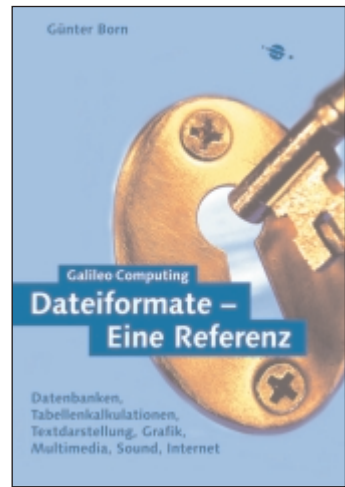


Dieses Kapitel stammt aus dem Buch
›Dateiformate – Eine Referenz‹
von Günter Born.

www.borncity.de

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



Informationen zum Buch
mit Bestellmöglichkeit

www.galileocomputing.de

Galileo Computing

© Copyright 2001 by Galileo Press

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

27 Das JPEG/JFIF-Format (JPG)

Das JPEG-Verfahren beschreibt nur die Komprimierungsmethoden für Bilder. Im JPEG-Standard (ISO DIS 10918-1) wird in Anhang B auch die Struktur der zugehörigen Dateien beschrieben. Allerdings erlaubt diese Definition viele Freiheiten, so daß ein Austausch von JPEG-Bilddaten zwischen verschiedenen Anwendungen und Plattformen relativ problematisch wird.

Mit dem JPEG File Interchange-Format wurde deshalb ein minimaler Standard definiert, um den Austausch von Bildern mit JPEG-Komprimierung über verschiedene Plattformen und Anwendungen zu organisieren. Das Format ist sowohl für PC, Macintosh als auch für Unix-Rechner implementierbar. So werden zum Beispiel keine Resource Forks des Macintosh benutzt, so daß ein Austausch mit anderen Plattformen möglich wird.

Das JFIF-Format baut auf dem JPEG-Standard (ISO DIS 10918-1) auf. Die folgende Beschreibung basiert auf der Version 1.02 der JFIF-Spezifikation. Dateien, die dem JPEG-Standard entsprechen, müssen in einzelne Blöcke, auch als Marker-Segmente bezeichnet, aufgeteilt werden. Jedes Marker-Segment besteht aus einer Kennung (Marker), gefolgt von den optionalen Parametern. Umfaßt ein Marker-Segment mehr als den Marker, findet sich im folgenden Feld die Längenangabe für das Segment.

Das JFIF-Format benutzt nur eine Untermenge der im Anhang B des Standards aufgeführten Marker für die einzelnen Segmente der Datei. Abbildung 27.1 zeigt den Aufbau einer JFIF-Datei aus den einzelnen Marker-Segmenten.

SOI-Segment
APP0-Segment
optionale JFIF Extension APP0-Segments
SOF-Segment
EOI-Segment

Abbildung 27.1 Struktur einer JFIF-Datei

Die Datei beginnt immer mit einem *Start of Image- (SOI-) Marker-Segment*. Daran schließt sich ein *Application- (APP0-) Marker-Segment* an. Dieses enthält zusätzlich Informationen zu den gespeicherten Bilddaten (z. B. Version, Auflösung etc.). Eine JFIF-Datei kann zusätzlich verkleinerte Bilder (Thumbnails) zur Vorschau enthalten. Für diesen Fall schließen sich an das erste *APP0-Marker-Segment* weitere optionale *Extension APP0-Marker-Segmente* an. Die komprimierten Bilddaten folgen in den *Start of Frame- (SOF-) Marker-Segmenten*. Eine JFIF-Datei wird immer mit einem *End of Image- (EOI-) Marker-Segment* abgeschlossen.

Die einzelnen Marker-Segmente enthalten zu Beginn einen 2-Byte-Marker, der den Typ identifiziert. Der Markercode besteht aus dem Wert FFxxH, wobei xx für eine Zahl zwischen C0H und FEH steht. Die Markercode sind im JPEG-Standard im Anhang B definiert. Alle Daten in der JFIF-Datei werden dabei im Motorola-Format gespeichert. Die Struktur der einzelnen Marker-Segmente einer JFIF-Datei werden nachfolgend vorgestellt.

Anmerkung: Die JPEG-Spezifikation sieht für das Austauschformat eine Reihe zusätzlicher Marker-Segmente für Komprimierungstabellen vor. Diese werden nachfolgend ebenfalls beschrieben, sollten aber in JFIF-Dateien nicht auftreten, da hier eine Baseline DCT-Komprimierung gefordert wird.

Die Marker einer JFIF-Datei

Innerhalb der JFIF-Datei sind folgende Marker vorgesehen:

Das Start of Image- (SOI-) Marker-Segment

Jede JFIF-Datei (und auch jede andere gemäß dem JPEG-Standard gespeicherte Datei) muß mit einem SOI-Marker-Segment beginnen. Dieses Segment besitzt folgende Struktur.

Offset	Bytes	Bemerkungen
00H	2	SOI-Signatur (FFD8H)

Tabelle 27.1 Struktur eines SOI-Marker-Segments

Das Segment umfaßt nur die zwei Bytes mit der Signatur. Diese Signatur ist im JPEG-Standard definiert.

Das End of Image- (EOI-) Marker-Segment

Jede JFIF-Datei (und auch jede andere gemäß dem JPEG-Standard gespeicherte Datei) ist mit einem EOI-Marker-Segment abzuschließen. Dieses Segment besitzt folgende Struktur.

Offset	Bytes	Bemerkungen
00H	2	EOI-Signatur (FFD9H)

Tabelle 27.2 Struktur eines EOI-Marker-Segments

Das Segment umfaßt ebenfalls nur zwei Bytes zur Signatur.

Das Application-(APP0-)Marker-Segment

Das zweite Segment einer JFIF-Datei muß ein APP0-Marker-Segment sein. Dieses Segment zeigt die Kompatibilität zur JFIF-Spezifikation an und dient auch zur Identifizierung der Datei.

Offset	Bytes	Bemerkungen
00H	2	APP0-Signatur (FFE0H)
02H	2	Segmentlänge
04H	5	ID = »JFIF«
09H	2	Version (0102H)
0BH	1	Units
0CH	2	X Density
0EH	2	Y Density
10H	1	X Thumbnail
11H	1	Y Thumbnail
12H	3*n	RGB-Thumbnail-Werte

Tabelle 27.3 Struktur des APP0-Marker-Segments

Das Marker-Segment beginnt mit der Signatur FFE0H. Daran schließt sich ein Wort mit der Längenangabe des Blocks an. Diese Längenangabe umfaßt alle Bytes des Blocks ohne die zwei ersten Bytes mit der Signatur. Ab Offset 04H beginnt eine Kennung von 5 Byte für das Marker-Segment. Bei einer JFIF-Datei muß hier die Signatur 4AH 46H 49H 46H 00H stehen. Diese Signatur entspricht dem ASCII-Z-String »JFIF«. Bei einer gültigen JFIF-Datei muß deshalb ab Offset 04H diese Signatur auftreten.

Das Wort ab Offset 09H enthält die Versionsnummer der JFIF-Spezifikation. Das erste Byte definiert dabei die Hauptversionsnummer. Die vorliegende Spezifikation besitzt die Versionsnummer 0102H.

Die Felder ab Offset 0BH definieren die Auflösung des Bildes. Das Byte ab Offset 0BH spezifiziert das Maßsystem (Units) für die angegebene Bildauflösung (Resolution). Hierbei gilt die Kodierung aus Tabelle 27.4.

Code	Einheit
0	keine
1	x,y dots per inch
2	x,y dots per Zentimeter

Tabelle 27.4 Einheit der Auflösung

Falls der Code auf 0 gesetzt ist, definieren x und y das Bildverhältnis (pixel aspect ratio). Ab Offset 0CH findet sich dann ein Wort mit der horizontalen Auflösung (horizontal pixel

density). Die vertikale Bildauflösung (vertical pixel density) wird ab Offset 0EH gespeichert.

Die JFIF-Datei kann ein verkleinertes Bild (Thumbnail) als RGB-Bitmap enthalten. Ab Offset 10H findet sich die Breite dieses Bildes in Pixel. Die Bildhöhe in Pixel wird ab Offset 11H gespeichert. Enthält die Datei keine Thumbnail-Bilder, sind die beiden Felder auf 0 zu setzen. Die Bilddaten schließen sich ab Offset 12H an. Diese werden dabei als RGB-Tabelle mit je drei Bytes (rot, grün, blau) gespeichert. Die Zahl der Einträge errechnet sich aus *Breite * Höhe*.

Das Extension APP0-(SOI-)Marker-Segment

An das APP0-Marker-Segment können sich ab der Version 1.02 optional weitere *Extension APP0-Marker-Segmente* anschließen. Die Syntax eines solchen *Extension APP0-Marker-Segments* ist in Tabelle 27.5 beschrieben.

Offset	Bytes	Bemerkungen
00H	2	APP0-Signatur FFE0H
02H	2	Segmentlänge
04H	5	ID = »JFXX«
09H	1	Extension Code
0AH	n	Datenbereich

Tabelle 27.5 Struktur eines Extension APP0

Das *Extension APP0-Marker-Segment* beginnt ebenfalls mit der Signatur FFE0H. Daran schließt sich ein Wort mit der Längenangabe des Blocks an. Diese Längenangabe umfaßt alle Bytes des Blocks ohne die zwei ersten Bytes mit der Signatur. Ab Offset 04H beginnt eine Kennung von 5 Byte für das *Extension Marker-Segment*. Bei einer JFIF-Datei muß hier die Signatur 4AH 46H 58H 58H 00H stehen. Diese Signatur entspricht dem ASCII-String »JFXX«.

Ab Offset 09H folgt dann der 1-Byte-Extension Code, der den Typ des *Extended APP0-Marker-Segments* spezifiziert. Hier sind bisher folgende Codes definiert:

Code	Bemerkungen
10H	Thumbnail JPEG-kodiert
11H	Thumbnail als 1 Byte/Pixel
13H	Thumbnail als 3 Byte/Pixel

Tabelle 27.6 Kodierung Extend APP0-Marker-Segment

Ab Offset 0AH schließt sich dann der Datenbereich an. Dieser Datenbereich variiert mit der jeweiligen Extension.

JFIF Extension: Thumbnail JPEG-kodiert

Diese Erweiterung dient zur Speicherung von Thumbnails, deren Bilddaten mit dem JPEG-Verfahren komprimiert wurden. Als Extension-Code wird der Wert 10H im Marker-Segment eingetragen. Das Extension-Datenfeld besitzt dann folgenden Aufbau:

SIO-Marker-Segment
 SOF-Marker-Segment
 EOI-Marker-Segment

Die Struktur dieser Segmente wird auf den folgenden Seiten beschrieben. Innerhalb des Datenbereiches dürfen keine Marker-Segmente mit den Signaturen »JFIF« oder »JFXX« auftreten.

JFIF-Extension: Thumbnail mit 1 Byte/Pixel

Bei dieser Erweiterung werden Thumbnails mit einem Byte pro Pixel abgelegt. Als Extension-Code wird der Wert 11H im Marker-Segment eingetragen. Das Extension-Datenfeld besitzt dann folgenden Aufbau:

Bytes	Bemerkungen
1	Thumbnail horizontale Pixelzahl (x)
1	Thumbnail vertikale Pixelzahl (y)
768	Palette mit 256 * 3 Byte (rot, grün, blau)
n	Datenbereich 1 Byte/Pixel für Thumbnail-Bild ($n = x*y$)

Tabelle 27.7 JFIF-Extension für Thumbnails (1 Byte/Pixel)

Die Daten werden unkomprimiert gespeichert.

JFIF-Extension: Thumbnail mit 3 Byte/Pixel

Bei dieser Erweiterung werden Thumbnails mit drei Byte pro Pixel abgelegt. Als Extension-Code wird der Wert 13H im Marker-Segment eingetragen. Das Extension-Datenfeld besitzt dann folgenden Aufbau:

Bytes	Bemerkungen
1	Thumbnail horizontale Pixelzahl (x)
1	Thumbnail vertikale Pixelzahl (y)
n	Datenbereich 3 Byte/Pixel für Thumbnail-Bild ($n = 3*x*y$)

Tabelle 27.8 JFIF-Extension für Thumbnails (3 Byte/Pixel)

Die Daten werden unkomprimiert gespeichert.

Das Define Huffman Table-(DHT-)Marker-Segment

Die JPEG-Spezifikation fordert, daß alle Tabellen mit Definitionen, die während der Dekodierung benötigt werden, vor der Benutzung im Datenstrom gespeichert werden müssen. Der JPEG-Standard beschreibt eine Reihe verschiedener Marker-Segmente mit Hilfstabellen zur Kodierung. Das *DHT-Marker-Segment* besitzt folgenden Aufbau:

Offset	Bytes	Bemerkungen
00H	2	DHT-Signatur (FFC4H)
02H	2	Segmentlänge
04H	1	Color Component Index
05H	16	Length Huffman-Tabelle
15H	n	Huffman-Tabelle

Tabelle 27.9 Struktur eines DHT-Marker-Segments

Das Marker-Segment umfaßt die zwei Bytes mit der Signatur, die Längenangabe (ohne die Signatur) und den Index auf die betreffende Farbkomponente. Dann wird die Länge der Huffman-Tabelle ab Offset 05H definiert. Daran schließt sich die eigentliche Tabelle an. Die Huffman-Tabellen sind in der JPEG-Spezifikation definiert, werden aber in JFIF-Dateien nicht benötigt.

Das Define Arithmetic Coding-(DAC-)Marker-Segment

Dieses Marker-Segment enthält die Informationen für die im Index angegebene Farbkomponente. Das *DAC-Marker-Segment* besitzt folgenden Aufbau:

Offset	Bytes	Bemerkungen
00H	2	DAC-Signatur (FFCCH)
02H	2	Segmentlänge
04H	1	Color Component Index
05H	1	Wert

Tabelle 27.10 Struktur eines DAC-Marker-Segments

Das Marker-Segment umfaßt die zwei Bytes mit der Signatur sowie die Längenangabe (ohne die Signatur). Nach dem Index auf die betreffende Farbkomponente wird die Arithmetikkodierung als Byte gespeichert. Dieses *Marker-Segment* wird aber in JFIF-Dateien nicht benötigt.

Das Define Quantisation Table-(DQT-)Marker-Segment

Dieses Marker-Segment definiert eine Quantisierungstabelle (DQT). Das *DQT-Marker-Segment* besitzt folgenden Aufbau:

Offset	Bytes	Bemerkungen
00H	2	DQT-Signatur (FFDBH)
02H	2	Segmentlänge
04H	1	Index
05H	64	Quantisierungstabelle

Tabelle 27.11 Struktur eines DQT-Marker-Segments

Das Marker-Segment umfaßt die zwei Bytes mit der Signatur und die Längenangabe (ohne die Signatur). Der Index liegt verschlüsselt vor. Die Bits 0 bis 3 definieren den Index in die Quantisierungstabelle, während der Wert der Bits 4 bis 7 das Format der Einträge angibt. Mit dem Wert 0 liegen die Einträge als Bytewerte vor, während der Wert 1 die Einträge als Word definiert. Dieses Marker-Segment sollte in JFIF-Dateien nicht benutzt werden.

Das Define Restart Interval-(DRI-)Marker-Segment

Innerhalb der Datenblöcke kann ein Restart der Kodierung erforderlich werden. Die Länge der Restart-Intervalle wird über ein DRI-Marker-Segment festgelegt.

Offset	Bytes	Bemerkungen
00H	2	DRI-Signatur (FFDDH)
02H	2	Segmentlänge
04H	2	Restart Interval

Tabelle 27.12 Struktur eines DRI-Marker-Segments

Das Marker-Segment umfaßt die zwei Bytes mit der Signatur und die Längenangabe (ohne die Signatur). Ab Offset 04H findet sich der Wert für das Restart-Intervall. Nach der im Intervall angegebenen *Minimum Code Units (MCU)* wird die JPEG-Kodierungsprozedur reinitialisiert. Dieses Marker-Segment sollte in JFIF-Dateien (Version 1.02) nicht auftreten.

Das Start of Frame-(SOF-)Marker-Segment

Die *SOF-Marker*-Segmente enthalten die Informationen zur Bildgröße und die Zuordnung der Quantisierungstabellen zu den einzelnen Farbkomponenten. Es gibt verschiedene *SOF-Marker*-Segmente, die mit den Kennungen 0, 1 ... 9, a, b, c, d, e und f bezeichnet werden. Die JFIF-Spezifikation sieht lediglich die *Baseline DCT*-Kodierung vor, so daß die oberen beschriebenen Marker-Segmente nicht auftreten sollten. Tabelle 27.13 enthält eine Aufstellung der verschiedenen *SOF-Marker*-Segmente.

Code	Marker-Segmente
--	- non-differential Huffman coding -
0	Baseline DCT
1	Extended sequential DCT
2	Progressive DCT
3	Spatial (sequential) lossless
--	- differential Huffman coding -
5	Differential sequential DCT
6	Differential progressive DCT
7	Differential spatial
--	- non-differential arithmetic coding -
8	reserviert
9	Extended Sequential DCT
a	Progressive DCT
b	Spatial (sequential) lossless
--	- differential arithmetic coding -
d	Differential sequential DCT
e	Differential progressive DCT
f	Differential spatial

Tabelle 27.13 Kodierung der SOF-Marker-Typen

Die in Tabelle 27.13 angegebenen Codes werden in der Spezifikation als *Index SOFx* angegeben. Weiterhin wird der Code als letzte Ziffer der Signatur (z. B. FFCaH = progressive DCT) verwendet. Ein *SOF-Marker-Segment* besitzt die in Tabelle 27.14 angegebene Struktur.

Offset	Bytes	Bemerkungen
00H	2	SOFx-Signatur (FFDxH)
02H	2	Segmentlänge
04H	1	Data Precision
05H	2	Bildhöhe in Pixel
07H	2	Bildbreite in Pixel
09H	1	Zahl der Komponenten
0AH	1	ID 1. Komponente
0BH	1	Abtastfaktor (sample factor) Bit 0-3: vertikal Bit 4-7: horizontal

Offset	Bytes	Bemerkungen
0CH	1	Nummer Quantisierungstabelle
0DH	1	ID 2. Komponente
0EH	1	Abtastfaktor Bit 0-3: vertikal Bit 4-7: horizontal
0FH	1	Nummer Quantisierungstabelle
10H	1	ID 3. Komponente
11H	1	Abtastfaktor Bit 0-3: vertikal Bit 4-7: horizontal
12H	1	Nummer Quantisierungstabelle
.....		

Tabelle 27.14 Struktur eines SOF-Marker-Segments

Das Marker-Segment umfaßt die zwei Bytes mit der Signatur. Hier wird für x ein Wert zwischen 0 bis FH gemäß Tabelle 27.13 eingesetzt. Wegen der geforderten Baseline DCT-Komprimierung sollte die Signatur FFD0H auftreten. Die Längenangabe beinhaltet nicht die 2 Bytes der Signatur. Ab Offset 04H findet sich der Wert für die Auflösung (data precision). Die Word-Daten ab Offset 05H definieren die Bildabmessungen in Pixel. Ab Offset 09H findet sich die Zahl der Farbkomponenten. Bei JFIF-Dateien sind die Werte 1 oder 3 erlaubt.

Die eigentlichen Bilddaten schließen sich ab Offset 0AH an (ID, Abtastfaktor, Nummer Quantisierungstabelle). Die Bilddaten werden dabei im YCbCr-Farbmodell in die Zahl der angegebenen Komponenten (1 oder 3) aufgeteilt.

Die Farbkodierung

JPEG-Grafiken werden im YCbCr-Farbmodell gespeichert. Dieses Farbmodell erlaubt eine Reduzierung der Informationen in der Farbebene (Crominanz), während die Helligkeit (Luminanz) erhalten bleibt. Der Standard-YCbCr-Farbraum ist gemäß CCIR 601 mit 256 Farbebenen festgelegt. Die RGB-Farben lassen sich linear aus dem YCbCr-Farbmodell ableiten. Eine Gammakorrektur ist nicht durchzuführen. Wird nur eine Komponente benutzt, sollte dies die Y-Komponente sein. Für die Komponenten des YCbCr-Farbsystems gilt:

$$Y = 256 * E'y$$

$$Cb = 256 * [E'Cb] + 128$$

$$Cr = 256 * [E'Cr] + 128$$

Abbildung 27.2 Die YCbCr-Farbkodierung

Die Werte E'y (0 ... 1.0), E'Cb (-0.5 ... +0.5) und E'Cr (-0.5 ... +0.5) sind in CCIR 601 definiert. Die 256 Farbstufen des YCbCr-Farbsystems lassen sich direkt aus den RGB-Werten berechnen:

$$Y = 0.299 R + 0.578 G + 0.114 B$$

$$Cb = -0.1687 R - 0.3313 G + 0.5 B + 128$$

$$Cr = 0.5 R - 0.4187 G - 0.0813 B + 128$$

Abbildung 27.3 Berechnung der YCbCr-Farbe

Die Werte R, G, B stehen dabei für die Farbanteile rot, grün und blau, die zwischen 0 und 255 liegen dürfen. Die Konvertierung vom YCbCr-Farbraum in die RGB-Ebene erfolgt über die Formeln:

$$R = Y + 1.402 (Cr-128)$$

$$G = Y - 0.34414 (Cb-128) - 0.71414 (Cr-128)$$

$$B = Y + 1.772 (Cb-128)$$

Abbildung 27.4 Berechnung der RGB-Farbe

Die Werte Y, Cb und Cr dürfen zwischen 0 und 255 liegen.

In den JFIF-Dateien ist die Bildorientierung immer von oben nach unten festgelegt. Die Bilddaten werden weiterhin von links nach rechts geschrieben.

Anmerkung: Bezüglich der Kodierung der einzelnen Daten und deren Bearbeitung möchte ich auf die JPEG-Spezifikation verweisen. In *W. Pennebaker, J. Mitchell, JPEG: Still image data compression standard, Van Nostrand Reinhold, ISBN 0-442-01272-1* finden Sie die JPEG-Spezifikation, die Definition der Datenformate und eine Beschreibung der Komprimierungsverfahren.

Das Start of Scan-(SOS-)Marker-Segment

Der SOS-Block leitet bei mehreren Bildern den Start eines Scans ein und besitzt folgende Struktur:

Offset	Bytes	Bemerkungen
00H	2	SOS-Signatur (FFDAH)
02H	2	Segmentlänge
04H	1	Zahl der Komponenten
05H	1	ID-Komponente
06H	1	Tabellenindex Bit 0-3: AC-Tabelle BIT 4-7: DC-Tabelle

Tabelle 27.15 Struktur eines SOS-Marker-Segments

Das Marker-Segment umfaßt die zwei Bytes mit der Signatur. Die Längenangabe beinhaltet nicht die 2 Bytes der Signatur. Ab Offset 04H findet sich das Byte mit der ID der benutzten Farbkomponente. Die Zuordnung der Tabellen zu der Komponente erfolgt über das Byte ab Offset 06H. Dieses Marker-Segment sollte in JFIF-Dateien jedoch nicht auftreten.

Anmerkung: Treten in einer JFIF-Datei unbekannte Marker-Segmente auf, sollte der JFIF-Leser diese überlesen können, ohne daß die Bildd Dekodierung beeinflußt wird.

Das SPIFF-Format

Bei dem JFIF handelt es sich um ein von Eric Hamilton (Firma C-Cube Microsystem) definiertes Format zum Austausch von JPEG-Daten. Mittlerweile wurde ein ISO-Standard 10918-2 »Digital Compression and Coding of Continuous-Tone Still Images: Extensions« verabschiedet. In diesem Standard wird auch ein Austauschformat für JPEG-Grafiken unter dem Namen SPIFF definiert. Eine Datei im SPIFF-Format besitzt dabei einen blockweisen Aufbau (Abbildung 27.5).



Abbildung 27.5 Struktur einer SPIFF-Datei

Der SPIFF-Header

Der Header eine SPIFF-Datei umfaßt 36 Bytes und besitzt den in Tabelle 27.16 gezeigten Aufbau:

Offset	Bytes	Bemerkungen
00H	4	Signatur FFD8FFE8H
04H	2	Header-Länge ohne Signatur
06H	6	ID-Wert »SPIFF«+00H
0CH	2	Version
0EH	1	Application Profile ID
0FH	1	Zahl der Farbkomponenten
10H	4	Bildhöhe
14H	4	Bildbreite

Offset	Bytes	Bemerkungen
18H	1	benutzter Farbraum
19H	1	Bit per Pixel
1AH	1	Komprimierungstyp
1BH	1	Resolution-Unit
1CH	4	vertikale Auflösung
20H	4	horizontale Auflösung

Tabelle 27.16 Struktur eines SPIFF-Headers (Version 1.0)

Die ersten vier Bytes enthalten eine feste Signatur FFD8FFE8H. Daran schließt sich die Header-Länge (ohne die Signatur) an. Der Wert wird auf 32 (20H) gesetzt. Danach folgt eine Zeichenkette mit der Signatur SPIFF. Die Zeichenkette wird durch ein Nullbyte abgeschlossen.

Als Versionsnummer kann 01H 00H eingetragen werden, was der Version 1.0 entspricht. Ab Offset 0EH kann die Anwendung einen ID-Code hinterlegen, der angibt, welches Profil der Leser unterstützen muß (0 kein Profil angegeben, 1 Continuous-Tone Base-Profil, 2 Continous-Ton Progressive-Profil, 3 Bi-Level Facsimile-Profil, 4 Continuous-Tone Facsimile-Profil).

Ab Offset 0FH wird die Zahl der Farbkanäle hinterlegt. Mit 1 wird ein monochromes Bild angegeben, und der Wert 3 steht für ein RGB- oder CMY-Bild. Die Bildabmessungen werden als Zahl der Bildzeilen und Zahl der Bildpunkte (Samples) angegeben. Der Farbraum (Offset 18H) für die Farbkodierung ist gemäß Tabelle 27.17 definiert.

Wert	Farbraum
0	Bi-Level
1	YCbCr, ITU-R BT 709, Video
2	kein Farbraum definiert
3	YCbCr, ITU-R BT 601-1, RGB
4	YCbCr, ITU-R BT 601-1, Video
5-7	reserviert
8	Graustufenbild
9	PhotoYCC
10	RGB
11	CMY
12	CMYK
13	YCKK
14	CIELab

Tabelle 27.17 Kodierung des Farbraums

Die Bilddaten können nach verschiedenen Komprimierungsverfahren in der Datei hinterlegt werden. Das Komprimierungsverfahren wird im Header ab Offset 1AH gemäß Tabelle 27.18 hinterlegt.

Wert	Komprimierungsverfahren
0	unkomprimiert
1	Modified Huffman
2	Modified READ
3	Modified Modified READ
4	JBIG
5	JPEG

Tabelle 27.18 Komprimierungstypen

Das Feld ab Offset 1BH gibt die *Resolution Unit* an (0 *Aspect Ratio* wird durch die vertikale und horizontale Auflösung definiert, 1 Dots per Inch, 2 Dots per Zentimeter). Die beiden letzten Felder des Headers enthalten dann die Auflösung (Resolution) für die X- und Y-Richtung (die Felder sind als DWORD ausgelegt). Wird der Wert der *Resolution Unit* auf 0 gesetzt, enthalten die beiden Felder den Zähler und den Nenner für den Wert des *Aspect Ratio* (Seitenverhältnis).

Der Directory-Teil

An den Header schließt sich eine Directory-Struktur mit folgendem Aufbau an:

Bytes	Bedeutung
2	Signatur FFE8H
2	Länge des Eintrags
4	TAG-ID des Eintrags (Bitfeld)
n	Daten (optional)

Tabelle 27.19 Aufbau der Directory-Struktur

Die ersten beiden Bytes dienen als Signatur zur Identifikation des Directory-Eintrags. Der Wert wird immer auf FFE8H gesetzt. Danach folgt ein Wort, das die Länge des Eintrags (ohne die 2 Bytes der Signatur) angibt. Der Wert kann zwischen 6 und 65534 Byte liegen. Die folgenden vier Bytes werden als Bitfeld interpretiert. Die Belegung dieser Bits ist folgendermaßen definiert:

Bit	Bedeutung
0–20	Tag-ID 01 = End of Directory 02 = Transfer Characteristics 03 = Component Registration

Bit	Bedeutung
	04 = Image Orientation 05 = Thumbnail Image 06 = Image Title 07 = Image Description 08 = Time Stamp 09 = Version Number 10 = Creator Identification 11 = Protection Indicator 12 = Copyright Information 13 = Contact Information 14 = Tile Index 15 = Scan Index 16 = Set Reference
21–23	Konformität 0 = SPIFF-Spezifikation-Definition 1–3 = ISO/IEC- und Common Text Generic-Standard 4 = ISO Application-Standard 5 = ITU-T Recommendation 6 = National Standards Bodies 7 = Application specific

Tabelle 27.20 Belegung des Bitfelds mit dem ID-Wert des Directory-Eintrags

An diesen Header können sich nun optional die Directory-Daten anschließen. Diese müssen in 4 Byte langen Datenstrukturen angegeben werden. Hierbei dürfen durchaus mehrere solcher Directory-Strukturen in der Datei auftreten. Alternativ kann ein Zeiger auf dem (indirekten) Datenbereich abgelegt sein. Die Einträge im Directory-Teil werden nicht durch Offset-Werte miteinander verbunden. Vielmehr liegen die einzelnen Einträge jeweils hintereinander. Das Ende des Directory-Bereichs wird durch den *End of Directory (EOD)*-Marker markiert. Dieser Marker beginnt mit der Signatur FFE8H, weist eine Länge von 8 Byte auf und besitzt im letzten DWORD das Tag-ID 1.

Anmerkung: SPIFF-Leser können alle Einträge mit unbekannten Tags überlesen.

Der Datenbereich

An den Directory-Teil (nach dem EOD-Marker) schließen sich die Bilddaten an. Es handelt sich um einen Datenstrom, der in der angegebenen Kodierung vorliegt. Dies können Daten in der Huffman-Kodierung (Huffman 1D), Modified READ (MR), Modified Modified READ (MMR) und JBIG-Kompression (gem. ITU-T T.82 Spezifikation) sein.

Anmerkung: Wenn Sie eine JFIF-Datei um einen SPIFF-Header und einen EOD-Marker ergänzen, haben Sie im Grunde eine SPIFF-Datei. (Problem hierbei ist aber, daß nicht alle JFIF-Reader diese Datei lesen können.) Bisher wird das SPIFF-Format nach meinem Wissen noch nicht durch Anwendungen unterstützt. Lediglich das von der JPEG-Gruppe implementierte Beispieldatensatz (JPEG-Library) enthält auch Dateien, um SPIFF-Dateien zu unterstützen.