

## 1 Einleitung

Mit dieser Datei können verschiedene Schriften und farbige Textinhalte gespeichert werden. Zusätzlich sind Angaben zur Textbreite, Ausrichtung und Trennzeichen im Dateiformat enthalten. Die Textdatei wird mit dem Programm "TextWriter" erstellt. Die Dateierweiterung wird ".tdft" (Two Dimension Format Text) genannt.

## 2 Wertetypen

Typ	Beschreibung	Bereich
INT8	8Bit mit Vorzeichen	-128 bis 127
INT16	16Bit mit Vorzeichen	-32.768 bis 32.767
INT32	32Bit mit Vorzeichen	-2.147.483.648 bis 2.147.483.647
INT64	64Bit mit Vorzeichen	-9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807
BYTE	8Bit ohne Vorzeichen	0 bis 255
UINT16	16Bit ohne Vorzeichen	0 bis 65.535
UINT32	32Bit ohne Vorzeichen	0 bis 4.294.967.295
UINT64	64Bit ohne Vorzeichen	0 bis 18.446.744.073.709.551.615
CHAR	8Bit Zeichen	0 bis 255
WCHAR	16Bit Zeichen	0 bis 65.535
FLOAT	32Bit Gleitkommazahl	± 1.5e-45 zu ± 3.4e38
DOUBLE	64Bit Gleitkommazahl	± 5.0e-324 zu ± 1.7e308
MEMORY	Speicher in Bytes	
...[]	Wertemenge	siehe Abschnitt 2.1
-> {	Beginn der Schleife	siehe Abschnitt 2.2
} <-	Ende der Schleife	siehe Abschnitt 2.2
...	Nächste Tabelle	siehe Abschnitt 2.3
!	Abhängigkeit	Siehe Abschnitt 2.4

Tabelle 2: Wertetypen

### 2.1 Wertemenge

Die Menge besteht aus einem bestimmten Wertetyp. Die Anzahl der Menge wird in den Informationen detailliert angegeben und ist meistens der vorherige Formatwert.

#### Beispiel:

Eine Menge INT16[] enthält eine bestimmte Anzahl von INT16 Werten { INT16, INT16, INT16, INT16, ... }.

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

### 2.2 Die Schleife

Bei einer Schleife wird das Format wiederholt durchlaufen. Die Anzahl der Durchläufe wird in den Informationen detailliert angegeben und ist meistens der vorherige Wert.

### 2.3 Nächste Tabelle

In dem angegebenen Abschnitt und der Tabelle wird das Dateiformat weiter fortgesetzt.

### 2.4 Abhängigkeit

Der Wert ist nur vorhanden, wenn ein bestimmtes Bit (Flags) gesetzt wurde.

### 3 Beschreibungen

#### 3.1 Dateiformat

Typ	Name	Beschreibung	Info
UINT32	IDNumber	Die Datei muss die ID Nummer (0x57544454) haben.	4.1
BYTE	Version	Die Dateiversion muss für diese Beschreibung 1 sein.	4.2
BYTE	Alignment	Die Ausrichtung des Textes.	4.3
BYTE	Flags	Die Bits zur Formatbeschreibung.	4.4
INT32!	ThumbnailSize	Die Größe des Miniaturbildes in Bytes.	4.5
MEMORY!	ThumbnailImage	Der Speicher ist eine Bilddatei.	4.6
WCHAR!	Separator	Das verwendete Trennzeichen im Text.	4.7
INT32!	TextWidth	Eine bestimmte Breite der Darstellung.	4.8
INT32	LetterLength	Die Anzahl der Zeichen im Text.	4.9
...		3.2 Schriftformat, Tabelle 3.2	

Tabelle 3.1: Dateiformat

#### 3.2 Schriftformat

Typ	Name	Beschreibung	Info
INT32	FontCount	Die Anzahl der Schriften.	5.1
-> {	Font		
INT32	FontNameLength	Die Anzahl der Zeichen im Namen des Schrifttyps.	5.2
WCHAR[]	FontName	Der Name der Schrift.	5.3
BYTE	FontStyle	Die Stile der Schrift.	5.4
FLOAT	FontSize	Die Höhe der Schrift in Bildpunkte.	5.5
} <-	Font		
...		3.3 Farbformat, Tabelle 3.3	

Tabelle 3.2: Schriftformat

#### 3.3 Farbformat

Typ	Name	Beschreibung	Info
INT32	ColorCount	Die Anzahl der Farben.	6.1
-> {	Color		
UINT32	ColorValue	Die ARGB Farben im Text.	6.2
} <-	Color		
...		3.4 Zeichenformat, Tabelle 3.4	

Tabelle 3.3: Farbformat

#### 3.4 Zeichenformat

Typ	Name	Beschreibung	Info
UINT16[]	LetterMemory	Alle Zeichen im Text.	7.1
BYTE[]	LetterInfoMemory	Die Informationen über jedes Zeichen im Text.	7.2
INT16[]	LetterSizeMemory	Die Größenangaben über jedes Zeichen im Text.	7.3

Tabelle 3.4: Zeichenformat

## 4 Informationen zum Dateiformat

### 4.1 Identifikationsnummer

Die Identifikationsnummer kennzeichnet das Dateiformat. Die Nummer kann auch mit 4 Buchstaben (TDTW: Two Dimension Text Writer) dargestellt werden.

### 4.2 Versionsnummer

Die Versionsnummer ist für diese Formatbeschreibung immer 1.

### 4.3 Textausrichtung

Der Wert gibt Auskunft darüber, wie der Text ausgerichtet ist.

Name	Werte	Beschreibung
Left	0	Der Text wird linksbündig ausgerichtet.
Center	1	Der Text wird mittig ausgerichtet.
Right	2	Der Text wird rechtsbündig ausgerichtet.
Justified	3	Der Text wird als Blocksatz dargestellt.

Tabelle 4.3: Textausrichtung

### 4.4 Formatbits

Die Bits bestimmen den weiteren Inhalt im Dateiformat. Ist das Bit (Compress) gesetzt, muss das Format für den komprimierten Text verwendet werden. Dieses Format heißt "TDCompressText".

Name	Werte	Beschreibung
Separator	0x01	Das Dateiformat enthält ein Trennzeichen (siehe 4.7).
TextWidth	0x02	Das Dateiformat enthält eine bestimmte Textbreite (siehe 4.8).
Compress	0x04	Das Dateiformat ist komprimiert (siehe Format: TDCompressText).
Thumbnail	0x08	Das Dateiformat enthält ein Miniaturbild (siehe 4.5 und 4.6).

Tabelle 4.4: Formatbits

### 4.5 Miniaturbildgröße

Die Größe wird in Bytes angegeben. Sie kann nicht negativ oder 0 sein (siehe 4.4 Formatbits).

### 4.6 Miniaturbild

Das Miniaturbild zeigt einen bestimmten Textausschnitt. Das Bild kann in den Formaten PNG, JPEG, TIFF oder BMP gespeichert sein. Als Standard wird das PNG Format verwendet.

### 4.7 Trennzeichen

Ein Trennzeichen wird standardmäßig als Bindestrich (-) angezeigt. Das Zeichen wird bei Wortsilben eingefügt, um einen Text optischer besser darzustellen. Ist dieser Wert nicht vorhanden, werden keine Wortumbrüche verwendet (siehe 4.4 Formatbits).

### 4.8 Darstellungsbreite

Der Wert gibt eine bestimmte Darstellungsbreite vor. Wird dieser Wert nicht angegeben, soll die aktuellen Fensterbreite als Textbreite verwendet werden (siehe 4.4 Formatbits).

### 4.9 Anzahl der Zeichen im Text

Der Wert gibt die Anzahl der Zeichen im Text an. Dazu gehören auch die Sonderzeichen für eine neue Zeile. Die Zahl kann nicht negativ sein. Ist der Wert 0 ist das Dateiformat hier beendet.

## 5 Informationen zum Schriftformat

### 5.1 Anzahl

Die Anzahl der enthaltenen Schriften. Der Wert kann nicht kleiner als 1 und größer als 256 sein. Der Durchlauf der Schleife (Font) wird mit diesem Wert bestimmt.

### 5.2 Namenlänge

Die Anzahl von Zeichen im Namen des Schrifttyps.

### 5.3 Name

Der Name besteht aus einer bestimmten Anzahl von Zeichen (Buchstaben). Die Namenlänge (5.2) bestimmt die Anzahl der Zeichen. Ein Zeichen ist ein 16Bit Wert ohne Vorzeichen. Die Speichergröße für den Namen ergibt sich aus der Namenlänge mal 2Bytes.

Speichergröße: `FontNameLength * 2Bytes (16Bit)`

### 5.4 Schriftstile

Die Schriftstile sind in Bits angegeben und können somit auch kombinierte werden.

Zum Beispiel: **Bold, Italic** (0x03)

Name	Werte	Beschreibung
Regular	0x00	Der Text wird ohne weitere Stile dargestellt.
Bold	0x01	Die Zeichen werden Fett gezeichnet.
Italic	0x02	Die Zeichen werden kursive gezeichnet.
Underline	0x04	Die Zeichen werden unterstrichen dargestellt.
Strikout	0x08	Die Zeichen werden durchgestrichen dargestellt.

Tabelle 5.4: Schriftstile

### 5.5 Schriftgröße

Die Schriftgröße wird in Bildpunkte angegeben. Der Wert ist als Gleitkommazahl definiert. Die Nachkommastellen werden nicht verwendet, weil die Schrifthöhe in Bildpunkte angegeben ist.

## 6 Informationen zum Farbformat

### 6.1 Anzahl

Die Anzahl der enthaltenen Farben im Text. Der Wert kann nicht kleiner als 1 oder größer als 256 sein. Der Durchlauf der Schleife (Color) wird mit diesem Wert bestimmt.

### 6.2 Farbe

Die Farbe (ARGB) wird als 32Bit Wert angegeben. Die Farbkomponente "Alpha" wird nicht verwendet.

## 7 Informationen zum Zeichenformat

### 7.1 Zeichenspeicher

Der Speicher enthält alle Zeichen als 16Bit Werte. Die Anzahl ist im Abschnitt 4.9 angegeben.

Speichergröße: `LetterLength * 2Bytes (16Bit)`

## 7.2 Informationen

Der Informationsspeicher enthält für jedes Zeichen (siehe Abschnitt 4.9) drei Bytewerte. Den ersten für den Zeichentyp, den zweiten als Schriftindex und den dritten als Farbindex.

Typ	Name	Beschreibung
BYTE	Typ	Der Zeichentyp wird in der Tabelle 7.2.2 beschrieben.
BYTE	Schrift	Der null basierte Index der gespeicherten Schriften.
BYTE	Farbe	Der null basierte Index der gespeicherten Farben.

Tabelle 7.2.1: Informationsspeicher

Der Zeichentyp enthält Zusatzinformationen zum verwendeten Zeichen. Ein Wort, dass geteilt werden kann, enthält an der Trennstelle den Wert 0x01 (Seperator). Zusätzlich wird nach dem Buchstaben ein Trennzeichen angezeigt (siehe Abschnitt 4.7). Bei einem Leerzeichen wird der Wert 0x02 (Space) angegeben. Ein Zeilenumbruch besteht aus zwei Sonderzeichen. Am Ende wird das Zeichen "Carriage Return" mit dem Wert 0x0D eingefügt. Das Zeichen am Anfang der Zeile hat den Wert 0x0A und wird mit "New Line" bezeichnet. Beide Sonderzeichen haben den Zeichentyp (Control) mit dem Wert 0x03. Der Anfang, der neuen Zeile wird zusätzlich mit dem Bit 0x04 (NewLine) gekennzeichnet, daraus ergibt sich der Zeichentyp mit dem Wert 0x07. Wird eine bestimmte Textbreite verwendet (siehe Abschnitt 4.4), kann das Bit (NewLine) auch beim ersten Buchstaben eines Wortes gesetzt worden sein.

Name	Wert	Beschreibung
Letter	0x00	Ein Zeichen ohne zusätzlichen Informationen.
Seperator	0x01	Das Wort kann mit einem Trennzeichen geteilt werden.
Space	0x02	Das Zeichen ist ein Leerzeichen.
Control	0x03	Das Sonderzeichen wird bei einem Zeilenumbruch angegeben.
NewLine	0x04	Das Bit wird gesetzt, wenn eine neue Zeile beginnt.

Tabelle 7.2.2: Zeichentyp

## 7.3 Größenangaben

Der Größenspeicher enthält für jedes Zeichen (siehe Abschnitt 4.9) drei 16Bit Werte. Diese Breiten werden mit a, b und c gekennzeichnet und sind in Bildpunkte angegeben. Die Werte a und c können negativ oder auch 0 sein.

Typ	Name	Beschreibung
INT16	a	Der Überhang am Anfang des Zeichen.
INT16	b	Die Breite des dargestellten Zeichens.
INT16	c	Der Überhang am Ende des Zeichen.

Tabelle 7.3: Größenspeicher

Beispiel:

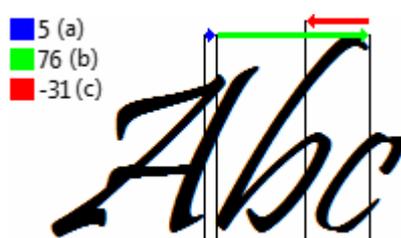
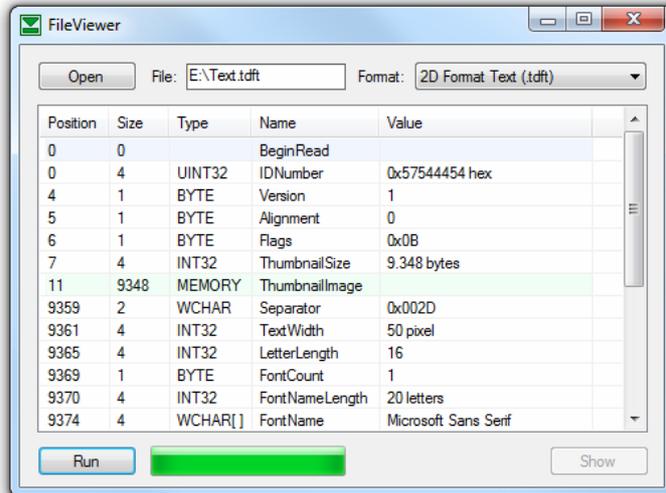


Bild 7.3: Buchstabenbreiten

Der Buchstabe "b" beginnt mit einem Abstand (a) zum vorherigen Zeichen "A". Die Breite (b) des Buchstaben geht über das folgenden Zeichen "c" hinweg. Am Ende des Buchstaben "b" wird noch der negativen Abstand (c) für den Überhang abgezogen. An der aktuellen Position wird dann das Zeichen "c" dargestellt.

## 8 Programm zum Auslesen des Dateiformats

Auf der Internetseite von PanotiSoft ist unter technische Dokumente ein Testprogramm vorhanden, mit dem das Dateiformat strukturiert ausgelesen werden kann. Zusätzlich kann auch der Programmcode herunter geladen werden. Das Programm wurde unter Visual Studio 2008 in der Programmiersprache C# geschrieben.



Programm: FileViewerX64.zip oder FileViewerX32.zip

Projektdatei: FileViewerCode.zip

Beschreibung: FileViewer.pdf

FileViewerCode:

Formatdatei: FileViewerFormat.cs

Formatklasse: FileViewerTDFormatText