## 1 Introduction

The file format describes the project file for the program "PicturePaint". The archive can contain all file types. For all image files, there are also different setting values in the file format. The file extension is called ".ppp" (Picture Paint Project).

## 2 Value types

Туре	Description	Area
INT8	8Bit with sign	-128 to 127
INT16	16Bit with sign	-32.768 to 32.767
INT32	32Bit with sign	-2.147.483.648 to 2.147.483.647
INT64	64Bit with sign	-9.223.372.036.854.775.808 to 9.223.372.036.854.775.807
BYTE	8Bit unsigned	0 to 255
UINT16	16Bit unsigned	0 to 65.535
UINT32	32Bit unsigned	0 to 4.294.967.295
UINT64	64Bit unsigned	0 to 18.446.744.073.709.551.615
CHAR	8Bit character	0 to 255
WCHAR	16Bit character	0 to 65.535
FLOAT	32Bit floating point	± 1.5e-45 to ± 3.4e38
DOUBLE	64Bit floating point	± 5.0e-324 to ± 1.7e308
MEMORY	Memory in bytes	
[]	Array	see section 2.1
-> {	Start of the loop	see section 2.2
} <-	End of the loop	see section 2.2
	Next table	see section 2.3
!	Table with condition	see section 2.4

Table 2: Value types

# 2.1 Array

The set consists of a specific value type. The count of the set is detailed in the information and is usually the previous format value.

## Example:

```
A Array INT16[] contains a certain count of INT16 values { INT16, INT16, INT16, INT16, ... }.
```

INT16[], BYTE[], UINT32[], WCHAR[], usw.

## 2.2 The loop

In a loop, the format is repeatedly run through. The count of run through is specified in detail in the information and is usually the previous value.

## 2.3 Next table

The file format is displayed further in the section and the table specified.

## 2.4 Next table with condition

The file format continues in the section and the table if the condition is met.

# 3 Description

# 3.1 File format

Туре	Name	Description	
UINT32	IDNumber	The file must have the ID number $(0x50504454)$ .	4.1
INT64	FileSize	The size of the entire file in bytes.	
BYTE	Version	The version number is 1 in this file format.	
INT32	PreviewImageSize	The size of the memory with the preview image in bytes.	4.4
MEMORY	PreviewImage	The memory with the preview image file.	4.5
		3.2 Data format, Table 3.2	

Table 3.1: File format

# 3.2 Data format

Туре	Name	Description	Info
INT32	FileCount	The number of files.	5.1
-> {	File		
BYTE	FileType	The file type as a number.	5.2
INT32	FileNameLength	The number of characters in the name of the file.	5.3
WCHAR[]	FileName	The name of the file.	5.4
INT32	FileExtentionLength	The number of characters in the name of the extension.	5.5
WCHAR[]	FileExtention	The name of the file extension.	5.6
!	FileType = 0	3.3 Image format, Table 3.2	
!	FileType = 1	3.5 Playing file format (Video), Table 3.5	
!	FileType = 2	3.5 Playing file format (Musik), Table 3.5	
!	FileType = 3	3.5 Playing file format (GIF), Table 3.5	
!	FileType = 4	3.6 Animation format (TDA), Table 3.6	
!	FileType = 5	3.7 Various file format, Table 3.7	
} <-	File		
		3.8 Image group format, Table 3.8	

## Table 3.2: Data format

# 3.3 Image format

Туре	Name	Description	Info
BYTE	ImageFormat	The image format as a number.	6.1
UINT16	ImageFlags	The setting bits for the program "PicturePaint".	6.2
INT32	ImagePosX	The image position in X coordinates.	6.3
INT32	ImagePosY	The image position in Y coordinates.	б.4
INT32	ImageWidth	The width of the image.	6.5
INT32	ImageHeight	The height of the image.	6.6
INT32	FileImageSize	The size of the image memory in bytes.	6.7
BYTE[]	FileImage	The memory with the image file.	6.8
		3.4 Image setting format, Table 3.4	

Table 3.3: Image format

# 3.4 Image setting format

Туре	Name	Description	Info
INT32	ToolIndex	Recently used tool while drawing.	7.1
BYTE	PaintAlpha	Last used alpha value while drawing.	7.2
UINT32	PaintColor	Last used color when drawing.	7.3
UINT32	BackColor	The set background color in the window.	7.4
UINT32	ImageColor	The set extension color for images.	7.5
UINT32	ImageTransparentColor	The transparent color when drawing images.	7.6
UINT32	TextColor	The font color when drawing text content.	7.7
INT32	PaintLineSize	The thickness of the lines when drawing.	7.8
INT32	PaintRoundWidth	Width of the rounding of a rounded rectangle.	7.9
INT32	PaintRoundHeight	Height of the rounding of a rounded rectangle.	7.10
INT32	PaintRotationLeverLength	Length of the lever with rotating color gradient.	7.11
DOUBLE	PaintRotationAngle	The angle of the rotating color gradient.	7.12
INT32	ImageRotationLeverLength	Length of the lever arm with the rotating image.	7.13
DOUBLE	ImageRotationAngle	The angle of the rotating image.	7.14
INT32	TextRotationLeverLength	Length of the lever arm with the rotating text.	7.15
DOUBLE	TextRotationAngle	The angle of the rotating text.	7.16
INT32	AlphaArrayCount	The number of alpha values when drawing.	7.17
BYTE[]	AlphaArray	The alpha values in the selection bar.	7.18
INT32	ColorArrayCount	The number of colors when drawing.	7.19
UINT32[]	ColorArray	The colors in the selection bar.	7.20
INT32	AlphaPaletteCount	The number of alpha values when drawing.	7.21
MEMORY	AlphaPalette	The alpha values in the user list.	7.22
INT32	ColorPaletteCount	The number of colors when drawing.	7.23
MEMORY	ColorPalette	The colors in the user list.	7.24
INT32	AlphaGradientCount	The number of alpha gradients when drawing.	7.25
MEMORY	AlphaGradient	The alpha gradients in the user list.	7.26
INT32	ColorGradientCount	The number of color gradients when drawing.	7.27
MEMORY	ColorGradient	The color gradients in the user list.	7.28
INT32	AlphaGradientIndex	The selected alpha gradient.	7.29
INT32	ColorGradientIndex	The selected color gradient.	7.30
INT32	TextAlphaPaletteCount	Is not used.	7.31
INT32	TextColorPaletteCount	The number of colors in the text editor.	7.32
MEMORY	TextColorPalette	The text colors in the user list.	7.33
INT32	TextAlphaGradientCount	Is not used.	7.34
INT32	TextColorGradientCount	Is not used.	7.35
INT32	FontCount	The number of fonts in text editing.	7.36
MEMORY	FontMemory	The fonts in the user list.	7.37
INT32	FontIndex	The selected font.	7.38
INT32	GroupIndex	The selected group in image editing.	7.39
		3.2 Data format, Table 3.2	

Table 3.4: Image setting format

# 3.5 Playing file format

Туре	Name	Description	Info
INT32	ImageWidth	The width of the picture.	8.1
INT32	ImageHeight	The height of the picture.	8.2
INT64	PlayerDuration	The playing time of the file.	8.3
INT32	FileImageSize	The size of the image memory in bytes.	8.4
MEMORY	FileImageMemory	The memory with the image file.	8.5
INT32	FileMemorySize	The size of the file memory in bytes.	8.6
BYTE[]	FileMemory	The memory with the file data.	8.7
		3.2 Data format, Table 3.2	

## Table 3.5: Playing file format

# 3.6 Animation format

Туре	Name	Description	
INT32	ImageWidth	The width of the picture.	9.1
INT32	ImageHeight	The height of the picture.	9.2
INT64	PlayerDuration	The playing time of the file.	9.3
UINT32	BackColor	The background color in the display window.	9.4
INT32	FileImageSize	The size of the image memory in bytes.	9.5
MEMORY	FileImageMemory	The memory with the image file.	9.6
INT32	FileMemorySize	The size of the file memory in bytes.	9.7
BYTE[]	FileMemory	The memory with the file data.	9.8
		3.2 Data format, Table 3.2	

## Table 3.6: Animation format

# 3.7 Various file format

Туре	Name	Description	Info
INT32	FileMemorySize	The size of the file memory in bytes.	10.1
BYTE[]	FileMemory	The memory with the file data.	10.2
		3.2 Data format, Table 3.2	

## Table 3.7: Various file format

# 3.8 Image group format

Туре	Name	Description	Info
INT32	GroupCount	The number of image groups.	11.1
-> {	Group		
INT32	GroupNameLength	The number of characters in the name of the group.	11.2
WCHAR[]	GroupName	The name of the group.	11.3
INT32	GroupItemCount	The number of image elements in a group.	11.4
MEMORY	GroupItems	The memory with the image element.	11.5
} <-	Group		

## 4 Information about the file format

## 4.1 Identification number

The identification number identifies the file format. The number can also be displayed with 4 letters (TDPP: Two Dimension PicturePaint Project).

## 4.2 The total file size

The size of the entire file in bytes. The value is given as a control and can not be different from the file size.

## 4.3 Version number

The version number is always 1 for this description.

## 4.4 The size of the preview image

The size of the image for the file preview in bytes. If the value is 0, no preview image is used. The value can not be less than 0 or greater than the remaining file size. The image uses a file from the project archive (see section 6.2: "ProjectPreview").

## 4.5 Speicher des Vorschaubildes

The memory with the file of the preview image. If the file size (see section 4.4) is 0, no preview image will be used. By default, an image should be selected in the format "PNG" or "JPEG" from the project archive. The file memory for the image is specified here. The image file then contains the memory size 0 (see section 6.7). In addition, the setting bit "ProjectPreview" (see section 6.2) must be set.

Note: The program "FileViewer" does not recognize the "2D Picture" format as a preview image.

## 5 Information about the data format

## 5.1 Number of files

The value indicates the number of files in the archive. He can not be less than 0. If the value is 0, there are no files. The next read value is then the number of image groups (see section 11.1). Again, this value must also be 0 because only images from the project archive will be used in the groups.

## 5.2 File type

The file type indicates the application of the file in the program "PicturePaint". Not all files are recognized by the program. Unassigned files must have the value 5 (Various).

Name	Number	Description	
Image	0	The file is a displayable image format.	
Video	1	The file is a playable video.	
Sound	2	The file is a playable sound.	
Gif	3	The file is a playable GIF animation.	
Animation	4	The animation (.tda) is created with the program "PicturPaint".	
Various	5	The file is not assigned by the PicturPaint program.	

## 5.3 Name length

The number of characters in the name of the file. The value can not be less than 1 or greater than 260. The maximum name length with file extension (see section 5.5) is specified by the file system.

## 5.4 Name

The name consists of a certain number of characters (letters). The file name may only be contained once in the project archive. The name length (see section 5.3) determines the number of characters. A character is a 16 bit unsigned value. The memory size for the name results from the name length times 2 bytes.

Memory size: FileNameLength \* 2Bytes (16Bit)

## 5.5 Name length of the file extension

The number of characters in the name of the file extension. The value can not be less than 0. The length of the name in section 5.3 and the file extension can not be greater than 260 together. The maximum name length is specified by the file system. If no file extension was specified, the value must be 0.

## 5.6 Name of the file extension

The name consists of a certain number of characters (letters). The first character in the extension is a dot. The name length (see section 5.5) determines the number of characters. A character is a 16 bit unsigned value. The extension does not exist if the length is 0. The memory size for the file extension results from the length times 2 bytes.

Memory size: FileExtentionLength \* 2Bytes (16Bit)

Example: ".png", ".jpeg", ".avi", ".mp4", ".wav"

## 6 Information about the image format

## 6.1 Image format

The number indicates the image format for the following file memory (see section 6.8). The type (see section 5.2) must have the value 0 for an image file. If the file format is not an image, the value 0 (Unknown) is specified here.

Name	Number	Description	
Unknown	0	An unknown or no image format.	
bmp	1	An uncompressed picture (Windows Bitmap).	
emf	2	A vector based image (Windows Enhanced Metafile).	
exif	3	An image file format (Exchangeable Image File Format).	
gif	4	A lossless compressed image (Graphics Interchange Format).	
icon	5	An uncompressed small picture (Windows Icon).	
jpeg	6	A lossy compressed image (Joint Photographic Experts G).	
png	7	A lossless compressed image (Protable Network Graphics).	
tiff	8	A lossless compressed image (Tagged Image File Format).	
wmf	9	A vector based image (Windows Metafile).	
tpd	10	An uncompressed picture with thumbnail (2D picture).	

Table 6.1: Image format

## 6.2 Setting bits

For the program "PicturePaint" different setting bits for picture files are possible. The "PaintPreview" bit is set if a preview image is to be saved in the "2D Picture" format. The "ProjectPreview" bit indicates that this image will be used as archive preview (see sections 4.5 and 6.7).

Name	Bit	Description
PaintAlpha	0x0001	To draw with alpha values is selected.
PaintOver	0x0002	Draw over the display image is selected.
PaintMultiColor	0x0004	Draw with color gradient is selected.
PaintEllipse	0x0008	The ends of a line are drawn as an ellipse.
PaintPreview	0x0010	The image format (.tdp) save a preview image.
PaintMultiColorRotation	0x0020	The rotating color gradient is selected.
PaintRotationLever	0x0040	The lever arm is displayed when the gradient is rotating.
ImageOver	0x0080	Draw the inserted image over the display image.
ImageSize	0x0100	Adjust the inserted image to the window size.
ImageAspectRatio	0x0200	Keep the aspect ratio of the inserted image.
ImageRotationLever	0x0400	The lever arm is displayed on the rotating picture.
TextOver	0x0800	Draw the text over the display image.
TextRotationLever	0x1000	The lever arm is displayed while the text is rotating.
ProjectPreview	0x8000	The file is used as a project preview.

Table 6.2: Setting bits

## 6.3 **Position in X-coordinates**

The position in X coordinates should originally be used when drawing with image clipping. The value must be 0 because the program "PicturePaint" does not use the position.

## 6.4 **Position in Y-coordinates**

The position in Y coordinates should originally be used when drawing with image clipping. The value must be 0 because the program "PicturePaint" does not use the position.

## 6.5 Image width

The width of the image in pixels. The value can not be less than 1.

## 6.6 Image height

The height of the image in pixels. The value can not be less than 1.

## 6.7 Size of the image memory

The size of the image memory in bytes. The value can not be less than 0 or greater than the remaining file size. If the value is 0, the setting bit "ProjectPreview" must be set (see table 6.2). Thus, the project preview (see section 4.6) is the image memory.

## 6.8 Image memory

The memory contains the file data. The size in bytes is given in section 6.7. If the image is used from project preview, the memory is not present. The size and image file are then defined in sections 4.4 and 4.5.

## 7 Information about the image setting format

The picture settings are specified for the program "PicturePaint". In the further description reference is made to the program. At the end of the information, the value range  $\{...\}$  and the default value  $\{, ...\}$  are defined.

## 7.1 Drawing tool

The value indicates the graphic object used when drawing (rectangle, circle, etc.). {0 ... 11, 2}

## 7.2 Alpha value for drawing

The current alpha value used for drawing. {0 ... 255, 255}

## 7.3 Color for drawing

The current color used to draw. {0x00000000 ... 0xFFFFFFF, 0xFFFF0000}

## 7.4 Background color in the display window

Under image settings, the background color in the display window can be changed. { 0xFF000000 ... 0xFFFFFFF, 0xFFFFFFF }

## 7.5 Color when creating or expanding images

Under image settings, the color can be changed when creating or expanding images. { 0x00000000 ... 0xFFFFFFF, 0xFFFFFFF }

## 7.6 Transparent color when drawing with images

In image editing, the transparent color can be specified under settings. { 0x00000000 ... 0xFFFFFFF, 0x00000000 }

## 7.7 Font color when drawing text content

The current font color used for drawing in text editing. { 0x00000000 ... 0xFFFFFFFF, 0xFF000000 }

## 7.8 Line thickness when drawing

The thickness of the line is specified under settings when drawing. {0 ... 999, 1}

## 7.9 Width of the rounded rectangle

In drawing settings, the width of the rounding is determined. {1 ... 1000000, 12}

# 7.10 Height of the rounded rectangle

In drawing settings, the height of the rounding is determined. {1 ... 1000000, 14}

# 7.11 Length of the lever arm for the rotating color gradient

In drawing settings, the length of the lever arm is specified. {5 ... 2000, 100}

# 7.12 Angle in the rotating color gradient

The angle is entered in the drawing settings. { 0.0 ... 360.0, 0.0 }

## 7.13 Length of the lever arm for the rotating image

In image editing under settings, the length of the lever arm is determined. {5 ... 2000, 100}

# 7.14 Angle in the rotating image

In image editing under settings, the angle is entered. { 0.0 ... 360.0, 0.0 }

## 7.15 Length of the lever arm for the rotating text

In text editing under settings, the length of the lever arm is determined. {5 ... 2000, 100}

## 7.16 Angle in the rotating text

In text editing under Settings the angle is entered. { 0.0 ... 360.0, 0.0 }

## 7.17 Number of alpha values in the display window

The number of custom alpha values. {0 ... 10000, 0}

## 7.18 Alpha values in the display window

All alpha values displayed in the window when drawing. {0x00 ... 0xFF}

## 7.19 Number of colors in the display window

The number of custom colors. {0 ... 10000, 0}

## 7.20 Colors in the display window

All colors displayed in the window when drawing. { 0x00000000 ... 0xFFFFFFF }

## 7.21 Number of alpha values in the user list

The number of custom alpha values. {0 ... 10000, 0}

## 7.22 Alpha values in the user list

All alpha values displayed when drawing in alpha mode in the user color list. The structure of the memory is described in table 7.22. The number of elements is given in section 7.21.

Туре	Name	Description	
BYTE	UserSelect	A value of 1 marks the alpha value (see 7.17 and 7.18).	
BYTE	Alpha	The alpha value from 0x00 to 0xFF.	

Table 7.22: Alpha element

## 7.23 Number of colors in the user list

The number of custom colors. {0 ... 10000, 0}

## 7.24 Colors in the user list

All colors included in the list of user colors when drawing. The structure of the memory is described in table 7.24. The number is given in section 7.23.

Туре	Name	Description	
BYTE	UserSelect	The value 1 marks the color (see 7.19 and 7.20).	
UINT32	Color	The color value from 0x00000000 to 0xFFFFFFFF.	

Table 7.24: Color element

## 7.25 Number of alpha gradients in the user list

The number of custom alpha gradients. {0 ... 10000, 0}

## 7.26 Alpha gradients in the user list

All alpha gradients displayed when drawing in alpha mode under user colors. The structure of the memory is described in table 7.26.1. The number of memory elements is given in section 7.25. Depending on the type, different values are contained in an element. The different display types are given in table 7.26.2.

Туре	Name	Description
BYTE	Туре	see Table 7.26.2 and section 12: Color gradients
BYTE	Begin	The alpha value at the beginning. (Type: 0,1,2,3,4)
BYTE	Between	The alpha value between beginning and middle. (Type: 4)
BYTE	Middle	The alpha value between beginning and end. (Type: 3,4)
BYTE	End	The alpha value at the end. (Type: 0,1,2,3,4)
INT32	Factor	The factor determines the slope. (Type: 1,2,3,4) { 0 100, 100}
INT32	FactorBetween	The factor determines position between beginning and middle. (Type: 4)
INT32	Length	The number of alpha values. (Type: 0,1,2,3,4)

## Table 7.26.1: Element to the alpha gradient

Value	Name	Description
0	Linear	A linear course with start and end value.
1	Cosine	Semicircular gradient with factor, start and end value as cosine curve.
2	Sine	Semicircular gradient with factor, start and end value as sine curve.
3	Round	A round gradient with factor, start, middle and end value.
4	RoundExtra	A round gradient with highlight as sine and cosine curve.

Table 7.26.2: Display types

## 7.27 Number of color gradients in the user list

The number of custom color gradients. {0 ... 10000, 0}

## 7.28 Color gradients in the user list

All color gradients displayed when drawing under user colors. A memory element is described in table 7.28. The number of elements can be found in section 7.27. An element can contain different values. All types are given in Table 7.26.2.

Туре	Name	Description
BYTE	Туре	See table 7.26.2 and section 12: Color gradients
UINT32	Begin	The color at the beginning. (Type: 0,1,2,3,4)
UINT32	Between	The color between beginning and middle. (Type: 4)
UINT32	Middle	The color between beginning and end. (Type: 3,4)
UINT32	End	The color in the end. (Type: 0,1,2,3,4)
INT32	Factor	The factor determines the slope. (Type: 1,2,3,4) {0 100, 100}
INT32	FactorBetween	The factor determines position between beginning and middle. (Type: 4)
INT32	Length	The number of colors. (Type: 0,1,2,3,4)

Table 7.28: Element to the color gradient

## 7.29 Selected alpha gradient

The alpha gradient is displayed as you draw in the display window. If no history is selected, the value is -1.  $\{-1 \dots 10000, -1\}$ 

## 7.30 Selected color gradient

The color gradient is displayed as you draw in the display window. If no history is selected, the value is -1.  $\{-1 ... 10000, -1\}$ 

## 7.31 Number of alpha values in the user list of text editing

The value is not used and must be 0.

## 7.32 Number of colors in the user list of text editing

The number of custom colors. {0 ... 10000, 0}

## 7.33 Colors in the user list of text editing

All font colors that are included under user colors in text editing. The structure of the memory is described in table 7.33. The number is given in section 7.32.

Туре	Name	Description	
BYTE	UserSelect	The value is not used and must be 0.	
UINT32	Color	The color value from 0x00000000 to 0xFF0000000.	

Table 7.33: Color element

## 7.34 Number of alpha gradients in text editing

The value is not used and must be 0.

# 7.35 Number of color gradients in text editing

The value is not used and must be 0.

## 7.36 Number of font types of the text editing

The number of custom fonts in the text editor. {0 ... 10000, 0}

## 7.37 Fonts in the user list of text editing

All fonts included in user fonts in text editing. The structure of the memory is described in table 7.37.1. The number of memory elements is given in section 7.36. The text line "InfoString" in the element indicates first the name of the font, then the size in pixels and finally a bit combination of the font styles. The font styles are described in table 7.37.2. The individual values are subdivided with a separator.

Example: "Arial | 30 | 1"

Туре	Name	Description	
INT32	InfoLength	The number of characters in the text line.	
WCHAR[]	InfoString	The text line with installation information.	

Table 7.37.1: Font element

Value	Name	Description	
0x00	Regular	The text is displayed normally.	
0x01	Bold	The text is displayed in bold.	
0x02	Italic	The text is displayed in italics.	
0x04	Underline	The text is displayed underlined.	
0x08	Strikeout	The text is displayed strike out.	

#### Table 7.37.2: Font styles

## 7.38 Selected font

The font is displayed in the text editor under fonts. If no font is selected, the value is -1.  $\{-1 \dots 10000, -1\}$ 

## 7.39 Selected image group

The image groups are described in section 11. The selected group is displayed in the image editor in the display window if the value is greater than -1. {-1 ... 10000, -1}

#### 8 Information about the playing file format

#### 8.1 Image with

The width of the image in pixels. The value can not be less than 0. The file type 2 (Sound) see section 5.2 can also contain a picture.

#### 8.2 Image height

The height of the image in pixels. The value can not be less than 0. The file type 2 (Sound) see section 5.2 can also contain a picture.

## 8.3 Playing time

The playing time is given in 100ns. The value can not be less than 1.

#### 8.4 Size of the image memory

The size of the preview image in bytes. The value can not be less than 0. If there is no image, the value 0 is specified.

#### 8.5 Image memory

The memory contains a preview image in "PNG" or "JPEG" format. The size in bytes is given in section 8.4. For video or animation, an image is created in the middle of the playing time.

#### 8.6 Size of the file memory

The size of the file memory in bytes. The value can not be less than 1 or greater than the remaining file size of the archive.

## 8.7 File memory

The memory contains the file data. The size in bytes is given in section 8.6.

## 9 Information about the animation format

## 9.1 Image with

The width of the image in pixels. The value can not be less than 0.

## 9.2 Image height

The height of the image in pixels. The value can not be less than 0.

## 9.3 Playing time

The playing time is given in 100ns. The value can not be less than 1.

## 9.4 Background color

The background color is specified in the program "PicturePaint" under settings. The color is displayed as background in display window during animation editing.

## 9.5 Size of the image memory

The size of the preview image in bytes. The value can not be less than 0. If there is no image, the value 0 is specified.

## 9.6 Image memory

The memory contains a preview image in "PNG" or "JPEG" format. The size in bytes is given in section 9.5.

## 9.7 Size of the file memory

The size of the file memory in bytes. The value can not be less than 1 or greater than the remaining file size of the archive.

## 9.8 File memory

The memory contains the file data. The size in bytes is given in section 9.7.

## 10 Information about the various file format

## 10.1 Size of the file memory

The size of the file memory in bytes. The value can not be less than 1 or greater than the remaining file size of the archive.

## 10.2 File memory

The memory contains the file data. The size in bytes is given in section 10.1.

## 11 Information about the image group format

## 11.1 Number of groups

The value indicates the number of image groups. The number can not be less than 0. If the value is 0, no image groups are used.

# 11.2 Name length

The number of characters in the name of the image group. The value can not be less than 1.

## 11.3 Name of the image group

The name consists of a certain number of characters (letters). The name length (see section 11.2) determines the number of characters. A character is a 16 bit unsigned value. The memory size for the name results from the name length times 2 bytes.

Memory size: GroupNameLength \* 2Bytes (16Bit)

#### 11.4 Number of image elements

The value indicates the number of elements in a image group. The number can not be less than 0. If the value is 0, there are no elements.

#### 11.5 Memory of a image element

The memory for a image element always consists of five values. The first four values indicate the position and size of the image section. The last value is a zero-based index that references an image file in the repository. All image files are counted and numbered when loading. The image groups are used in the "PicturePaint" program in image editing.

Туре	Name	Description	
INT32	ImagePosX	The position in X coordinates in the image.	11.5.1
INT32	ImagePosY	The position in Y coordinates in the image.	11.5.2
INT32	ImageWidth	The width of the image section.	11.5.3
INT32	ImageHeight	The height of the image section.	11.5.4
INT32	ImageIndex	The zero-based index for mapping the image file.	11.5.5

#### Table 11.5: Image element

#### 11.5.1 X position of the image element

The position is indicated in pixels. The value must not be less than 0. The X position and the width of the image section (see 11.5.3) must not exceed the width of the original image.

#### 11.5.2 Y position of the image element

The position is indicated in pixels. The value must not be less than 0. The Y position and the height of the image section (see 11.5.4) must not exceed the height of the original image.

#### 11.5.3 Width of the image element

The width is specified in pixels. The value can not be less than 1. The X position (see 11.5.1) and the width of the image section must not exceed the width of the original image.

#### 11.5.4 Height of the image element

The height is specified in pixels. The value can not be less than 1. The Y position (see 11.5.2) and the height of the image section must not exceed the height of the original image.

#### 11.5.5 The reference number for the image file

The assignment number refers to an image file in the repository. The number can not be less than 0 or greater than the number of image files in the archive. The image files are counted and numbered when loading.

### 12 Methods for creating color gradients

#### 12.1 Linear gradient

The method creates a linear gradient of two specified colors.

```
public UInt32[] CreateLinearColor(Int32 Count, Color Begin, Color End) {
  if(Count < 1) return new UInt32[0];</pre>
  UInt32[] ColorArray = new UInt32[Count];
  if(ColorArray.Length == 1) {
    ColorArray[0] = (UInt32) Begin.ToArgb();
    return ColorArray;
  }
  double A = End.A - Begin.A;
  double R = End.R - Begin.R;
  double G = End.G - Begin.G;
  double B = End.B - Begin.B;
  for(Int32 i = 0; i < Count; i++) {</pre>
    double Value;
    if(i == Count - 1)
      Value = 1.0;
    else
      Value = ((double) i) / (Count - 1);
    ColorArray[i] = (UInt32) Color.FromArgb(Convert.ToInt32(Begin.A + Value * A),
                                  Convert.ToInt32(Begin.R + Value * R),
                                  Convert.ToInt32(Begin.G + Value * G),
                                  Convert.ToInt32(Begin.B + Value * B)).ToArgb();
  }
  return ColorArray;
}
```

#### 12.2 Sine gradient

The first method creates a sine wave gradient of two colors. The second method uses an additional color as a highlight. For the factor see section 12.4.

```
public void CreateSineColor(UInt32[] ColorArray, Int32 Index, Int32 Count,
                                          Double Factor, Color Begin, Color End) {
  if(ColorArray.Length == 1) {
    ColorArray[0] = (UInt32) Begin.ToArgb();
    return;
  }
  double A = End.A - Begin.A;
  double R = End.R - Begin.R;
  double G = End.G - Begin.G;
double B = End.B - Begin.B;
  double dCount = (Count - 1) * (Count - 1);
  for(Int32 i = 0; i < Count; i++) {</pre>
    double Value = Math.Pow(1.0 - i * i / dCount, Factor);
    ColorArray[Index + Count - 1 - i] =
         (UInt32) Color.FromArgb(Convert.ToInt32(Begin.A + Value * A),
                                   Convert.ToInt32(Begin.R + Value * R),
                                   Convert.ToInt32(Begin.G + Value * G),
                                   Convert.ToInt32(Begin.B + Value * B)).ToArgb();
  }
}
```

```
public void CreateSineColor(UInt32[] ColorArray, Int32 Index, Int32 Count,
      Double Factor, Double FactorBegin, Color Begin, Color Middle, Color End) {
  if(Count < 4) {
    ColorArray[Index] = (UInt32) Begin.ToArgb();
    if(Count > 1) ColorArray[Index + 1] = (UInt32) Middle.ToArgb();
    if(Count > 2) ColorArray[Index + 2] = (UInt32) End.ToArgb();
    return;
  }
  if(FactorBegin < 0) FactorBegin = 0;</pre>
  if(FactorBegin > 1.0) FactorBegin = 1.0;
  Int32 CountA = Convert.ToInt32(Count * FactorBegin);
  Int32 CountB = Count - CountA;
  double dCount = (Count - 1) * (Count - 1);
  if(CountA < 3) {
    if(CountA > 0) ColorArray[Index] = (UInt32) Begin.ToArgb();
    if(CountA == 2) ColorArray[Index + 1] = (UInt32) Middle.ToArgb();
    Index += CountA;
  } else {
    double dRef = 1.0 / Math.Pow(1.0 - CountB * CountB / dCount, Factor);
    double A = Middle.A - Begin.A;
    double R = Middle.R - Begin.R;
    double G = Middle.G - Begin.G;
    double B = Middle.B - Begin.B;
    for(Int32 i = Count - 1; i >= CountB; i--, Index++) {
      double Value = Math.Pow(1.0 - i * i / dCount, Factor) * dRef;
      ColorArray[Index] = (UInt32) Color.FromArgb(
                                 Convert.ToInt32(Begin.A + Value * A),
                                 Convert.ToInt32(Begin.R + Value * R),
                                 Convert.ToInt32(Begin.G + Value * G),
                                 Convert.ToInt32(Begin.B + Value * B)).ToArgb();
   }
  }
  if(CountB < 3) {
    if(CountB > 0) ColorArray[Index + CountB - 1] = (UInt32) End.ToArgb();
    if(CountB == 2) ColorArray[Index + CountB - 2] = (UInt32) Middle.ToArgb();
  } else {
    double dRef = Math.Pow(1.0 - (CountB - 1) * (CountB - 1) / dCount, Factor);
    double A = End.A - Middle.A;
    double R = End.R - Middle.R;
    double G = End.G - Middle.G;
    double B = End.B - Middle.B;
    for(Int32 i = CountB - 1; i >= 0; i--, Index++) {
      double Value = (Math.Pow(1.0 - i * i / dCount, Factor)
                                                     - dRef) * 1.0 / (1.0 - dRef);
      ColorArray[Index] = (UInt32) Color.FromArgb(
                                 Convert.ToInt32(Middle.A + Value * A),
                                 Convert.ToInt32(Middle.R + Value * R),
                                 Convert.ToInt32(Middle.G + Value * G),
                                 Convert.ToInt32(Middle.B + Value * B)).ToArgb();
   }
 }
}
```



## 12.3 Cosine gradient

The method creates a cosine curve gradient of two colors. For the factor see 12.4.

```
public void CreateCosineColor(UInt32[] ColorArray, Int32 Index, Int32 Count,
                                        Double Factor, Color Begin, Color End) {
  if(ColorArray.Length == 1) {
    ColorArray[0] = (UInt32) Begin.ToArgb();
    return;
  }
  double A = End.A - Begin.A;
  double R = End.R - Begin.R;
  double G = End.G - Begin.G;
  double B = End.B - Begin.B;
  double dCount = (Count - 1) * (Count - 1);
  for(Int32 i = 0; i < Count; i++) {</pre>
    double Value = Math.Pow(1.0 - i * i / dCount, Factor);
    ColorArray[Index + i] = (UInt32) Color.FromArgb(
                                  Convert.ToInt32(End.A - Value * A),
                                 Convert.ToInt32(End.R - Value * R),
                                 Convert.ToInt32(End.G - Value * G),
                                  Convert.ToInt32(End.B - Value * B)).ToArgb();
  }
}
```

## 12.4 Round gradient

The round gradient consists of a sine and cosine curve. The second method uses an additional color as a highlight. The two factors from the file format (see sections 7.26 and 7.28) are divided by 100. The factor (INT32) can be a number between 0 and 100. The factor (Double) between 0.0 and 1.0.

```
public UInt32[] CreateRoundColor(Int32 Count, Double Factor, Color Begin,
                                                      Color Middle, Color End) {
  if(Count < 1) return new UInt32[0];</pre>
  UInt32[] ColorArray = new UInt32[Count];
  if(Count < 4) {
    if(Count == 1) {
      ColorArray[0] = (UInt32) Middle.ToArgb();
      return ColorArray;
    }
    if(Count == 2) {
      ColorArray[0] = (UInt32) Middle.ToArgb();
      ColorArray[1] = (UInt32) End.ToArgb();
      return ColorArray;
    }
    ColorArray[0] = (UInt32) Begin.ToArgb();
    ColorArray[1] = (UInt32) Middle.ToArgb();
    ColorArray[2] = (UInt32) End.ToArgb();
    return ColorArray;
  }
  Int32 SizeA = Count / 2;
  Int32 SizeB = Count - SizeA;
  this.CreateSineColor(ColorArray, 0, SizeA, Factor, Begin, Middle);
  this.CreateCosineColor(ColorArray, SizeA, SizeB, Factor, Middle, End);
  return ColorArray;
}
```

```
File format: 2D Picture Paint Project
                                                             Version 1 05.12.2017 Page 18
public UInt32[] CreateRoundColor(Int32 Count, Double Factor, Double FactorBetween,
                           Color Begin, Color Between, Color Middle, Color End) {
  if(Count < 1) return new UInt32[0];</pre>
  UInt32[] ColorArray = new UInt32[Count];
  if(Count < 5) {
    if(Count == 1) {
      ColorArray[0] = (UInt32) Middle.ToArgb();
      return ColorArray;
    }
    if(Count == 2) \{
      ColorArray[0] = (UInt32) Middle.ToArgb();
      ColorArray[1] = (UInt32) End.ToArgb();
      return ColorArray;
    }
    if(Count == 3) \{
      ColorArray[0] = (UInt32) Begin.ToArgb();
      ColorArray[1] = (UInt32) Middle.ToArgb();
      ColorArray[2] = (UInt32) End.ToArgb();
      return ColorArray;
    }
    ColorArray[0] = (UInt32) Begin.ToArgb();
    ColorArray[1] = (UInt32) Between.ToArgb();
    ColorArray[2] = (UInt32) Middle.ToArgb();
    ColorArray[3] = (UInt32) End.ToArgb();
    return ColorArray;
  }
  Int32 SizeA = Count / 2;
```

## 13 Program for reading the file format

Int32 SizeB = Count - SizeA;

On the website of PanotiSoft under technical documents a test program is available, with which the file format can be read structured. In addition, the program code can also be downloaded. The program was written under Visual Studio 2008 in the programming language C #.

Open	File:	E:\Test.ppp	Format: Pict	ture Paint Project (.ppp)	
Position	Size	Туре	Name	Value	
0	0		BeginRead		
0	4	UINT32	IDNumber	0x50504454 hex	
4	8	INT64	FileSize	43.593.135 bytes	
12	1	BYTE	Version	1	
13	4	INT32	PreviewImageSize	2.702 bytes	
17	2702	MEMORY	PreviewImage		
2719	4	INT32	FileCount	12	
2723	1	BYTE	FileType	0 (Image)	
2724	4	INT32	FileNameLength	7 letters	
2728	14	WCHAR[]	FileName	Sanduhr	
2742	4	INT32	FileExtentionLength	4 letters	
2746	8	WCHAR[]	FileExtention	.png	
2754	1	BYTE	ImageFormat	7 (png)	
2755	2	UINT16	ImageFlags	0x1FD2	

Program:	FileViewerX64.zip oder FileViewerX32.zip	
Project file:	FileViewerCode.zip	
Description:	FileViewer.pdf	
FileViewerCod	<u>e</u> :	
Format file: Format class:	FileViewerFormat.cs FileViewerTDPicturePaintProject	