffMBS

f=GetFolderItem("test.S0.tif")
```

```

t(4)=f.OpenAsTiffMBS

for i=0 to 4
nC(i)=dC(i)*1000.0
nM(i)=dM(i)*1000.0
nY(i)=dY(i)*1000.0
nK(i)=dK(i)*1000.0
next

f=GetFolderItem("resultCMYK.tif")
ot=new TiffPictureMBS
if ot.Create(f) then
v=t(0)
u=ot
u.Width=v.Width
u.Height=v.Height
u.RowsPerStrip=v.Height
u.BitsPerSample=8
u.SamplesPerPixel=4
u.ResolutionUnit=v.ResolutionUnit
u.HorizontalPosition=v.HorizontalPosition
u.HorizontalResolution=v.HorizontalResolution
u.VerticalPosition=v.VerticalPosition
u.VerticalResolution=v.VerticalResolution
u.Orientation=v.Orientation
u.PlanarConfig=PLANARCONFIG_ CONTIG
u.Photometric=PHOTOMETRIC_ SEPARATED
u.FillOrder=FILLORDER_ MSB2LSB
Title=str(CombineTiffCMYKtoCMYKMBS(ot, nC,nM,nY,nK,t))
ot.Close
end if

//Neues Bild anzeigen
pnach=f.OpenAsPicture
if pnach<>Nil then
Window1.Width=pNach.Width
Window1.Height=pNach.Height
Window1.Backdrop=pNach
end if

```

Notes:

All arrays have the same size specifying for each 8bit grayscale source image the colors to be used in the final image.

Result image is written to the current tiffpicture. Error code is returned which is 0 for no error.

You need to set yourself all needed tiffpicture parameters for the output image and you must create it before. The plugin sets width and height for the output tiff.

CombineTiffCMYKtoRGBMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, TiffData() as TiffPictureMBS) as integer

Plugin Version: 12.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Combines Grayscale tiff images to one RGB tiff.

Example:

```
const PLANARCONFIG_ CONTIG=1
const PHOTOMETRIC_ RGB=2
const FILLORDER_ MSB2LSB=1

dim u,v,ot,t(4) as TiffPictureMBS
dim f as FolderItem
dim pnach as Picture
dim dC(4), dM(4), dY(4), dK(4) As Double
dim nC(4), nM(4), nY(4), nK(4) as integer
dim i as integer

//Cyan
dK(0)=0.0
dC(0)=1.0
dM(0)=0.0
dY(0)=0.0
//Magenta
dK(1)=0.00
dC(1)=0.0
dM(1)=1.0
dY(1)=0.0
//Yellow
dK(2)=0.00
dC(2)=0.0
dM(2)=0.0
dY(2)=1.0
//Black
dK(3)=1.00
dC(3)=0.0
dM(3)=0.0
dY(3)=0.0
//Pantone, S0
dK(4)=0.00
dC(4)=0.60
```

```

dM(4)=0.35
dY(4)=0.15

//Bilder
f=GetFolderItem("test.Cyan.tif")
t(0)=f.OpenAsTiffMBS

f=GetFolderItem("test.Magenta.tif")
t(1)=f.OpenAsTiffMBS

f=GetFolderItem("test.Yellow.tif")
t(2)=f.OpenAsTiffMBS

f=GetFolderItem("test.Black.tif")
t(3)=f.OpenAsTiffMBS

f=GetFolderItem("test.S0.tif")
t(4)=f.OpenAsTiffMBS

for i=0 to 4
nC(i)=dC(i)*1000.0
nM(i)=dM(i)*1000.0
nY(i)=dY(i)*1000.0
nK(i)=dK(i)*1000.0
next

f=GetFolderItem("resultRGB.tif")
ot=new TiffPictureMBS
if ot.Create(f) then
v=t(0)
u=ot
u.Width=v.Width
u.Height=v.Height
u.RowsPerStrip=v.Height
u.BitsPerSample=8
u.SamplesPerPixel=3
u.ResolutionUnit=v.ResolutionUnit
u.HorizontalPosition=v.HorizontalPosition
u.HorizontalResolution=v.HorizontalResolution
u.VerticalPosition=v.VerticalPosition
u.VerticalResolution=v.VerticalResolution
u.Orientation=v.Orientation
u.PlanarConfig=PLANARCONFIG_ CONTIG
u.Photometric=PHOTOMETRIC_ RGB
u.FillOrder=FILLORDER_ MSB2LSB
Title=str(CombineTiffCMYKtoRGBMBS(ot, nC,nM,nY,nK,t))
ot.Close
end if

```

```
//Neues Bild anzeigen
pnach=f.OpenAsPicture
if pnach<>Nil then
Window1.Width=pNach.Width
Window1.Height=pNach.Height
Window1.Backdrop=pNach
end if
```

Notes:

All arrays have the same size specifying for each 8bit grayscale source image the colors to be used in the final image.

Result image is written to the current tiffpicture. Error code is returned which is 0 for no error.

You need to set yourself all needed tiffpicture parameters for the output image and you must create it before.

TIFFStringToPictureMBS(data as string) as picture

Plugin Version: 6.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens a tiff string and returns the RGB image for this tiff file.

Notes:

Returns nil on any error.

This function works with most Tiff formats, but has problems with some like 16 bit CMYK.

TIFFStringToTiffPictureMBS(data as string) as TiffPictureMBS

Plugin Version: 6.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens a tiff string and returns a TiffPictureMBS object.

Notes:

if you want to read the picture after this line you need to call ReadBW, ReadRGB or use the Scanline methods.

Returns nil on any error.

3.2 class TiffPictureMBS

class TiffPictureMBS

Console & Web: Yes Mac: Yes, Win: No, Linux: Yes, . **Function:** A class for a Tiff picture.

Notes: Bases on libtiff.

3.2.1 Methods

AddCustomTag(Tag as integer, FieldReadCount as integer, FieldWriteCount as integer, FieldType as integer, FieldBit as integer, OkToChange as integer, PassCount as integer, FieldName as string) as boolean

Plugin Version: 8.7 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Adds a custom tag.

Example:

```

const TIFFTAG_ ASCIITAG = 40666
const TIFFTAG_ LONGTAG = 40667
const TIFFTAG_ SHORTTAG = 40668
const TIFFTAG_ RATIONALTAG = 40669
const TIFFTAG_ FLOATTAG = 40670
const TIFFTAG_ DOUBLETAG = 40671
const TIFFTAG_ BYTE = 40672

const TIFFTAG_ SOFTWARE=305

const TIFF_ BYTE=1
const TIFF_ ASCII=2
const TIFF_ SHORT=3 // integer 16 bit signed
const TIFF_ LONG=4 // integer 32 bit
const TIFF_ FLOAT=11
const TIFF_ DOUBLE=12

const FIELD_ CUSTOM = 65

dim t as new TiffPictureMBS

// open tiff

if not t.AddCustomTag(TIFFTAG_ ASCIITAG, -1, -1, TIFF_ ASCII, FIELD_ CUSTOM, 1, 0, "MyString")
then
MsgBox "AddCustomTag failed1."

```

```
end if
```

Notes: See tiff documentation for details.

AddImage as boolean

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes existing picture and header information to file and starts a new one.

Example:

```
dim t as TiffPictureMBS
dim f as FolderItem
dim p as Picture

f=SpecialFolder.Desktop.Child("test.tif")
t=new TiffPictureMBS
if t.Create(f) then

p=NewPicture(100,100,32)
p.Graphics.ForeColor=rgb(255,0,0)
p.Graphics.FillOval 0,0,100,100
t.Pict=p
if t.WriteRGB then
if t.AddImage then
p=NewPicture(100,100,32)
p.Graphics.ForeColor=rgb(0,0,255)
p.Graphics.FillOval 0,0,100,100
t.Pict=p
if t.WriteRGB then
MsgBox "Written multi picture tiff."
end if
end if
end if
end if
```

Notes:

Returns true on success and false on any error.

Calls TIFFWriteDirectory internally.

close

Plugin Version: 3.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Closes the Tiff handle.

Notes:

In 5.1 and older the destructor.

In 5.2 and later only closes the tiff handle so you can still read the pictures, the output or input buffer.

CombinePictureWithMask as picture

Plugin Version: 6.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns a new picture which is created using the picture and it's mask.

Example:

```
dim t as TiffPictureMBS
' ...
canvas1.backdrop=t.CombinePictureWithMask
```

Create(file as folderitem) as boolean

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new empty tiff file.

Example:

```
dim p as Picture
dim f as FolderItem
dim t as TiffPictureMBS

p=NewPicture(100,100,32)

f=SpecialFolder.Desktop.Child("test.tif")

t=new TiffPictureMBS
t.Pict=p
```

```

if t.Create(f) then
if t.WriteRGB then
t.Close
MsgBox "Ok"
f.Launch
end if
end if

```

Notes:

Returns true on success.

This function uses paths like all Tiff functions, so be sure there are not two volumes named equal on a Mac OS Classic system.

See also:

- 3.2.1 Create(file as folderitem, endian as integer) as boolean

25

Create(file as folderitem, endian as integer) as boolean

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new empty tiff file.

Notes:

Returns true on success.

Endian settings:

- 0 Default (System)
- 1 BigEndian (Mac)
- 2 LittleEndian (Win)

This function uses paths like all Tiff functions, so be sure there are not two volumes named equal on a Mac OS Classic system.

See also:

- 3.2.1 Create(file as folderitem) as boolean

24

CreateString(Size as integer) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new string based tiff writer.

Example:

```
Function PictureToTiffStringMBS(pic as picture) As string
dim t as TiffPictureMBS
```

```
t=new TiffPictureMBS
if t.CreateString(& h100000) then
t.Pict=pic
if t.WriteRGB then
t.Close
Return t.OutputBuffer
end if
end if
End Function
```

Notes:

Same as the Create() function, but memory based. You can now use functions like Scanline(), WriteSW() or WriteRGB() to put the picture data.

Returns true on success.

The Warning and Error events may show you reasons why it does not work.

The size parameter you pass in is a guess for the initial size of the memory block used. If more data is written, the memory block is resized, but it is quite slow to resize a memoryblock, so make a good guess!

You can and should use this function to write yourself a PictureToTiffString function. The plugin can not well make such a function as there are thousands of possible parameters combination you may want to use. (compared to the JPEG library where you only have the compression level parameter.)

See also:

- 3.2.1 CreateString(Size as integer, Mode as string) as boolean

CreateString(Size as integer, Mode as string) as boolean

Plugin Version: 5.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Creates a new string based tiff writer.

Notes:

Same as the other CreateString method, but you can pass a mode string to the library.

mode="wb" for big endian and mode="wl" for little endian.

See also:

- 3.2.1 CreateString(Size as integer) as boolean

26

Flush

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Flush causes any pending writes for the specified file (including writes for the current directory) to be done.

Notes: In normal operation this call is never needed - the library automatically does any flushing required.

FlushData

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** FlushData flushes any pending image data for the specified file to be written out; directory-related data are not flushed.

Notes: In normal operation this call is never needed — the library automatically does any flushing required.

GetColorMap(byref red as memoryblock, byref green as memoryblock, byref blue as memoryblock) as boolean

Plugin Version: 9.6 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Queries the color map for a paletten image.

Notes:

The memoryblock must be $2^{\text{bitspersample}} * 2$ bytes big.

Returns true on success and false on failure.

GetColorProfile as string

Plugin Version: 7.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the color profile data stored in the tiff file.

Example:

```
dim f as FolderItem
dim t as TiffPictureMBS
dim s as string
dim p as CMPProfileMBS

f=SpecialFolder.Desktop.Child("horsehead_ steinberg_ big.tif")

t=f.OpenAsTiffMBS

s=t.GetColorProfile

p=LCMSMBS.OpenProfileFromData(s)

MsgBox p.Name
```

Notes: Returns "" on any error.

GetData(Tag as integer) as string

Plugin Version: 8.5 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the string stored for this tag.

GetField(Tag as integer, mem as memoryblock) as boolean

Plugin Version: 7.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

The memoryblock you pass in must be big enough to hold whatever data the library stores there.

GetFieldByte(Tag as integer, byref value as integer) as boolean

Plugin Version: 8.7 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldCount(Tag as integer, byref count as integer, mem as memoryblock) as boolean

Plugin Version: 8.7 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Example:

```
'TIFF Tag ImageID
```

```
,
```

```
'IFD Image
```

```
'Code 32781 (hex 0x800D)
```

```
'Name ImageID
```

```
'Type ASCII
```

```
'Count N
```

```
'Default None
```

```
'Description
```

```
,
```

```
'OPI-related.
```

```
,
```

```
'ImageID is the full pathname of the original, high-resolution image, or any other identifying string that uniquely identifies the original image.
```

```
,
```

```
'The high-resolution image is not required to be in TIFF format. It can be in any format that an OPI Consumer wishes to support.
```

```
DIm tiffImport As TiffPictureMBS
```

```
Dim xx as string
```

```
tiffImport = New TiffPictureMBS
```

```
Call tiffImport.Open(SpecialFolder.Desktop.Child("test.tif"))
```

```

// the memoryblock is a storage for the data. In this case a pointer to the CString
dim m as MemoryBlock=NewMemoryBlock(4)
dim count as integer

if tiffImport.GetFieldCount(32781, count, m) then
  MsgBox str(count)
  MsgBox m.Ptr(0).CString(0)
end if

tiffImport.close

```

Notes:

This is the special version using memoryblock so you can use it for reading values with a count value.

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldDefaultedByte(Tag as integer, byref value as integer) as boolean

Plugin Version: 8.7 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success. May return a default value.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldDefaulted* is identical to GetField*, except that if a tag is not defined in the current directory and it has a default value, then the default value is returned.

GetFieldDefaultedDouble(Tag as integer, byref value as double) as boolean

Plugin Version: 7.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success. May return a default value.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldDefaulted* is identical to GetField*, except that if a tag is not defined in the current directory and it has a default value, then the default value is returned.

GetFieldDefaultedInteger(Tag as integer, byref value as integer) as boolean

Plugin Version: 7.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success. May return a default value.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldDefaulted* is identical to GetField*, except that if a tag is not defined in the current directory and it has a default value, then the default value is returned.

GetFieldDefaultedShort(Tag as integer, byref value as integer) as boolean

Plugin Version: 7.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success. May return a default value.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldDefaulted* is identical to GetField*, except that if a tag is not defined in the current directory and it has a default value, then the default value is returned.

GetFieldDefaultedSingle(Tag as integer, byref value as Single) as boolean

Plugin Version: 8.7 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success. May return a default value.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldDefaulted* is identical to GetField*, except that if a tag is not defined in the current directory and it has a default value, then the default value is returned.

GetFieldDefaultedString(Tag as integer, byref value as String) as boolean

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Example:

```
dim t as TiffPictureMBS
dim f as FolderItem
dim s as string

f=SpecialFolder.Desktop.Child("ChristianSchmitz.tif")
t=f.OpenAsTiffMBS

const TIFFTAG_SOFTWARE=305

if t.GetFieldDefaultedString(TIFFTAG_SOFTWARE, s) then
MsgBox "TIFFTAG_SOFTWARE" +EndOfLine+s
end if
```

Notes:

Please look for Tag values in the tiff specification.

Returns true on success. May return a default value.

Please make sure to use the correct setter depending on data type associated with the tag.

The string is returned with ascii encoding. You may need to define a different encoding if this is not correct.

GetFieldDefaulted* is identical to GetField*, except that if a tag is not defined in the current directory and it has a default value, then the default value is returned.

GetFieldDouble(Tag as integer, byref value as double) as boolean

Plugin Version: 7.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldInteger(Tag as integer, byref value as integer) as boolean

Plugin Version: 7.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldMemory(Tag as integer, byref ItemCount as integer) as memoryblock

Plugin Version: 8.7 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

This is the special version using memoryblock so you can use it for reading values with a count value.

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldShort(Tag as integer, byref value as integer) as boolean

Plugin Version: 7.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldSingle(Tag as integer, byref value as Single) as boolean

Plugin Version: 8.7 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

GetFieldString(Tag as integer, byref value as string) as boolean

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the value associated with the given tag id.

Example:

```

dim t as TiffPictureMBS
dim f as FolderItem
dim s as string

f=SpecialFolder.Desktop.Child("test.tif")
t=f.OpenAsTiffMBS

const TIFFTAG_SOFTWARE=305

if t.GetFieldString(TIFFTAG_SOFTWARE, s) then
MsgBox "TIFFTAG_SOFTWARE"+EndOfLine+s
end if

const TIFFTAG_HOSTCOMPUTER=316

```

```

if t.GetFieldString(TIFFTAG_ HOSTCOMPUTER, s) then
MsgBox "TIFFTAG_ SOFTWARE" +EndOfLine+s
end if

const TIFFTAG_ IMAGEDESCRIPTION=270

if t.GetFieldString(TIFFTAG_ IMAGEDESCRIPTION, s) then
MsgBox "TIFFTAG_ IMAGEDESCRIPTION" +EndOfLine+s
end if

const TIFFTAG_ MAKE=271

if t.GetFieldString(TIFFTAG_ MAKE, s) then
MsgBox "TIFFTAG_ MAKE" +EndOfLine+s
end if

const TIFFTAG_ ARTIST=315

if t.GetFieldString(TIFFTAG_ ARTIST, s) then
MsgBox "TIFFTAG_ ARTIST" +EndOfLine+s
end if

```

Notes:

Please look for Tag values in the tiff specification.

Returns true on success. May return a default value.

Please make sure to use the correct setter depending on data type associated with the tag.

The string is returned with ascii encoding. You may need to define a different encoding if this is not correct.

GetXMP as string

Plugin Version: 7.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the xmp metadata package stored in the tiff file.

Example:

```

dim f as FolderItem
dim t as TiffPictureMBS
dim s as string

f=SpecialFolder.Desktop.Child("test.tif")

```

```
t=f.OpenAsTiffMBS
```

```
s=t.GetXMP
```

```
MsgBox left(s,500)
```

Notes: Returns "" on any error.

ImageCount as integer

Plugin Version: 4.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Returns the number of images in the TIFF file.

Notes: Returns 0 on any error.

ImageIndex as integer

Plugin Version: 4.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The current image index.

Notes: 0 based.

IsLastImage as boolean

Plugin Version: 4.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Whether the current picture is the last picture.

Notes: Useful if you walk through all pictures using NextImage.

MirrorVertical(output as TiffPictureMBS) as boolean

Plugin Version: 7.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Mirrors the current picture to another tiff object.

Notes:

You may be able to pass the current tiff file as the output one if you have it open for read and write.
Returns true on success and false on failure.
Works for any color depth or color mode.

NextImage as boolean

Plugin Version: 4.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Loads the next image in the TIFF file.

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

See also:

- 3.2.1 NextImage(HeaderOnly as boolean) as boolean 37

NextImage(HeaderOnly as boolean) as boolean

Plugin Version: 4.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the next image.

Notes: If HeaderOnly is false the current picture is read into the pict& mask properties.

See also:

- 3.2.1 NextImage as boolean 37

Open(file as folderitem) as boolean

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens a tiff file for readonly access.

Notes:

You need to use the ReadRGB method or the Scanline property to get data from this file.

Returns true on success.

See also:

- 3.2.1 Open(file as folderitem, Mode as string) as boolean 38

Open(file as folderitem, Mode as string) as boolean

Plugin Version: 5.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens a tiff file for readonly access.

Example:

```
dim t as new TiffPictureMBS
if t.Open(file, "r") then
msgbox "OK"
end if
```

Notes:

Same as the other Open method, but you can pass a mode string to the library.

The open mode parameter can include the following flags in addition to the "r" (Read), "w" (Write), and "a" (Append) flags. Note however that option flags must follow the read-write-append specification.

See also:

- 3.2.1 Open(file as folderitem) as boolean

37

OpenString(data as string) as boolean

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens a tiff image located in a binary string.

Example:

Function TiffStringToPicture(data as string) As picture

```
dim t as MyTiffPictureMBS

t=new MyTiffPictureMBS

if t.OpenString(data) then
if t.ReadRGB then
return t.Pict
// you could add the mask here.
end if
end if
```

```
Return nil // failed
End Function
```

Notes:

Same as the Open() function, but memory based. You can now use functions like Scanline(), ReadSW() or ReadRGB() to get the picture data.

Returns true on success.

The Warning and Error events may show you reasons why it does not work.

The string you pass is saved in the data property of the class for later use.

You can and should use this function to write yourself a TiffStringToPicture function.

See also:

- 3.2.1 OpenString(data as string, Mode as string) as boolean

39

OpenString(data as string, Mode as string) as boolean

Plugin Version: 5.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Opens a tiff image located in a binary string.

Notes: Same as the other OpenString method, but you can pass a mode string to the library.

See also:

- 3.2.1 OpenString(data as string) as boolean

38

ReadBW as boolean

Plugin Version: 4.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the current picture in BW mode.

Example:

```
dim f as FolderItem
dim p as Picture
dim t as TiffPictureMBS
```

```
f=SpecialFolder.Desktop.Child("Multipage fax.tif")
t=f.OpenAsTiffMBS(true)

if t.ReadBW then
title=str(t.ImageCount)

p=t.pict
Title=str(p.Depth)
Canvas1.Backdrop=p
end if
```

Notes:

Returns true on success and false on any error.

Only if the current picture is a 1bit black& white picture, a new picture is created in the pict property and the data is copied inside.

Data in the TIFF file must be in 1 bit BW mode. Else use ReadRGB which does a lot of converting.

This method uses the YieldTicks property and may yield time to other threads.

See also:

- 3.2.1 ReadBW(left as integer, top as integer, width as integer, height as integer) as boolean 40

ReadBW(left as integer, top as integer, width as integer, height as integer) as boolean

Plugin Version: 8.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the current picture in BW mode.

Example:

```
dim f as FolderItem
dim p as Picture
dim t as TiffPictureMBS

f=SpecialFolder.Desktop.Child("Multipage fax.tif")
t=f.OpenAsTiffMBS(true)

if t.ReadBW(0,0,t.width/2,t.height/2) then
title=str(t.ImageCount)
```

```

p=t.pict
Title=str(p.Depth)
Canvas1.Backdrop=p
end if

```

Notes:

Returns true on success and false on any error.

Only if the current picture is a 1bit black& white picture, a new picture is created in the pict property and the data is copied inside.

Data in the TIFF file must be in 1 bit BW mode. Else use ReadRGB which does a lot of converting.

This method uses the YieldTicks property and may yield time to other threads.
See also:

- 3.2.1 ReadBW as boolean

39

ReadPreviewBW as boolean

Plugin Version: 9.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the current picture in BW mode as a preview.

Notes:

This method is faster than ReadBW, because it reads only every 8th line and every 8th pixel in each row. So the picture you get is much smaller, but may be enough for a preview.

Returns true on success and false on any error.

Only if the current picture is a 1bit black& white picture, a new picture is created in the pict property and the data is copied inside.

Data in the TIFF file must be in 1 bit BW mode. Else use ReadRGB which does a lot of converting.

This method uses the YieldTicks property and may yield time to other threads.
See also:

- 3.2.1 ReadPreviewBW(left as integer, top as integer, width as integer, height as integer) as boolean 42

ReadPreviewBW(left as integer, top as integer, width as integer, height as integer) as boolean

Plugin Version: 9.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the current picture in BW mode as a preview.

Notes:

This method is faster than ReadBW, because it reads only every 8th line and every 8th pixel in each row. So the picture you get is much smaller, but may be enough for a preview.

Returns true on success and false on any error.

Only if the current picture is a 1bit black& white picture, a new picture is created in the pict property and the data is copied inside.

Data in the TIFF file must be in 1 bit BW mode. Else use ReadRGB which does a lot of converting.

This method uses the YieldTicks property and may yield time to other threads.

See also:

- 3.2.1 ReadPreviewBW as boolean

41

ReadPreviewRGB(ReduceFactor as integer) as boolean

Plugin Version: 9.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads a CMYK or RGB picture and stores it in the pict and name properties.

Notes:

This method is faster than ReadRGB, because it reads only every ReduceFactorth line and every ReduceFactorth pixel in each row. So the picture you get is much smaller, but may be enough for a preview.

Returns true on success.

ReadRGB does converting on the picture data if needed so you can read CMYK, RGB, BW and other image data using this function.

This function works with most Tiff formats, but has problems with some like 16 bit CMYK.

ReadRGB as boolean

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads a CMYK or RGB picture and stores it in the pict and name properties.

Notes:

Returns true on success.

ReadRGB does converting on the picture data if needed so you can read CMYK, RGB, BW and other image data using this function.

This function works with most Tiff formats, but has problems with some like 16 bit CMYK.

See also:

- 3.2.1 ReadRGB(byref ErrorMessage as string) as memoryblock 43

ReadRGB(byref ErrorMessage as string) as memoryblock

Plugin Version: 9.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the image into a memoryblock as RGB.

Notes:

The Scanline() array gives you the raw uncompressed data. But this method decodes the data into a RGBA image.

The Memoryblock has the size of 4*width*height. Each pixel has one byte for red, one byte for green, one byte for blue and one byte for alpha. If no alpha is in the file, all pixels have the same alpha of 255.

If the decompression fails the function returns nil. (e.g. out of memory).

If the decoding fails, you also get an error message.

This function works with most Tiff formats, but has problems with some like 16 bit CMYK.

See also:

- 3.2.1 ReadRGB as boolean 43

ReadRGBMemoryBegin(byref ErrorMessage as string) as boolean

Plugin Version: 10.5 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Starts a RGBA reader for the given TIFF picture.

Notes:

Like ReadRGB, but with ReadRGBMemoryBegin, ReadRGBMemoryEnd and ReadRGBMemoryStep you can read the RGBA data in portions.

The Memoryblock returned by ReadRGBMemoryStep has the size of 4*width*height. Each pixel has one byte for red, one byte for green, one byte for blue and one byte for alpha. If no alpha is in the file, all pixels have the same alpha of 255.

If the decompression fails the function returns false. (e.g. out of memory).
If the decoding fails, you also get an error message.

This function works with most Tiff formats, but has problems with some like 16 bit CMYK.

ReadRGBMemoryEnd

Plugin Version: 10.5 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Cleans up the internal data structures created by ReadRGBMemoryBegin.

ReadRGBMemoryStep(x as integer, y as integer, width as integer, height as integer) as memoryblock

Plugin Version: 10.5 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the image into a memoryblock as RGBA.

Example:

```
dim t as new TiffPictureMBS
dim f as FolderItem = SpecialFolder.Desktop.Child("test.tif")
dim e as string

if t.Open(f) then
if t.ReadRGBMemoryBegin(e) then
```

```

dim w as integer = t.Width
dim h as integer = t.Height

dim p as new PictureMBS(w, h, PictureMBS.ImageFormatRGB)

for y as integer = 0 to h-1 step 100

dim m as MemoryBlock = t.ReadRGBMemoryStep(0, y, w, 100)

for i as integer = 0 to 99
p.RowInFormat(i+y, PictureMBS.ImageFormatRGBA)=m
next

next

t.ReadRGBMemoryEnd
Backdrop = p.coppicture(0,0,1000,1000) // display a portion
end if

end if

```

Notes:

As you see in the example, you can use this method to easily read huge TIFF files into a PictureMBS object.

If the decompression fails the function returns nil. (e.g. out of memory).

Please make sure that the range in x and y, width and height is right.

ReadWithLUT(ColorLookupTable() as color) as boolean

Plugin Version: 9.8 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the current picture with a Color Lookup Table.

Notes:

Returns true on success and false on any error.

Works with 8, 16 and 32 bits per sample. And with 1 or 3 samples per pixels (Gray or RGB). The lookup table has 256 entries for 8 bit and 65536 entries for 16/32 bit.

This method uses the YieldTicks property and may yield time to other threads.

See also:

- 3.2.1 ReadWithLUT(ColorLookupTable() as color, left as integer, top as integer, width as integer, height as integer) as boolean 46

ReadWithLUT(ColorLookupTable() as color, left as integer, top as integer, width as integer, height as integer) as boolean

Plugin Version: 9.8 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads the current picture with a Color Lookup Table.

Notes:

Returns true on success and false on any error.

Works with 8, 16 and 32 bits per sample. And with 1 or 3 samples per pixels (Gray or RGB). The lookup table has 256 entries for 8 bit and 65536 entries for 16/32 bit.

This method uses the YieldTicks property and may yield time to other threads.

See also:

- 3.2.1 ReadWithLUT(ColorLookupTable() as color) as boolean 45

RewriteDirectory as boolean

Plugin Version: 8.7 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes existing picture and header information to file replacing the old data.

Notes:

Returns true on success and false on any error.

As the data is added to the file and just the reference to the old data is deleted, the filesize will grow.

Calls TIFFRewriteDirectory internally. Same as SaveImage.

SaveImage as boolean

Plugin Version: 7.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes existing picture and header information to file replacing the old data.

Notes:

Returns true on success and false on any error.

As the data is added to the file and just the reference to the old data is deleted, the filesize will grow.

Calls TIFFRewriteDirectory internally. Same as RewriteDirectory.

Scanline(index as integer, sample as integer = 0) as memoryblock

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An image scanline.
Notes:

Returns nil on any error.

If you set the value, make sure the Memoryblock is big enough. The BytesPerRow functions returns the number of bytes needed.

The data for this property is stored in the Tiff file.

The first scanline has the index of 0.

(Read and Write computed property)

See also:

- 3.2.1 Scanline(mem as memoryblock, index as integer, sample as integer = 0) as boolean 47

Scanline(mem as memoryblock, index as integer, sample as integer = 0) as boolean

Plugin Version: 11.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reads an image scanline in a memoryblock.

Notes:

Returns false on any error.

The Memoryblock must be big enough. The BytesPerRow functions returns the number of bytes needed.

The first scanline has the index of 0.

This is the fastest way to read scanlines. You provide the memoryblock and you can reuse it for all calls to this method for one tiff.

See also:

- 3.2.1 Scanline(index as integer, sample as integer = 0) as memoryblock 47

Scanlines(index as integer, count as integer, sample as integer = 0) as memoryblock

Plugin Version: 4.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Several image scanlines.

Notes:

Returns nil on any error.

If you set the value, make sure the Memoryblock is big enough. The BytesPerRow functions returns the number of bytes needed.

The data for this property is stored in the Tiff file.

The first scanline has the index of 0. Count is the number of scanlines you want to read/write.

(Read and Write computed property)

SetColorMap(red as memoryblock, green as memoryblock, blue as memoryblock) as boolean

Plugin Version: 9.6 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the color map for a paletten image.

Notes:

The memoryblock must be $2\hat{\text{bitspersample}} * 2$ bytes big.

Returns true on success and false on failure.

SetColorProfile(ProfileData as String) as boolean

Plugin Version: 7.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the color profile for this tiff file.

Notes: Returns true on success and false on failure.

SetData(Tag as integer, data as string) as boolean

Plugin Version: 8.5 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes the string for this tag.

SetFieldByte(Tag as integer, value as integer) as boolean

Plugin Version: 8.7 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the tiff data field with the given tag to the given value.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

SetFieldDouble(Tag as integer, value as double) as boolean

Plugin Version: 7.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the tiff data field with the given tag to the given value.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

SetFieldInteger(Tag as integer, value as integer) as boolean

Plugin Version: 7.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the tiff data field with the given tag to the given value.

Example:

```

dim t as TiffPictureMBS
dim f as FolderItem

const TIFFTAG_IMAGELENGTH=257 // integer
const TIFFTAG_IMAGEWIDTH=256 // integer

f=GetTemporaryFolderItem
t=new TiffPictureMBS

if t.Create(f) then

if t.SetFieldInteger(TIFFTAG_IMAGELENGTH,100) then
if t.Height=100 then
MsgBox "ok"
end if
end if

end if

```

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

SetFieldMemory(Tag as integer, ItemCount as integer, data as memoryblock) as boolean

Plugin Version: 8.7 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the tiff data field with the given tag to the given value.

Notes:

This is the special version of the setter which passes a memoryblock and a count value so you can set an array using this method.

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

SetFieldShort(Tag as integer, value as integer) as boolean

Plugin Version: 7.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the tiff data field with the given tag to the given value.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

SetFieldSingle(Tag as integer, value as Single) as boolean

Plugin Version: 8.7 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the tiff data field with the given tag to the given value.

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

SetFieldString(Tag as integer, value as string) as boolean

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets one of the fields to the value of the string.

Example:

```

dim m,p as Picture
dim f as FolderItem
dim t as TiffPictureMBS
dim g as Graphics
dim s as string

// make pictures
p=NewPicture(100,100,32)
m=NewPicture(100,100,32)

p.Graphics.ForeColor=Rgb(0,255,0) // fill green
p.Graphics.Fillrect 0,0,100,100

p.Graphics.ForeColor=Rgb(255,0,0) // fill red
p.Graphics.FillOval 0,0,100,100

m.Graphics.ForeColor=Rgb(0,0,0) // fill black (invisible so green not seen)
m.Graphics.Fillrect 0,0,100,100

m.Graphics.ForeColor=Rgb(255,255,255) // fill white (Visible)
m.Graphics.FillOval 0,0,100,100

// save
f=SpecialFolder.Desktop.Child("test.tif")

t=new TiffPictureMBS
t.Pict=p
t.Mask=m

if t.Create(f) then
if t.WriteRGB then

const TIFFTAG_SOFTWARE=305

s="Example Software"
call t.SetFieldString TIFFTAG_SOFTWARE,s

```

```

const TIFFTAG_ HOSTCOMPUTER=316

s="Example HostComputer"
call t.SetFieldString TIFFTAG_ HOSTCOMPUTER,s

const TIFFTAG_ IMAGEDESCRIPTION=270

s="Example ImageDescription"
call t.SetFieldString TIFFTAG_ IMAGEDESCRIPTION,s

const TIFFTAG_ MAKE=271

s="Example Make"
call t.SetFieldString TIFFTAG_ MAKE,s

const TIFFTAG_ ARTIST=315

s="Example Artist"
call t.SetFieldString TIFFTAG_ ARTIST,s

t.Close
MsgBox "Ok"
f.Launch
end if
end if

```

Notes:

Please look for Tag values in the tiff specification.

Returns true on success.

Please make sure to use the correct setter depending on data type associated with the tag.

You have to pass in the string with the correct encoding.

SetImageIndex(index as integer) as boolean

Plugin Version: 4.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the current image index and reads the new image.

Example:

```
dim t as TiffPictureMBS // your tiff picture
```

```

if t.SetImageIndex(1) then
Canvas1.Backdrop=t.pict
end if

```

Notes:

Reads automatically a RGB picture for you, so the pict property is filled on success.

Returns true on success.

Index is zero based.

See also:

- 3.2.1 SetImageIndex(index as integer, HeaderOnly as boolean) as boolean

53

SetImageIndex(index as integer, HeaderOnly as boolean) as boolean

Plugin Version: 4.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the current image index.

Notes:

If HeaderOnly is false the current picture is read into the pict& mask properties.

Returns true on success.

See also:

- 3.2.1 SetImageIndex(index as integer) as boolean

52

SetXMP(ProfileData as String) as boolean

Plugin Version: 7.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Sets the xmp metadata package for this tiff file.

Notes: Returns true on success and false on failure.

WriteBW as boolean

Plugin Version: 5.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Write a black & white image.

Example:

```

dim p as Picture
dim f as FolderItem
dim t as TiffPictureMBS

p=NewPicture(100,100,32)
p.Graphics.ForeColor=rgb(0,0,0)
p.Graphics.FillOval 0,0,100,100

f=SpecialFolder.Desktop.Child("test.tif")

t=new TiffPictureMBS

if t.Create(f) then
t.Pict=p
if t.WriteBW then
MsgBox "ok"
end if
end if

Backdrop=p

```

Notes:

Uses the pictures in the pict property to write a picture.
Currently masks are not supported.

The following settings are made before the image data is written:

```

PlanarConfig = PLANARCONFIG_ CONTIG
Photometric = PHOTOMETRIC_ MINISBLACK
BitsPerSample = 1
SamplesPerPixel = 1
FillOrder = FILLORDER_ MSB2LSB
VerticalResolution = 72
HorizontalResolution = 72
Orientation = ORIENTATION_ TOPLEFT
ResolutionUnit = RESUNIT_ INCH
Compression = COMPRESSION_ NONE

```

You may change settings before or later. For example if you set Compression before it should be used for writing image data to the file.

Returns true on success.

This method uses the YieldTicks property and may yield time to other threads.

WriteGray as boolean

Plugin Version: 7.5 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Write a grayscale image with 256 colors.

Example:

```
dim t as TiffPictureMBS
dim f as FolderItem

f=SpecialFolder.Desktop.Child("test.tif")

t=new TiffPictureMBS
if t.Create(f) then
t.Pict=LogoMBS(500)
call t.WriteGray
t.Close
end if
```

Notes:

Uses the pictures in the pict property to write a picture.
Currently masks are not supported.

The following settings are made before the image data is written:

```
PlanarConfig = PLANARCONFIG_ CONTIG
Photometric = PHOTOMETRIC_ MINISBLACK
BitsPerSample = 8
SamplesPerPixel = 1
FillOrder = FILLORDER_ MSB2LSB
VerticalResolution = 72
HorizontalResolution = 72
Orientation = ORIENTATION_ TOPLEFT
ResolutionUnit = RESUNIT_ INCH
Compression = COMPRESSION_ NONE
```

You may change settings before or later. For example if you set Compression before it should be used for writing image data to the file.

Returns true on success.

This method uses the YieldTicks property and may yield time to other threads.

WriteRGB as boolean

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Writes a RGB image.

Example:

```

dim m,p as Picture
dim f as FolderItem
dim t as TiffPictureMBS
dim g as Graphics

p=NewPicture(100,100,32)
m=NewPicture(100,100,32)

p.Graphics.ForeColor=Rgb(0,255,0) // fill green
p.Graphics.Fillrect 0,0,100,100

p.Graphics.ForeColor=Rgb(255,0,0) // fill red
p.Graphics.FillOval 0,0,100,100

m.Graphics.ForeColor=Rgb(0,0,0) // fill black (invisible so green not seen)
m.Graphics.Fillrect 0,0,100,100

m.Graphics.ForeColor=Rgb(255,255,255) // fill white (Visible)
m.Graphics.FillOval 0,0,100,100

f=SpecialFolder.Desktop.Child("test.tif")

t=new TiffPictureMBS
t.Pict=p
t.Mask=m

if t.Create(f) then
if t.WriteRGB then
t.Close
MsgBox "Ok"
f.Launch
end if
end if

```

Notes:

Uses the pictures in the mask and pict properties to write a picture.
If mask is set, the picture is saved with an alpha channel.

The following settings are made before the image data is written:

```
PlanarConfig = PLANARCONFIG_ CONTIG
Photometric = PHOTOMETRIC_ RGB
BitsPerSample = 8
SamplesPerPixel = 3
FillOrder = FILLORDER_ MSB2LSB
VerticalResolution = 72
HorizontalResolution = 72
Orientation = ORIENTATION_ TOPLEFT
ResolutionUnit = RESUNIT_ INCH
Compression = COMPRESSION_ NONE
```

You may change settings before or later. For example if you set Compression before it should be used for writing image data to the file.

Returns true on success.

This method uses the YieldTicks property and may yield time to other threads.

3.2.2 Properties**BitsPerSample as integer**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The count of bits for each pixel component.

Notes:

Should normally be 8 for RGB images.
(Read and Write property)

BytesPerRow as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The number of bytes needed for each row in a scan line.

Notes:

0 on any error.
(Read only property)

Compression as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The compression used.

Notes:

some constants:

The data for this property is stored in the Tiff file.
(Read and Write property)

Copyright as String

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The copyright notice of this image.

Notes:

The data for this property is stored in the Tiff file.
(Read and Write property)

DateTime as String

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Date and time of the TIFF file.

Notes:

Check some TIFF files for the format used.

The data for this property is stored in the Tiff file.
(Read and Write property)

DocumentName as String

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The name of the document.

Notes:

The data for this property is stored in the Tiff file.
(Read and Write property)

ExtraSamples as MemoryBlock

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The settings for the extra samples.

Notes:

A memoryblock filled with an array of shorts (16bit integers).

constants:

The data for this property is stored in the Tiff file.
(Read and Write property)

FillOrder as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The fill order of the bits in a byte.

Notes:

constants:

(MSB = Most significant bit, LSB = Least significant bit)

The data for this property is stored in the Tiff file.
(Read and Write property)

height as integer

Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The height of the picture.
Notes: (Read and Write property)

HorizontalPosition as Single

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The x Offset of this image in the drawing area.
Notes:

The data for this property is stored in the Tiff file.
(Read and Write property)

HorizontalResolution as Single

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The horizontal resolution used.
Notes:

Value depends on ResolutionUnit value.

The data for this property is stored in the Tiff file.
(Read and Write property)

HostComputer as String

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Name of the machine where the tiff file was created.

Notes:

The data for this property is stored in the Tiff file.
(Read and Write property)

ImageDescription as String

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Information about the image.

Notes:

The data for this property is stored in the Tiff file.
(Read and Write property)

InputBuffer as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The string passed for OpenString.

Notes:

Used for the read requests from the Tiff library.
(Read only property)

Make as String

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Scanner manufacturer name.

Notes:

The data for this property is stored in the Tiff file.
(Read and Write property)

mask as picture

Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The mask of the picture.

Notes:

May be nil.
(Read and Write property)

Model as String

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Scanner model name/number.

Notes:

The data for this property is stored in the Tiff file.
(Read and Write property)

Orientation as integer

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The orientation of this image.

Example:

```
// creates a tif file with a horizontal flipped image
// red rectangle is on the left in the image data, but with orientation tag it should be displayed on the right
// side.
// Mac OS X Preview.app shows it correctly.
```

```
dim p as Picture
```

```
dim t as TiffPictureMBS
```

```
dim f as FolderItem
```

```
const ORIENTATION_TOPLEFT = 1 /* row 0 top, col 0 lhs */
const ORIENTATION_TOPRIGHT = 2 /* row 0 top, col 0 rhs */
const ORIENTATION_BOTRIGHT = 3 /* row 0 bottom, col 0 rhs */
const ORIENTATION_BOTLEFT = 4 /* row 0 bottom, col 0 lhs */
const ORIENTATION_LEFTTOP = 5 /* row 0 lhs, col 0 top */
const ORIENTATION_RIGHTTOP = 6 /* row 0 rhs, col 0 top */
const ORIENTATION_RIGHTBOT = 7 /* row 0 rhs, col 0 bottom */
const ORIENTATION_LEFTBOT = 8 /* row 0 lhs, col 0 bottom */
```

```

p=NewPicture(150,100,32)

p.Graphics.ForeColor=rgb(255,0,0)
p.Graphics.FillRect 0,0,10,10

t=new TiffPictureMBS
f=SpecialFolder.Desktop.Child("test.tif")

if t.Create(f) then
t.Pict=p

t.Orientation=ORIENTATION_ TOPRIGHT
if t.WriteRGB then
end if
t.Orientation=ORIENTATION_ TOPRIGHT

t.Close
end if

```

Notes:

Orientation:

The orientation of the image with respect to the rows and columns.

Tag = 274 (112.H)

Type = SHORT

N = 1

1 = The 0th row represents the visual top of the image, and the 0th column represents the visual left-hand side.

2 = The 0th row represents the visual top of the image, and the 0th column represents the visual right-hand side.

3 = The 0th row represents the visual bottom of the image, and the 0th column represents the visual right-hand side.

4 = The 0th row represents the visual bottom of the image, and the 0th column represents the visual left-hand side.

5 = The 0th row represents the visual left-hand side of the image, and the 0th column represents the visual top.

6 = The 0th row represents the visual right-hand side of the image, and the 0th column represents the visual top.

7 = The 0th row represents the visual right-hand side of the image, and the 0th column represents the visual bottom.

8 = The 0th row represents the visual left-hand side of the image, and the 0th column represents the visual bottom.

Default is 1.

Support for orientations other than 1 is not a Baseline TIFF requirement.

(This text was sent in by David Austin)

(Read and Write property)

OutputBuffer as String

Plugin Version: 5.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The current output data from the CreateString function.

Notes:

Between CreateString and Close the plugin will record all the output data and you can get a copy using this property.

(Read only property)

PageName as String

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The name of the current page.

Notes:

The data for this property is stored in the Tiff file.

(Read and Write property)

Photometric as integer

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A property of the TIFF image.

Example:

```
// read and write a BW fax tiff with compression  
  
dim d as new OpenFileDialog()  
dim f as FolderItem = GetFolderItem("myfax.tiff")  
if f = nil then  
return
```

```

end if

dim tiff as new TiffPictureMBS()
if not tiff.Open(f) then
MsgBox("Error while opening.")
return
end if

if not tiff.ReadBW() then
MsgBox("Error reading.")
end if

dim tiff2 as TiffPictureMBS = new TiffPictureMBS()

dim f2 as FolderItem = SpecialFolder.Desktop.Child("Copy.tiff")
if not tiff2.Create(f2) then
MsgBox("Error creating file.")
end if

const COMPRESSION_ CCITTFAX3=3
const WhiteIsZero=0
const BlackIsZero=1

tiff2.Pict = tiff.pict
tiff2.Photometric=WhiteIsZero
tiff2.Compression=COMPRESSION_ CCITTFAX3
if not tiff2.WriteBW() then
MsgBox("Error writing.") // Error
end if

tiff2.Close()

```

Notes:

PhotometricInterpretation:
The color space of the image data.
Tag = 262 (106.H)
Type = SHORT
N = 1

0 = WhiteIsZero. For bilevel and grayscale images: 0 is imaged as white. $2^{**}\text{BitsPerSample}-1$ is imaged as black. This is the normal value for Compression=2.

1 = BlackIsZero. For bilevel and grayscale images: 0 is imaged as black. $2^{**}\text{BitsPerSample}-1$ is imaged as white. If this value is specified for `Compression=2`, the image should display and print reversed.

2 = RGB. In the RGB model, a color is described as a combination of the three primary colors of light (red, green, and blue) in particular concentrations. For each of the three components, 0 represents minimum intensity, and $2^{**}\text{BitsPerSample} - 1$ represents maximum intensity. Thus an RGB value of (0,0,0) represents black, and (255,255,255) represents white, assuming 8-bit components. For `PlanarConfiguration = 1`, the components are stored in the indicated order: first Red, then Green, then Blue. For `PlanarConfiguration = 2`, the `StripOffsets` for the component planes are stored in the indicated order: first the Red component plane `StripOffsets`, then the Green plane `StripOffsets`, then the Blue plane `StripOffsets`.

3= Palette color. In this model, a color is described with a single component. The value of the component is used as an index into the red, green and blue curves in the `ColorMap` field to retrieve an RGB triplet that defines the color. When `PhotometricInterpretation=3` is used, `ColorMap` must be present and `SamplesPerPixel` must be 1.

4 = Transparency Mask. This means that the image is used to define an irregularly shaped region of another image in the same TIFF file. `SamplesPerPixel` and `BitsPerSample` must be 1. `PackBits` compression is recommended. The 1-bits define the interior of the region; the 0-bits define the exterior of the region. A reader application can use the mask to determine which parts of the image to display. Main image pixels that correspond to 1-bits in the transparency mask are imaged to the screen or printer, but main image pixels that correspond to 0-bits in the mask are not displayed or printed. The image mask is typically at a higher resolution than the main image, if the main image is grayscale or color so that the edges can be sharp.

There is no default for `PhotometricInterpretation`, and it is required. Do not rely on applications defaulting to what you want.

(This text was sent in by David Austin)

For more details see:

<http://partners.adobe.com/asn/developer/pdfs/tn/TIFF6.pdf>

(Read and Write property)

pict as picture

Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The picture data of the picture.

Notes: (Read and Write property)

PlanarConfig as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The storage organization used.

Notes:

Value is 1 for a single image plane and 2 for separated planes.

The data for this property is stored in the Tiff file.
(Read and Write property)

ResolutionUnit as Integer

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Which unit is used for the Resolution values.

Notes:

constants:

The data for this property is stored in the Tiff file.
(Read and Write property)

RowsPerStrip as Integer

Plugin Version: 4.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** A property of the TIFF image.

Notes:

Should be same as the height for our uses. (with one strip)
(Read and Write property)

SamplesPerPixel as integer

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The count of components used for each pixel.

Notes:

SamplesPerPixel= 1=Grayscale, 3=RGB, 4=CMYK, ...

Photometric= 0 or 1=Grayscale depending on white point, 2=RGB, 5=CMYK, ...

For other formats see:

<http://partners.adobe.com/asn/developer/pdfs/tn/TIFF6.pdf>

(Read and Write property)

Software as String

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The software name used to make this image.

Notes:

The data for this property is stored in the Tiff file.

(Read and Write property)

Version as Integer

Plugin Version: 5.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The version of the Tiff library used.

Notes:

Updated to 3.7.1 in plugin version 5.0.

(Read only property)

VersionString as String

Plugin Version: 5.0 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The version of the Tiff library used.

Example:

```
dim z as new TiffPictureMBS
```

```
MsgBox z.VersionString
```

```
// shows for example:  
// LIBTIFF, Version 3.9.4  
// Copyright (c) 1988-1996 Sam Leffler  
// Copyright (c) 1991-1996 Silicon Graphics, Inc.
```

Notes: (Read only property)

VerticalPosition as Single

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The y Offset of this image in the drawing area.

Notes:

The data for this property is stored in the Tiff file.
(Read and Write property)

VerticalResolution as Single

Plugin Version: 4.2 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The vertical resolution used.

Notes:

Value depends on ResolutionUnit value.

The data for this property is stored in the Tiff file.
(Read and Write property)

width as integer

Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** The width of the picture.

Notes: (Read and Write property)

YieldTicks as Integer

Plugin Version: 7.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** How much time is given back to REALbasic for other ticks.

Example:

```
dim t as new TiffPictureMBS
t.YieldTicks=6 // only use 1/10th of a second
```

Notes:

If value is greater than zero, the application will yield to another RB thread after the given number of ticks have passed. 60 ticks are one second. Using a small value can slow down processing a lot while a big value keeps your application not responding to mouse clicks.

If you use this property with e.g. 6 as the value, you may also want to use this method in a thread so you can handle mouse events or let REALbasic redraw a progressbar.

(Read and Write property)

3.2.3 Events**Error(libModule as string, message as string)**

Plugin Version: 4.2 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An event called whenever an error is to be reported.

Progress(line as integer, total as integer)

Plugin Version: 6.2 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Reports changes in progress.

Notes: Used in CombineBitCMYKtoCMYK, CombineTiffCMYKtoRGB and CombineTiffCMYKtoCMYK methods.

Warning(libModule as string, message as string)

Plugin Version: 4.2 Console & Web: No Mac: Yes, Win: Yes, Linux: Yes, . **Function:** An event called whenever a warning is to be reported.

- l When creating a new file force information be written with Little-Endian byte order (but see below). By default the library will create new files using the native CPU byte order.
- b When creating a new file force information be written with Big-Endian byte order (but see below). By default the library will create new files using the native CPU byte order.
- L Force image data that is read or written to be treated with bits filled from Least Significant Bit (LSB) to Most Significant Bit (MSB). Note that this is the opposite to the way the library has worked from its inception.
- B Force image data that is read or written to be treated with bits filled from Most Significant Bit (MSB) to Least Significant Bit (LSB); this is the default.
- H Force image data that is read or written to be treated with bits filled in the same order as the native CPU.
- M Enable the use of memory-mapped files for images opened read-only. If the underlying system does not support memory-mapped files or if the specific image being opened cannot be memory-mapped then the library will fallback to using the normal system interface for reading information. By default the library will attempt to use memory-mapped files.
- m Disable the use of memory-mapped files.
- C Enable the use of "strip chopping" when reading images that are comprised of a single strip or tile of uncompressed data. Strip chopping is a mechanism by which the library will automatically convert the single-strip image to multiple strips, each of which has about 8 Kilobytes of data. This facility can be useful in reducing the amount of memory used to read an image because the library normally reads each strip in its entirety. Strip chopping does however alter the apparent contents of the image because when an image is divided into multiple strips it looks as though the underlying file contains multiple separate strips. Finally, note that default handling of strip chopping is a compile-time configuration parameter. The default behaviour, for backwards compatibility, is to enable strip chopping.
- c Disable the use of strip chopping when reading images.

COMPRESSION_ NONE	1	dump mode
COMPRESSION_ CCITTRLE	2	CCITT modified Huffman RLE
COMPRESSION_ CCITTFAX3	3	CCITT Group 3 fax encoding
COMPRESSION_ CCITT_ T4	3	CCITT T.4 (TIFF 6 name)
COMPRESSION_ CCITTFAX4	4	CCITT Group 4 fax encoding
COMPRESSION_ CCITT_ T6	4	CCITT T.6 (TIFF 6 name)
COMPRESSION_ LZW	5	Lempel-Ziv & Welch
COMPRESSION_ OJPEG	6	!6.0 JPEG
COMPRESSION_ JPEG	7	% JPEG DCT compression
COMPRESSION_ NEXT	32766	NeXT 2-bit RLE
COMPRESSION_ CCITTRLEW	32771	# 1 w/ word alignment
COMPRESSION_ PACKBITS	32773	Macintosh RLE
COMPRESSION_ THUNDERSCAN	32809	ThunderScan RLE
COMPRESSION_ IT8CTPAD	32895	IT8 CT w/padding
COMPRESSION_ IT8LW	32896	IT8 Linework RLE
COMPRESSION_ IT8MP	32897	IT8 Monochrome picture
COMPRESSION_ IT8BL	32898	IT8 Binary line art
COMPRESSION_ PIXARFILM	32908	Pixar companded 10bit LZW
COMPRESSION_ PIXARLOG	32909	Pixar companded 11bit ZIP
COMPRESSION_ DEFLATE	32946	Deflate compression
COMPRESSION_ ADOBE_ DEFLATE	8	Deflate compression, as recognized by Adobe
COMPRESSION_ DCS	32947	Kodak DCS encoding
COMPRESSION_ JBIG	34661	ISO JBIG
COMPRESSION_ SGILOG	34676	SGI Log Luminance RLE
COMPRESSION_ SGILOG24	34677	SGI Log 24-bit packed

EXTRASAMPLE_ UNSPECIFIED	0	unspecified data
EXTRASAMPLE_ ASSOCALPHA	1	associated alpha data (pre multiplied)
EXTRASAMPLE_ UNASSALPHA	2	unassociated alpha data (mask in RB)

FILLORDER_ MSB2LSB	1 (default)
FILLORDER_ LSB2MSB	2

RESUNIT_ NONE	1	no meaningful units
RESUNIT_ INCH	2	english
RESUNIT_ CENTIMETER	3	metric

Chapter 4

List of all classes

- TiffPictureMBS

22

Chapter 5

List of all global methods

- 3.1 CombineBitCMYKtoCMYKMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, Files() as FolderItem, scale as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, CacheSizeRead as integer) as integer 13
- 3.1 CombineBitCMYKtoRGBMBS(CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, Files() as FolderItem, scale as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, byref output as picture, CacheSizeRead as integer) as integer 13
- 3.1 CombineTiff1BitCMYKtoTiffMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, TiffData() as TiffPictureMBS, scale as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, ditherMode as integer = 0) as integer 15
- 3.1 CombineTiff1BitCMYKtoTiffMBS(dest as TiffPictureMBS, TiffData as TiffPictureMBS, scalex as double, scaley as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, ditherMode as integer = 0) as integer 15
- 3.1 CombineTiff1BitCMYKtoTiffSWMBS(dest as TiffPictureMBS, a as UInt32, TiffData as TiffPictureMBS, b as UInt32, c as UInt32, d as UInt32) as integer 16
- 3.1 CombineTiff8BitCMYKtoTiffMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, TiffData() as TiffPictureMBS, scale as double, width as integer, height as integer, X1 as integer, Y1 as integer, X2 as integer, Y2 as integer, ditherMode as integer = 0) as integer 16
- 3.1 CombineTiffCMYKtoCMYKMBS(dest as TiffPictureMBS, CyanChannel() as integer, MagentaChannel() as integer, YellowChannel() as integer, BlackChannel() as integer, TiffData() as TiffPictureMBS) as integer 16

- 3.1 `CombineTiffCMYKtoRGBMBS`(`dest` as `TiffPictureMBS`, `CyanChannel()` as integer, `MagentaChannel()` as integer, `YellowChannel()` as integer, `BlackChannel()` as integer, `TiffData()` as `TiffPictureMBS`) as integer 19
- 2.1 `Split1BitFileMBS`(`f` as folderitem, `fc` as folderitem, `fm` as folderitem, `fy` as folderitem, `fk` as folderitem, `width` as integer, `height` as integer, `CallbackTarget` as object, `CacheSizeRead` as integer, `CacheSizeWrite` as integer) as integer 9
- 2.1 `Split1BitFileMBS`(`f` as folderitem, `fc` as folderitem, `fm` as folderitem, `fy` as folderitem, `fk` as folderitem, `width` as integer, `height` as integer, `CallbackTarget` as object, `CacheSizeRead` as integer, `CacheSizeWrite` as integer, `ReadLines` as integer, `WriteLines` as integer) as integer 10
- 3.1 `TIFFStringToPictureMBS`(`data` as string) as picture 21
- 3.1 `TIFFStringToTiffPictureMBS`(`data` as string) as `TiffPictureMBS` 21