

WAV ファイルフォーマット

全体構造

Windows 標準の音楽, 音声フォーマット WAV 形式は, RIFF というフォーマットの1つです。
RIFF 形式は, [このよう](#)になっていて, 「データの説明」と「データ」が順に記録されています。
実際の WAV ファイルは[このよう](#)になっています。

実際に, ヘッダにどのような内容が書かれているかは, [maxim digital audio: WaveTools](#) の RIFF File Viewer を使って見ることができます。

RIFF フォーマット全体の構造

4 byte	RIFF形式の識別子 'RIFF'
4 byte	ファイルサイズ(byte単位)
4 byte	RIFFの種類を表す識別子 'WAVE'
4 byte	タグ 1 参照
4 byte	データの長さ 1
n byte	データ 1
4 byte	タグ 2 参照
4 byte	データの長さ 2
n byte	データ 2
4 byte	タグ 3 参照
4 byte	データの長さ 3
n byte	データ 3
	(以下同様)

WAV ファイルヘッダ情報

4 byte	R' 'I' 'F' 'F'	RIFFヘッダ	
4 byte	これ以降のファイルサイズ (ファイル サイズ - 8)		
4 byte	W' 'A' 'V' 'E'	WAVEヘッダ	RIFFの種類がWAVEであることを あらわす
4 byte	f' 'm' 't' ' ' (スペースも含む)	fmt チャンク	フォーマットの定義
4 byte	バイト数	fmt チャンクのバイト数	リニアPCM ならば 16(10 00 00 00)
2 byte	フォーマットID	参照	リニアPCM ならば 1(01 00)
2 byte	チャンネル数		モノラル ならば 1(01 00) ステレオ ならば 2(02 00)
4 byte	サンプリングレート	Hz	44.1kHz ならば 44100(44 AC 00 00)
4 byte	データ速度 (Byte/sec)		44.1kHz 16bit ステレオ ならば $44100 \times 2 \times 2 = 176400$ (10 B1 02 00)
2 byte	ブロックサイズ (Byte/sample × チャン ネル数)		16bit ステレオ ならば $2 \times 2 = 4$ (04 00)
2 byte	サンプルあたりのビット数 (bit/sample)	WAV フォーマットでは 8bit か 16bit	16bit ならば 16(10 00)
2 byte	拡張部分のサイズ		リニアPCMならば存在しない
n byte	拡張部分		リニアPCMならば存在しない
4 byte	d' 'a' 't' 'a'	data チャンク 参照	
4 byte	バイト数n	波形データのバイト数	
n byte	波形データ	参照	

'fmt' チャンク構造体

```
#define FormatID 'fmt ' /* chunkID for Format Chunk. NOTE: There is a space at the end of this ID. */
typedef struct {
    ID          chunkID;
    long        chunkSize;
    short       wFormatTag; unsigned short wChannels;
    unsigned long dwSamplesPerSec; unsigned long dwAvgBytesPerSec;
    unsigned short wBlockAlign; unsigned short wBitsPerSample;
    /* Note: there may be additional fields here, depending upon wFormatTag. */
} FormatChunk;
```

'data' チャンク構造体

```
#define DataID 'data' /* chunk ID for data Chunk */
typedef struct {
    ID          chunkID;
    long        chunkSize;
    unsigned char waveformData[];
} DataChunk;
```

チャンクサイズについて

WAVファイルによっては、ヘッダで指定されたチャンクサイズと実際のチャンクサイズが異なっていることがあります。

Windows Platform SDK の [mmioAscend](#) が影響しているようです。

mmioCreateChunk 関数が呼び出されたときに、チャンクサイズが MMCKINFO 構造体の cksize メンバにある値と異なる場合、mmioAscend 関数は、チャンクから退出する前にファイルのチャンクサイズを修正します。

チャンクサイズが奇数の場合、mmioAscend 関数は、チャンクの終わりに NULL の埋め込みバイトを書き込みます。

タグ情報

WAV フォーマットで使われるタグとして、[このような](#)ものがあります。
最低限 "fmt " と "data" があれば良いようです。

フォーマットIDとしては、[このような](#)ものがあります。
このほかにもたくさん存在するようですが、無圧縮、リニアPCM は「1」です。
「0x0001」の場合には多少異なる部分があるので注意("fmt " の拡張部分が存在しない)。

ADPCM = Adaptive Delta Pulse Coded Modulation

波形を標本化するとき隣り合うサンプル値の差分を量子化する手法をDPCM(differential pulse code modulation)というが、このとき量子化のステップ・サイズを波形の振幅によって変える方式。(「情報・通信新語辞典98年版」)

例: 44.1kHz, 16bit, ステレオ, リニアPCM の場合のヘッダ内容

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	
	00	52	49	46	46	--	--	--	--	57	41	56	45	66	6D	74	20
(内容)	R	I	F	F	ファイルサイズ - 8				W	A	V	E	f	m	t		
	01	10	00	00	00	01	00	02	00	44	AC	00	00	10	B1	02	00
(内容)	16				1		2		44100				176400				
	02	04	00	10	00	64	61	74	61	--	--	--	--	--	--	--	--
(内容)	4	16		d	a	t	a	データサイズ				以後データ					

WAVEで使われる主なタグ

4 byte	"fmt "	フォーマット定義 最初のタグ 必須 4文字目の ' '(スペース)も含まれるので注意
4 byte	"fact"	全サンプル数 波形データの前に存在する 無くても良い
4 byte	"data"	波形データ 必須
4 byte	"LIST"	コメントや著作権情報 無くても良い

PCM の種類		
0x0000	unknown	
0x0001	PCM	Windows 標準サポート
0x0002	MS ADPCM	Windows 標準サポート
0x0005	IBM CSVD	
0x0006	A-Law	Windows 標準サポート
0x0007	μ -Law	Windows 標準サポート
0x0010	OKI ADPCM	
0x0011	IMA/DVI ADPCM	Windows 標準サポート
0x0012	MediaSpace ADPCM	
0x0013	Sierra ADPCM	
0x0014	ADPCM (G.723)	
0x0015	DIGISTD	
0x0016	DIGIFIX	
0x0020	YAMAHA ADPCM	
0x0021	SONARC	
0x0022	TrueSpeech	Windows 標準サポート
0x0023	Echo Speech1	
0x0024	AF36 (Audiofile)	
0x0025	Apix	
0x0026	AF10 (Audiofile)	
0x0030	AC2 (Dolby)	
0x0031	GSM 6.10	Windows 標準サポート
0x0033	ANTEX ADPCM	
0x0034	VQLPC (Control Resources)	
0x0035	DIGIREAL	
0x0036	DIGIADPCM	
0x0037	CR10 (Control Resources)	
0x0040	ADPCM (G.721)	
0x0101	IBM μ -LAW	
0x0102	IBM A-LAW	
0x0103	IBM ADPCM	
0x0200	Creative Labs ADPCM	
0x0300	FM TOWNS	
0x1000	Olivetti GSM	
0x1001	Olivetti ADPCM	
0x1002	Olivetti CELP	
0x1003	Olivetti SBC	
0x1004	Olivetti OPR	

データ形式

"data" チャンクのデータ内容は、8bit または 16bit PCM データが時間順に記録されています。

- ステレオならば L R L R L R ... の順番
- 数値の並べ型は Intel バイトオーダー ([バイトオーダーの変換方法について](#))

- ビット数は 8bit と 16bit
 - 8bit ならば符号無し unsigned (0 ~ 255, 無音は 128)
 - 16bit ならば符号付き signed (-32768 ~ +32767, 無音は 0)

その他

データ読み出し手順の例

1. RIFFとWAVEをチェック
2. チャンクの種類とチャンクデータサイズをチェック
 1. チャンクが "fmt " なら, フォーマット情報を読み込む
 2. チャンクが "fact" なら, サンプル数を読み込む
 3. チャンクが "data" なら, PCM データをブロック単位で読み込む

ネットニュースに投稿された記事

- [WAVEファイルの読み込み、書き換え。](#)
- [WAV データの波形表示について](#)

拙作プログラム

ASCII テキストデータを WAVE 形式に変換します

- [txt2wav.c](#) (2000.11.13)

WAVE 形式からテキストデータに変換するには [WAVE file format](#) の dumpwave.c があります。
(txt2wav.c は dumpwave.c を参考にさせていただきました)

ステレオ WAVE データを右チャンネル・左チャンネルに分割します

- [separate.c](#) (2000.12.15)

WAVE 形式ファイルを連結(時間長, チャンネル)します

- [combine.c](#) (2000.12.15)

WAVE 形式ファイルのヘッダ情報を出力します

- [wavhdchk.c](#) (2001.05.03)

関連ドキュメント

[The Programmer's File Format Collection](#)

- [mpidata](#) Multimedia Programming Interface and Data Specifications [IBM/ Microsoft]
- [wave](#) WAVE File Format [Microsoft]
- [wavecomp](#) WAVE formats and compression types [Microsoft]

関連リンク

- [.wav フォーマットについて](#)
- [WAVE file format](#)
- [Windows wav ファイルについて](#)
- [COMP630: WAV file format](#)

wav フォーマットファイル用の信号処理ソフト

- [WaveSpectra](#) スペクトルアナライザ
- [Spectrogram](#) スペクトログラム
- [WaveTools](#) (16bit アプリケーション) ([maxim digital audio](#))

他のフォーマットについて

WAV フォーマット以外の AIFF(SGI, Machintosh の標準) や AU(SUN の標準) などのフォーマットも無圧縮, リニアPCM のフォーマットがあるので, ヘッド情報と [バイトオーダー](#) を変換すれば, 相互に変換可能です。

AIFF フォーマットについて

- [aiff.txt](#)

AU フォーマットについて

- [au.txt](#)
- [wavフォーマットとauフォーマットの説明](#)

その他

- [Audio File Formats FAQ](#) (英文)
- [AudioFormats](#) (英文)
- [Audio.Formats.part1](#) (英文)
- [Audio.Formats.part2](#) (英文)
- [File Formats](#) (日本語)
- [SoX: Sound eXchange](#) 音声フォーマット変換ソフトウェア
SoX は上記サイトでソースコードが公開されているので参考になると思います
- [spwave](#) 音声ファイルエディタ
spwave もソースコードが公開されています

HOME

kondo@kk.ij4u.or.jp