1 Introduction

Various fonts and colored text content can be saved with this file. In addition, information about the text width, text alignment and separator are included in the file format. The text file is created using the program "TextWriter". The file extension is called ".tdft" (Two Dimension Format Text).

2 Value types

Type	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
!	Dependence	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 Dependence

The value is only available if a certain bit (flags) has been set.

3 Description

3.1 File format

Type	Name	Description	Info
UINT32	IDNumber	The file must have the ID number (0x57544454).	4.1
BYTE	Version	The file version must be 1 for this description.	4.2
BYTE	Alignment	The alignment of the text.	4.3
BYTE	Flags	The bits for the format description.	4.4
INT32!	ThumbnailSize	The size of the thumbnail in bytes.	4.5
MEMORY!	ThumbnailImage	The memory is an image file.	4.6
WCHAR!	Separator	The separator used in the text.	4.7
INT32!	TextWidth	A certain width of the presentation.	4.8
INT32	LetterLength	The number of characters in the text.	4.9
...		3.2 Font format, Table 3.2	

Table 3.1: File format

3.2 Font format

Type	Name	Description	Info
INT32	FontCount	The number of fonts.	5.1
-> {	Font		
INT32	FontNameLength	The number of characters in the name of the font.	5.2
WCHAR[]	FontName	The name of the font.	5.3
BYTE	FontStyle	The styles of the font.	5.4
FLOAT	FontSize	The height of the font in pixels.	5.5
} <-	Font		
...		3.3 Color format, Table 3.3	

Table 3.2: Font format

3.3 Color format

Type	Name	Description	Info
INT32	ColorCount	The number of color.	6.1
-> {	Color		
UINT32	ColorValue	The ARGB colors in the text.	6.2
} <-	Color		
...		3.4 Letter format, Table 3.4	

Table 3.3: Color format

3.4 Letter format

Type	Name	Description	Info
UINT16[]	LetterMemory	All characters in the text.	7.1
BYTE[]	LetterInfoMemory	The information about each character in the text.	7.2
INT16[]	LetterSizeMemory	The size information about each character in the text.	7.3

Table 3.4: Letter 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 (TDTW: Two Dimension Text Writer).

4.2 Version number

The version number is always 1 for this format description.

4.3 Text alignment

The value tells you how the text is aligned.

Name	Value	Description
Left	0	The text is aligned left justified.
Center	1	The text is centered.
Right	2	The text is aligned right justified.
Justified	3	The text is displayed as a justified sentence.

Table 4.3: Text alignment

4.4 Format bits

The bits determine the further content in the file format. If the bit (Compress) is set, the format for the compressed text must be used. This format is called "TDCompressText".

Name	Value	Description
Separator	0x01	The file format contains a separator (see 4.7).
TextWidth	0x02	The file format contains a specific text width (see 4.8).
Compress	0x04	The file format is compressed (see format: TDCompressText).
Thumbnail	0x08	The file format contains a thumbnail image (see 4.5 and 4.6).

Table 4.4: Format bits

4.5 Thumbnail size

The size is specified in bytes. It can not be negative or 0 (see 4.4 Format bits).

4.6 Thumbnail image

The thumbnail shows a specific text section. The image can be saved in PNG, JPEG, TIFF or BMP formats. By default, the PNG format is used.

4.7 Separator

A separator appears as a hyphen (-) by default. The character is inserted in word syllables to better represent a text visually. If this value is not present, no word breaks are used (see 4.4 Format bits).

4.8 Display width

The value specifies a certain display width. If this value is not specified, the current window width should be used as the text width (see 4.4 Format bits).

4.9 Number of characters in the text

The value indicates the number of characters in the text. This includes the special characters for a new line. The number can not be negative. If the value is 0, the file format is terminated here.

5 Information about the font format

5.1 Number

The number of fonts contained. The value can not be less than 1 and greater than 256. The run of the loop (Font) is determined by this value.

5.2 Name length

The number of characters in the name of the font.

5.3 Name

The name consists of a certain number of characters (letters). The name length (5.2) 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: `FontNameLength * 2Bytes (16Bit)`

5.4 Font styles

The font styles are given in bits and can therefore be combined.

For example: ***Bold, Italic*** (0x03)

Name	Value	Description
Regular	0x00	The text is displayed without further styles.
Bold	0x01	The characters are drawn in bold.
Italic	0x02	The characters are drawn in italics.
Underline	0x04	The characters are underlined.
Strikout	0x08	The characters are crossed out.

Table 5.4: Font styles

5.5 Font size

The font size is specified in pixels. The value is defined as a floating point number. The decimal places are not used because the font height is specified in pixels.

6 Information about the color format

6.1 Number

The number of colors in the text. The value can not be less than 1 or greater than 256. The run of the loop (Color) is determined by this value.

6.2 Color

The color (ARGB) is given as a 32bit value. The color component "Alpha" is not used.

7 Information about the character format

7.1 Character memory

The memory contains all characters as 16bit values. The number is given in section 4.9.

Memory size: `LetterLength * 2Bytes (16Bit)`

7.2 Character information

The information memory contains three byte values for each character (see section 4.9). The first for the character type, the second as the font index and the third as the color index.

Type	Name	Description
BYTE	Type	The character type is described in table 7.2.2.
BYTE	Fonts	The zero based index of the stored fonts.
BYTE	Colors	The zero based index of stored colors.

Table 7.2.1: Information memory

The character type contains additional information about the character used. A word that can be split contains the value 0x01 (Seperator) at the separation point. In addition, a separator is displayed after the letter (see section 4.7). For a space, the value 0x02 (Space) is specified. A line break consists of two special characters. At the end the character "Carriage Return" with the value 0x0D is inserted. The character at the beginning of the line has the value 0x0A and is called "New Line". Both special characters have the character type (Control) with the value 0x03. The beginning, the new line is additionally marked with the bit 0x04 (NewLine), resulting in the character type with the value 0x07. If a specific text width is used (see section 4.4), the bit (NewLine) may also have been set at the first letter of a word.

Name	Value	Description
Letter	0x00	A character without additional information.
Seperator	0x01	The word can be divided with a separator.
Space	0x02	The character is a space.
Control	0x03	The special character is specified for a line break.
NewLine	0x04	The bit is set when a new line starts.

Table 7.2.2: Character type

7.3 Size information

The size memory contains three 16bit values for each character (see section 4.9). These widths are labeled a, b and c and are indicated in pixels. The values a and c can be negative or 0.

Type	Name	Description
INT16	a	The overhang at the beginning of the character.
INT16	b	The width of the displayed character.
INT16	c	The overhang at the end of the character.

Table 7.3: Size memory

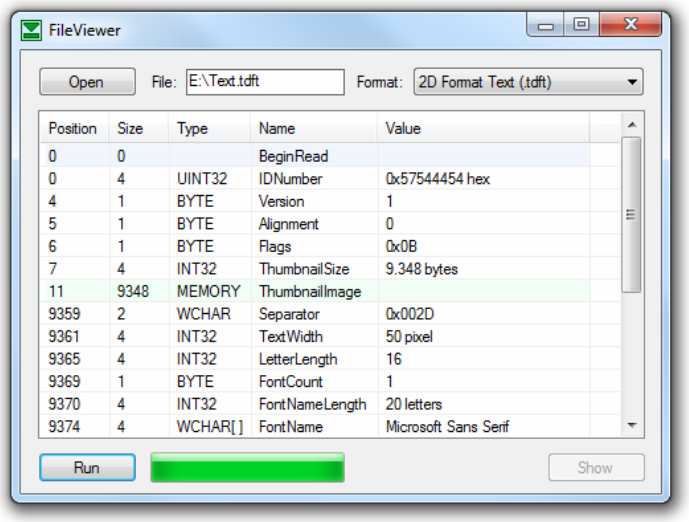
Example:

Figure 7.3: Letter widths

The letter "b" starts with a distance (a) to the previous character "A". The width (b) of the letter passes over the following character "c". At the end of the letter "b" the negative distance (c) for the overhang is deducted. The character "c" is displayed at the current position.

8 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#.



Program: FileViewerX64.zip oder FileViewerX32.zip

Project file: FileViewerCode.zip

Description: FileViewer.pdf

FileViewerCode:

Format file: FileViewerFormat.cs
Format class: FileViewerTDFormatText