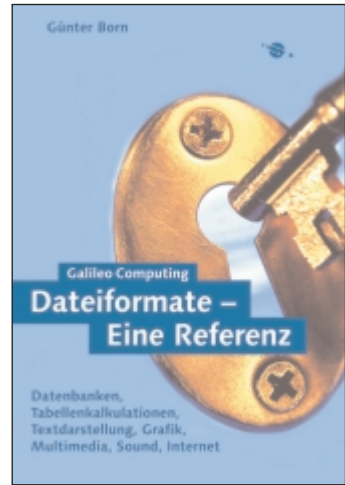


Dieses Kapitel stammt aus dem Buch  
›Dateiformate – Eine Referenz‹  
von Günter Born.

[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.

# 46 Das Windows WAV-Format

Dieses Format stellt eine der Implementierungen des von Microsoft vorgeschlagenen Resource Interchange-Formates (RIFF) dar. Beim RIFF-Format handelt es sich um einen Formatrahmen (Container), in den verschiedene Daten wie Grafik, Sound etc. gepackt werden. Die RIFF-Dateien werden dabei in Anlehnung an das IFF-Format in CHUNKs aufgeteilt. Die CHUNKs bestimmen den Inhalt der Daten.

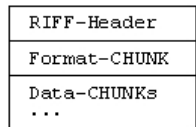


Abbildung 46.1 Struktur einer WAV-Datei

Jeder CHUNK wird durch eine 4-Byte-Kennung eingeleitet. An diese Kennung schließt sich ein 4-Byte-Feld mit der Länge des CHUNKs an. Anschließend folgt eine variable Anzahl an Parametern. Da das RIFF-Format die Daten wortweise organisiert, muß ein CHUNK mit einer ungeraden Anzahl an Bytes bis zur Wortgrenze mit Nullbytes (Padbytes) gefüllt werden. Zu beachten ist aber, daß dieses Füllbyte in der Längenangabe nicht berücksichtigt wird. RIFF-Dateien enthalten die gespeicherten Daten im Intel-Format (little-endian).

## Der WAV-Header

Der Header einer WAV-Datei beginnt mit einem RIFF-CHUNK mit folgendem Aufbau.

Offset	Bytes	Bemerkungen
00H	4	CHUNK-Name 'RIFF'
04H	4	CHUNK-Länge
08H	4	RIFF-Typ 'WAVE'

Tabelle 46.1 Struktur eines WAV-Headers

Die ersten 4 Byte der WAV-Datei enthalten die RIFF-Signatur. Hierbei handelt es sich lediglich um die vier Buchstaben 'RIFF'. Anschließend folgt ein 4-Byte-Feld mit der Länge des Headers. Der Header wird anschließend mit einem weiteren 4-Byte-Feld abgeschlossen, welches den Typ der Datei angibt. Für WAV-Dateien steht hier der Eintrag 'WAVE'. Damit sind die folgenden CHUNKs auf WAV-CHUNKs festgelegt.

## Der FMT-CHUNK

Auf den Header folgt der Format-CHUNK. Dieser CHUNK besitzt folgenden Aufbau.

Offset	Bytes	Bemerkungen
00H	4	CHUNK-Name 'FMT '
04H	4	CHUNK-Länge (10H)
08H	2	Formattyp 0 = Mono 1 = Stereo
0AH	2	Kanalzahl
0CH	4	Sample Rate (in Herz)
10H	4	Bytes pro Sekunde
14H	2	Bytes pro Sample 1 = 8 Bit Mono 2 = 8 Bit Stereo oder 16 Bit Mono 4 = 16 Bit Stereo
16H	2	Bits per Sample

**Tabelle 46.2** Struktur eines FMT-CHUNK

Die ersten 4 Bytes des FMT-CHUNK enthalten die Signatur 'FMT '. Anschließend folgt ein 4-Byte-Feld mit der Länge des CHUNKs. Das folgende Wort definiert das Aufzeichnungsformat (1 = Mono, 2 = Stereo).

Die nächsten Felder definieren die Anzahl der benutzten Kanäle, die Aufzeichnungsrate in Herz und die Anzahl der benötigten Bytes pro Sekunde. Die Anzahl der Byte pro Sample kann zwischen 1 (Mono) und 4 (16 Bit Stereo) schwanken. Das letzte Feld definiert die Zahl der Bits pro Sample (8, 12 oder 16). Über diese Informationen lässt sich der folgende Daten-CHUNK lesen.

## Der DATA-CHUNK

Auf den FMT-CHUNK folgen die eigentlichen Sounddaten. Diese werden in einem DATA-CHUNK gespeichert. Der CHUNK besitzt folgenden Aufbau.

Offset	Bytes	Bemerkungen
00H	4	CHUNK-Name 'data'
04H	4	CHUNK-Länge
08H	n	Datenbereich

**Tabelle 46.3** Struktur eines DATA-CHUNK

Die ersten 4 Bytes des DATA-CHUNK enthalten die Signatur 'data'. Anschließend folgt ein 4-Byte-Feld mit der Längenangabe für den Datenbereich. Die DATA-CHUNKs besitzen bei RIFF-Dateien immer den gleichen Aufbau, d.h., an die Längenangabe schließt

sich der Datenbereich an. Die Interpretation der Daten erfolgt über den vorherigen FMT-CHUNK. Bei den Sounddateien sind die Daten sequentiell abgelegt. Für jedes Sample sind  $n$  Bits zu lesen. Bei einer 8-Bit-Mono-Aufzeichnung sind die Daten Byte für Byte zu lesen und auszugeben. Bei Stereoaufzeichnungen werden die Werte für den linken und rechten Kanal wechselweise gespeichert.

**Anmerkung:** Bei WAV-Dateien tritt in der Regel nur ein DATA-CHUNK auf. Es besteht aber die Möglichkeit, mehrere CHUNKs hintereinander anzuordnen, wobei durchaus mehrere DATA-CHUNKs auftreten dürfen.

