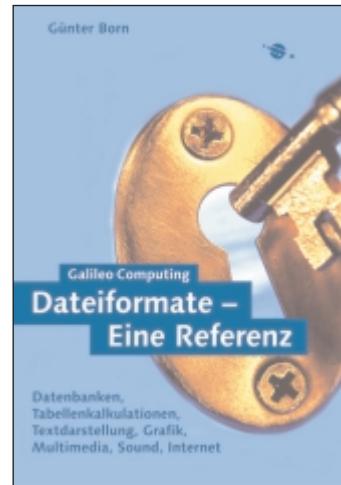


Dieses Kapitel stammt aus dem Buch  
›Dateiformate – Eine Referenz‹  
von Günter Born.

[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**

## 21 Das Dr. Halo-Format (PIC, CUT, PAL)

Das Dr. Halo-Format dient zur Speicherung von Bitmap-Grafiken. Den Maustreibern der Firma Genius liegt ein Programm (Dr. Halo III oder Dr. Genius) bei, welches Bilder im PIC-Format unterstützt. Weiterhin können verschiedene Grafikkonvertierungsprogramme CUT-Bilder erstellen und konvertieren.

Das PIC-Dateiformat lehnt sich stark an die verwendete Hardware an, d.h., ein auf EGA-Karten erstelltes Bild kann kaum auf VGA-Karten ausgegeben werden. Die praktische Verwendbarkeit ist daher stark eingeschränkt. Anders sieht es bei dem CUT-Format aus, welches Bildausschnitte aus Dr. Halo-Bildern speichert. Die Palettes werden in einer dritten Datei (PAL) gespeichert. Tabelle 21.1 gibt die Dateierweiterungen der verschiedenen Varianten an.

Extension	Bemerkungen
CUT	Cut eines Dr. Halo-Bildes
PIC	Dr. Halo-Bild
PAL	Palette für CUT und PIC

Tabelle 21.1 Datei-Extensionen für Dr. Halo-Dateien

### Das PIC-Format

Eine Dr. Halo-Bilddatei besteht aus einem Header mit der Kennung und einigen Feldern mit der Bildbeschreibung (Image Descriptor), gefolgt von den eigentlichen Bilddaten. Die Bilddaten werden dabei in Blöcke zu 512 Byte aufgeteilt, die gemeinsam zu lesen und zu schreiben sind. Tabelle 21.2 gibt den Aufbau des PIC-Headers wieder.

Offset	Bytes	Bemerkungen
00H	2	Signatur 'AH'
02H	2	Versionsnummer (Integer) der Dr. Halo-Treiber-Lib (immer Code E3H (π))
04H	2	unbenutzt (00 00)
06H	1	Image-Flag (2 = Image File)
07H	1	Device Code
08H	2	unbenutzt (00 00)
0AH	n	adapterspezifischer Datenbereich für IBM EGA-Adapter 2 Byte Driver Modus 0 = 320 x 200, 4 Farben 1 = 640 x 200, 2 Farben

Tabelle 21.2 Struktur eines PIC-Headers

Das dritte Byte im Header wird meist mit der Signatur  $p$  belegt. Ab Offset 07H findet sich ein Code für den verwendeten Treiber (Tabelle 21.3).

Code	Treiber
1BH	HALODEBA, AT&T DEB
1CH	HALOINDA, AT&T Indigenous
15H	HALOIBME, Atronics Mega Graph, IBM EGA, STB EGA Plus
49H	HALOHERC, Hercules Mono
01H	HALOIBM, IBM CGA
47H	HALOIBMV, Sigma VGA
36H	HALOIBMP, Sigma VGA
13H	HALOIBMP, STB VGA Extra

Tabelle 21.3 Treibercodes

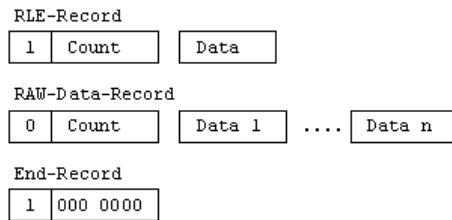
Die Treibermodi mit den zugehörigen Auflösungen sind ab Offset 0AH definiert. Tabelle 21.4 gibt die für die Dr. Halo-Treiber definierten Werte wieder.

Treiber	Nummer	Mode	Auflösung	Farbe	BIOS-Mode
HALOIBM	01H	0	320 x 200	4	04H
		1	640 x 200	2	06H
HALOIBME	15H	0	320 x 200	4	04H
		1	640 x 200	2	06H
		2	320 x 200	16	0DH
		3	640 x 200	16	0EH
		4	640 x 350	16	10H
		5	640 x 800	16	--
		A	640 x 350	4	10H
HALOIBMG	1AH	0	320 x 200	4	04H
		1	640 x 200	2	06H
HALOIBMP	3CH	0	320 x 200	4	04H
		1	640 x 200	2	06H
		2	640 x 480	2	11H
		3	320 x 200	256	13H
HALOIBMV	47H	4	320 x 200	16	0DH
		5	640 x 200	16	06H
		6	640 x 350	16	10H
		7	640 x 480	16	12H
HALODEBA	1BH	2	640 x 200	8	--
		3	640 x 400	8	--
		4	640 x 400	16	--
		5	640 x 400	16	--

Tabelle 21.4 Auflösungen und Modi

An den Header schließt sich dann der Bereich mit den Bilddaten an. Bei EGA-Bildern beginnen die Bilddaten ab Offset 12 (0CH). Dieser Bilddatenbereich wird dabei in Blöcken zu 512 Byte geschrieben und gelesen.

Die Daten innerhalb eines 512-Byte-Blockes werden im RLE-Verfahren komprimiert. Dies setzt eine bestimmte Recordstruktur innerhalb eines 512-Byte-Blockes voraus. Abbildung 21.1 gibt die einzelnen Recordstrukturen wieder.



**Abbildung 21.1** Recordformat der RLE-Kodierung (PIC)

Jeder Record besteht aus einem Headerbyte, gefolgt von einem oder mehreren Datenbytes. Gleiche Datenbytes in einem Bild werden im PIC-Format in einem RLE-Record abgelegt. Hierbei ist das oberste Bit des Headerbytes gesetzt (1). Die restlichen 7 Bits des Headerbytes definieren dann einen Zähler, der die Zahl der Wiederholungen angibt. Mit diesen 7 Bits sind damit Wiederholungsfaktoren zwischen 1 und 127 möglich. Im Folgebyte steht dann das zu kopierende Bitmuster. Die Hexadezimalsequenz

83 02

wird demnach zu

02 02 02

expandiert (83H = 3 Wiederholungen).

Lassen sich mehrere Bilddatenbytes nicht zusammenfassen, sind diese im sogenannten Rohformat (RAW-Format) zu speichern. In diesem Fall wird das oberste Bit des Headers auf 0 gesetzt. Die restlichen 7 Bits definieren einen Zähler (n), der die Zahl der Folgebytes angibt. Auf den Header folgen dann n Bytes mit unkomprimierten Daten. Die Sequenz

90 0F 08 0F 00 0F 0A 0F 09 0F 0C 88 0F

wird dann folgendermaßen expandiert:

0F 16 \* 0F  
 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F  
 08 9 unkomprimierte Bytes 0F 8 \* 0F  
 08 0F 00 0F 0A 0F 09 0F 0C 0F 0F 0F 0F  
 0F

**Abbildung 21.2** Expandierte Bilddaten

Die Bilddaten werden immer in Blöcken zu 512 Byte geschrieben oder gelesen. Die RLE- oder RAW-Records dürfen nicht über eine solche 512-Byte-Grenze hinausreichen. Deshalb bleiben am Ende dieses Bereichs in der Regel einige unbelegte Bytes zurück. Dieser unbelegte Bereich wird durch ein Endbyte markiert, welches den Wert 80H enthält. Dieser Wert kann bei den RLE- und RAW-Records nicht auftreten. Die Folgebytes bis zur Blockgrenze sind dann unbelegt. Das nächste Bilddatenbyte beginnt im folgenden 512-Byte-Block.

Auf Grund der Hardwareabhängigkeit lässt sich eine Grafikdatei im PIC-Format kaum auf andere Rechner übertragen und nur mit dem gleichen Dr. Halo-Treiber bearbeiten. Ich habe das Format der Vollständigkeit halber hier aufgenommen.

### Das CUT-Format

Interessanter für den Austausch von Bildern im Dr. Halo-Format sind die CUT-Dateien. Diese werden in Dr. Halo III durch Ausschneiden und Speichern eines Bildbereiches erzeugt. Das CUT-Format ist unabhängig von der Hardware und wird von verschiedenen Grafikprogrammen unterstützt. Eine CUT-Datei besteht aus einem 6-Byte-Header, gefolgt von den Bilddaten. Tabelle 21.5 spezifiziert den Aufbau des CUT-Headers.

Offset	Bytes	Bemerkungen
00H	2	Bildbreite in Pixel
02H	2	Bildhöhe in Pixel
04H	2	unbenutzt (00 00)

Tabelle 21.5 Struktur eines CUT-Headers

Der Rest der Datei enthält die Bilddaten, die in Records mit folgender Struktur aufgeteilt sind:

Bytes	Bemerkungen
2	Länge Datenbereich in Byte
n	Datenbereich

Tabelle 21.6 Struktur eines CUT-Records

Der Datenbereich innerhalb eines CUT-Records wird wieder per RLE-Verfahren komprimiert, d.h., die Bilddaten liegen in einer Struktur wie beim PIC-Format vor.

### Das PAL-Format

Die PIC- und CUT-Dateien enthalten keine Informationen über die verwendete Farbpalette. Vielmehr wird eine getrennte Palette-Datei mit der Erweiterung PAL, aber dem gleichen Dateinamen wie die CUT- oder PIC-Datei benutzt. Eine PAL-Datei besteht aus einem 40-Byte-Header, gefolgt von den Palette-Daten.

Offset	Bytes	Bemerkungen
00H	2	Signatur 'AH'
02H	2	Versionsnummer (Integer) der Dr. Halo-Treiber-Library meist Code E3H (p)
04H	2	Dateigröße – Headerlänge
06H	1	Signatur für PAL-Files (0AH)
07H	1	Palette File Subtyp 0 = generisch 1 = adapterspezifisch
08H	2	Adapter-ID-Nummer (unbenutzt, falls Palette-File-Subtype = generisch)
0AH	2	Graphikmodus
0CH	2	Palette-Einträge 15 = EGA, 255 = VGA
0EH	2	Maximum Rot (nur EGA) 3 = 100 %, 2 = 66 %, 1 = 33 %
10H	2	Maximum Grün (nur EGA) 3 = 100 %, 2 = 66 %, 1 = 33 %
12H	2	Maximum Blau (nur EGA) 3 = 100 %, 2 = 66 %, 1 = 33 %
14H	19	Palette-ID (ASCII-Text)

**Tabelle 21.7** Struktur eines PAL-Headers

Die ersten Bytes des Headers stimmen weitgehend mit der CUT-Struktur überein. Sofern im Byte ab Offset 07H der Wert 0 (generic) steht, sind gerätespezifische Informationen nicht relevant. Der Palette-ID-String (Offset 14H) enthält meist die Signatur »Dr. Halo«.

An den Header schließt sich die Tabelle mit den Palette-Daten an. Die Zahl der Einträge in der Palette steht im Header ab Offset 0CH. Jeder Eintrag umfaßt drei Integerwerte mit den Farbanteilen in der Reihenfolge Rot, Grün und Blau. Beachten Sie aber, daß nur das untere Byte belegt wird, d.h., die Werte liegen zwischen 0 und 255 (0 bis 100 %). Die Farbe Schwarz wird zu 00 00, 00 00, 00 00 definiert. Weiß ist als FF 00, FF 00, FF 00 festgelegt.

