

1 Einleitung

Das Dateiformat beschreibt die Projektdatei für das Programm "PicturePaint". Das Archiv kann alle Dateitypen enthalten. Für alle Bilddateien sind zusätzlich verschiedene Einstellungswerte im Dateiformat vorhanden. Die Dateierweiterung wird ".ppp" (Picture Paint Project) 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
...!	Tabelle mit Bedingung	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 Nächste Tabelle mit Bedingung

Das Dateiformat wird in dem Abschnitt und der Tabelle fortgesetzt, wenn die Bedingung erfüllt ist.

3 Beschreibungen

3.1 Dateiformat

Typ	Name	Beschreibung	Info
UINT32	IDNumber	Die Datei muss die ID Nummer (0x50504454) haben.	4.1
INT64	FileSize	Die Größe der gesamten Datei in Bytes.	4.2
BYTE	Version	Die Versionsnummer ist in diesem Dateiformat gleich 1.	4.3
INT32	PreviewImageSize	Die Größe des Speichers mit dem Vorschaubild in Bytes.	4.4
MEMORY	PreviewImage	Der Speicher mit der Vorschaubilddatei.	4.5
...		3.2 Datenformat, Tabelle 3.2	

Tabelle 3.1: Dateiformat

3.2 Datenformat

Typ	Name	Beschreibung	Info
INT32	FileCount	Die Anzahl der Dateien.	5.1
-> {	File		
BYTE	FileType	Der Dateityp als Nummer.	5.2
INT32	FileNameLength	Die Anzahl der Zeichen im Namen der Datei.	5.3
WCHAR[]	FileName	Der Name der Datei.	5.4
INT32	FileExtentionLength	Die Anzahl der Zeichen im Namen der Dateierweiterung.	5.5
WCHAR[]	FileExtention	Der Name der Dateierweiterung.	5.6
...!	FileType = 0	3.3 Bildformat, Tabelle 3.2	
...!	FileType = 1	3.5 Abspielendes Dateiformat (Video), Tabelle 3.5	
...!	FileType = 2	3.5 Abspielendes Dateiformat (Musik), Tabelle 3.5	
...!	FileType = 3	3.5 Abspielendes Dateiformat (GIF), Tabelle 3.5	
...!	FileType = 4	3.6 Animationsformat (TDA), Tabelle 3.6	
...!	FileType = 5	3.7 Verschiedene Dateiformate, Tabelle 3.7	
} <-	File		
...		3.8 Bildgruppenformat, Tabelle 3.8	

Tabelle 3.2: Datenformat

3.3 Bildformat

Typ	Name	Beschreibung	Info
BYTE	ImageFormat	Das Bildformat als Nummer.	6.1
UINT16	ImageFlags	Die Einstellungsbits für das Programm "PicturePaint".	6.2
INT32	ImagePosX	Die Bildposition in X-Koordinaten.	6.3
INT32	ImagePosY	Die Bildposition in Y-Koordinaten.	6.4
INT32	ImageWidth	Die Breite des Bildes.	6.5
INT32	ImageHeight	Die Höhe des Bildes.	6.6
INT32	FileImageSize	Die Größe des Bildspeichers in Bytes.	6.7
BYTE[]	FileImage	Der Speicher mit der Bilddatei.	6.8
...		3.4 Bildeinstellungsformat, Tabelle 3.4	

Tabelle 3.3: Bildformat

3.4 Bildeinstellungsformat

Typ	Name	Beschreibung	Info
INT32	ToolIndex	Zuletzt verwendetes Werkzeug beim Zeichnen.	7.1
BYTE	PaintAlpha	Zuletzt verwendeter Alphawert beim Zeichnen.	7.2
UINT32	PaintColor	Zuletzt verwendete Farbe beim Zeichnen.	7.3
UINT32	BackColor	Die eingestellte Hintergrundfarbe im Fenster.	7.4
UINT32	ImageColor	Die eingestellte Erweiterungsfarbe für Bilder.	7.5
UINT32	ImageTransparentColor	Die transparente Farbe beim Zeichnen von Bildern.	7.6
UINT32	TextColor	Die Schriftfarbe beim Zeichnen von Textinhalten.	7.7
INT32	PaintLineSize	Die Dicke der Linien beim Zeichnen.	7.8
INT32	PaintRoundWidth	Breite der Rundung eines abgerundeten Rechtecks.	7.9
INT32	PaintRoundHeight	Höhe der Rundung eines abgerundeten Rechtecks.	7.10
INT32	PaintRotationLeverLength	Länge des Hebelarms beim rotierenden Farbverlauf.	7.11
DOUBLE	PaintRotationAngle	Der Winkel beim rotierenden Farbverlauf.	7.12
INT32	ImageRotationLeverLength	Länge des Hebelarms beim rotierenden Bild.	7.13
DOUBLE	ImageRotationAngle	Der Winkel beim rotierenden Bild.	7.14
INT32	TextRotationLeverLength	Länge des Hebelarms beim rotierenden Text.	7.15
DOUBLE	TextRotationAngle	Der Winkel beim rotierenden Text.	7.16
INT32	AlphaArrayCount	Die Anzahl von Alphawerten beim Zeichnen.	7.17
BYTE[]	AlphaArray	Die Alphawerte in der Auswahlleiste.	7.18
INT32	ColorArrayCount	Die Anzahl von Farben beim Zeichnen.	7.19
UINT32[]	ColorArray	Die Farben in der Auswahlleiste.	7.20
INT32	AlphaPaletteCount	Die Anzahl von Alphawerten beim Zeichnen.	7.21
MEMORY	AlphaPalette	Die Alphawerte in der Benutzerliste.	7.22
INT32	ColorPaletteCount	Die Anzahl von Farben beim Zeichnen.	7.23
MEMORY	ColorPalette	Die Farben in der Benutzerliste.	7.24
INT32	AlphaGradientCount	Die Anzahl von Alphaverläufen beim Zeichnen.	7.25
MEMORY	AlphaGradient	Die Alphaverläufe in der Benutzerliste.	7.26
INT32	ColorGradientCount	Die Anzahl von Farbverläufen beim Zeichnen.	7.27
MEMORY	ColorGradient	Die Farbverläufe in der Benutzerliste.	7.28
INT32	AlphaGradientIndex	Der ausgewählte Alphaverlauf.	7.29
INT32	ColorGradientIndex	Der ausgewählte Farbverlauf.	7.30
INT32	TextAlphaPaletteCount	Wird nicht verwendet.	7.31
INT32	TextColorPaletteCount	Die Anzahl von Farben in der Textbearbeitung.	7.32
MEMORY	TextColorPalette	Die Textfarben in der Benutzerliste.	7.33
INT32	TextAlphaGradientCount	Wird nicht verwendet.	7.34
INT32	TextColorGradientCount	Wird nicht verwendet.	7.35
INT32	FontCount	Die Anzahl von Schriften in der Textbearbeitung.	7.36
MEMORY	FontMemory	Die Schrifttypen in der Benutzerliste.	7.37
INT32	FontIndex	Der ausgewählte Schrifttyp.	7.38
INT32	GroupIndex	Die ausgewählte Gruppe in der Bildbearbeitung.	7.39
...		3.2 Datenformat, Tabelle 3.2	

Tabelle 3.4: Bildeinstellungsformat

3.5 Abspielendes Dateiformat

Typ	Name	Beschreibung	Info
INT32	ImageWidth	Die Breite des Bildes.	8.1
INT32	ImageHeight	Die Höhe des Bildes.	8.2
INT64	PlayerDuration	Die Abspielzeit der Datei.	8.3
INT32	FileImageSize	Die Größe des Bildspeichers in Bytes.	8.4
MEMORY	FileImageMemory	Der Speicher mit der Bilddatei.	8.5
INT32	FileMemorySize	Die Größe des Dateispeichers in Bytes.	8.6
BYTE[]	FileMemory	Der Speicher mit den Dateidaten.	8.7
...		3.2 Datenformat, Tabelle 3.2	

Tabelle 3.5: Abspielendes Dateiformat

3.6 Animationsformat

Typ	Name	Beschreibung	Info
INT32	ImageWidth	Die Breite des Bildes.	9.1
INT32	ImageHeight	Die Höhe des Bildes.	9.2
INT64	PlayerDuration	Die Abspielzeit der Datei.	9.3
UINT32	BackColor	Die Hintergrundfarbe im Darstellungsfenster.	9.4
INT32	FileImageSize	Die Größe des Bildspeichers in Bytes.	9.5
MEMORY	FileImageMemory	Der Speicher mit der Bilddatei.	9.6
INT32	FileMemorySize	Die Größe des Dateispeichers in Bytes.	9.7
BYTE[]	FileMemory	Der Speicher mit den Dateidaten.	9.8
...		3.2 Datenformat, Tabelle 3.2	

Tabelle 3.6: Animationsformat

3.7 Verschiedenes Dateiformat

Typ	Name	Beschreibung	Info
INT32	FileMemorySize	Die Größe des Dateispeichers in Bytes.	10.1
BYTE[]	FileMemory	Der Speicher mit den Dateidaten.	10.2
...		3.2 Datenformat, Tabelle 3.2	

Tabelle 3.7: Verschiedenes Dateiformat

3.8 Bildgruppenformat

Typ	Name	Beschreibung	Info
INT32	GroupCount	Die Anzahl der Bildgruppen.	11.1
-> {	Group		
INT32	GroupNameLength	Die Anzahl der Zeichen im Namen der Gruppe.	11.2
WCHAR[]	GroupName	Der Name der Gruppe.	11.3
INT32	GroupItemCount	Die Anzahl der Bildelemente in einer Gruppe.	11.4
MEMORY	GroupItems	Der Speicher mit dem Bildelement.	11.5
} <-	Group		

Tabelle 3.8: Bildgruppenformat

4 Informationen zum Dateiformat

4.1 Identifikationsnummer

Die Identifikationsnummer kennzeichnet das Dateiformat. Die Nummer kann auch mit 4 Buchstaben (TDPP: Two Dimension PicturePaint Project) dargestellt werden.

4.2 Die gesamte Dateigröße

Die Größe der gesamten Datei in Bytes. Der Wert wird zur Kontrolle angegeben und kann nicht ungleich der Dateigröße sein.

4.3 Versionsnummer

Die Versionsnummer ist für diese Beschreibung immer 1.

4.4 Die Größe des Vorschaubildes

Die Größe des Bildes für die Dateivorschau in Bytes. Ist der Wert 0 wird kein Vorschaubild verwendet. Der Wert kann nicht kleiner als 0 oder größer als die restliche Dateigröße sein. Als Bild wird eine Datei aus dem Projektarchiv verwendet (siehe Abschnitt 6.2: "ProjectPreview").

4.5 Speicher des Vorschaubildes

Der Speicher mit der Datei des Vorschaubildes. Wenn die Dateigröße (siehe Abschnitt 4.4) gleich 0 ist, wird kein Vorschaubild verwendet. Als Standard sollte ein Bild in dem Format "PNG" oder "JPEG" aus dem Projektarchiv gewählt werden. Der Dateispeicher für das Bild wird hier angegeben. Die Bilddatei enthält dann die Speichergröße 0 (siehe Abschnitt 6.7). Zusätzlich muss das Einstellungsbit "ProjectPreview" (siehe Abschnitt 6.2) gesetzt sein.

Hinweis: Das Programm "FileViewer" erkennt das Format "2D Picture" nicht als Vorschaubild.

5 Informationen zum Datenformat

5.1 Anzahl der Dateien

Der Wert gibt die Anzahl der Dateien im Archiv an. Er kann nicht kleiner als 0 sein. Ist der Wert 0 sind keine Dateien vorhanden. Der nächste gelesene Wert ist dann die Anzahl der Bildgruppen (siehe Abschnitt 11.1). Auch dieser Wert muss ebenfalls 0 sein, weil nur Bilder aus dem Projektarchiv in den Gruppen verwendet werden.

5.2 Dateityp

Der Dateityp kennzeichnet die Anwendung der Datei im Programm "PicturePaint". Nicht alle Dateien werden vom Programm erkannt. Nicht zugeordnete Dateien müssen den Wert 5 (Various) haben.

Name	Nummer	Beschreibung
Image	0	Die Datei ist ein darstellbares Bildformat.
Video	1	Die Datei ist ein abspielbares Video.
Sound	2	Die Datei ist eine abspielbare Musikdatei.
Gif	3	Die Datei ist eine abspielbare GIF Animation.
Animation	4	Die Animation (.tda) wird mit dem Programm "PicturPaint" erstellt.
Various	5	Die Datei wird vom Programm "PicturPaint" nicht zugeordnet.

Tabelle 5.2: Dateityp

5.3 Namenlänge

Die Anzahl von Zeichen im Namen der Datei. Der Wert kann nicht kleiner als 1 oder größer als 260 sein. Die maximale Namenlänge mit Dateierweiterung (siehe Abschnitt 5.5) wird durch das Dateisystem vorgegeben.

5.4 Name

Der Name besteht aus einer bestimmten Anzahl von Zeichen (Buchstaben). Der Dateiname darf nur einmal im Projektarchiv enthalten sein. Die Namenlänge (siehe Abschnitt 5.3) 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: FileNameLength * 2Bytes (16Bit)

5.5 Namenlänge der Dateierweiterung

Die Anzahl von Zeichen im Namen der Dateierweiterung. Der Wert kann nicht kleiner als 0 sein. Die Länge des Namens aus Abschnitt 5.3 und der Dateierweiterung können zusammen nicht größer als 260 sein. Die maximale Namenlänge wird durch das Dateisystem vorgegeben. Wenn keine Dateierweiterung angegeben wurde, muss der Wert 0 sein.

5.6 Name der Dateierweiterung

Der Name besteht aus einer bestimmten Anzahl von Zeichen (Buchstaben). Das erste Zeichen in der Erweiterung ist ein Punkt. Die Namenlänge (siehe Abschnitt 5.5) bestimmt die Anzahl der Zeichen. Ein Zeichen ist ein 16Bit Wert ohne Vorzeichen. Die Erweiterung ist nicht vorhanden, wenn die Länge 0 ist. Die Speichergröße für die Dateierweiterung ergibt sich aus der Länge mal 2Bytes.

Speichergröße: FileExtentionLength * 2Bytes (16Bit)

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

6 Informationen zum Bildformat

6.1 Bildformat

Die Nummer gibt das Bildformat für den folgenden Dateispeicher an (siehe Abschnitt 6.8). Der Dateityp (siehe Abschnitt 5.2) muss für eine Bilddatei den Wert 0 haben. Ist das Dateiformat kein Bild wird hier der Wert 0 (Unknown) angegeben.

Name	Nummer	Beschreibung
Unknown	0	Ein unbekanntes oder kein Bildformat.
bmp	1	Ein unkomprimiertes Bild (Windows Bitmap).
emf	2	Ein auf Vektordaten basiertes Bild (Windows Enhanced Metafile).
exif	3	Ein Dateiformat für Bilddaten (Exchangeable Image File Format).
gif	4	Ein verlustfreies komprimiertes Bild (Graphics Interchange Format).
icon	5	Ein unkomprimiertes kleines Bild (Windows Icon).
jpeg	6	Ein verlustbehaftetes komprimiertes Bild (Joint Photographic Experts G.).
png	7	Ein verlustfreies komprimiertes Bild (Portable Network Graphics).
tiff	8	Ein verlustfreies komprimiertes Bild (Tagged Image File Format).
wmf	9	Ein auf Vektordaten basiertes Bild (Windows Metafile).
tpd	10	Ein unkomprimiertes Bild mit Vorschau (2D Picture).

Tabelle 6.1: Bildformat

6.2 Einstellungsbits

Für das Programm "PicturePaint" sind verschiedene Einstellungsbits für Bilddateien möglich. Das Bit "PaintPreview" wird gesetzt, wenn in dem Format "2D Picture" ein Vorschaubild gespeichert werden soll. Das Bit "ProjectPreview" gibt an, das dieses Bild als Archivvorschau verwendet wird (siehe die Abschnitte 4.5 und 6.7).

Name	Bit	Beschreibung
PaintAlpha	0x0001	Mit Alphawerten zeichnen ist ausgewählt.
PaintOver	0x0002	Über das Darstellungsbild zeichnen ist ausgewählt.
PaintMultiColor	0x0004	Mit Farbverlauf zeichnen ist ausgewählt.
PaintEllipse	0x0008	Die Enden einer Linie werden als Ellipse gezeichnet.
PaintPreview	0x0010	Das Bildformat (.tdp) speichert ein Vorschaubild mit.
PaintMultiColorRotation	0x0020	Der rotierende Farbverlauf ist ausgewählt.
PaintRotationLever	0x0040	Der Hebelarm wird beim rotierenden Farbverlauf angezeigt.
ImageOver	0x0080	Das eingefügte Bild über das Darstellungsbild zeichnen.
ImageSize	0x0100	Das eingefügte Bild an die Fenstergröße anpassen.
ImageAspectRatio	0x0200	Das Seitenverhältnis beim eingefügten Bild beibehalten.
ImageRotationLever	0x0400	Der Hebelarm wird beim rotierende Bild angezeigt.
TextOver	0x0800	Den Text über das Darstellungsbild zeichnen.
TextRotationLever	0x1000	Der Hebelarm wird beim rotierende Text angezeigt.
ProjectPreview	0x8000	Die Datei wird als Projektvorschau verwendet.

Tabelle 6.2: Einstellungsbits

6.3 Position in X-Koordinaten

Die Position in X-Koordinaten sollte ursprünglich beim Zeichnen mit Bildausschnitten verwendet werden. Der Wert muss 0 sein, weil das Programm "PicturePaint" die Position nicht benutzt.

6.4 Position in Y-Koordinate

Die Position in Y-Koordinaten sollte ursprünglich beim Zeichnen mit Bildausschnitten verwendet werden. Der Wert muss 0 sein, weil das Programm "PicturePaint" die Position nicht benutzt.

6.5 Bildbreite

Die Breite der Darstellung in Bildpunkte (Pixel). Der Wert kann nicht kleiner als 1 sein.

6.6 Bildhöhe

Die Höhe der Darstellung in Bildpunkte (Pixel). Der Wert kann nicht kleiner als 1 sein.

6.7 Größe des Bildspeichers

Die Größe des Bildspeichers in Bytes. Der Wert kann nicht kleiner als 0 oder größer als die restliche Dateigröße sein. Ist der Wert gleich 0 muss das Einstellungsbit "ProjectPreview" gesetzt sein (siehe Tabelle 6.2). Somit ist die Projektvorschau (siehe Abschnitt 4.6) der Bildspeicher.

6.8 Bildspeicher

Der Speicher enthält die Dateidaten. Die Größe in Bytes wird im Abschnitt 6.7 angegeben. Wird das Bild aus der Projektvorschau verwendet ist der Speicher nicht vorhanden. Die Größe und die Bilddatei sind dann in den Abschnitten 4.4 und 4.5 definiert.

7 Informationen zum Bildeinstellungsformat

Die Bildeinstellungen werden für das Programm "PicturePaint" angegeben. In der weiteren Beschreibung wird Bezug auf das Programm genommen. Am Ende der Information ist der Wertebereiche { ... } und der Standardwert { ,...} definiert.

7.1 Werkzeug zum Zeichnen

Der Wert gibt das verwendete Grafikobjekt beim Zeichnen an (Rechteck, Kreis, usw). { 0 ... 11, 2 }

7.2 Alphawert zum Zeichnen

Der aktuelle Alphawert, der zum Zeichnen verwendet wird. { 0 ... 255, 255 }

7.3 Farbe zum Zeichnen

Die aktuelle Farbe, die zum Zeichnen verwendet wird. { 0x00000000 ... 0xFFFFFFFF, 0xFFFF0000 }

7.4 Hintergrundfarbe im Darstellungsfenster

Unter Bildeinstellungen kann die Hintergrundfarbe im Darstellungsfenster geändert werden. { 0xFF000000 ... 0xFFFFFFFF, 0xFFFFFFFF }

7.5 Farbe beim Erstellen oder Erweitern von Bildern

Unter Bildeinstellungen kann die Farbe beim Erstellen oder Erweitern von Bildern geändert werden. { 0x00000000 ... 0xFFFFFFFF, 0xFFFFFFFF }

7.6 Transparente Farbe beim Zeichnen mit Bildern

In der Bildbearbeitung kann unter Einstellungen die transparente Farbe angegeben werden. { 0x00000000 ... 0xFFFFFFFF, 0x00000000 }

7.7 Schriftfarbe beim Zeichnen von Textinhalten

Die aktuelle Schriftfarbe, die zum Zeichnen in der Textbearbeitung verwendet wird. { 0x00000000 ... 0xFFFFFFFF, 0xFF000000 }

7.8 Liniendicke beim Zeichnen

Die Dicke der Linie wird unter Einstellungen beim Zeichnen angegeben. { 0 ... 999, 1 }

7.9 Breite der Rundung beim abgerundeten Rechtecks

In den Einstellungen beim Zeichnen wird die Breite der Rundung bestimmt. { 1 ... 1000000, 12 }

7.10 Höhe der Rundung beim abgerundeten Rechtecks

In den Einstellungen beim Zeichnen wird die Höhe der Rundung bestimmt. { 1 ... 1000000, 14 }

7.11 Länge des Hebelarms beim rotierenden Farbverlauf

In den Einstellungen beim Zeichnen wird die Länge des Hebelarms angegeben. { 5 ... 2000, 100 }

7.12 Winkel beim rotierenden Farbverlauf

In den Einstellungen beim Zeichnen wird der Winkel eingegeben. { 0.0 ... 360.0, 0.0 }

7.13 Länge des Hebelarms beim rotierenden Bild

In der Bildbearbeitung unter Einstellungen wird die Länge des Hebelarms bestimmt. { 5 ... 2000, 100 }

7.14 Winkel beim rotierenden Bild

In der Bildbearbeitung unter Einstellungen wird der Winkel eingegeben. { 0.0 ... 360.0, 0.0 }

7.15 Länge des Hebelarms beim rotierenden Text

In der Textbearbeitung unter Einstellungen wird die Länge des Hebelarms bestimmt. { 5 ... 2000, 100 }

7.16 Winkel beim rotierenden Text

In der Textbearbeitung unter Einstellungen wird der Winkel eingegeben. { 0.0 ... 360.0, 0.0 }

7.17 Anzahl der Alphawerte im Darstellungsfenster

Die Anzahl von benutzerdefinierten Alphawerte. { 0 ... 10000, 0 }

7.18 Alphawerte im Darstellungsfenster

Alle Alphawerte, die im Darstellungsfenster beim Zeichnen angezeigt werden. { 0x00 ... 0xFF }

7.19 Anzahl der Farben im Darstellungsfenster

Die Anzahl von benutzerdefinierten Farben. { 0 ... 10000, 0 }

7.20 Farben im Darstellungsfenster

Alle Farben, die im Darstellungsfenster beim Zeichnen angezeigt werden.
{ 0x00000000 ... 0xFFFFFFFF }

7.21 Anzahl der Alphawerte in der Benutzerliste

Die Anzahl von benutzerdefinierten Alphawerte. { 0 ... 10000, 0 }

7.22 Alphawerte in der Benutzerliste

Alle Alphawerte, die beim Zeichnen im Alphamodus in der Liste mit den Benutzerfarben angezeigt werden. Der Aufbau des Speichers wird in Tabelle 7.22 beschrieben. Die Anzahl der Elemente wird in Abschnitt 7.21 angegeben.

Typ	Name	Beschreibung
BYTE	UserSelect	Bei dem Wert 1 ist der Alphawert markiert (siehe 7.17 und 7.18).
BYTE	Alpha	Der Alphawert von 0x00 bis 0xFF.

Tabelle 7.22: Alphaelement

7.23 Anzahl der Farben in der Benutzerliste

Die Anzahl von benutzerdefinierten Farben. { 0 ... 10000, 0 }

7.24 Farben in der Benutzerliste

Alle Farben, die in der Liste mit den Benutzerfarben beim Zeichnen enthalten sind. Der Aufbau des Speichers wird in Tabelle 7.24 beschrieben. Die Anzahl wird in Abschnitt 7.23 angegeben.

Typ	Name	Beschreibung
BYTE	UserSelect	Bei dem Wert 1 ist die Farbe markiert (siehe 7.19 und 7.20).
UINT32	Color	Der Farbwert von 0x00000000 bis 0xFFFFFFFF.

Tabelle 7.24: Farbelement

7.25 Anzahl der Alphaverläufe in der Benutzerliste

Die Anzahl von benutzerdefinierten Alphaverläufe. { 0 ... 10000, 0 }

7.26 Alphaverläufe in der Benutzerliste

Alle Alphaverläufe, die beim Zeichnen im Alphamodus unter Benutzerfarben angezeigt werden. Der Aufbau des Speichers wird in der Tabelle 7.26.1 beschrieben. Die Anzahl der Speicherelemente ist im Abschnitt 7.25 angegeben. In Abhängigkeit vom Typ sind in einem Element verschiedene Werte enthalten. Die verschiedenen Darstellungstypen sind in der Tabelle 7.26.2 angegeben.

Typ	Name	Beschreibung
BYTE	Type	Siehe Tabelle 7.26.2 und Abschnitt 12: Farbverläufe
BYTE	Begin	Der Alphawert am Anfang. (Type: 0,1,2,3,4)
BYTE	Between	Der Alphawert zwischen Anfang und Mitte. (Type: 4)
BYTE	Middle	Der Alphawert zwischen Anfang und Ende. (Type: 3,4)
BYTE	End	Der Alphawert am Ende. (Type: 0,1,2,3,4)
INT32	Factor	Der Faktor bestimmt die Steigung. (Type: 1,2,3,4) { 0 ... 100, 100 }
INT32	FactorBetween	Der Faktor bestimmt die Position zwischen Anfang und Mitte. (Type: 4)
INT32	Length	Der Anzahl der Alphawerte. (Type: 0,1,2,3,4)

Tabelle 7.26.1: Element zum Alphaverlauf

Wert	Name	Beschreibung
0	Linear	Ein linearer Verlauf mit Anfang- und Endwert.
1	Cosine	Ein halbrunder Verlauf mit Faktor, Anfang- und Endwert als Kosinuskurve.
2	Sine	Ein halbrunder Verlauf mit Faktor, Anfang- und Endwert als Sinuskurve.
3	Round	Ein runder Verlauf mit Faktor, Anfang-, Mitte- und Endwert.
4	RoundExtra	Ein runder Verlauf mit Highlight als Sinus- und Kosinuskurve.

Tabelle 7.26.2: Darstellungstypen

7.27 Anzahl der Farbverläufe in der Benutzerliste

Die Anzahl von benutzerdefinierten Farbverläufe. { 0 ... 10000, 0 }

7.28 Farbverläufe in der Benutzerliste

Alle Farbverläufe, die beim Zeichnen unter Benutzerfarben angezeigt werden. Ein Speicherelement wird in der Tabelle 7.28 beschrieben. Die Anzahl der Element ist im Abschnitt 7.27 zu finden. Ein Element kann verschiedene Werte enthalten. Alle Typen sind in der Tabelle 7.26.2 angegeben.

Typ	Name	Beschreibung
BYTE	Type	Siehe Tabelle 7.26.2 und Abschnitt 12: Farbverläufe
UINT32	Begin	Die Farbe am Anfang. (Type: 0,1,2,3,4)
UINT32	Between	Die Farbe zwischen Anfang und Mitte. (Type: 4)
UINT32	Middle	Die Farbe zwischen Anfang und Ende. (Type: 3,4)
UINT32	End	Die Farbe am Ende. (Type: 0,1,2,3,4)
INT32	Factor	Der Faktor bestimmt die Steigung. (Type: 1,2,3,4) { 0 ... 100, 100 }
INT32	FactorBetween	Der Faktor bestimmt die Position zwischen Anfang und Mitte. (Type: 4)
INT32	Length	Der Anzahl der Farben. (Type: 0,1,2,3,4)

Tabelle 7.28: Element zum Farbverlauf

7.29 Ausgewählter Alphaverlauf

Der Alphaverlauf wird beim Zeichnen im Darstellungsfenster angezeigt. Wenn kein Verlauf ausgewählt wurde, ist der Wert -1. {-1 ... 10000, -1}

7.30 Ausgewählter Farbverlauf

Der Farbverlauf wird beim Zeichnen im Darstellungsfenster angezeigt. Wenn kein Verlauf ausgewählt wurde, ist der Wert -1. {-1 ... 10000, -1}

7.31 Anzahl der Alphawerte in der Benutzerliste der Textverarbeitung

Der Wert wird nicht verwendet und muss 0 sein.

7.32 Anzahl der Farben in der Benutzerliste der Textverarbeitung

Die Anzahl von benutzerdefinierten Farben. { 0 ... 10000, 0 }

7.33 Farben in der Benutzerliste der Textverarbeitung

Alle Schriftfarben, die unter Benutzerfarben in der Textverarbeitung enthalten sind. Der Aufbau des Speichers wird in Tabelle 7.33 beschrieben. Die Anzahl wird in Abschnitt 7.32 angegeben.

Typ	Name	Beschreibung
BYTE	UserSelect	Der Wert wird nicht verwendet und muss 0 sein.
UINT32	Color	Der Farbwert von 0x00000000 bis 0xFF000000.

Tabelle 7.33: Farbelement

7.34 Anzahl der Alphaverläufe in der Textverarbeitung

Der Wert wird nicht verwendet und muss 0 sein.

7.35 Anzahl der Farbverläufe der Textverarbeitung

Der Wert wird nicht verwendet und muss 0 sein.

7.36 Anzahl der Schrifttypen der Textverarbeitung

Die Anzahl von benutzerdefinierten Schrifttypen in der Textverarbeitung. { 0 ... 10000, 0 }

7.37 Schrifttypen in der Benutzerliste der Textverarbeitung

Alle Schrifttypen, die unter Benutzerschriften in der Textverarbeitung enthalten sind. Der Aufbau des Speichers wird in Tabelle 7.37.1 beschrieben. Die Anzahl der Speicherelemente wird in Abschnitt 7.36 angegeben. Die Textzeile "InfoString" im Element gibt zuerst den Namen der Schrift, dann die Größe in Bildpunkte (Pixel) und zum Schluss eine Bitkombination der Schriftstile an. Die Schriftstile sind in der Tabelle 7.37.2 beschrieben. Die einzelnen Werte sind mit einem Trennzeichen unterteilt.

Beispiel: "Arial|30|1"

Typ	Name	Beschreibung
INT32	InfoLength	Die Anzahl der Zeichen in der Textzeile.
WCHAR[]	InfoString	Die Textzeile mit Installationsinformationen.

Tabelle 7.37.1: Schriftelement

Wert	Name	Beschreibung
0x00	Regular	Die Text wird normal angezeigt.
0x01	Bold	Die Text wird fett angezeigt.
0x02	Italic	Die Text wird kursiv angezeigt.
0x04	Underline	Die Text wird unterstrichen angezeigt.
0x08	Strikeout	Die Text wird durchgestrichen angezeigt.

Tabelle 7.37.2: Schriftstile

7.38 Ausgewählter Schrifttyp

Der Schrifttyp wird in der Textverarbeitung unter Schrift angezeigt. Wenn kein Schrifttyp ausgewählt wurde, ist der Wert -1. {-1 ... 10000, -1}

7.39 Ausgewählte Bildgruppe

Die Bildgruppen werden im Abschnitt 11 beschrieben. Die ausgewählte Gruppe wird in der Bildbearbeitung im Darstellungsfenster angezeigt, wenn der Wert größer als -1 ist. {-1 ... 10000, -1}

8 Informationen zum abspielenden Dateiformat

8.1 Bildbreite

Die Breite der Darstellung in Bildpunkte (Pixel). Der Wert kann nicht kleiner als 0 sein. Der Dateityp 2 (Sound) siehe Abschnitt 5.2 kann auch ein Bild enthalten.

8.2 Bildhöhe

Die Höhe der Darstellung in Bildpunkte (Pixel). Der Wert kann nicht kleiner als 0 sein. Der Dateityp 2 (Sound) siehe Abschnitt 5.2 kann auch ein Bild enthalten.

8.3 Abspielzeit

Die Abspielzeit wird in 100ns angegeben. Der Wert kann nicht kleiner als 1 sein.

8.4 Größe des Bildspeichers

Die Größe des Vorschaubildes in Bytes. Der Wert kann nicht kleiner als 0 sein. Wenn kein Bild vorhanden ist, wird der Wert 0 angegeben.

8.5 Bildspeicher

Der Speicher enthält ein Vorschaubild in "PNG" oder "JPEG" Format. Die Größe in Bytes wird im Abschnitt 8.4 angegeben. Beim Video oder der Animation wird ein Bild in der Mitte der Abspielzeit erstellt.

8.6 Größe des Dateispeichers

Die Größe des Dateispeichers in Bytes. Der Wert kann nicht kleiner als 1 oder größer als die restliche Dateigröße des Archivs sein.

8.7 Dateispeicher

Der Speicher enthält die Dateidaten. Die Größe in Bytes wird im Abschnitt 8.6 angegeben.

9 Informationen zum Animationsformat

9.1 Bildbreite

Die Breite der Darstellung in Bildpunkte (Pixel). Der Wert kann nicht kleiner als 1 sein.

9.2 Bildhöhe

Die Höhe der Darstellung in Bildpunkte (Pixel). Der Wert kann nicht kleiner als 1 sein.

9.3 Abspielzeit

Die Abspielzeit wird in 100ns angegeben. Der Wert kann nicht kleiner als 1 sein.

9.4 Hintergrundfarbe

Die Hintergrundfarbe wird im Programm "PicturePaint" unter Einstellungen angegeben. Die Farbe wird bei der Animationsbearbeitung als Hintergrund in Darstellungsfenster angezeigt.

9.5 Größe des Bildspeichers

Die Größe des Vorschaubildes in Bytes. Der Wert kann nicht kleiner als 0 sein. Wenn kein Bild vorhanden ist, wird der Wert 0 angegeben.

9.6 Bildspeicher

Der Speicher enthält ein Vorschaubild in "PNG" oder "JPEG" Format. Die Größe in Bytes wird im Abschnitt 9.5 angegeben.

9.7 Größe des Dateispeichers

Die Größe des Dateispeichers in Bytes. Der Wert kann nicht kleiner als 1 oder größer als die restliche Dateigröße des Archivs sein.

9.8 Dateispeicher

Der Speicher enthält die Dateidaten. Die Größe in Bytes wird im Abschnitt 9.7 angegeben.

10 Informationen zum verschiedenen Dateiformat

10.1 Größe des Dateispeichers

Die Größe des Dateispeichers in Bytes. Der Wert kann nicht kleiner als 1 oder größer als die restliche Dateigröße des Archivs sein.

10.2 Dateispeicher

Der Speicher enthält die Dateidaten. Die Größe in Bytes wird im Abschnitt 10.1 angegeben.

11 Informationen zum Bildgruppenformat

11.1 Anzahl der Gruppen

Der Wert gibt die Anzahl der Bildgruppen an. Die Anzahl kann nicht kleiner als 0 sein. Wenn der Wert 0 ist, werden keine Bildgruppen verwendet.

11.2 Namenlänge

Die Anzahl von Zeichen im Namen der Bildgruppe. Der Wert kann nicht kleiner als 1 sein.

11.3 Name der Bildgruppe

Der Name besteht aus einer bestimmten Anzahl von Zeichen (Buchstaben). Die Namenlänge (siehe Abschnitt 11.2) 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: `GroupNameLength * 2Bytes (16Bit)`

11.4 Anzahl der Bildelemente

Der Wert gibt die Anzahl der Elemente in einer Bildgruppe an. Die Anzahl kann nicht kleiner als 0 sein. Wenn der Wert 0 ist, sind keine Elemente vorhanden.

11.5 Speicher eines Bildelements

Der Speicher für eine Bildelement besteht immer aus fünf Werten. Die ersten vier Werte geben die Position und Größe des Bildausschnittes an. Der letzte Wert ist ein nullbasierter Index, der auf eine Bilddatei im Projektarchiv verweist. Alle Bilddateien werden beim Laden durchgezählt und nummeriert. Die Bildgruppen werden im Programm "PicturePaint" in der Bildbearbeitung verwendet.

Typ	Name	Beschreibung	Info
INT32	ImagePosX	Die Position in X-Koordinaten im Bild.	11.5.1
INT32	ImagePosY	Die Position in Y-Koordinaten im Bild.	11.5.2
INT32	ImageWidth	Die Breite des Bildausschnittes.	11.5.3
INT32	ImageHeight	Die Höhe des Bildausschnittes.	11.5.4
INT32	ImageIndex	Der nullbasierte Index für die Zuordnung der Bilddatei.	11.5.5

Tabelle 11.5: Bildelement

11.5.1 Die X-Position des Bildelements

Die Position wird in Bildpunkte angegeben. Der Wert darf nicht kleiner als 0 sein. Die X-Position und die Breite des Bildausschnittes (siehe 11.5.3) dürfen die Breite des Originalbildes nicht überschreiten.

11.5.2 Die Y-Position des Bildelements

Die Position wird in Bildpunkte angegeben. Der Wert darf nicht kleiner als 0 sein. Die Y-Position und die Höhe des Bildausschnittes (siehe 11.5.4) dürfen die Höhe des Originalbildes nicht überschreiten.

11.5.3 Die Breite des Bildelements

Die Breite wird in Bildpunkte angegeben. Der Wert kann nicht kleiner als 1 sein. Die X-Position (siehe 11.5.1) und die Breite des Bildausschnittes dürfen die Breite des Originalbildes nicht überschreiten.

11.5.4 Die Höhe des Bildelements

Die Höhe wird in Bildpunkte angegeben. Der Wert kann nicht kleiner als 1 sein. Die Y-Position (siehe 11.5.2) und die Höhe des Bildausschnittes dürfen die Höhe des Originalbildes nicht überschreiten.

11.5.5 Die Zuordnungsnummer für die Bilddatei

Die Zuordnungsnummer verweist auf eine Bilddatei im Projektarchiv. Die Nummer kann nicht kleiner als 0 oder größer gleich die Anzahl der Bilddateien im Archiv. Die Bilddateien werden beim Laden durchgezählt und nummeriert.

12 Methoden zum Erstellen von Farbverläufen

12.1 Linearverlauf

Die Methode erstellt einen linearen Farbverlauf aus zwei angegebene Farben.

```
public UInt32[] CreateLinearColor(Int32 Count, Color Begin, Color End) {
    if(Count < 1) return new UInt32[0];

    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++) {
        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 Sinusverlauf

Die erste Methode erstellt einen Farbverlauf mit Sinuskurve aus zwei Farben. Bei der zweiten Methode wird eine zusätzliche Farbe als Highlight verwendet. Für den Faktor siehe Abschnitt 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++) {
        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;
    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 Kosinusverlauf

Die Methode erstellt einen Farbverlauf mit Kosinuskurve aus zwei Farben. Für den Faktor siehe 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++) {
        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 Runder Verlauf

Der runde Verlauf besteht aus einer Sinus- und Kosinuskurve. Bei der zweiten Methode wird eine zusätzliche Farbe als Highlight verwendet. Die beiden Faktoren aus dem Dateiformat (siehe Abschnitte 7.26 und 7.28) werden durch 100 geteilt. Der Faktor (INT32) kann eine Zahl zwischen 0 und 100 sein. Der Faktor (Double) zwischen 0.0 und 1.0.

```
public UInt32[] CreateRoundColor(Int32 Count, Double Factor, Color Begin,
                                 Color Middle, Color End) {
    if(Count < 1) return new UInt32[0];

    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;
}
```

```

public UInt32[] CreateRoundColor(Int32 Count, Double Factor, Double FactorBetween,
                                Color Begin, Color Between, Color Middle, Color End) {
    if(Count < 1) return new UInt32[0];

    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;
    }

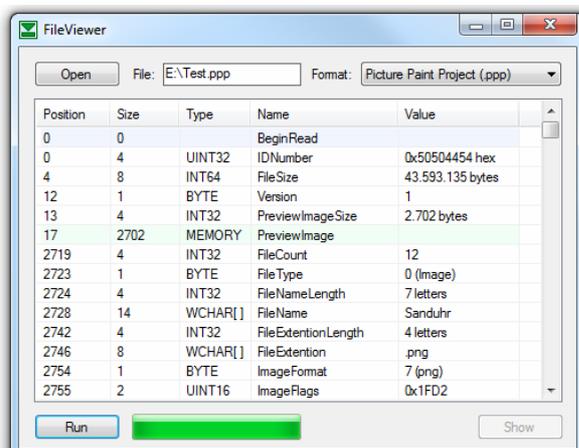
    Int32 SizeA = Count / 2;
    Int32 SizeB = Count - SizeA;

    this.CreateSineColor(ColorArray, 0, SizeA, Factor, FactorBetween, Begin,
                        Between, Middle);
    this.CreateCosineColor(ColorArray, SizeA, SizeB, Factor, Middle, End);
    return ColorArray;
}

```

13 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: FileViewerTDPicturePaintProject