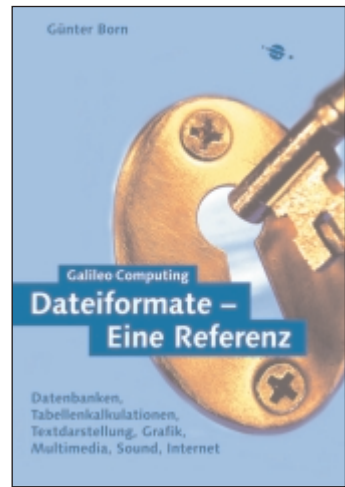


Dieses Kapitel stammt aus dem Buch  
›Dateiformate – Eine Referenz‹  
von Günter Born.

[www.borncity.de](http://www.borncity.de)

ISBN 3-934358-83-7  
119,90 DM



Informationen zum Buch  
mit Bestellmöglichkeit

[www.galileocomputing.de](http://www.galileocomputing.de)

**Galileo Computing**

© Copyright 2001 by Galileo Press

Verlag und Autor schließen jede Haftung beim Gebrauch dieser Informationen aus.

# 40 Das Windows Metafile-Format (WMF)

Bitmap-Dateien können nur Punktgrafiken abbilden. Windows ermöglicht zusätzlich die Speicherung von Vektorgrafiken, wenn auch keine Standardwerkzeuge mitgeliefert werden, die dies unterstützen. Die Vektorgrafiken werden dabei als Metadateien abgespeichert. Eine Windows Metafile-Datei besteht aus einer Reihe von Anweisungen für das Graphics Device Interface (GDI). Damit richten sich die Parameter der Metaanweisungen nach den Aufrufkonventionen der einzelnen GDI-Funktionen. Die Metaanweisungen dürfen dabei im Speicher oder in einer Datei stehen. Nachfolgende Beschreibung bezieht sich auf das Dateiformat (im Speicher liegt ein analoges Format vor). Die Metadatei besteht aus einem Header und einem Datenteil mit den eigentlichen Metarecords.

## Der Metafile-Header

Der Header eines Metafiles besitzt folgende Struktur (Tabelle 40.1):

Offset	Bytes	Name	Bedeutung
00H	2	mtType	Metafile-Typ
02H	2	mtHeader	Länge des Headers in Worten
04H	2	mtVersion	Windows-Version
06H	4	mtSize	Dateilänge in Worten
0AH	2	mtNoObj	Maximale Zahl der Elemente
0CH	4	mtMaxRec	Maximale Recordlänge
10H	2	mtnoPar	unbenutzt

Tabelle 40.1 Aufbau des WMF-Headers

Zu beachten ist, daß alle Längenangaben in Metafiles in Worten (2 Byte) erfolgen. Das erste Wort enthält den Typ des Metafiles:

- 1 = Metafile wird im Hauptspeicher gehalten.
- 2 = Metafile wird in Datei gespeichert.

Die Struktur der Metarecords ist für beide Varianten gleich.

Im folgenden Wort *mtHeaderSize* steht die Größe des Metafile-Headers in Worten. Ab Offset 04H findet sich ein Wort, das die Windows-Version aufnimmt, unter der die Metadatei erzeugt wurde. Diese Version wird als BCD-Zahl abgebildet (z. B. 300H entspricht 3.0).

Ab Offset 06H folgt ein Doppelwort mit der Länge der Metadatei in Worten. Daran schließt sich die maximale Zahl der Objekte an, die gleichzeitig in der Metadatei existieren können. Dies ist vor allem für Metafiles wichtig, die im Speicher gehalten werden.

Das Feld *mtMaxRecord* (Offset 0CH) definiert die Länge (in Worten) des größten Metafile-Records der Datei. Das letzte Wort im Header ist unbelegt.

### Placeable metafiles

Neben den Standard-Metafiles wurden noch die sogenannten *placeable metafiles* (Aldus) definiert. Hierbei handelt es sich um Standard-Metafiles, denen ein 22 Byte langer Header vorangestellt wurde. Dieser Header enthält zusätzliche Informationen über die Originalgröße der Metadatei und die Seitenverhältnisse (Aspect Ratio) des Bildes. Der Header besitzt dabei folgende Struktur:

Offset	Bytes	Bemerkungen
00H	4	Binary-Schlüssel Dateityp (9AC6CDD7H)
04H	2	Handle (auf 0 gesetzt)
06H	8	Bounding-Box in Units 2 Byte linker Rand 2 Byte oberer Rand 2 Byte rechter Rand 2 Byte unterer Rand
0EH	2	Metafile Units per Zoll (Wert sollte kleiner 1440 sein)
10H	4	reserviert (muß 0 sein)
14H	2	Checksumme

Tabelle 40.2 Placeable Metafile-Header

Ab Offset 06H findet sich die Angabe über die *Bounding-Box* des Bildes. Diese wird als *Rect*-Struktur (4 Integerwerte) definiert und gibt die Koordinaten der linken oberen und rechten unteren Ecke des Rechtecks an. Der Wert muß im Bereich -32767 und +32767 liegen. Die Maßeinheit wird dabei im Wort ab Offset 0EH definiert und bezieht sich auf *n* Einheiten pro Zoll. Hierbei soll der Wert unter 1440 bleiben. Die Checksumme wird aus der XOR-Verknüpfung der ersten 10 Worte des Headers gebildet. Um eine solche Metadatei zu lesen, muß eine Windows-Anwendung den 22-Byte-Header entfernen und eine Standard-Metafile speichern.

### Die Metafile-Records

An den Header schließt sich der Bereich mit den Metafile-Records an. Diese Records besitzen folgenden Aufbau:

Bytes	Bedeutung
4	Größe des Metafile-Records (in Word-Einträgen)
2	Typ des Metafile-Records
n*2	Feld mit n (Word-Einträgen) Parametern

Tabelle 40.3 Aufbau des Metafile-Records

Die einzelnen Recordtypen eines Metafiles sind für Windows 3.0 der zu entnehmen.

Viele dieser Strukturen sind sehr speziell auf Windows 3.x ausgerichtet. Daher möchte ich an dieser Stelle auf die Beschreibung der Parameter der einzelnen GDI-Funktionen bzw. Metarecords verzichten. Sofern diese für Sie von Interesse ist, sind die Parameter der GDI-Aufrufe in den meisten Programmierunterlagen für Windows (z.B. Win32 SDK, [msdn.microsoft.com](http://msdn.microsoft.com)) beschrieben.

Code	Recordname
0201H	SETBKCOLOR
0102H	SETBKMODE
0103H	SEMAPMODE
0104H	SETROP2
0105H	SETRELABS
0106H	SETPOLYFILLMODE
0107H	SETSTRETCHBLTMODE
0108H	SETTEXTCHAREXTRA
0209H	SETTEXTCOLOR
020AH	SETTEXTJUSTIFICATION
020BH	SETWINDOWORG
020CH	SETWINDOWEXT
020DH	SETVIEWPORTORG
020EH	SETVIEWPORTEXT
020FH	OFFSETWINDOWORG
0400H	SCALEWINDOWEXT
0211H	OFFSETVIEWPORTORG
0412H	SCALEVIEWPORTEXT
0213H	LINETO
0214H	MOVETO
0415H	EXCLUDECLIPRECT
0416H	INTERSECTCLIPRECT
0817H	ARC
0418H	ELLIPSE
0419H	FLOODFILL
081AH	PIE
041BH	RECTANGLE
061CH	ROUNDRECT
061DH	PATBLT
001EH	SAVEDC

Code	Recordname
041FH	SETPIXEL
0220H	OFFSETCLIPRGN
0521H	TEXTOUT
0922H	BITBLT
0B23H	STRETCHBLT
0324H	POLYGON
0325H	POLYLINE
0626H	ESCAPE
0127H	RESTOREDC
0228H	FILLREGION
0429H	FRAMEREGION
012AH	INVERTREGION
012BH	PAINTREGION
012CH	SELECTCLIPREGION
012DH	SELECTOBJECT
012EH	SETTEXTALIGN
062FH	DRAWTEXT
0830H	CHORD
0231H	SETMAPPERFLAGS
0A32H	EXTTEXTOUT
0D33H	SETDIBTODEV
0234H	SELECTPALETTE
0035H	REALIZEPALETTE
0436H	ANIMATEPALETTE
0037H	SETPALENTRIES
0538H	POLYPOLYGON
0139H	RESIZEPALETTE
0940H	DIBBITBLT
0B41H	DIBSTRETCHBLT
0142H	DIBCREATEPATTERNBRUSH
0F43H	STRETCHDIB
01F0H	DELETEOBJECT
00F7H	CREATEPALETTE
00F8H	CREATEBRUSH
01F9H	CREATEPATTERNBRUSH
02FAH	CREATEPENINDIRECT

Code	Recordname
02FBH	CREATEFONTINDIRECT
02FCH	CREATEBRUSHINDIRECT
02FDH	CREATEBITMAPINDIRECT
06FEH	CREATEBITMAP
06FFH	CREATEREGION

**Tabelle 40.4** Recordtypen einer WMF-Datei

Nachfolgend finden Sie Auszüge mit den Strukturen der jeweiligen Metafile-Records.

## AnimatePalette

Hierbei handelt es sich um einen speziellen Metafile-Record.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (variabel)
04H	2	Recordtyp (0436H)
06H	2 2 n*4	Parameter erster zu animierender Eintrag Zahl (n) der Animationseinträge Palette Entry Array mit: 1 Byte rot 1 Byte grün 1 Byte blau 1 Byte Flag .....

**Tabelle 40.5** Der AnimatePalette-Record

Der Record definiert die Einträge der logischen Palette.

## Arc

Dieser Record definiert einen Kreis (arc) in einer Metadatei.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (08H)
04H	2	Recordtyp (0817H)
06H	2 2 2 2 2 2 2 2	Parameter X1: X-Koord. oben links Y1: Y-Koord. oben links X2: X-Koord. unten rechts Y2: Y-Koord. unten rechts X3: log. X-Koord. Kreis Startpunkt Y3: log. Y-Koord. Kreis Startpunkt X4: log. X-Koord. Kreis Endpunkt Y4: log. Y-Koord. Kreis Endpunkt

**Tabelle 40.6** Der Arc-Record

Alle Parameter sind als Integer definiert, und die absoluten Werte von X1-X2 und Y2-Y1 dürfen 32767 Units nicht überschreiten.

**BitBlt**

Dieser Record existiert in zwei Versionen. Records, die von Windows-Versionen vor 3.0 erzeugt werden, enthalten einen Device-Dependant Bitmap. Ab Windows 3.0 enthält der Record eine Device-Independent-Bitmap (DIB). Der BitBlt-Record aus Windows-Versionen vor 3.0 besitzt folgende Struktur:

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (variabel)
04H	2	Recordtyp (0922H)
06H		Parameter
	2	High Order Word Raster Operation
	2	Y-Koordinate Source Origin
	2	X-Koordinate Source Origin
	2	Destination Y-Extend
	2	Destination X-Extend
	2	Y-Koordinate Destination Origin
	2	X-Koordinate Destination Origin
	2	Bitmap Width in Pixel
	2	Bitmap Height in Pixel
	2	Bytes per Raster-Line
	2	Color Planes Bitmap
	2	Adjacent Color Bits
	n	Device Dependent Bitmap Bits

Tabelle 40.7 Der BitBlt-Record (Windows 2.x)

In Windows 3.0 und höher wird folgende Struktur für den Record verwendet:

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (variabel)
04H	2	Recordtyp (0940H)
06H		Parameter
	2	High Order Word Raster Operation
	2	Y-Koordinate Source Origin
	2	X-Koordinate Source Origin
	2	Destination y-Extend
	2	Destination x-Extend
	2	Y-Koordinate Destination Origin
	2	X-Koordinate Destination Origin
	n	BITMAPINFO-Struktur
	n	Device Independent Bitmap Bits

Tabelle 40.8 Der BitBlt-Record (Windows 3.x)

Die BITMAP\_INFO-Struktur ist in Kapitel 17 über das Windows BMP-Format (Tabelle 17.2) beschrieben.

## CHORD

Diese Funktion zeichnet einen Kreisbogen (chord). Die Parameter  $X1$ ,  $Y1$  und  $X2$ ,  $Y2$  definieren die obere linke und untere rechte Ecke der Bounding-Box einer Ellipse.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (0BH)
04H	2	Recordtyp (0830H)
06H	2	Parameter
	2	X1-Koord. oben links Box
	2	Y1-Koord. oben links Box
	2	X2-Koord. unten rechts Box
	2	Y2-Koord. unten rechts Box
	2	X3-Koord. Ende 1. Liniensegment
	2	Y3-Koord. Ende 1. Liniensegment
	2	X4-Koord. Ende 2. Liniensegment
	2	Y4-Koord. Ende 2. Liniensegment

**Tabelle 40.9** Der CHORD-Record

Die Punkte  $x3$ ,  $y3$  und  $x4$ ,  $y4$  definieren die Endpunkte des Liniensegments, das die Ellipsenlinie schneidet. Der Kreisabschnitt (chord) wird mit dem aktuellen Stift (pen) und dem aktuellen Füllmuster (brush) gezeichnet.

## CREATEBITMAP

Dieser Record erzeugt eine Device-Dependant-Bitmap.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (variabel)
04H	2	Recordtyp (06FEH)
06H	2	Parameter
	2	Bitmap Breite in Pixel
	2	Bitmap Höhe in Pixel
	1	Color Planes
	1	Bits per Pixel
	n*2	Array mit Bitmap Init-String

**Tabelle 40.10** Der CREATEBITMAP-Record

## CREATEBITMAPINDIRECT

Dieser Record erzeugt eine Device-Dependant Bitmap.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (variabel)
04H	2	Recordtyp (02FDH)



Offset	Bytes	Bemerkungen
06H		Parameter
	2	Bitmap-Typ (0)
	2	Bitmap Breite in Pixel
	2	Bitmap Höhe in Pixel
	2	Bytes per Raster-Linie
	1	Color Planes
	1	Bits per Pixel
	n*2	Array mit Bitmap

**Tabelle 40.11** Der CREATEBITMAPINDIRECT-Record

## CREATEBRUSH

Die Struktur dieses Records (F8H) ist unbekannt.

## CREATEBRUSHINDIRECT

Dieser Record definiert ein logisches Zeichenwerkzeug (brush) mit einer Form (style), einer Farbe (color) und einem Muster (pattern).

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (07H)
04H	2	Recordtyp (02FCH)
06H		Parameter
	2	Style
	4	Farbwert
	2	Hatch

**Tabelle 40.12** Der CREATEBRUSHINDIRECT-Record

Der Style ist dabei als DIB-Bitmap definiert, wobei die Typen *hatched brush*, *hollow brush*, *pattern brush* oder *solid brush* zulässig sind. Der Farbwert besteht aus vier Bytes, welche die Farbe angeben. Der Farbwert ist dabei folgendermaßen zu interpretieren:

- Ist das oberste Byte 0, enthalten die unteren drei Bytes die Werte für die Farbanteile (Blau, Grün, Rot). Der Wert 00FF0000H steht z. B. für ein reines Blau.
- Ist das oberste Byte 1, muß das nächste Byte 0 sein. Die beiden unteren Bytes enthalten dann einen Index in die virtuelle Farbpalette.
- Ist das oberste Byte 2, enthalten die restlichen 3 Bytes eine RGB-Farbe. Im Gegensatz zum ersten Fall (oberstes Byte = 0) wird die RGB-Farbe an die ähnlichste Farbe in der virtuellen Palette angeglichen.

### CREATEFONTINDIRECT

Dieser Record vereinbart einen logischen Font.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (variabel)
04H	2	Recordtyp (02FBH)
06H	2	Font Parameter
	2	Höhe in logischen Units
	2	Breite in logischen Units
	2	Escapement (Winkel 1/10 Grad zwischen Basislinie und x-Achse)
	2	Orientation (Wert wird ignoriert)
	2	Fontformat (Weight)
		100 don't care
		100 thin
		200 extralight
		300 light
		400 normal
		500 medium
		600 semibold
		700 bold
		800 extrabold
		900 heavy
	1	Italic (IF <> 0)
	1	Underline (IF <> 0)
	1	Strikeout (IF <> 0)
	1	Character Set
		0 ANSI
		1 default
		2 Symbol
		128 SHIFTJIS
		255 OEM
	1	Output Precision
	1	Clip Precision
	1	Quality
	1	Pitch and Family
	n	Face Name

Tabelle 40.13 Der CREATEFONTINDIRECT-Record

### CREATEPALETTE

Diese Funktion erzeugt eine logische Palette.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (variabel)
04H	2	Recordtyp (0F7H)

Offset	Bytes	Bemerkungen
06H		Parameter
	2	Version
	2	Einträge (n)
	4*n	Color Palette, 1 Byte rot, grün, blau, Flags

**Tabelle 40.14** Der CREATEPALETTE-Record

## CREATEPATTERNBRUSH

Diese Funktion erzeugt in Windows 2.x ein logisches Zeichenwerkzeug (brush), das als Muster in einer Bitmap definiert ist.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (variabel)
04H	2	Recordtyp (01F9H)
06H		Parameter
	2	Breite Bitmap
	2	Höhe Bitmap
	2	Bytes per Rasterlinie
	1	Color Planes
	1	Bits per Pixel
	4	Pointer auf Bitwerte
	n	Bitmuster

**Tabelle 40.15** Der CREATEPATTERNBRUSH-Record (Windows 2.x)

In Windows 3.x wird die folgende modifizierte Struktur verwendet:

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (variabel)
04H	2	Recordtyp (0142H)
06H		Parameter
	2	Bitmap-Typ
	2	Color-Tabellen-Typ
	n	BITMAPINFO-Struktur
	n	Bitmap

**Tabelle 40.16** Der CREATEPATTERNBRUSH-Record (Windows 3.x)

Die BITMAPINFO-Struktur wird im Kapitel über das BMP-Format (Tabelle 17.2) beschrieben.

## CREATEPENINDIRECT

Dieser Record erzeugt einen logischen Stift (pen) mit einer Form (style), einer Strichbreite (width) und einer Farbe (color).

Offset	Bytes	Bemerkungen
00H	4	Recordgröße (08H)
04H	2	Recordtyp (02FAH)
06H	2	Parameter
	2	Style (solid, dash, dot etc.)
	2	Pen Width (x) in log. Units
	2	unbenutzt
	4	Color

**Tabelle 40.17** Der CREATEPENINDIRECT-Record

Die Farbe für Pen wird mit 4 Bytes angegeben. Die Kodierung der Farbwerte wird einige Seiten vorher beim Objekt CREATEBRUSHINDIRECT beschrieben.

## CREATEREGION

Der Record mit dem Opcode 6FFH definiert einen rechteckigen Bereich (region).

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (06FFH)
06H	2	Parameter
	2	X oben links
	2	Y oben links
	2	X unten rechts
	2	Y unten rechts

**Tabelle 40.18** Der CREATEREGION-Record

Der Wert X-Y darf 32767 Einheiten (units) nicht überschreiten.

## DELETEOBJECT

Der Record mit dem Opcode 1F0H löscht ein Objekt. Der Satz enthält ein Handle als einen Parameter zur Objekttabelle.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (01F0H)
06H	2	Parameter
	2	Handle

**Tabelle 40.19** Der DELETEOBJECT-Record

## DRAWTEXT

Über diesen Record läßt sich ein formatierter Text in einem Rechteck ausgeben.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (062FH)
06H	2 2 8 n	Parameter Format Stringlänge (-1 ASCII-Z-String) Koordinaten Rechteck String

Tabelle 40.20 Der DRAWTEXT-Record

## ELLIPSE

Über diesen Record läßt sich eine Ellipse definieren.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0418H)
06H	2 2 2 2	Parameter X1-Koordinate oben links Y1-Koordinate oben links X2-Koordinate unten rechts Y2-Koordinate unten rechts

Tabelle 40.21 Der ELLIPSE-Record

Der Mittelpunkt der Ellipse liegt in der Mitte des Rechtecks der Bounding-Box.

## ESCAPE

Dieser Record spezifiziert eine Escape-Sequenz zum Zugriff auf spezielle Geräte.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0626H)
06H	2 2 n	Parameter Escape-Nummer Zahl Folgebytes Escape-Sequenz

Tabelle 40.22 Der ESCAPE-Record

**EXCLUDECLIPRECT**

Dieser Record erzeugt eine neue Clipping Region, die auf der existierenden Clipping Region minus eines spezifizierten Rechtecks basiert.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0415H)
06H		Parameter
	2	X1-Koordinate oben links
	2	Y1-Koordinate oben links
	2	X2-Koordinate unten rechts
	2	Y2-Koordinate unten rechts

Tabelle 40.23 Der EXCLUDECLIPRECT-Record

**EXTTEXTOUT**

Gibt einen Zeichenstring mit dem aktuellen Font in einem rechteckigen Bereich aus.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0A32H)
06H		Parameter
	2	Y1-Koordinate erstes Zeichen
	2	X1-Koordinate erstes Zeichen
	2	Stringlänge
	2	Rechtecktyp
	8	Rechteckstruktur (x1,y1,x2,y2)
	n	String
	n	Word Array mit Abstand zwischen Zeichen

Tabelle 40.24 Der EXTTEXTOUT-Record

**LINETO**

Der Record definiert eine Linie vom aktuellen zum definierten Punkt.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0213H)
06H		Parameter
	2	X-Koordinate Endpunkt
	2	Y-Koordinate Endpunkt

Tabelle 40.25 Der LINETO-Record

Der Endpunkt ist nicht Bestandteil der Linie.

## MOVETO

Über diesen Record wird die aktuelle Position zum definierten Punkt verschoben.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0214H)
06H	2 2	Parameter X-Koordinate neuer Punkt Y-Koordinate neuer Punkt

Tabelle 40.26 Der MOVETO-Record

## OFFSETCLIPRGN

Dieser Record verschiebt die Clipping Region eines gegebenen Geräts um einen definierten Offset.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0220H)
06H	2 2	Parameter Move x Units Move y Units

Tabelle 40.27 Der OFFSETCLIPRGN-Record

## OFFSETVIEWPORTORG

Der Record 412H modifiziert den Ursprung-Viewport relativ zu den aktuellen Werten. Es wird die Struktur des OFFSETCLIPRGN-Records benutzt.

## OFFSETWINDOWORG

Der Record 40FH modifiziert den Ursprung des Fensters relativ zu den aktuellen Werten. Es gilt die Struktur des OFFSETCLIPRGN-Records.

## PAINTREGION

Der Record füllt den angegebenen Bereich mit dem ausgewählten Zeichenwerkzeug (Brush).

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (012BH)
06H	n	Parameter Füllbereich

Tabelle 40.28 Der PAINTREGION-Record

### PATBLT

Dieser Record definiert ein Muster (Bitpattern) für das angegebene Gerät.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (061DH)
06H	2 2 2 2 8	Parameter X-Koordinate oben links Y-Koordinate oben links Breite Höhe Raster Operationscode

Tabelle 40.29 Der PATBLT-Record

Das Muster (pattern) wird durch ein Rechteck (in logical Units) und den Rastercode definiert. Der Rastercode definiert, welche Funktionen auf die Bitmap anwendbar sind (alles schwarz, alles weiß, Invertieren, Kopieren, OR).

### PIE

Der Opcode 81AH definiert einen Kreisausschnitt (Pie). Dabei wird die Struktur des Ellipse-Records benutzt.

### POLYGON

Der Record definiert ein Polygon, bestehend aus zwei oder mehr Punkten (vertices).

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0324H)
06H	2 n*4	Parameter Zahl der Punkte n Punkte (X, Y)

Tabelle 40.30 Der POLYGON-Record

### POLYLINE

Der Record definiert eine Polylinie (Serie aus Liniensegmenten) und benutzt folgende Recordstruktur:

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0325H)



Offset	Bytes	Bemerkungen
06H	2 n*4	Parameter Zahl der Punkte n Punkte (X, Y)

Tabelle 40.31 Der POLYLINE-Record

## POLYPOLYGON

Der Record definiert mehrere Polygone und benutzt folgende Struktur:

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0538H)
06H	2 n*2 n*4	Parameter Zahl der Polygone (n) Zahl der Punkte Polygon n Liste der Punkte (X, Y)

Tabelle 40.32 Der POLYPOLYGON-Record

## RECTANGLE

Dieser Record definiert ein Rechteck.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (041BH)
06H	2 2 2 2	Parameter X1-Koordinate oben links Y1-Koordinate oben links X2-Koordinate unten rechts Y2-Koordinate unten rechts

Tabelle 40.33 Der RECTANGLE-Record

Das Rechteck wird mit dem aktuellen Pinsel (brush) gefüllt und mit dem gewählten Stift (pen) gezeichnet.

## RESICEPALETTE

Der Record verändert die Größe der aktuellen Palette.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0139H)
06H	2	Parameter Neue Zahl Paletteneinträge

Tabelle 40.34 Der RESICEPALETTE-Record

## ROUNDRECT

Der Record beschreibt ein Rechteck mit abgerundeten Ecken.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (061CH)
06H	2	Parameter
	2	X1-Koordinate oben links
	2	Y1-Koordinate oben links
	2	X2-Koordinate unten rechts
	2	Y2-Koordinate unten rechts
	2	Ellipse Breite für Ecken
	2	Ellipse Höhe für Ecken

Tabelle 40.35 Der ROUNDRECT-Record

Das Rechteck wird mit dem aktuellen Muster (brush) gefüllt und mit dem selektierten Stift (pen) gezeichnet.

## SCALEVIEWPORT

Der Record definiert die Skalierung des Viewports.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0412H)
06H	2	Parameter
	2	Aktueller Multiplikator x
	2	Aktueller Divisor x
	2	Aktueller Multiplikator y
	2	Aktueller Divisor y

Tabelle 40.36 Der SCALEVIEWPORTEXT-Record

Ein neuer Wert wird mit dem Ausdruck:

$$neu = alt * Multiplikator / Divisor$$

berechnet.

## SCALEWINDOWEXT

Der Record (Opcode 400H) verwendet die gleiche Struktur wie der SCALEVIEWPORTEXT-Record und modifiziert die Windows-Extends.

**SETBKCOLOR**

Der Record (Opcode 201H) definiert eine neue Hintergrundfarbe. Der Record enthält die Farbe (4 Bytes) als Parameter.

**SETBKMODE**

Der Record (Opcode 102H) definiert einen neuen Hintergrundmodus. Der Record enthält ein Wort als Parameter für den neuen Modus (opak oder transparent).

**SETDIBTODEV**

Der Record enthält die Bits einer Device-Independant-Bitmap (DIB) für eine Einheit (Device).

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0D33H)
06H		Parameter
	2	Flag Color-Palette
	2	Zahl der Scanlines DIB
	2	erste Scanline DIB
	2	Y-Koordinate Source für DIB
	2	X-Koordinate Source für DIB
	2	Höhe Rechteck für DIB
	2	Breite Rechteck für DIB
	2	Y-Koordinate Origin Destination Rechteck für DIB
	2	X-Koordinate Origin Destination Rechteck für DIB
	n	BITMAP_INFO-Struktur
	n	Bitmap

Tabelle 40.37 Der SETDIBTOSTODEVICE-Record

Die BITMAP\_INFO-Struktur wird in Kapitel 17 im Abschnitt über das BMP-Format (Tabelle 17.2) beschrieben.

**SETPALENTRIES**

Der Record setzt einen RGB-Farbwert und definiert einen zu setzenden Bereich für den Eintrag in der logischen Palette.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0037H)
06H		Parameter
	2	erster Eintrag in Palette
	2	zu setzende Einträge (n)
	n*4	1 Byte für rot, grün, blau, Flag

Tabelle 40.38 Der SETPALENTRIES-Record

### SETPIXEL

Der Record definiert ein Pixel für einen x-, y-Punkt in der angegebenen Farbe.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (041FH)
06H	2 2 4	Parameter X-Position Y-Position Farbe für Pixel (rot, grün, blau, Flag)

Tabelle 40.39 Der SETPIXEL-Record

### SETPOLYFILLMODE

Der Record definiert einen neuen Füllmodus für Polygone. Der Record enthält einen Parameter (Word) mit dem neuen Füllmodus und besitzt den Opcode 106H.

### SETROP2

Der Record definiert einen neuen Zeichenmodus (drawing mode). Der Modus wird als Integer (Word) übergeben, und der Recordtyp ist 104H.

### SETSTRETCHBLTMODE

Der Record setzt den Stretching-Modus (Black on White, Color on Color, White on Black). Verwendet wird der Opcode 107H, und der Modus ist als Integer (Word) im Record gespeichert.

### SETTEXTALIGN

Der Record (Opcode 12EH) enthält ein Flag (Word) für die Textjustierung (text align).

### SETTEXTCHAREXTRA

Der Record definiert einen zusätzlichen Zeichenabstand zwischen Textzeichen. Der Wert wird in logischen Units als Word angegeben. Der Record besitzt den Opcode 108H.

### SETTEXTCOLOR

Der Record (Typ 209H) definiert die Textfarbe. Diese wird als 4-Byte-Parameter (rot, grün, blau, Flag) gespeichert.

### SETTEXTJUSTIFICATION

Der Record enthält einen Parameter zur Justierung des Textes.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (020AH)
06H	2 2	Parameter Extra Space Zahl der Break Characters

**Tabelle 40.40** Der SETTEXTJUSTIFICATION-Record

## SETWINDOWEXT

Der Record definiert die Window-Extends des zugehörigen Fensters.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (020CH)
06H	2 2	Parameter X-Extend in logical Units Y-Extend in logical Units

**Tabelle 40.41** Der SETWINDOWEXT-Record

## SETWINDOWORG

Der Record definiert einen neuen Ursprung für das verbundene Fenster.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (020BH)
06H	2 2	Parameter X-Origin in logical Units Y-Origin in logical Units

**Tabelle 40.42** Der SETWINDOWORG-Record

## STRETCHBLT

Der Record enthält eine Device-Independent-Bitmap (DIB). Hierbei sind zwei Varianten für Windows 2.x (Code B23H) und Windows 3.x (Code B41H) definiert.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0B41H)

Offset	Bytes	Bemerkungen
06H		Parameter
	2	Low Order Word Rasteroperation
	2	High Order Word Rasteroperation
	2	Source y-Extend
	2	Source x-Extend
	2	Y-Koordinate Source Origin
	2	X-Koordinate Source Origin
	2	Destination y-Extend
	2	Destination x-Extend
	2	Y-Koordinate Destination Origin
	2	X-Koordinate Destination Origin
	n	BITMAPINFO-Struktur
	n	Bitmap

**Tabelle 40.43** Der STRETCHBLT-Record (Windows 3.x)

Die BITMAP\_INFO-Struktur wird in Kapitel 17 im Abschnitt über das BMP-Format (Tabelle 17.2) vorgestellt. In Windows 2.x wird diese BITMAP\_INFO-Struktur im Record nicht benutzt. Es wird die gleiche Struktur wie beim BitBlt-Record verwendet.

## STRETCHDIB

Der Record enthält eine Device-Independant-Bitmap (DIB), die in ein Fenster (Window) verschoben wird.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0F43H)
06H		Parameter
	2	anzuwendende Rasteroperation
	2	Flag (Color)
	2	Höhe Bitmap (Source)
	2	Breite Bitmap (Source)
	2	Y-Koordinate Source Origin
	2	X-Koordinate Source Origin
	2	Höhe Destination Rechteck
	2	Breite Destination Rechteck
	2	Y-Koordinate Destination Origin
	2	X-Koordinate Destination Origin
	n	BITMAP_INFO-Struktur
	n	Bitmap

**Tabelle 40.44** Der STRETCHDIBITS-Record

Die BITMAP\_INFO-Struktur wird in Kapitel 17 im Abschnitt über das BMP-Format (Tabelle 17.2) beschrieben.

### TEXTOUT

Der Record enthält eine Zeichenkette, die im aktuellen Font auszugeben ist.

Offset	Bytes	Bemerkungen
00H	4	Recordgröße
04H	2	Recordtyp (0521H)
06H	2	Parameter
	2	Stringlänge
	2	String
	2	Y-Koordinate Startpunkt
	2	X-Koordinate Startpunkt

Tabelle 40.45 Der TEXTOUT-Record

**Anmerkung:** Ein WMF-Reader oder -Writer sollte die Windows-Metafile-Funktionen zur Bearbeitung der Daten verwenden. Die erforderlichen Datenstrukturen werden im Windows-SDK beschrieben.

### Das Enhanced Metafile-Format (EMF)

Das unter Windows 3.x eingeführte Metafile-Format ist geräteabhängig. Die Metadatei enthält keine Informationen zum benutzten Koordinatensystem und zur Lage der Grafik. Um diesen Nachteil zu umgehen, wurde von Aldus das *Placeable Metafile*-Format mit einem erweiterten Header definiert. Trotz dieser Erweiterung ließen sich die Nachteile des WMF-Formats nicht beheben. Die Microsoft-Entwickler haben deshalb in Win32 (Windows 95 und Windows NT) ein erweitertes Metafile-Format (EMF) eingeführt. Eine solche EMF-Datei besitzt einen erweiterten Header, einen optionalen Beschreibungstext, eine Metafile-Palette sowie eine größere Menge an GDI-Funktionen (Abbildung 40.1). Dadurch lassen sich EMF-Dateien allerdings nicht mehr unter Windows 3.x mit den Betriebssystemfunktionen »abspielen«. Unter Win32 lassen sich Enhanced Metafiles aber skalieren und beschneiden (Clipping), was bedeutet: Es ist eine Geräteabhängigkeit gegeben.

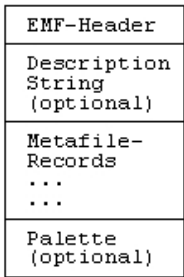


Abbildung 40.1 Struktur einer EMF-Datei

### Der Enhanced Metafile-Header

Eine EMF-Datei besitzt einen erweiterten Header, dessen Struktur hier wiedergegeben wird.

Typ	Name	Bemerkung
DWORD	iType	Recordtyp (01 00 00 00)
DWORD	nSize	Record-Size in Byte
RECTL	rcIBounds	Inclusive Bounds in Device Units
RECTL	rcIFrame	Inclusive Picture Frame des Metafile in der Einheit 0,01 mm
DWORD	dSignature	Signatur (» EMF«)
DWORD	nVersion	Versionsnummer (00 00 01 00)
DWORD	nBytes	Größe der Metadatei in Byte
DWORD	nRecords	Zahl der Records im Metafile
WORD	nHandles	Zahl der Handles in der Handle-Tabelle; Handle-Index 0 ist reserviert
WORD	sReserved	reserviert (muß 0 sein)
DWORD	nDescription	Zahl der Zeichen im Unicode Description-String (oder 0, falls der String fehlt)
DWORD	offDescription	Offset zum Satz des Description String (oder 0, falls der String fehlt)
DWORD	nPalEntries	Zahl der Einträge in der Metafile-Palette
SIZEL	szlDevice	Größe der Referenzeinheit in Pixel
SIZEL	szlMillimeters	Größe der Referenzeinheit in Millimeter

**Tabelle 40.46** Struktur des EMF-Headers

Der Wert *rcIBounds* gibt die Abmessungen der Zeichenfläche in Geräteeinheiten wieder. Der folgende Wert legt den Rahmen für das Metafile-Bild in 0,01 mm-Einheiten fest. Beide Werte werden als RECTL-Struktur definiert (4 \* 4 Bytes mit den Koordinaten der Fensterecken). Enthält die Metadatei einen Description-String, wird dessen Länge und Lage im Header in den Einträgen *nDescription* und *offDescription* angegeben. Enthält die EMF-Datei eine Palette, wird die Zahl der Einträge im Header im Feld *nPalEntries* eingetragen. Die eigentliche Palette befindet sich dann im EMR-EOF-Record am Ende der EMF-Datei hinter den Metafile-Records.

### Enhanced Metafile-Records

Ein Record in einem Enhanced Metafile besitzt den hier gezeigten Aufbau.

DWORD	iType	Recordtyp
DWORD	nSize	Recordgröße in Byte
DWORD	dParm[1]	Feld mit den Parametern des Records

**Tabelle 40.47** Struktur eines EMF-Records



Ein Vergleich ergibt eine ähnliche Recordstruktur wie bei WMF-Dateien. Beachten Sie aber, daß einmal die beiden Felder für den Recordtyp und die Recordgröße beim EMF-Format gegenüber dem WMF-Format vertauscht sind. Weiterhin wird beim WMF-Format die Recordgröße in Worten angegeben, während im EMF-Format die Recordgröße in Byte gespeichert ist. Zusätzlich werden in den EMF-Records DWORD-Datentypen für die Felder verwendet. Die Anzahl und Art der Felder hängt von den jeweiligen Recordtypen ab. Hierbei bildet jeder Record die Parameter des korrespondierenden GDI-Aufrufs ab. enthält eine Übersicht über die EMF-Records und die zugehörigen Recordtypen.

Record	Rec.-Typ	Record	Rec.-Typ
EMR_ABORTPATH	68	EMR_POLYLINE	4
EMR_ANGLEARC	41	EMR_POLYLINE16	87
EMR_ARC	45	EMR_POLYLINETO	6
EMR_ARCTO	55	EMR_POLYLINETO16	89
EMR_BEGINPATH	59	EMR_POLYPOLYGON	8
EMR_BITBLT	76	EMR_POLYPOLYGON16	91
EMR_CHORD	46	EMR_POLYPOLYLINE	7
EMR_CLOSEFIGURE	61	EMR_POLYPOLYLINE16	90
EMR_CREATEBRUSHINDIRECT	39	EMR_POLYTEXTOUTA	96
EMR_CREATEDIBPATTERNBRUSHPT	94	EMR_POLYTEXTOUTW	97
EMR_CREATEMONOBRUSH	93	EMR_REALIZEPALETTE	52
EMR_CREATEPALETTE	49	EMR_RECTANGLE	43
EMR_CREATEPEN	38	EMR_RESIZEPALETTE	51
EMR_DELETEOBJECT	40	EMR_RESTOREDC	34
EMR_ELLIPSE	42	EMR_ROUNDRECT	44
EMR_ENDPATH	60	EMR_SAVEDC	33
EMR_EOF	14	EMR_SCALEVIEWPORTEXTEX	31
EMR_EXCLUDECLIPRECT	29	EMR_SCALEWINDOWEXTEX	32
EMR_EXTCREATEFONTINDIRECTW	82	EMR_SELECTCLIPPATH	67
EMR_EXTCREATEPEN	95	EMR_SELECTOBJECT	37
EMR_EXTFLOODFILL	53	EMR_SELECTPALETTE	48
EMR_EXTSELECTCLIPRGN	75	EMR_SETARCDIRECTION	57
EMR_EXTTEXTOUTA	83	EMR_SETBKCOLOR	25
EMR_EXTTEXTOUTW	84	EMR_SETBKMODE	18
EMR_FILLPATH	62	EMR_SETBRUSHORGE	13
EMR_FILLRGN	71	EMR_SETCOLORADJUSTMENT	23
EMR_FLATTENPATH	65	EMR_SETDIBITSTODEVICE	80
EMR_FRAMERGN	72	EMR_SETMAPMODE	17

Record	Rec.-Typ	Record	Rec.-Typ
EMR_GDICOMMENT	70	EMR_SETMAPPERFLAGS	16
EMR_HEADER	1	EMR_SETMETARGN	28
EMR_INTERSECTCLIPRECT	30	EMR_SETMITERLIMIT	58
EMR_INVERTRGN	73	EMR_SETPALETTEENTRIES	50
EMR_LINETO	54	EMR_SETPIXELV	15
EMR_MASKBLT	78	EMR_SETPOLYFILLMODE	19
EMR_MODIFYWORLDTRANSFORM	36	EMR_SETROP2	20
EMR_MOVETOEX	27	EMR_SETSTRETCHBLTMODE	21
EMR_OFFSETCLIPRGN	26	EMR_SETTEXTALIGN	22
EMR_PAINTRGN	74	EMR_SETTEXTCOLOR	24
EMR_PIE	47	EMR_SETVIEWPORTEXTEX	11
EMR_PLGBLT	79	EMR_SETVIEWPORTORGE	12
EMR_POLYBEZIER	2	EMR_SETWINDOWEXTEX	9
EMR_POLYBEZIER16	85	EMR_SETWINDOWORGE	10
EMR_POLYBEZIERTO	5	EMR_SETWORLDTRANSFORM	35
EMR_POLYBEZIERTO16	88	EMR_STRETCHBLT	77
EMR_POLYDRAW	56	EMR_STRETCHDIBITS	81
EMR_POLYDRAW16	92	EMR_STROKEANDFILLPATH	63
EMR_POLYGON	3	EMR_STROKEPATH	64
EMR_POLYGON16	86	EMR_WIDENPATH	66

**Tabelle 40.48** Enhanced Metafile-Records

Informationen zu den Parametern der einzelnen Records finden Sie in den Win32 API-Beschreibungen (z. B. im Windows 32 SDK oder bei den API-Beschreibungen der Compiler). Als erster Satz muß immer der Metafile-Header auftreten. Der Wert für den Recordtyp ist zu EMR\_HEADER (1) festgelegt (siehe auch Tabelle 40.48). Der letzte Satz in einer EMF-Datei ist der End-of-File-Record, dessen Recordtyp EMR\_EOF (14) ist.

Zusätzlich dürfen zum EMF-Header, dem EOF-Record und den Metafile-Records zwei zusätzliche Informationen (der Description-String und die Metafile-Palette) enthalten sein. Der optionale Description-String folgt auf den Enhanced Metafile-Header. Es handelt sich dabei um einen Unicode-String, dessen Länge und Lage im Header angegeben wird. In diesem String können Sie zusätzliche Informationen zum Dateiinhalt (ein beschreibender Name für das Bild und der Name der Anwendung, welche die EMF-Datei erzeugt hat) hinterlegen. Die beiden Einträge müssen durch ein Nullbyte getrennt werden. Der Description-String endet mit zwei Nullbytes (z. B. »Born-Edit«+00H+»Umsatzgrafik«+00H+00H).

Die optionale Farbpalette wird in einem speziellen Enhanced Metafile-Record (es wird der EMR\_EOF-Record benutzt) untergebracht. Diese Palette lässt sich mit den GDI-Funktionen *CreatePalette*, *SelectPalette* und *RealizePalette* erzeugen. Die Palette lässt sich unter Win32 mit der Funktion *GetEnhMetaFilePaletteEntries* aus der EMF-Datei lesen. Alternativ können Sie im Header prüfen, ob eine Palette vorhanden ist. Falls ja, befindet sich die Palette im letzten EMR\_EOF-Record. Der EMR\_EOF-Record mit Palette besitzt folgenden Aufbau:

Typ	Name	Bemerkung
EMR	emr	Einträge für den Basis Enhanced Metafile-Record
DWORD	nPalEntries	Zahl der Paletteneinträge
DWORD	offPalEntries	Offset zu den Paletteneinträgen
DWORD	nSizeLast	enthält den gleichen Wert wie das nSize-Feld im ersten Feld der <i>emr</i> -Struktur. Falls Paletteneinträge vorliegen, finden sich diese vor diesem DWORD.

**Tabelle 40.49** Struktur des EMR\_EOF-Records

Das letzte DWORD (*nSizeLast*) im EMR\_EOF-Record gibt den Offset vom Dateiende zum Anfang des EMR\_EOF-Records an. Damit lässt sich der Wert *offPalEntries* lesen, und Sie erhalten einen Zeiger auf die Palettendaten.

**Anmerkung:** Weitere Informationen über das EMF-Format bzw. über die API-Aufrufe zum Erzeugen der EMF-Records erhalten Sie im Win32 SDK.