1 Introduction

The file format describes a simple listing of files that can be identified by name and therefore differentiate. The file archive contains all file data and can be read by other programs. The additional information, such as the image width, height or the playing time, are only needed for the program "PicturePaint". The file extension is called ".ppc" (Picture Paint Collection).

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 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.

3 Description

3.1 File format

Туре	Name	Description	Info
UINT32	IDNumber	The file must have the ID number $(0x43504454)$.	4.1
INT64	FileSize	The size of the entire file in bytes.	4.2
BYTE	Version	The version for this file format is 1.	4.3
		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 file extension.	5.5
WCHAR[]	FileExtention	The name of the file extension.	5.6
BYTE	ImageFormat	The picture format as a number.	5.7
INT32	ImageWidth	The width of the picture.	5.8
INT32	ImageHeight	The height of the picture.	5.9
INT64	PlayerDuration	The playing time of the file.	5.10
INT32	FileMemorySize	The size of the file memory in bytes.	5.11
BYTE[]	FileMemory	The memory with the file data.	5.12
} <-	File		

Table 3.2: Data format

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 (TDPC: Two Dimension PicturePaint Collection).

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

The version number is always 1 for this description.

5 Information about the data format

5.1 Number of files

The number of files in the archive. The value can not be less than 0. If the value is 0, there are no files in the archive.

5.2 File type

The file type identifies the format of the file in memory (see section 5.12). Not all file formats are detected by the program "PicturePaint". All files that can not be assigned must have the number 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 animation.
Animation	4	The animation (.tda) is created with the program "PicturePaint".
Various	5	The file is not assigned by the PicturePaint program.

Table 5.2: File type

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 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 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 the file extension does not exist, the value is 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)

5.7 Image format

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

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 (2D picture).

Table 5.7: Image format

5.8 Image width

The width of the representation in pixels. The value can not be less than 0. The file type 2 (Sound) see section 5.2 can contain a picture. For type 5 (Various) the value is always 0.

5.9 Image height

The height of the representation in pixels (pixels). The value can not be less than 0. The file type 2 (Sound) see section 5.2 can contain a picture. For type 5 (Various) the value is always 0.

5.10 Playing time

The playing time is given in 100ns. The value can not be less than 0. If the file type (see section 5.2) indicates a picture or different files, the playing time is 0.

5.11 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.

5.12 File memory

The memory contains the file data. The size in bytes is given in section 5.11. The first image file in the listing is used as a preview image. The file type (section 5.2) must be 0. By default, a PNG or JPEG image format should be selected.

6 Simple Program

The program was written under Visual Studio 2008 in the programming language C #. It is kept simple and reads only the file data by name.

```
public static Dictionary<String, MemoryStream> Load(String FilePath) {
  //Dictionary: System.Collections.Generic
  //FileStream, BinaryReader, MemoryStream: System.IO
  try {
    FileStream Stream = new FileStream(FilePath, FileMode.Open, FileAccess.Read);
    BinaryReader Reader = new BinaryReader(Stream);
    if(Reader.ReadUInt32() != 0x43504454) return null;
    if(Reader.ReadInt64() != Stream.Length) return null;
    Dictionary<String, MemoryStream> FileDictionary = new Dictionary<String,</pre>
                                                                     MemoryStream>();
    Int32 FileCount = Reader.ReadInt32();
    for(Int32 i = 0; i < FileCount; i++) {</pre>
      Stream.Position++;
      Int32 FileNameLength = Reader.ReadInt32();
      if(FileNameLength < 1 || FileNameLength > 260) return null;
      String FileName = String.Empty;
      while(FileNameLength > 0) {
        FileName += (Char) Reader.ReadUInt16();
        FileNameLength--;
      }
      Int32 FileExtentionLength = Reader.ReadInt32();
      if(FileExtentionLength < 0 || FileExtentionLength > 260) return null;
      Stream.Position += FileExtentionLength * 2 + 1 + 4 + 4 + 8;
      Int32 FileSize = Reader.ReadInt32();
      if(FileSize < 1 || Stream.Position + FileSize > Stream.Length) return null;
      FileDictionary[FileName] = new MemoryStream(Reader.ReadBytes(FileSize));
    }
    if(Stream.Position != Stream.Length) return null;
    Stream.Close();
    return FileDictionary;
  } catch(Exception) {
  return null;
}
```

7 Program for reading the file format

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

Open	File:	E:\Test.ppc	Format: F	Picture Paint Collection (.p	opc) 🔻
Position	Size	Туре	Name	Value	*
0	0		BeginRead		E
0	4	UINT32	IDNumber	0x43504454 hex	
4	8	INT64	FileSize	41.382.120 bytes	
12	4	INT32	ItemCount	12	
16	1	BYTE	FileType	0	
17	4	INT32	FileNameLength	4 letters	
21	8	WCHAR[]	FileName	Bild	
29	4	INT32	FileExtentionLength	4 letters	
33	8	WCHAR[]	FileExtention	.png	
41	1	BYTE	ImageFormat	7	
42	4	INT32	ImageWidth	300 bytes	
46	4	INT32	Image Height	300 bytes	
50	8	INT64	PlayerDuration	00:00:00	-

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

Format file:	FileViewerFormat.cs
Format class:	FileViewerTDPicturePaintCollection