

# ***FX2WAV***

*- PC-FX/ADPCM to WAVE/PCM コンバータ -*  
***Ver.0.08***

**オンラインマニュアル**

(c)1998 -1999 こうFX

# 目次

1. 概要	8
2. 同梱ファイル	8
3. 動作環境	9
4. 製作環境及び動作確認環境	9
5. インストール方法	10
6. 使用方法	11
6.1. 引数表記法の解説	11
6.1.1. 整数表記	11
6.1.2. 小数表記	11
6.1.3. アドレス表記	11
6.1.4. MAKELST 形式	12
6.2. FX2WAV	13
6.2.1. オプション	14
6.2.1.1. 入力ファイルフォーマットの指定 (/FMT=xxxx)	14
ヘッダ自動解析 (xxxx=AUTO)	14
ADPCM 後置・フレームサイズ固定形式 (xxxx=LAST[ffff[,aaaa[,ssss]])	14
アルバレアの乙女動画形式 (仮称) (xxxx=ARBULEA)	15
虚空漂流ニルゲンツ動画形式 (仮称) (xxxx=NIRGENDS)	15
ヘッダ無し ADPCM 形式 (xxxx=ADP)	15
ヘッダ無し PCM 形式 (xxxx=PCM[n][,aaaa])	15
6.2.1.2. オーバーフロー処理 (/Ovf=xxxx)	16
直接出力 (xxxx=Straight)	16
切り捨て (xxxx=Limit)	16
6.2.1.3. 入力サンプリング周波数の指定 (/Fn)	16
6.2.1.4. 出力サンプリング周波数の指定 (/Ofn)	16
6.2.1.5. ノイズの重畳 (/noise=aaaa[,bbbb][,cccc][,x])	16
6.2.1.6. 飛び石補間 (/SSInp[=Off])	16
6.2.1.7. 補間方法の指定 (/INP=xxxx)	17

直線補間(xxxx=Linear)	17
2次関数による補間(xxxx=Parabola)	17
ベーススプラインによる補間(xxxx=BaseSpline)	17
三角関数による補間(xxxx=Trigonometric[n])	17
高次関数への最小二乗フィットによる補間(xxxx=LSM[[n],[m])	17
<b>6.2.1.8. 平滑化法の指定(/SM=xxxx)</b>	<b>18</b>
平滑化しない(xxxx=Off)	18
単純移動平均法(xxxx=Simple[[n]])	18
重み関数を三角波とした移動平均法(xxxx=Triangle[n][,y])	18
重み関数を2次多項式とした移動平均法(xxxx=Parabola[n][,y])	18
重み関数を余弦波とした移動平均法(xxxx=Cosine[n][,y])	19
重み関数を指数関数とした移動平均法(xxxx=Exponential[n][,[a][,1]])	19
2次多項式適合移動平均法(xxxx=Parabola[n],,1)	20
適応化平滑化法(xxxx=Adaptive[n][,y])	20
位相シフト平滑化(仮称)(xxxx=PhaseShift[l][,m])	20
<b>6.2.1.9. フィルタの指定(/Fil=xxxx)</b>	<b>21</b>
フィルタをかけない(xxxx=Off)	21
ローパスフィルタをかける(xxxx=LP[m,[n]])	21
ハイパスフィルタをかける(xxxx=HP[m,[n]])	21
バンドパスフィルタをかける(xxxx=BP[k,l[,m,n]])	22
グライコモード(xxxx=GEx1,y1[,x2,y2[,x3,y3...])	22
拡張グライコモード(xxxx=AGE[L R]x1,y1,a1[,x2,y2,a2[,x3,y3,a3...])	23
<b>6.2.1.10. 出力ファイルの量子化ビット数の指定</b>	<b>24</b>
16ビット出力(/16)	24
8ビット出力(/8)	24
<b>6.2.1.11. ADPCMの解析順序の指定</b>	<b>24</b>
上位4bit 下位4bitの順で解析(/HF)	24
下位4bit 上位4bitの順で解析(/LF)	24
<b>6.2.1.12. ループ回数の指定(/LPn[,init])</b>	<b>24</b>
<b>6.2.1.13. チャンネル数の指定</b>	<b>25</b>
モノラル入力、モノラル出力(/M)	25
ステレオ入力、ステレオ出力(/S)	25
ステレオ入力、両チャンネル加算モノラル出力(/SA)	25
ステレオ入力、左チャンネルモノラル出力(/SL)	25
ステレオ入力、右チャンネルモノラル出力(/SR)	25
疑似ステレオ出力(/SS=[l][,[m][,[n][,[x1,y1][,x2,y2][,...]]]])	25
<b>6.2.1.14. ブロックサイズの指定(/BSn)</b>	<b>26</b>
<b>6.2.1.15. オフセット(/On)</b>	<b>26</b>
<b>6.2.1.16. 変換開始位置・終了位置の指定</b>	<b>26</b>
変換開始位置の指定(/TSn)	26
変換終了位置の指定(/TEn)	26
<b>6.2.1.17. 正規化の指定</b>	<b>28</b>

両チャンネルから最大値を取得(/MB)	28
各チャンネルごとに最大値を取得(/ME)	28
正規化しない(/MM)	28
<b>6.2.1.18. ゼロ補正の指定</b>	<b>28</b>
ゼロ補正をしない(/NZA)	28
ゼロ補正をする(/ZA)	28
<b>6.2.1.19. 音量の指定</b>	<b>28</b>
増幅率による指定(/Xn)	28
音量(デシベル単位)による指定(/dBn)	28
<b>6.2.1.20. データの反転</b>	<b>28</b>
左チャンネル反転(/L-)	28
右チャンネル反転(/R-)	28
<b>6.2.1.21. 左右の音量バランス(/Bn)</b>	<b>29</b>
<b>6.2.1.22. 無音部の指定</b>	<b>29</b>
先頭無音部時間(/StNSn)	29
終端無音部時間(/EdNSn)	29
<b>6.2.1.23. フェードイン・フェードアウトの指定</b>	<b>29</b>
フェードインの時間(/FIn)	29
フェードアウトの時間(/FOn)	29
<b>6.2.1.24. 上書き確認の指定</b>	<b>29</b>
上書き確認をする(/CF)	29
上書き確認をしない(/NC)	29
<b>6.2.1.25. 変換処理中の画面表示の指定</b>	<b>29</b>
すべて表示する(/V)	29
データの表示のみする(/D0)	29
経過表示のみする(/P0)	29
なにも表示しない(/NV)	29
<b>6.2.1.26. ログファイルの指定</b>	<b>30</b>
ログファイルを作成する(/Log)	30
ログファイルを作成しない(/Log=Off)	30
<b>6.2.1.27. FFT 演算精度の指定(/FFT=n)</b>	<b>30</b>
<b>6.2.1.28. チャンネル合成(/MIXn)</b>	<b>30</b>
<b>6.2.1.29. 著作権情報の付加</b>	<b>31</b>
アーカイブの場所(/IARL=xxxx)	31
アーティスト(/IART=xxxx)	31
コミッション(/ICMS=xxxx)	31
コメント(/ICMT=xxxx)	31
著作権(/ICOP=xxxx)	31
作成日(/ICRD=xxxx)	31
クロップ(/ICRP=xxxx)	31
大きさ(/IDIM=xxxx)	31
ドット/インチ(/IDPI=xxxx)	31

エンジニア ( / IENG=xxxx )	31
ジャンル ( / IGNR=xxxx )	31
キーワード ( / IKEY=xxxx )	31
明るさ ( / ILGT=xxxx )	31
メディア ( / IMED=xxxx )	31
名前 ( / INAM=xxxx )	31
パレット設定 ( / IPLT=xxxx )	31
製品 ( / IPRD=xxxx )	31
タイトル ( / ISBJ=xxxx )	31
ソフトウェア ( / ISFT=xxxx )	31
鮮明度 ( / ISHP=xxxx )	31
ソース ( / ISRC=xxxx )	31
ソースの形式 ( / ISRF=xxxx )	31
技術者 ( / ITCH=xxxx )	31
<b>6.2.2. 注意事項</b>	<b>32</b>
<b>6.3. MAKELST ( オマケ )</b>	<b>33</b>
<b>6.3.1. MAKELST フォーマット</b>	<b>33</b>
<b>6.3.1.1. データ部</b>	<b>33</b>
<b>6.3.1.2. 制御部</b>	<b>34</b>
トラックの指定 (*Track=n)	34
オフセットの指定 (*Offset=n)	34
開始アドレスの丸め方の指定 (*StartPoint=xxx)	34
終了アドレスの丸め方 (*EndPoint=xxx)	34
サンプリング周波数の指定 (*Frequency=n)	34
出力サンプリング周波数の指定 (*OutputFrequency=n)	34
チャンネル数の指定 (*IOStyle=xxx)	35
ブロックサイズの指定 (*BlockSize=n)	35
ADPCM 解析順序の指定 (*ReadMode=xxx)	35
サンプリング周波数変換時の補間方法の指定 (*InpMethod=xxx)	35
FX2WAV ファイル内変換位置の指定	35
ループ回数の指定 (*Loop=n)	35
正規化の方法 (*MAXSeek=xxx)	36
ゼロ補正の指定 (*ZeroAdjust=xxx)	36
データ反転の指定	36
音量の指定 (*Volume=x[dB])	36
バランスの指定 (*Balance=x)	36
フェードイン・フェードアウト時間	36
(FX2WAV) 上書き確認の指定 (*WriteConfirm=xxx)	37
(FX2WAV) 情報表示の指定 (*Information=xxx)	37
FX2WAV にそのまま渡すオプション文字列の指定 (*Inline=xxx)	37
<b>6.3.2. 注意事項</b>	<b>38</b>

7. 動画フォーマット抽出確認状況 .....	39
7.1.MIX フォーマット .....	39
7.2.RTF フォーマット .....	39
7.3.ARS フォーマット .....	40
7.3.1.32KHz ステレオデータ .....	40
7.3.2.32KHz モノラルデータ .....	40
7.4.ADPCM 後置フレームサイズ固定形式 (仮称) .	40
7.5. アルバレアの乙女動画形式 (仮称) .....	41
7.6. 虚空漂流ニルゲンツ動画形式 (仮称) .....	41
8. バージョンアップ履歴 .....	42
8.1.Ver.0.01(1998/6/6) .....	42
8.2.Ver.0.02(1998/6/8) .....	42
8.2.1.FX2WAV .....	42
8.2.2.MAKELST .....	43
8.3.Ver.0.03(1998/6/21) .....	44
8.3.1.FX2WAV .....	44
8.3.2.MAKELST .....	46
8.4.Ver.0.04 (1998/7/3) .....	47
8.4.1.FX2WAV .....	47
8.4.2.MAKELST .....	49
8.5.Ver.0.05(1998/7/14) .....	50
8.5.1.FX2WAV .....	50
8.5.2.MAKELST .....	50
8.6.Ver.0.06(1998/11/16) .....	51
8.6.1.FX2WAV .....	51

8.6.2.MAKELST .....	54
8.7.Ver.0.07(1998/12/28) .....	55
8.7.1.FX2WAV .....	55
8.7.2.MAKELST .....	58
8.8.Ver.0.08(1999/3/7) .....	60
8.8.1.FX2WAV .....	60
8.8.2.MAKELST .....	61
9. 参考文献 .....	62
10. 免責等 .....	63
索引 .....	64

# 1. 概要

PC-FX/PC エンジンのヘッダ無し ADPCM データから WAVE ファイルに変換するツールです。  
また、一部の動画フォーマット (MIX/RTF 形式、ARS 形式、アルバレアの乙女、虚空漂流ニルゲンツ、ミラークルム) からの音声抽出にも対応しています。  
出力はリニア PCM (サンプリング周波数任意、量子化ビット数 8bit もしくは 16bit) のみとなります。

なお、FX の CD-ROM からのデータ抽出は、このツールではサポートしておりません。別途、CDREAD.EXE<sup>\*1</sup> などを用いて抜き出してください。

## 2. 同梱ファイル

• README.TXT	...	最初に読むファイル
• FX2WAV.EXE	...	FX2WAV 本体
• FX2WAV.CFG	...	ユーザデフォルト設定用コンフィグファイル
• FX2WAV.PDF	...	このファイル。マニュアルですね。
• MAKELST.EXE	...	補助ツール MAKELST (おまけ / CDREAD.EXE と組み合わせる場合に使用)
• SAMPLE.LZH	...	PC エンジン、PC-FX からの MAKELST ファイルのサンプル集
AFX2WAV.BAT	...	FX2WAV 一括処理バッチファイル(インダイレクトファイル用)
AMAKELST.BAT	...	MAKELST 一括処理バッチファイル
ANGE2.TXT	...	『アンジェリーク Special2』用 MAKELST ファイル (音楽)
ANGE2X.TXT	...	同疑似ステレオ化バージョン
ANGEREQ.TXT	...	『アンジェリーク 天空の鎮魂歌』用 MAKELST ファイル (音楽)
ANGEREQX.TXT	...	同疑似ステレオ化バージョン
ARBULEA.TXT	...	『アルバレアの乙女』用 MAKELST ファイル (音楽)
ARBULEAX.TXT	...	同疑似ステレオ化バージョン
BB.TXT	...	『ブルーブレイカー』用 MAKELST ファイル (音楽)
BBX.TXT	...	同疑似ステレオ化バージョン
EMEDRA.TXT	...	『エメラルドドラゴン』用 MAKELST ファイル (音声)
FKS.TXT	...	『ファースト Kiss 物語』用 MAKELST ファイル (音楽)
FKSX.TXT	...	同疑似ステレオ化バージョン
LANG.TXT	...	『デア ラングリッサー F X』用 MAKELST ファイル (音楽)
LANGX.TXT	...	同疑似ステレオ化バージョン
LINDA3.TXT	...	『Linda3』用 MAKELST ファイル (音声)
MATOI.TXT	...	『ファイアーウーマン 纏組』用 MAKELST ファイル (音楽)
MATOIX.TXT	...	同疑似ステレオ化バージョン
MIRA.TXT	...	『ミラークルム』用 MAKELST ファイル (音楽)
MIRAX.TXT	...	同位相シフト平滑化 (仮称) バージョン
NIRGENDS.TXT	...	『虚空漂流ニルゲンツ』用 MAKELST ファイル (音楽)
NIRGENDX.TXT	...	同疑似ステレオ化バージョン
PEDO?.TXT	...	『プライベート・アイ・ドル』用 MAKELST ファイル (音声)
PEDREAD.BAT	...	同 CDREAD 一括処理バッチファイル
TENGA12.TXT	...	『天外魔境 II』用 MAKELST ファイル (音声)
TOKIO?.TXT	...	『ときめきメモリアル』用 MAKELST ファイル (音声)
TOKIREAD.BAT	...	同 CDREAD 一括処理バッチファイル
TOKITKN.TXT	...	『ときめきメモリアル体験版』用 MAKELST ファイル (音声)

---

<sup>\*1</sup>CDREAD.EXE は W.M. 氏の論理フォーマットによらず、CD-ROM のデータを読むことができる便利なツールです。有用なソフトを開発してくださった W.M. 氏には感謝します。なお、ダウンロード先が分からないという問い合わせがありましたので、一応参考までに書いておきますと、Vector のホームページでダウンロード出来ます。MS-DOS/ハードウェア/CD-ROM 関係と辿って行けば、入手出来ます。

### 3. 動作環境

Windows 98 DOS 窓専用 (爆)

プログラミングスタイルが変わったわけでもないのに、多分Windows 95でも動くと思いますが、なにせ確認が取れないもので...

なお、FXのCDからの読み込みにCDREAD.EXEをお使いになる場合、DOS窓では上手くいかない場合も多いようなので、基本的にDOSモードで起動するようにしてください。

### 4. 製作環境及び動作確認環境

#### 開発環境 & 動作確認環境 1

本体 : NEC / PC-9821V13S5RC  
CPU : AMD K6-2/333MHz (下駄 : MELCO HK6-MX233-Nの下駄部)  
メモリ : 96MB  
ドライブ構成 : A:(FAT32) EIDE P-MASTER: HDD 4.3GB (MELCO/DBI-N4300)  
B:(FAT32) EIDE S-MASTER: HDD 2.5GB (MELCO/DBI-N2500)  
C:(FAT32) SCSI2-ID0: HDD 6.4GB (MELCO/DSC-UE6.4G)  
D:(FAT32) SCSI2-ID1: HDD 4.0GB (Logitec/SHD-U4000)  
E: FDD 3.5" -2HD  
F: SCSI2-ID2: MO 640MB/OW (I-O DATA/MOF-640)  
Q: SCSI2-ID5: CD-ROM 24xR (I-O DATA/CDV-PX24)  
R: SCSI2-ID6: CD-R/RW 2xW/6xR (I-O DATA/CDRW-RX26)  
拡張スロット : PCIバス: グラフィックカード (I-O DATA/GA-VDB16/PCI)  
Cバス1: SCSIカード (I-O DATA/SC-98IIIP)  
Cバス2: 開発環境兼ゲームカード (NEC Home Electronics/PC-FXGA)  
モニタ : MITSUBISHI/RD17GZR (1600x1200pixels/32bpp/65Hz)  
COMポート : COM1: MIDI音源 (YAMAHA/MU100)  
COM2: モデム (MOTOROLA/ModemSURFR 56K External)  
LPTポート : LPT1: プリンタ (EPSON/PM-770C)  
OS : Microsoft Windows 98  
Internet Explorer 5.00.0910.1309 ( 版)  
DirectX 6.0  
開発言語 : Microsoft Visual C++ 6.0 (Visual Studio 6.0 Enterprise)  
ドキュメント作成 : Microsoft Word 97 (Office 97 Standard)  
Microsoft 数式エディタ Ver.3.0 (Office 97 Standard)  
川村久雄氏作 Plots32 Version 1.35  
Adobe Acrobat Distiller Ver.3.02 (Acrobat 3.0J)  
Adobe PageMaker 6.53J  
Village Center WZ Editor Ver.3.00

## 動作確認環境 2

本体： IBM / ThinkPad600 (2645-53J)  
CPU： モバイルPentiumII 266MHz  
メモリ： 64MB  
ドライブ構成： A: FDD 3.5" -2HD  
C: (FAT32) ATA / hda1(3.0GB)  
(参考) ハードディスクのパーティション構成  
総容量：6.4GB(東芝製)  
hda1: FAT32 (3.0GB) / Windows98  
hda2: Be FS (1.0GB) / BeOS Release 4J  
hda3: Linux native (2.0GB) / TurboLinux 3.0  
hda4: Linux swap (200MB)  
Q: ATAPI / CD-ROM 24xR (TOSHIBA/XM-1702B)  
OS： Microsoft Windows 98  
DirectX 6.0

## 5. インストール方法

任意のフォルダに解凍して使ってください。

(あり得ないとは思いますが...(^\_^;) 使用頻度が高いようでしたら、パスの通ったディレクトリ(例えば、%WinDir%\commandとか)に入れてお使いください。

なお、おまけのMAKELSTで掃き出されるBATファイル、LSTファイルは、パスが通ったディレクトリか、カレントディレクトリにFX2WAVがあることを想定していますので、ご注意ください。

## 6. 使用方法

### 6.1. 引数表記法の解説

FX2WAV 及び MAKELST ではオプションの種類により、数値を与える場合の引数の形式が異なりますので、それについて簡単にではありますが解説します。

#### 6.1.1. 整数表記

整数のみが扱える場合をこう呼びます。平滑化処理の平滑化する際のデータ数などがこの形式にあたります。

#### 6.1.2. 小数表記

整数の他、小数点を含む数字も扱える場合をこう呼びます。バランスや音量などをこの形式で与えます。

#### 6.1.3. アドレス表記

以下の2つの書式が扱える場合、その引数形式を、便宜上、アドレス表記形式と呼ぶことにします。

0xhhhhhhhh	... バイト単位、16進整数で指定します (C言語の16進数の形式で与えてください)。
n	... バイト単位、10進整数で指定します。

## 6.1.4. MAKELST 形式

以下の7つの書式が扱える場合、その引数形式を、便宜上、MAKELST形式と呼ぶことにします。主にCD-ROM内の論理アドレスを指定する場合に使用しています。

- 0xhhhhhhhh ... バイト単位、16進整数で指定します  
n ... バイト単位、10進整数で指定します。
- B0xhhhhhhhh ... ブロック単位、16進整数で指定します。  
Bn ... ブロック単位、10進整数で指定します。  
これらは、ステレオ時のブロックサイズ(『6.1.2.14 ブロックサイズの指定』参照)のチャンネル数倍を基本単位とします。
- S0xhhhhhhhh ... セクタ単位、16進整数で指定します。  
Sn ... セクタ単位、10進整数で指定します。  
FXのCD-ROMのセクタあたりのバイト数2048バイトを基本単位とします。
- t[[hh:]mm:]ss ... 時:分:秒という時間を基本単位とします。  
mm:ssと書かれていた場合は、分:秒と判断し、ssと書かれていた場合は、秒と判断します。時と分は整数のみで、秒数だけは小数点以下の指定も出来ます。

なお、ここで指定された時間の秒数と周波数、チャンネル数を掛け合わせて、2で割ったものを(FXのADPCMの1バイト=2サンプルより)求めるアドレスとします。ですので、周波数やチャンネル数を変えて指定する場合は十分ご注意ください。

表1 ARS形式のサンプリング周波数

ヘッダ情報	サンプリング周波数
32k	31560Hz
16k	15780Hz
8k	7890Hz
4k	3945Hz

表2 ARS形式、MIX/RTF形式、WAV/ADP形式の出力チャンネル数

パラメータ指定	ヘッダ情報	実際の出力
/m	ステレオ	両チャンネル加算
	モノラル	モノラル
/s	ステレオ	ステレオ
	モノラル	モノラル
/sa	ステレオ	両チャンネル加算
	モノラル	モノラル
/sl	ステレオ	左チャンネルのみ
	モノラル	モノラル
/sr	ステレオ	右チャンネルのみ
	モノラル	モノラル

## 6.2.FX2WAV

DOS プロンプトを起動して、以下の書式に沿った形で実行してください。

書式 : FX2WAV [オプション] InputFile [OutputFile]  
もしくは : FX2WAV [追加オプション] @IndFile

Input : 入力ファイル

Output : 出力ファイル (WAVE ファイル)

省略した場合は、入力ファイルの拡張子を wav に変更したものを出力ファイルとします。

また、Windows95 の長いファイル名にも対応しているようです (んな、人ごとみたいに... (^; )。半角スペース入りのファイル名を指定する場合は、"(ダブルクォーテーションマーク) で囲んで指定してください。

IndFile : インダイレクトファイル

コマンドラインで渡すパラメータをそのまま記述したテキストファイルです。複数の変換をいっぺんに行いたい場合などにお薦めです。ひとつの変換ごとに改行を入れて指定してください。

なお、現状では制限がありまして、いっぺんに変換できるファイル数は **2048 ファイルまで** となっています (それ以上記述すると一般保護例外で死んでしまいます (^; )。明らかな設計ミスではありますが、これだけの数をいっぺんに変換する必要性はそうざらには無いのではないかと思います。このままとさせていただきます。フルボイスのゲームからすべての音声を抜き出したい場合など、非常に多くのファイルが作成される場合は、1 ファイル 2048 行以内におさめた複数のインダイレクトファイルを作成するようにしてください。

## 6.2.1. オプション

### 6.2.1.1. 入力ファイルフォーマットの指定(/FMT=xxxx)

#### ヘッダ自動解析 (xxxx=AUTO)

ヘッダのついている音声ファイル、あるいは、一部の動画フォーマットからヘッダを自動解析します。ヘッダが見つからなかった場合、あるいは未対応のファイル形式だった場合は、ヘッダ無しADPCM形式として処理されます。Ver.0.06よりこのAUTOがデフォルトとなりました。

現在の対応フォーマットは

ARS形式 (HuneXのゲームでよく使われています)

MIX形式 (PC-FXGAの標準動画フォーマット)

RTF形式 (バトルヒートなどで使われています)

ADP形式 (PC-FXGAの標準ADPCMフォーマット)

WAV形式 (ただし、8ビットもしくは16ビットのPCMのみ)

となります。ヘッダの位置が1バイトでもずれていると抽出出来ませんので、ご注意ください。

なお、抽出処理の都合上、これらのフォーマットが入力された場合は、/TS、/TE、/BSの各オプションの指定が無効となります。

/LPによるループ処理は元々FXのADPCMのために用意されたものですので、WAV形式で指定しても無効となります。

MIX形式、ADP形式、WAV形式の場合、サンプリング周波数をヘッダ情報から取得しますので、/Fオプションの指定は無効となります。

ARS形式の場合も、/Fオプションの指定は無効となり、ヘッダに書かれてあるサンプリング周波数から、入力サンプリング周波数を、表1のような値としてみなします。

ADP形式、ARS形式、MIX形式、RTF形式、WAV形式の場合、チャンネル数の指定は、表2のように自動的に変更されますので、注意してください。

#### ADPCM後置・フレームサイズ固定形式(xxxx=LAST[ffff[,aaaa[,ssss]]])

1フレームのサイズをffffバイト固定とし、画像データを前に、ADPCMデータを後ろに配置したようなものを便宜上、こう呼ぶこととし、そのフォーマットからADPCMを抽出し、変換します。

ADPCMの領域には、aaaa[バイト]割かれるものとし、そのうちデータとしてssss[バイト]抽出するものとします。

アドレス表記で指定してください。

デフォルトは、ffff = 0x4400、aaaa = 0x600、ssss = 0x41cとなります。

実際のゲームでは『ミラクルム』で使われています。抽出処理の都合上、/TS、/TEオプションの指定は無効となります。

### アルバレアの乙女動画形式(仮称)(xxxx=ARBULEA)

『アルバレアの乙女』の動画から抽出するための専用オプションです。  
抽出処理の都合上、/TS, /TE オプションの指定は無効となります。

### 虚空漂流ニルゲンツ動画形式(仮称)(xxxx=NIRGENDS)

『虚空漂流ニルゲンツ』の動画から抽出するための専用オプションです。  
抽出処理の都合上、/TS, /TE オプションの指定は無効となります。

### ヘッダ無しADPCM形式(xxxx=ADP)

普段は、/FMT=AUTOで問題ないかと思われませんが、ヘッダ無しADPCM形式を明示的に指定する場合に用います。

### ヘッダ無しPCM形式(xxxx=PCM[n][,aaaa])

ヘッダ無しPCM形式です。

nは量子化ビット数を示し、8を指定した場合は量子化ビット数8ビットとして、16を指定した場合は量子化ビット数16ビットとして処理します。

aaaaは符号の指定で、complementを指定した場合はデータを2の補数表現、signedを指定した場合は符号付き絶対値、unsignedを指定した場合は符号なし整数としてそれぞれ処理します(表3、表4参照)。

量子化ビット数のデフォルトは16ビットで、符号のデフォルトは、16ビット時がcomplement、8ビット時がunsignedとなります。

また、量子化ビット数16ビットを指定した場合は、/lf, /hfによって、リトルエンディアン(/hf)、ビッグエンディアン(/lf)の指定も可能となります。

それから、ステレオデータを指定する場合は、/BSnオプションで1チャンネルあたりのブロックサイズを指定してください。例えば、16ビットのWAVファイルと同一のフォーマットの場合は、/BS2となります。

表3 PCMデータの値(16ビット)

2の補数表現	符号なし整数	符号付き絶対値	出力
0x7fff ⋮	0xffff ⋮	0x7fff ⋮	+32767 ⋮
0x0000	0x8000	0x0000	0
0xffff ⋮	0x7fff ⋮	0x8000 ⋮	-1 ⋮
0x8000	0x0000	0xffff	-32768

表4 PCMデータの値(8ビット)

2の補数表現	符号なし整数	符号付き絶対値	出力 <sup>*1</sup>
0x7f ⋮	0xff ⋮	0x7f ⋮	+255 ⋮
0x00	0x80	0x00	0
0xff ⋮	0x7f ⋮	0x80 ⋮	-1 ⋮
0x80	0x00	0xff	-256

\*1) 内部処理の都合で、実際にはこの値を256倍したものとなります。

## 6.2.1.2. オーバーフロー処理 (/Ovf=xxxx)

### 直接出力(xxxx=Straight)

入力レベルがオーバーフローしたとき、その値をそのまま出力します(デフォルト)。

### 切り捨て(xxxx=Limit)

入力レベルがオーバーフローしたとき、オーバーフローした部分を切り捨てます。

## 6.2.1.3. 入力サンプリング周波数の指定 (/Fn)

サンプリング周波数 n Hz の指定をします。

n は整数で指定してください。

デフォルトは 31560Hz となります。

## 6.2.1.4. 出力サンプリング周波数の指定(/Ofn)

出力サンプリング周波数 nHz の指定をします。

n は整数で指定してください。

デフォルトは入力ファイルのサンプリング周波数と同一となります。

## 6.2.1.5. ノイズの重畳(/noise=aaaa[,bbbb][,cccc][,x])

波形にノイズを重畳します。

aaaa に “ rand ” と指定したら、乱数によるノイズを、“ white ” と指定したら、疑似ホワイトノイズ(ただし、実際にはM系列信号そのままです)を重畳します。aaaa に “ Off ” を指定した場合は、ノイズ重畳処理を行いません。デフォルトは “ Off ” となります。

bbbb に “ add ” と指定したら、上記ノイズを加算により重畳し、“ mul ” と指定したら、上記ノイズを乗算により重畳します。デフォルトは加算となります。

cccc に “ max ” と指定したら、想定している WAV ファイルの限界値...つまり 32768 を、“ dif ” と指定したら一つ前のデータとの差分を、“ inst ” と指定したら、そのデータの瞬時値をベースに重畳を行います。つまり、WAV ファイルの限界値を指定した場合は、一定振幅のノイズを重畳し、差分を指定した場合は、前のデータとの差が大きいかほど大きなレベルのノイズを重畳し、瞬時値を指定した場合は、瞬時値が大きいかほど大きなレベルのノイズを重畳することになります。なお、デフォルトは、前のデータとの差分となります。

x はノイズ重畳のレベルとなり、0 ~ 100 のパーセンテージ(小数表記)で指定します。上記ノイズ適応のベース値にこの値(の 1/100)を掛けたものが実際のノイズレベルとなります。デフォルトは 10% です。

## 6.2.1.6. 飛び石補間(/SSInp[=Off])

あるサンプル点の前後2点のデータと過去に求めた一つ前のサンプル点での1次微係数、そして、一つ後と2つ後のサンプル点を結ぶ直線の傾きを一つ後のサンプル点での1次微係数と見立てて、この4つのパラメータから3次曲線を求めて、そのサンプル点でのデータを置き換えます。

/SSInp=Off と指定した場合は、飛び石補間を行いません。デフォルトは /SSInp=Off となります。

## 6.2.1.7. 補間方法の指定 (/INP=xxxx)

### 直線補間(xxxx=Linear)

サンプリング周波数の変換に直線補間を適用します(デフォルト)。

直線補間と言って良いのかわかりませんが、とにかく、求めたいデータの近傍2点のデータ間を結ぶ直線上の点を拾います。

なお、互換性のため、Ver.0.05以前の/ISLも残してあります。

### 2次関数による補間(xxxx=Parabola)

サンプリング周波数の変換に2次関数による補間を適用します。こちらは、近傍3点を通る2次曲線から求めます。

なお、互換性のため、Ver.0.05以前の/ITDも残してあります。

### ベーススプラインによる補間(xxxx=BaseSpline)

サンプリング周波数の変換にベーススプラインによる補間を適用します。

ただ、それなりに複雑なことをしているわりには、音質的には、上の直線補間や2次関数補間の方が全然良いです。...もしかして、式を間違えているんですかねえ...

### 三角関数による補間(xxxx=Trigonometric[n])

サンプリング周波数の変換にフーリエ級数展開を用いた三角関数による補間を適用します。求めるサンプル点の近傍n点をフーリエ級数展開し、補間値を求めます。nのデフォルトは25となります。

ただし、サンプル点1点1点につきフーリエ級数展開をしているため、とてつもなく時間がかかります。

### 高次関数への最小二乗フィットによる補間(xxxx=LSM[[n],m])

サンプリング周波数の変換に最小二乗法を用いた、高次関数への近似を行い、補間を実行します。近傍n点のサンプルから、m (n-1) 次関数への近似を行います。nのデフォルトは2で、mのデフォルトは1となっています。

サンプル点1点1点につき、連立方程式を解いている(Gauss法)ため、三角関数補間ほどではないにしても、その処理には膨大な時間がかかりますので、ご注意ください。

なお、m=n-1のときは、n点のサンプル点を通るm次関数となります。つまり、論理的には、n=2, m=1のときは直線補間と同一で、n=3, m=2の時は2次関数補間と同一となります。ただし、プログラミングの都合で、全く同一というわけではありません。

## 6.2.1.8. 平滑化法の指定 (/SM=xxxx)

平滑化しない(xxxx=0ff)

平滑化処理を行いません (デフォルト)。

単純移動平均法(xxxx=Simple[[n]])

近傍  $n$  点の平均をとって平滑化する方法です。  
 $n$  は整数で、デフォルト値は 3 となります。

重み関数を三角波とした移動平均法(xxxx=Triangle[n][,y])

重み関数を三角波とした荷重平均を近傍  $n$  点でとって平滑化します。  
また、重み係数が 0 となる位置の平滑化範囲 (厳密にはその  $1/2$ ) に対する割合を  $y$  (小数表記) で与えます。デフォルトは、 $n$  が 5 で  $y$  が 0.8 となります。  
式としては、サンプル点  $x$  でのデータを  $y(x)$ 、重み関数を  $w(x)$ 、出力データを  $Y(x)$  とすれば、

$$Y(x) = \frac{1}{W} \sum_{i=-2/n}^{2/n} w(i) \cdot y(x+i)$$
$$W = \sum_{i=-2/n}^{2/n} w(i)$$

となり、重み関数  $w(x)$  は、例えば、 $y=0.8$  のときは下図のようになります。

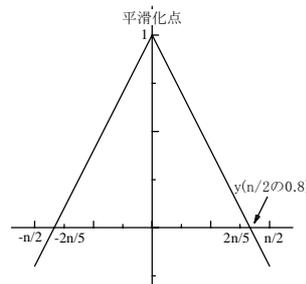


図1  $y=0.8$  のときの三角波の重み関数

重み関数を 2 次多項式とした移動平均法(xxxx=Parabola[n][,y])

重み関数を 2 次多項式とした加重平均を近傍  $n$  点でとって平滑化します。  
また、重み係数が 0 となる位置の平滑化範囲 (厳密にはその  $1/2$ ) に対する割合を  $y$  (小数表記) で与えます。  
デフォルトは、 $n$  が 5 で  $y$  が 0.8 となります。  
重み関数は、例えば、 $y=0.8$  の場合、下図のようになります。

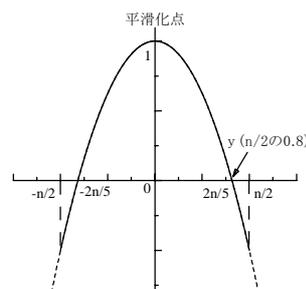
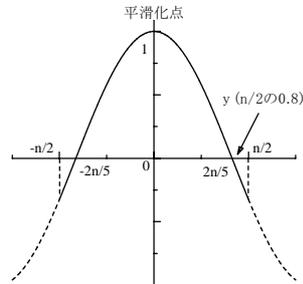


図2  $y = 0.8$  とした時の 2 次多項式の重み関数

### 重み関数を余弦波とした移動平均法( $xxxx=Cosine[n][,y]$ )

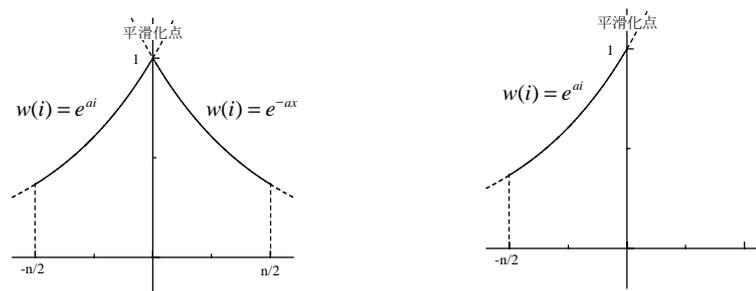
重み関数を余弦波とした加重平均を近傍  $n$  点でとって平滑化します。  
 また、重み係数が 0 となる位置の平滑化範囲（厳密にはその  $1/2$ ）に対する割合を  $y$ （小数表記）で与えます。  
 デフォルトは、 $n$  が 5 で  $y$  が 0.8 となります。  
 重み関数は、例えば、 $y=0.8$  の場合、**図 3** のようになります。



**図 3  $y=0.8$  のときの余弦波の重み関数**

### 重み関数を指数関数とした移動平均法( $xxxx=Exponential[n][,[a][,1]]$ )

重み関数を指数関数とした加重平均を近傍  $n$  点でとって平滑化します。  
 また、指数関数の係数を  $a$ （小数表記）で与えます。  
 サブオプションの第 3 項に '1' を指定した場合は、負の値、言い替えれば、過去の値についてのみ、加重平均をとります。  
 デフォルトは、 $n$  が 5 で  $a$  が  $1/n$  となり、第 3 項には 1 は付きません。  
 重み関数は、**図 4** のようになります。



**(a) 正負両方荷重平均をとる場合      (b) 負の値のみ荷重平均をとる場合**

**図 4 指数関数の重み関数**

## 2次多項式適合移動平均法 (xxxx=Parabola[n],,1)

近傍  $n$  点で最小二乗近似し、2次多項式にあてはめるものです。  
なお、内部的には、重み関数を2次多項式とした移動平均法と一緒にとなります。  
 $n$  のデフォルトは5となります。

## 適応化平滑化法 (xxxx=Adaptive[n][,y])

データの場所による性質の違いに応じて重みを変化させ、平滑化を行うものです。  
データ数を  $n$  とし、予測される雑音成分の分散を  $y$  (小数表記) として、処理します。  
デフォルトは、 $n$  が5となり、 $y$  が100となっています。

## 位相シフト平滑化(仮称) (xxxx=PhaseShift[l][,m])

周波数軸上で、一定周期で位相を変化(振動)させて音のザラつきを軽減しようというものです。  
内部処理としては、疑似ステレオ化処理とほぼ同一となります。異なるのは、疑似ステレオ化の場合、左右の位相の変化が正負逆になっているのに対し、こちらは同一方向に変化させているところです。

$l$  に位相シフトの周期、 $m$  に位相シフトの最大角を指定します。デフォルトは  $l$  が16で  $m$  が  $90^\circ$  となります。

なお、この位相シフト平滑化は、他の移動平均による平滑化方法とは異なり、FFTフィルタ内で処理を行いますので、ご注意ください。

ちなみに、FFTフィルタとも移動平均による平滑化とも、いずれの場合とも併用が効きます。

## 6.2.1.9. フィルタの指定(/Fil=xxxx)

FFT（高速フーリエ変換）を用いて、周波数領域でのフィルタを実行します。/FFT オプションにより精度を変化させることもできます。

フィルタをかけない場合、FX2WAVの出力はプラス側がマイナス側に偏ることが多いのですが、それを補正するためにも有用です（具体的には0Hzをカットするハイパスフィルタ）。

フィルタをかけない(xxxx=Off)

フィルタをかけません（デフォルト）。

ローパスフィルタをかける(xxxx=LP[m[,n]])

ローパスフィルタ（低域通過型フィルタ、つまり、高域をカットします）をかけます。

m, n 共に与えられない場合は、出力周波数の8割から減衰させ始め、出力周波数の9割で出力レベルが0となるようなフィルタをかけます。なお、ここでの出力周波数とは、出力サンプリング周波数ではなく、実際に出てくる音の周波数、つまり、出力サンプリング周波数の1/2の周波数となります。以下についても同様です。

mのみが与えられた場合、mHzから減衰し始め、出力周波数で出力レベルを0とするようなフィルタをかけます。

m, n 共に与えられた場合、mHzから減衰し始め、nHzで出力レベルを0とするようなフィルタをかけます。

例えば、m=10000[Hz], n=12000[Hz]とした場合、**図5**のような特性になります。

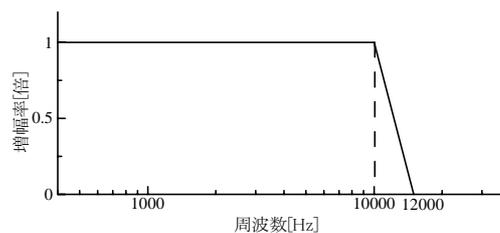


図5 ローパスフィルタ

ハイパスフィルタをかける(xxxx=HP[m[,n]])

ハイパスフィルタ（高域通過型フィルタ、つまり、低域をカットします）をかけます。

m, n 共に与えられない場合は、40Hzから減衰させ始め、20Hzで出力レベルが0となるようなフィルタをかけます。

mのみが与えられた場合、mHzから減衰し始め、0Hzで出力レベルを0とするようなフィルタをかけます。

m, n 共に与えられた場合、mHzから減衰し始め、nHzで出力レベルを0とするようなフィルタをかけます。

例えば、m=40[Hz], n=50[Hz]とした場合、**図6**のような特性になります。

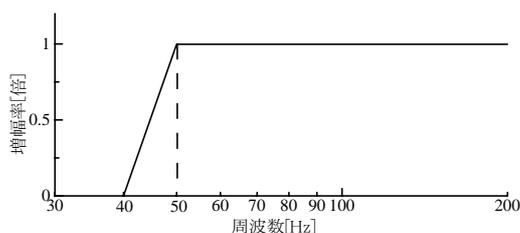


図6 ハイパスフィルタ

## バンドパスフィルタをかける (xxxx=BP[k, l[, m, n]])

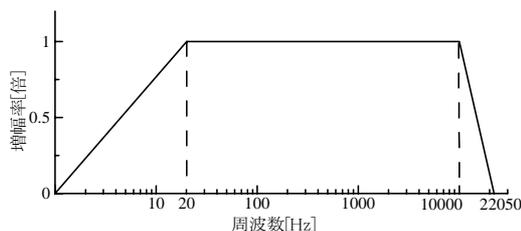
バンドパスフィルタ（帯域通過型フィルタ、つまり、高域と低域をカットします）をかけます。kとl、mとnは対で与えるようにしてください。

k, l, m, nすべて与えられない場合は、20Hzで出力レベルを0とし、40Hzで出力レベル1、そして、出力周波数の8割で減衰させ始め、出力周波数の9割で出力レベルを0とするフィルタをかけます。

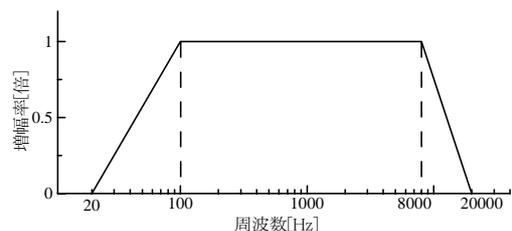
k, lのみが与えられた場合、0Hzで出力レベルを0とし、kHzで出力レベル1、そして、1Hzで出力レベル1、出力周波数で出力レベル0となるようなフィルタをかけます。

k, l, m, nすべて与えられた場合、kHzで出力レベルを0とし、1Hzで出力レベル1、そして、mHzで出力レベル1、nHzで出力レベル0となるようなフィルタをかけます。

例えば、出力周波数を44.1kHzとし、k=20[Hz], l=10000[Hz]として与えた場合は、**図7 (a)**のように、k=20[Hz], l=100[Hz], m=8000[Hz], l=20000[Hz]として与えた場合は、**図7 (b)**のようになります。



(a) サブオプションが2項目の場合



(b) サブオプションが4項目の場合

図7 バンドパスフィルタ

## グライコモード (xxxx=GEx1, y1[, x2, y2[, x3, y3...]])

グラフィックイコライザーモードとなります。バンドパスフィルタでの指定よりもっと細かく周波数特性を決定したい場合に指定します。必ず周波数 xn（整数表記）と増幅率 yn（小数表記）を対にして指定してください。最大20対までの指定が出来ます。

例えば、低域と高域を強調するために、**図8**のようなことも可能です。

ここでは、

x1 = 20,	y1 = 0
x2 = 50,	y2 = 1.5
x3 = 400,	y3 = 1.5
x4 = 800,	y4 = 1
x5 = 4000,	y5 = 1
x6 = 12000,	y6 = 3
x7 = 15000,	y7 = 0.1
x8 = 20000,	y8 = 0

と指定しています。

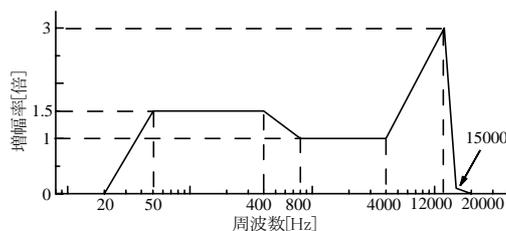


図8 グライコ

### 拡張グライコモード (xxxx=AGE[L|R]x1,y1,a1[,x2,y2,a2[,x3,y3,a3...]])

グライコモードよりさらに細かく指定出来るようにしたもので、左右のチャンネル別々にグライコをかけることができるようになり、さらに位相も変化させることができます。  
“AGE”としたときは両チャンネルに対して、“AGEL”としたときは左チャンネルに対して、“AGER”としたときは右チャンネルに対して、それぞれ拡張グライコをかけます。左右のチャンネル別々にグライコをかけたいときは、“AGEL”と“AGER”を両方指定してください。その後続く周波数特性のパラメータは、必ず周波数  $x_n$  (整数表記) と増幅率  $y_n$  (小数表記) そして、位相 (単位: 度、小数表記) をセットで指定してください。最大 20 セットまでの指定が出来ます。

## 6.2.1.10. 出力ファイルの量子化ビット数の指定

### 16 ビット出力(/16)

出力する量子化ビット数を 16bit とします (デフォルト)

### 8 ビット出力(/8)

出力する量子化ビット数を 8bit とします

## 6.2.1.11. ADPCM の解析順序の指定

### 上位 4bit 下位 4bit の順で解析(/HF)

上位 4 bit 下位 4bit の順で ADPCM を解析します (デフォルト)

FX の場合、こちらになります。

また、リトルエンディアンのヘッダ無し 16bit PCM 形式を指定する場合もこちらになります。

### 下位 4bit 上位 4bit の順で解析(/LF)

下位 4bit 上位 4bit の順で ADPCM を解析します。

PC エンジンの場合がこれにあたりますが、音質はよくありませんので、専用のツールの使用をお薦めします。

なお、ビッグエンディアンのヘッダ無し 16bit PCM 形式を指定する場合はこちらになります。

## 6.2.1.12. ループ回数の指定(/LPn[, init])

ループ回数  $n$  を指定します。  $n$  は整数で指定してください。

デフォルトは 1 となります。

`init` を与えた場合、毎ループごとに PCM 値とスケールレベルの初期化を行います。

なお、このループ処理は FX の ADPCM ファイル専用のオプションとなっており、wav ファイルには使用出来ません。あらかじめご了承ください。

## 6.2.1.13. チャンネル数の指定

### モノラル入力、モノラル出力(/M)

モノラル出力します (デフォルト)。

入力がステレオデータの場合は、両チャンネル加算モノラル出力となります。  
ヘッダ無しADPCM、あるいはヘッダ情報にチャンネル数が記載されていないフォーマットからの変換の場合、入力をモノラルと見立てて処理します。

### ステレオ入力、ステレオ出力(/S)

ステレオ出力します。

入力がモノラルデータの場合は、モノラル出力となります。  
ヘッダ無しADPCM、あるいはヘッダ情報にチャンネル数が記載されていないフォーマットからの変換の場合、入力をステレオと見立てて処理します。

### ステレオ入力、両チャンネル加算モノラル出力(/SA)

両チャンネルのデータを加え合わせ、モノラルとして出力します。

入力がモノラルデータの場合は、そのモノラルデータのまま出力します。

片チャンネルを反転 (/L-, /R-) すると、減算処理となります。

ヘッダ無しADPCM、あるいはヘッダ情報にチャンネル数が記載されていないフォーマットからの変換の場合、入力をステレオと見立てて処理します。

### ステレオ入力、左チャンネルモノラル出力(/SL)

左チャンネルのデータのみ取り出し、モノラルとして出力します。

入力がモノラルデータの場合は、そのモノラルデータのまま出力します。

ヘッダ無しADPCM、あるいはヘッダ情報にチャンネル数が記載されていないフォーマットからの変換の場合、入力をステレオと見立てて処理します。

### ステレオ入力、右チャンネルモノラル出力(/SR)

右チャンネルのデータのみ取り出し、モノラルとして出力します。

入力がモノラルデータの場合は、そのモノラルデータのまま出力します。

ヘッダ無しADPCM、あるいはヘッダ情報にチャンネル数が記載されていないフォーマットからの変換の場合、入力をステレオと見立てて処理します。

### 疑似ステレオ出力(/SS[=[l][,][m][,][n][,][x1,y1][,][x2,y2][,][...]]])

モノラルのデータからステレオっぽい音を生成します。処理としては、各周波数成分ごとに、少しずつ位相を変化させつつ、左右の音量に差を持たせるものとなっています。

lで位相変化の周期 (整数表記) を、mで位相変化の最大角 (整数表記) を、nで音量の変動周期 (整数表記) を、xn,ynで音量変動率の周波数特性 (整数表記, 小数表記) を指定し、詳細を設定出来るようにはしていますが、上手く説明する方法を思い浮かびませんでしたので、説明しないことにします(爆)。/FFTオプションを指定したり、サンプリング周波数が変化させたりした場合、目に見えて音色が変わってしまう上、プチプチとノイズが乗って困る場合にこれらのパラメータを変化させることが回避策の一つとなるので、ホントは書かなければならないところなのですが...(^^;

とりあえず、この/SSオプションは、44.1kHz専用で、/FFTオプションの値を1増した場合、lとnを2倍に、/FFTオプションの値を1減じた場合、lとnを1/2にすると、同一条件で変換出来る、とでも認識しておいて下さいm(..)m

なお、他のチャンネル指定とは独立して指定することができますが、その動作は通常の動作とは異なってきますので、ご注意下さい。ここでは、疑似ステレオ音声のソースとなるものが、右チャンネルか、左チャンネルか、あるいはモノラル音声であるかを指定することとなり、出力は常に疑似ステレオ出力となります。

## 6.2.1.14. ブロックサイズの指定(/BSn)

(ヘッダ無しステレオ ADPCM時の)ブロックサイズ。...と言って良いのか分かりませんが、とにかく、片チャンネルが連続して書き込まれている場合の、片チャンネルあたりのバイト数です。アドレス表記での指定が出来ます。

デフォルトは、0x4000 バイト (16384 バイト) となります。

## 6.2.1.15. オフセット(/On)

ADPCM のデコードを開始する位置をバイト単位で指定します。

また、MAKELST 表記も、アドレスの加減算を除き、可能です。

負の値の場合、ファイル終端からの位置となります。

また、ステレオデータの場合、同時に、この位置からブロックサイズごとに、左右のチャンネルが配置されるものとします。

このオプションは、指定した値そのまま使うため、ステレオデータの場合は注意が必要です。基本的に、多数の ADPCM データが含まれる大きなファイルに対し、それらを個別に切り出す用途には向いていますが、ひとつの ADPCM データの中から一部を切り出す用途には向きません。そういった場合は、/TS オプションをお使いください。

## 6.2.1.16. 変換開始位置・終了位置の指定

変換開始位置の指定(/TSn)

変換終了位置の指定(/TEn)

変換の開始位置と終了位置を指定します。いずれも n はバイト単位となります。

また、MAKELST 表記も、アドレスの加減算を除き、可能です。

負の値の場合ファイル終端からの位置となります。

なお、オフセットから開始位置までは、データにしないだけで、ADPCM のデコードは実行します(「読み飛ばし中」と表示されるのがそれです)。つまり、時間は、その分かかりますのでご注意ください。

また、ステレオデータの場合、片チャンネルのブロック内に位置を指定したときは、もう片方のチャンネルのデータも同じサイズだけ読み込んで変換します。

終了ブロックにおいて、右チャンネルのデータが無い場合は、左チャンネルも変換せず、右チャンネルのデータが途中で切れている場合は、左チャンネルもそのサイズだけ変換します。

処理としては、オフセットを OF バイト、変換開始位置を TS バイト、変換終了位置を TE バイト、ブロックサイズを BS、チャンネル数を CH (ここでは 2 とします) として与えた場合、

$$Ns = \frac{TS}{BS \times CH}, \quad s = TS \bmod BS$$

$$Ne = \frac{TE}{BS \times CH}, \quad e = TE \bmod BS$$

ただし、“mod” は余りを示す。

として、開始ブロック  $Ns$ 、開始ブロックからの開始位置までの距離  $s$ 、終了ブロック  $Ne$ 、終了ブロックからの終了位置までの距離  $e$  を求めます。

そして、実際に変換するのは、最初のブロックを  $(BS-s)$  バイト、最後のブロックを  $e$  バイト、として、両チャンネル同じサイズのみ変換します(イメージとしては、 参照)。

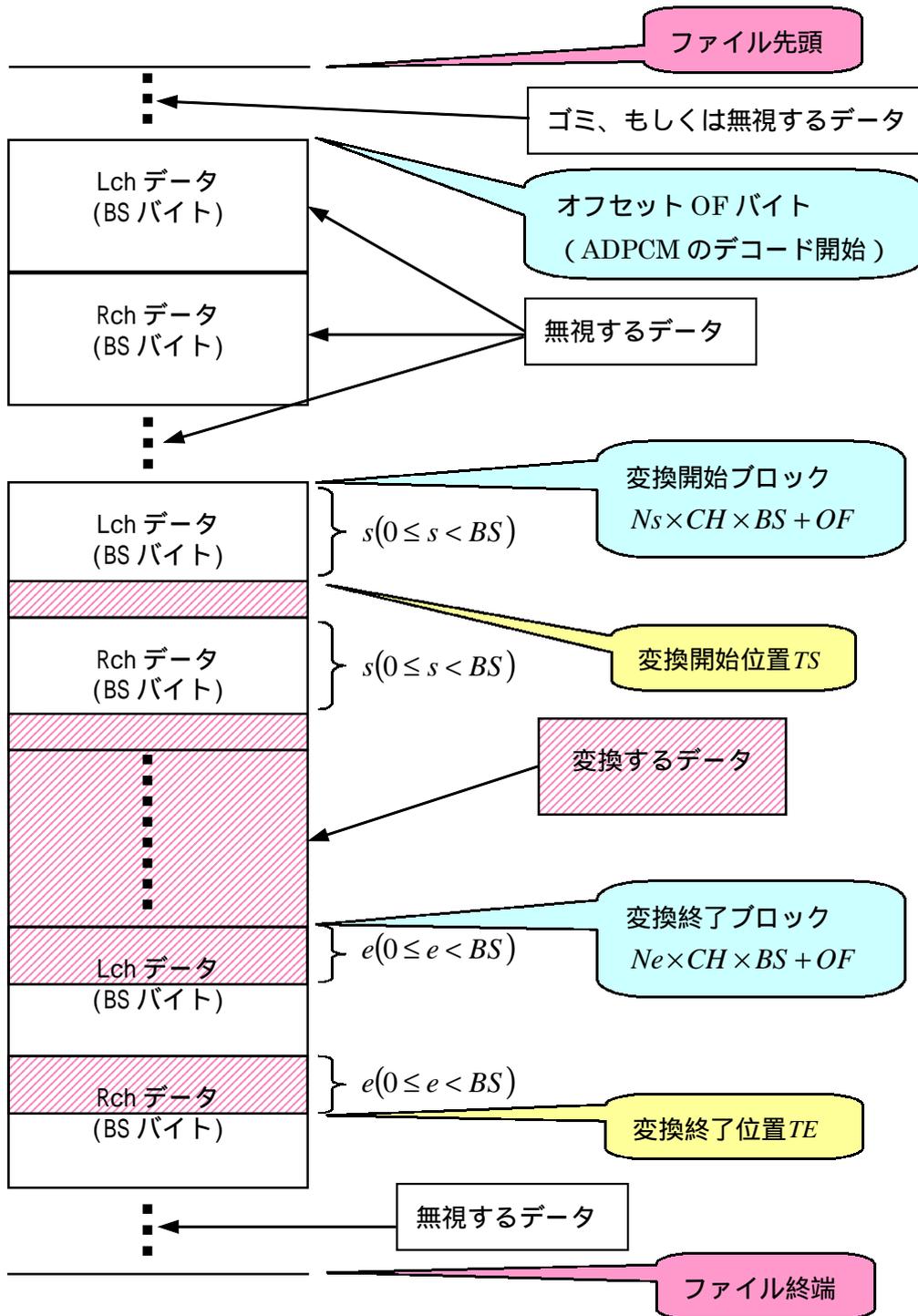


図9 ステレオデータの変換位置指定のイメージ

## 6.2.1.17. 正規化の指定

### 両チャンネルから最大値を取得(/MB)

FX用ADPCMの両方のチャンネルから最大の値を取得し、その値をWAVEの最大音量とします(デフォルト)。

### 各チャンネルごとに最大値を取得(/ME)

FX用ADPCMの各チャンネルごとに最大値を取得し、そのそれぞれの値をWAVEの最大音量とします。

左右の音量が酷く異なるものがあつたのでつけたのですが...、実はあまり使えません(^^;

### 正規化しない(/MM)

最大値からの正規化をせず、単純に変換します。

## 6.2.1.18. ゼロ補正の指定

### ゼロ補正をしない(/NZA)

ゼロ補正をしません(デフォルト)

### ゼロ補正をする(/ZA)

データの平均を取り、その値を引いたものを出力します。

## 6.2.1.19. 音量の指定

### 増幅率による指定(/Xn)

元データの音量に対する変換後の音量のをn倍します。  
10進数値で指定してください。小数点数表記も出来ます。  
デフォルトは1.0倍となります。

なお、"元データ"はしていますが、音量の決定は最後となりますので、ここでの"元データ"は、各種補正をした後でのデータとなります。

### 音量(デシベル単位)による指定(/dBn)

上記/Xn オプションのデシベル単位版です。  
nは、-90dB ~ +90dB の範囲で指定してください。  
デフォルトは0.0dBとなります。

## 6.2.1.20. データの反転

### 左チャンネル反転(/L-)

### 右チャンネル反転(/R-)

いずれも、データのプラスマイナスを単純に反転します。

/SA オプションと組み合わせて、両チャンネルの差を取りたい場合に使用します。...まず、使うことはありませんが...

### 6.2.1.21. 左右の音量バランス(/Bn)

右左の音量のバランス(左:-1.0 ~ 右:+1.0)を指定します(小数表記)。デフォルトは0.0となります。

中で何の計算をしているのか、解りづらいかも知れませんが、要は、左右のいずれかに値を振らした場合、振らした方がそのままの音量となるように、逆のチャンネルの方を相対的に下げています。

### 6.2.1.22. 無音部の指定

先頭無音部時間(/StNSn)

終端無音部時間(/EdNSn)

先頭の無音部とファイル終端の無音部の時間をそれぞれミリ秒単位(整数表記)で与えます。nには10進の整数のみ指定することができます。デフォルトは、いずれも0ミリ秒となっています。

### 6.2.1.23. フェードイン・フェードアウトの指定

フェードインの時間(/FI n)

フェードアウトの時間(/FO n)

フェードイン時間と、フェードアウト時間をそれぞれミリ秒単位(整数表記)で与えます。nには10進の整数のみ指定することができます。デフォルトは、フェードインが1ミリ秒で、フェードアウトが1000ミリ秒となります。

### 6.2.1.24. 上書き確認の指定

上書き確認をする(/CF)

出力ファイルと同名のファイルが存在した場合、上書きしてよいか確認をします(デフォルト)。

上書き確認をしない(/NC)

確認を抑制し、上書きを強制するためのオプションです。

### 6.2.1.25. 変換処理中の画面表示の指定

すべて表示する(/V)

途中経過(何%まで終了したかのメッセージ)、データ表示共々行います(デフォルト)。

データの表示のみする(/D0)

途中の経過表示をせず、データの表示のみ行います。

経過表示のみする(/P0)

データの表示をせず、途中の経過表示のみ行います。

なにも表示しない(/NV)

途中の経過表示も、データの表示も行わない。

## 6.2.1.26. ログファイルの指定

ログファイルを作成する(/Log)

出力ファイル名と同名で、拡張子のみ “.log” としたファイルにログを出力します。

ログファイルを作成しない(/Log=0ff)

## 6.2.1.27. FFT 演算精度の指定 (/FFT=n)

FFTのサンプル数を指定するもので、nには2に対する累乗の値を指定します。デフォルトは、n=12 (  $2^{12}=4096$  ) となります。

このnの値を大きくすることにより、精度が増して音質が良くなりますが、1増やすだけでも大幅に時間がかかるようになりますので、ご注意ください。

## 6.2.1.28. チャンネル合成 (/MIXn)

ステレオ時に他チャンネルのデータを各々n%だけ合成します。

nの範囲は、-100% ~ 100% となり、小数表記で指定してください。デフォルトは100% となります。

nに100%を指定した場合は、両チャンネルとも同一のデータとなりますので、右チャンネルは削除し、左チャンネルのデータのみをモノラルデータとして出力します。

なお、FX2WAVには、既にほぼ同一の操作が行える /SA オプション ( 及び /L- オプション、/R- オプションとの組み合わせ ) が用意されていますが、このオプションが異なるのは、デコード直後に合成処理を行うという点です。波形に処理を加える前にチャンネルの合成をしたい場合は、こちらのオプションをお使い下さい。

## 6.2.1.29. 著作権情報の付加

WAV ファイルの詳細プロパティや Windows Media Player の「クリップ」項目、「著作権」項目で表示される、LIST チャンクの著作権情報を付加します。いずれの場合も、下記のように /Ixxx=... の形式でテキストを指定してやることで、付加が可能となります。半角スペース入りのテキストの場合、ダブルクォーテーションマーク(") でくくって指定してください。

入力が WAV ファイルで、LIST チャンクが存在する場合は、それをデフォルト値として継承して出力します。

なお、各項目の詳しい説明は省きます（よく分からないだけなんですけどね(^;）。必要な場合は、適当な文献をご参照ください（MSDN ライブラリの AAFILE.C の項目なんか載っています）。ちなみに、明らかに WAV ファイルには不適當なものも含まれていますが、せっかくなんで入れておきました。また、決まった書式に沿わないテキストを記載した場合（FX2WAV では書式の制限をかけていません）もかししたら、問題が出てくるアプリもあるかもしれません。もし、そのようなアプリをお持ちでしたら、そのアプリで正常に読めるファイルを参考に指定をしてください。

アーカイブの場所 (/IARL=xxxx)  
アーティスト (/IART=xxxx)  
コミッション (/ICMS=xxxx)  
コメント (/ICMT=xxxx)  
著作権 (/ICOP=xxxx)

作成日 (/ICRD=xxxx)

デフォルトは、WAV ファイルに変換した日付で、yyyy-mm-dd 形式の書式で出力します。

クロップ (/ICRP=xxxx)  
大きさ (/IDIM=xxxx)  
ドット/インチ (/IDPI=xxxx)  
エンジニア (/IENG=xxxx)  
ジャンル (/IGNR=xxxx)  
キーワード (/IKEY=xxxx)  
明るさ (/ILGT=xxxx)  
メディア (/IMED=xxxx)  
名前 (/INAM=xxxx)  
パレット設定 (/IPLT=xxxx)  
製品 (/IPRD=xxxx)

タイトル (/ISBJ=xxxx)

Windows Media Player では、「クリップ」の項目に表示されます。

ソフトウェア (/ISFT=xxxx)

デフォルトは、FX2WAV Ver.x.xx となります。

鮮明度 (/ISHP=xxxx)  
ソース (/ISRC=xxxx)  
ソースの形式 (/ISRF=xxxx)  
技術者 (/ITCH=xxxx)

## 6.2.2. 注意事項

FX2WAV.EXE と同じディレクトリに FX2WAV.CFG があった場合、ユーザデフォルト値の指定をそこから読み込みます。

Ver.0.04 からオプション指定の出来るタイミングが増えましたが、その優先度は以下のようになります。

- システムデフォルト
- <ユーザデフォルト (コンフィグファイル)
- <インダイレクトファイル
- <コマンドライン

パラメータを渡さずに起動した場合、ヘルプとなりますが、数画面分一気に表示してしまいます。ですので、ヘルプを表示したい場合は、MORE コマンドを併用し、

```
fx2wav|more
```

とされることをお勧めします。

変換途中で Ctrl+C を押して処理を中断した場合、(あまり考えたくはありませんが) 不正な処理を行ったとかで強制終了させられた場合などでは、テンポラリファイルが実行した際のレントディレクトリに残ってしまいます。“~l?????”、“~r?????” といったファイルがそれに当たりますので、その場合は手動で削除してください。

音楽ファイルを一括変換する場合は、扱うファイルのサイズが大きくなりがちで、かつ、変換に時間がかかりがちになります。このため、裏のタスクで FX2WAV の変換をさせて、表のタスクでなにかをしようとしても、重くて使い物にならない場合があります。また、最悪の場合、システムが不安定になることもありますので、夜寝る前にスクリーンセーバーを切って (Direct3D 対応などの重いものであれば) 変換を開始させ、夜中じゅう変換させておくなど、なるべく他の作業とのマルチタスクはしないようにしたほうが無難です。

音声ファイルを一括変換する場合は、扱うファイル数が非常に多くなりがちです。このため、変換後に、エクスプローラなどでファイル検索やコピーをする場合(とくにウイルススキャンソフトが常駐している場合)、スキャンディスクや DiskDoctor、デフラグや SpeedDisk をかける場合などで、時間がかかるようになることが多いのでご注意ください。

## 6.3.MAKELST ( オマケ )

指定の書式のテキストファイルから CDREAD.EXE 用の lst ファイルと FX2WAV 用の ind ファイル ( もしくは、バッチファイル ) を作成する補助ツールです。CDREAD.EXE の読み込みアドレスの計算が面倒だったため、そのあたりを半自動化する目的で作りました。

DOS プロンプトを起動して、以下の書式に沿った形で実行してください。

書式 : MAKELST [/i|b] 入力ファイル [リストファイル [FX2WAV ファイル]]

入力ファイル : MAKELST フォーマットのテキストファイル

リストファイル : CDREAD 用のリストファイルです。

省略した場合は、入力ファイルの拡張子を “.lst ” としたものを出力します。

FX2WAV ファイル : バッチファイル、もしくはインダイレクトファイルを出力します。

どちらを出力するかは、オプション指定によります。

省略した場合は、インダイレクトファイルを出力するようになっていた場合は、リストファイルの拡張子を 祢 ND 猥二したものを、バッチファイルを出力するようになっていた場合は、リストファイルの拡張子を “.wav ” としたものを出力します。

オプション : /i ... インダイレクトファイル出力 ( デフォルト )

/b ... バッチファイル出力

このオプションは FX2WAV ファイルの出力方法の指定をするだけです。つまり、CDREAD のバッチファイル出力はサポートしていません。

### 6.3.1.MAKELST フォーマット

MAKELST にかませるリストファイルで、演算の基本となるデータや FX2WAV 用のオプションを記述した制御部、出力先ファイルとアドレスを記述したデータ部からなります。いずれも、行単位での記述となり、複数行にまたがった指定は出来ません。

#### 6.3.1.1. データ部

タブ区切り ( **スペース区切りでは正常動作しません** ) で WAV 用ファイル名、( CD-ROM からの ) 読み込み開始アドレス、読み込み終了アドレスを記述したものとなります。アドレスの指定には、MAKELST 形式 ( 6.1.4 参照 ) が使用出来、演算も加減算のみはサポートしています。

なお、読み込み開始アドレスに関しては、最後の行を除いて、省略出来ます。その場合、次の開始アドレスの 1 バイト手前を終端と判断します。

書式は次のようなイメージとなります。

書式 :

<WAVE ファイル名>                      <開始アドレス>[                      <終了アドレス>]

## 6.3.1.2. 制御部

アスタリスク ( \* ) で始まる行を制御部と判断し、以下に述べるような書式を使用出来ます。

なお、制御部で指定した各種制御は指定した次の行から有効になります。  
このため、データ部で MAKELST の表記方法にのっとった表記をする場合は、それ以前の行に必要なパラメータをきちんと設定しておかなければ、正しく計算してくれません。

また、ブロック単位の丸めの処理が入る時は、それ以前の行で “ \*BlockSize ” の設定を行っていないと、アドレスとをきちんと丸めてくれません。

### トラックの指定 (\*Track=n)

CDREAD に渡すトラック数の指定を整数表記で与えます。  
デフォルトは 1 で、1 以上の数を指定できます。

### オフセットの指定 (\*Offset=n)

データ部の開始アドレスのこの値を上乗せして CDREAD に渡します。  
FX2WAV には、このデータは渡されません。  
n にはアドレス表記で渡し、デフォルトは 0x00000000 となります。

### 開始アドレスの丸め方の指定 (\*StartPoint=xxx)

xxx =	BLOCK+	...	ブロック単位に切り上げる
	BLOCK-	...	ブロック単位に切り捨てる
	SECTOR+	...	セクタ単位に切り上げる
	SECTOR-	...	セクタ単位に切り捨てる
	BYTE	...	バイト単位のまま (デフォルト)

### 終了アドレスの丸め方 (\*EndPoint=xxx)

xxx =	BLOCK+	...	ブロック単位に切り上げる
	BLOCK-	...	ブロック単位に切り捨てる (デフォルト)
	SECTOR+	...	セクタ単位に切り上げる
	SECTOR-	...	セクタ単位に切り捨てる
	BYTE	...	バイト単位のまま

### サンプリング周波数の指定 (\*Frequency=n)

サンプリング周波数 nHz を整数表記で与えます。  
FX2WAV に渡すデータとなります。  
MAKELST 内では、アドレスの時間表記の計算に使用します。  
デフォルトは 31600Hz

### 出力サンプリング周波数の指定 (\*OutputFrequency=n)

FX2WAV に渡す /of オプションに相当します。  
MAKELST 内の演算では使用しません。

### チャンネル数の指定(\*IOStyle=xxx)

入力チャンネル数、出力チャンネル数、及び、その変換形式の指定をします。  
FX2WAV に渡すデータとなります。  
MAKELST 内では、アドレスの時間表記、ブロック表記の演算に使用します。

xxx =	MONORAL	...	モノラル入力、モノラル出力
	STEREO	...	ステレオ入力、ステレオ出力 (デフォルト)
	ADDOUT	...	ステレオ入力、両チャンネル加算モノラル出力
	LOUT	...	ステレオ入力、左チャンネルのみをモノラル出力
	ROUT	...	ステレオ入力、右チャンネルのみをモノラル出力

なお、互換性のため、\*Channel=n の書式も残してあります。

### ブロックサイズの指定(\*BlockSize=n)

FX2WAV に渡すデータで、アドレス表記で与えます。  
MAKELST 内ではアドレスのブロック表記の演算にも使用します。

### ADPCM 解析順序の指定(\*ReadMode=xxx)

FX2WAV の /hf, /lf に相当します。  
MAKELST 内部では使用しません。

xxx =	HighFast	...	上位 4bit	下位 4bit の順で解析 (デフォルト)
	LowFast	...	下位 4bit	上位 4bit の順で解析

### サンプリング周波数変換時の補間方法の指定(\*InpMethod=xxx)

FX2WAV の /ISL, /ITD に相当します。  
MAKELST 内部では使用しません。

xxx =	SLI	...	直線補間 (デフォルト)
	TDI	...	2 次関数補間

### FX2WAV ファイル内変換位置の指定

\*Offset/FX2WAV=n  
\*StartPoint/FX2WAV=n  
\*EndPoint/FX2WAV=n

それぞれ FX2WAV の /On, /TSn, /TEn に相当します。n の表記は MAKELST 表記となりますが、MAKELST 内部では演算せず、FX2WAV にそのまま渡します。  
デフォルトはいずれも 0 となり、ファイルの先頭をブロック配置の先頭と見立ててファイルすべてを変換します。

### ループ回数の指定(\*Loop=n)

FX2WAV に渡すデータで /LPn に相当します。  
MAKELST の演算では使用しません。

### 正規化の方法 (\*MAXSeek=xxx)

FX2WAV の /Mx オプションに相当します。  
MAKELST 内部の演算では使用しません。

xxx = BOTH  
FX 用 ADPCM の両方のチャンネルから最大の値を取得し、その値を WAVE の最大音量とします (デフォルト)。  
EACH  
FX 用 ADPCM の各チャンネルごとに最大値を取得し、そのそれぞれの値を WAVE の最大音量とします。  
MANUAL  
正規化しない。

### ゼロ補正の指定 (\*ZeroAdjust=xxx)

FX2WAV の /ZA, /NZA に相当します。  
MAKELST 内部の演算では使用しません。

xxx = ON ... ゼロ補正あり  
OFF ... ゼロ補正無し (デフォルト)

### データ反転の指定

\*ReverseL=xxx : 左チャンネル反転  
\*ReverseR=xxx : 右チャンネル反転

FX2WAV の /L-, /R- オプションに相当します。  
MAKELST 内部の演算では使用しません。

xxx = ON ... 反転する。  
OFF ... 反転しない (デフォルト)

### 音量の指定 (\*Volume=x[dB])

音量を x 倍します。また、最後に “ dB ” が付いていたら、x[dB]とします。  
デフォルトは 1.0 倍  
FX2WAV の /Xx, /dBx に相当します。  
MAKELST 内部の演算には使用しません。

### バランスの指定 (\*Balance=x)

バランス 左 : -1.0 ~ 右 : +1.0  
FX2WAV に渡す値で小数表記で与えます。  
MAKELST 内部の演算には使用しません。

### フェードイン・フェードアウト時間

\*FadeIn=n : フェードインの時間 (デフォルト : 1 ミリ秒)  
\*FadeOut=n : フェードアウトの時間 (デフォルト : 1000 ミリ秒)

いずれも FX2WAV に渡すデータで、MAKELST 内部の演算には使用しません。

### (FX2WAV) 上書き確認の指定(\*WriteConfirm=xxx)

(FX2WAV での) 同名ファイルが存在した際の動作

xxx = NotConfirm ... 上書きを強行する  
CONFIRM ... 上書き確認をする (デフォルト)

### (FX2WAV) 情報表示の指定(\*Information=xxx)

(FX2WAV での) 情報表示

xxx = OFF ... 情報表示をしない  
DataInfoOnly ... データ情報の表示のみ  
ProgressiveInfoOnly ... 経過表示のみ  
VERBOSE ... すべての情報を表示する (デフォルト)

### FX2WAV にそのまま渡すオプション文字列の指定(\*Inline=xxx)

FX2WAV にそのまま渡すオプション文字列を指定します。  
当然ながら、MAKELST の内部計算には使いません。

いちおう、仕様としては以上ようになりますが、かなり解りづらいかと思しますので、サンプルを添付しておきました。参考までにご覧ください。

なお、『アルバレアの乙女』については、もしかしたら英語の綴りを間違っているかも知れません(^\_^; どこにも英語で書かれているところがないので...

## 6.3.2. 注意事項

このツールにより掃き出されるBATファイルは、FX2WAVがパスの通ったディレクトリにあるか、カレントディレクトリにある場合を想定しています。あらかじめ、ご了承ください。

このツールはDOS窓専用となっておりますが、CDREADについては、基本的にDOSモードで起動してください。DOS窓だと、FXのCD-ROMをきちんと読むことができないようです。(Windows98のバグ?)

FX2WAVとは異なり、すでに同名ファイルが存在する場合も、上書き確認はしてくれません。十分気をつけてお使いください。

パラメータを与えずに起動した時はヘルプとなりますが、数画面分一気に表示してしまいます。ですので、MORE コマンドを併用し、“make1st|more”などとされることをお勧めします。

## 7. 動画フォーマット抽出確認状況

Ver.0.05からサポートし始めた動画からの抽出ですが、動画って、ゲームによってフォーマットがバラバラですので、どんなデータが抜き出せたかっのを参考として挙げることにします。

なお、参考として挙げてあるファイルの命名については詳しくは秘密です(ネタバレになるものもあるかも、なので...)

### 7.1.MIXフォーマット

MIXフォーマットへの対応は順番としては5番目となったのですが、よくよく考えてみれば、これを最優先させた方が良かったかも知れませんね(謎)。

以下のようなファイルにて抽出出来ることを確認しました。

PC-FXGA 付属「アマチュア提供 CD-ROM」より		
イメージアニメーション	F60_3_32.MIX	(32KHz/ ステレオ)
	FXGA4.MIX	(16KHz/ ステレオ)
	FXGA2.MIX	(16KHz/ ステレオ)
CGA 作品 1 : A DRAGONFLY	D60_3_32.MIX	(32KHz/ ステレオ)
	DRAGON4.MIX	(16KHz/ ステレオ)
	DRAGON2.MIX	(16KHz/ ステレオ)
CGA 作品 2 : CHESS	CHESS4.MIX	(16KHz/ ステレオ)
	CHESS2.MIX	(16KHz/ ステレオ)

### 7.2.RTFフォーマット

これについては、ほとんどMIXフォーマットと一緒にすることで対応させてしまいました。なんか、ファイルヘッダの一部が記述されていない(もしくは、別の意味で使われている?)ようですが...、実際のところ、どうなのでしょうね。

以下のデータにて抽出出来ることを確認しました。

『バトルヒート』より  
    オープニング (Ever Lasting Dance)

『アンジェリーク Special 2』より  
    光栄ロゴ  
    プロローグ

## 7.3.ARS フォーマット

ファイルヘッダが2項目、フレームヘッダも2項目しか結局分かっていませんので、かなりアヤしいです。特にステレオデータについては、ヘッダのどこを読めばサイズとかが分かるのかよく分かんなくて、かなり強引に位置指定しています。

ま、とりあえず、以下については抽出出来ることが確認出来ました(16KHz以下のサンプリング周波数はどのゲームに使われているのか、また、ARSのフォーマットに本当にそんなものがあるのか、分からないので、試していません)。

### 7.3.1.32KHz ステレオデータ

『ファースト Kiss 物語』より  
HuneX  
PC-FX  
オープニング1(君のファーストキス)  
オープニング2(だから涙をふいて)

『ファイアーウーマン 纏組』より  
オープニング(LET'S GET LOVE)

### 7.3.2.32KHz モノラルデータ

『ファースト Kiss 物語』より  
「長女の香奈です。よろしく...、お願いします。」  
『ブルーブレイカー ~ 剣よりも微笑みを~』より  
オープニング(笑顔の約束)

## 7.4.ADPCM 後置フレームサイズ固定形式(仮称)

もともとは、「ミラークルム動画形式(仮称)」として、追加オプションなども指定しなくて良い形式にするつもりだったのですが...。オープニングとそれ以外とで、フレームのサイズが異なっていたんですね、これが...

さらに、普通に考えて、ヘッダを記述しない方法が一番プログラミングしやすい形式に思えた(他のゲームでも使われているかも?)ので、このような呼び方にしてしまったという...。とりあえず、以下のようなデータで抽出出来ることを確認しました。

『ミラークルム - ザ・ラスト・レベレーション - 』より

RayForce	(0x4000,0x600,0x41c)
オープニング	(0x4400,0x600,0x41c)
プロローグ1(?)	(0x4000,0x600,0x41c)
プロローグ2(?)	(0x4000,0x600,0x41c)

## 7.5. アルバレアの乙女動画形式（仮称）

こっちは、すべての動画データで同じ形式を取っていましたので、「アルバレアの乙女形式」と名乗ることにしておきます。以下のデータについて、抽出出来ることを確認しました。

『アルバレアの乙女』より  
オープニング  
レオンエンディング  
カインエンディング  
ジャンエンディング  
マハトエンディング  
ロテールエンディング  
エンディング  
聖乙女エンディング

## 7.6. 虚空漂流ニルゲンツ動画形式（仮称）

フォーマットのにはステレオに対応しているようですが...、このゲームはモノラル音声ばかりだったと記憶していますので、とりあえず、モノラルの形式のみサポートしています。もっとも、ステレオの形式をサポートしようとしても、プログラムの構造上ちょっと問題があるんですけど...(^;)

それにしても...、普通にプログラミングしてすべて変換すると、後ろの方で豪快なプチノイズ（というより、プチノイズとでも言った方がいいかも？(^; )）を発していたんですよえ、これが...。本来ADPCM音声があるべきところ（という認識が間違っているのかも知れませんが...(^; )）に0xffが連続で書き込まれているんだもんなあ。とりあえず、0xffが3つ連続で続いた場合、抽出を終了するという強引な方法で対処しましたが...、他のゲームでは使えないかも？ま、その場合は適当にまた考えれば良いか...

以下のデータについて、抽出出来ることを確認しました。

『虚空漂流ニルゲンツ』より  
オープニング  
プロローグ  
『獵犬』導入アニメ  
許しはしない、あなたを  
リスじゃあるまいし  
オペレータとして人間として  
しっかりしてデュン  
これからは私の指示に  
みんないい人なのに...  
約束のもんだ  
私にはできないよ  
あれはホテルだったな  
信じてる、あなたを  
海  
もう少しこのままで  
エンディング

# 8. バージョンアップ履歴

## 8.1.Ver.0.01(1998/6/6)

初版

## 8.2.Ver.0.02(1998/6/8)

### 8.2.1.FX2WAV

リミッタをかけるタイミングを増やした

Ver.0.01ではADPCMデータの解析のときのみリミッタをかけていたのですが、浮動小数点の値から16ビットのPCMデータにする時にもリミッタをかけるように変更しました。

これにより、そのまま出力しても(倍率1)プチノイズが発生しなくなりました(^^;

音量のデフォルトを1に変更

プチノイズが発生しないんだってことで、おかしい仕様であるデフォルト0.8を1に変更しました(^^;

音量の表示にdB単位を追加

値を渡す時はちょっと面倒なのでパスするとしても、とりあえず、表示だけはつけました。ただ...、16ビットデータを現実的な範囲で増減させたところで、その倍率はたかが知れていますので、デシベルで表示するまでもないんですけどね(^^;

最大音量の表示を16ビット整数値での値に変更

変換時に表示される最大音量についてですが、整数部12ビット、小数部3ビット(?)の固定小数点での値ではなく、16ビット整数値の方がわかりやすくなって変更しました。

再生時間の表示がおかしかったのを修正

ドキュメントの免責等に一部項目を追加

## 8.2.2.MAKELST

音量のデフォルトを FX2WAV の変更に合わせて 1 に変更

同様に、MIRA.TXT のボリューム情報 “ \*Volume= ” を “ 1 ” に変更

ドキュメントの免責等に一部項目を追加

ドキュメント、ヘルプで丸め方のデフォルト値が間違っていたのを修正

終了アドレスのデフォルトは「ブロックサイズに切り捨てる」という “ BLOCK- ” だったので、 “ BLOCK+ ” になっていました。

終了アドレスを省略した場合にも丸めを適用

Ver.0.01では終了アドレスを省略した場合は、丸め方の指定を無視して次の開始アドレスの 1 バイト手前を終了アドレスとしていましたが、今回からは、丸め方の指定にそった値になるように変更しました。

セクタ単位の丸め方を追加

ついでなんで...(^^;

丸めの仕方にバグがあったので修正(したつもり)

ちゃんと丸めてくれない時などがあったのですが、それを修正。

今度こそちゃんと動くことを祈る(^^;

でも、正直な話、丸め処理とエラー処理については現状でもかなりアヤしいです。そう、実は、FX2WAV よりこっちの方が開発に手間取っているんですよ。3 倍くらい…。ほんと、どっちがおまけなんだか(^^;

あ、いえ、おまけだからこそ、少々挙動不審な部分があっても良いと割り切れるのではないかと...(^ ^);;

なお、今回上記丸め方の仕様変更及びバグフィックスにより、添付の MAKELST ファイルのサンプルについても、Ver.0.01 と Ver.0.02 とでは出力結果が異なってしまいます。あらかじめご了承ください(出てくる音に大差はないと思いますが...(^^);)

## 8.3.Ver.0.03(1998/6/21)

### 8.3.1.FX2WAV

#### メインルーチンの書き直し

バージョンアップする時に、あまりにもソースが見つらく、大変でしたので、書き直しました(新たなバグが発生する可能性は無きにしもあらず)。勢いだけで作ったスパゲッティソースがこんなにも早くあだになってしまうとは...(^\_^;

でも...、パラメータ読み込みの関数だけでもソースが数画面分になっている...(涙)。「関数は一画面に収まる大きさに」とはよく言われますが、それってどう対処すればいいものでしょ?無理やり分けると、かえって解りづらくなるだろうし...

「アクティブなプロジェクトの構成」を“Release”にしてビルド...

あ、いえ、知らなかったんです。ごめんなさ~い<m(\_ \_)m>  
もちろん、バイナリエディタで覗いても、妙に無駄が多く、デバッグモードでコンパイルされているのかな、とは思っていましたが。でも、それらしき(コンパイル)オプションが見当たらず、諦めていたんですが...。こんなところで設定するんだったのですね(^\_^;  
...マニュアルを見ていないことがバレバレですね(^\_^; だって、Visual C++ のマニュアルって分厚くってどこから手をつけて良いのか分からなくて...f^\_^;

#### ステレオ入力時の変換方法を増やした

両チャンネルを足してモノラル出力する方法と、片チャンネルのみ出力する方法を増やしました。

で...、今まで、いくら頑張っても上手く変換出来なかった『アルバレアの乙女』ですが...、結局は、モノラル(右チャンネルオンリーか左チャンネルオンリー)で出力するのが一番みたいです(^\_^; 最大値の取得とか、ゼロ補正とか、実はこの『アルバレアの乙女』のデータを変換するのが目的で付け足して行ったものなんですけど、なんともあっけない幕切れにちょっとガッカリ...

#### 反転出力に対応

上に書いた「両チャンネルを足してモノラル出力」に絡んだことなのですが、逆に差をとるために追加しました。必要性はなさそうですけど...(^\_^;

#### 増幅率の指定に dB 単位を追加

初め考えていたのは、/x オプション指定で、最後に dB と付いていたらデシベルと判断する、という物だったのですが、よくよく考えてみれば、“x” というスイッチは、倍率から連想したものなので、それでデシベルを渡すってのは変なんですよ(^\_^;

じゃあ、/dBってオプションを別に作れば良いや、だったら1分とかからずに修正作業が終わる...、ってことで、追加しちゃいました。

#### 平均値をとってゼロ補正

プラス方向やマイナス方向にずれている音声データってどうも気持ちが悪いので...(^\_^;  
あ、いえ、音がってわけではなく、単なる気持ちの持ちようですが...(^\_^;

なお、何の工夫もせずに long double 型ですべての値を合計したあと、サンプル数で割って平均を出していますので、データによっては(オーバーフロー、情報落ち等で)正しくない値をたたき出してくれることもあるかも知れません。

## 出力先の省略に対応

これで、  
FOR %a IN(\*.T\*) DO FX2WAV %a  
などの一括指定が出来るようになりましたね(^-^)  
実は、今まで、ファイル名の分割結合をする関数が見つからなかったんです。もちろん、自作するのもいいのですが、そうなってくると、マルチバイト対応でまた面倒になってくるし...。コード的に0x5C “¥” が含まれるものの対処とか...(^  
とりあえず、標準添付の\_splitpath関数や\_makepath関数はマルチバイト対応を謳っていますので、多分、問題は出て来ないのではないかと...

## 上書き確認するようにした

なんか、ようやくまともになって来つつありますね(^; ただし、メッセージでは、“No”の場合、ESCキーを押してくれて出て来ますが、Caps Lockがかかっていると、反応してくれません。...なぜ？

## 指定位置から指定位置までの変換に対応

仕様にはかなり悩んだんですが...、今のような仕様になりました。  
例によって、境界付近はかなりアヤしい動きをしますが、無いよりはマシってことで...(^  
なお、このオプションについては、もし、次のバージョンがあるとするなら、仕様変更される可能性が非常に高いです。

## 途中経過と上書き確認等を直接表示に

今までは画面に表示される情報すべてを標準出力に出していましたが、途中経過と上書き確認のみ直接表示に変更しました。とりあえず、リダイレクトしても、安心して見れるようにはなりました。

## 初期値のバグをフィックス

最大値などを取得した後、ADPCMの初期値とスケールレベルの初期値をクリアせずに変換していました。今回からはちゃんと対応出来ている、かな？

## リミッタのしきい値を変えた(これもバグ?)

Ver.0.02までは演算途中は-2048 ~ 2047の範囲で計算していましたが(本来なら、符号部1ビット、整数部11ビット、小数部4ビットの固定小数点なのですが、私は浮動小数点で演算しています)、±2047.9375の範囲に変更しました。  
いえ、何でって言われても困りますけど、なんとなく、こっちの方が良いかなって...

## 画面表示、その他諸々の変更

また表示内容が変わってしまいました。すみません <m(\_ \_)m>  
どうも、この狭い中にすべての情報を表示させるるのが大変で...(^;

## 8.3.2.MAKELST

「アクティブなプロジェクトの構成」を“Release”にしてビルド...

\*offset をバイト単位のみに変更

よくよく考えてみたら、“アドレスの表記方法”でのデータ(時間単位、ブロック単位)はこの \*offset の値も参照するんですね。前バージョンまでは、\*offset に限り \*offset を参照しないようになっていましたが、まぎらわしいので、\*offset 指定にはバイト単位のみ可能ということにしました。

もっとも、コードを変えたわけではなく、単にヘルプとドキュメントを変えただけですが... (^;)

FX2WAV の仕様追加に併せて、

```
*ZeroAdjust=  
*IOStyle=  
*ReverseL=  
*ReverseR=  
*WriteConfirm=  
*Information=
```

といった制御コードを追加

\*Volume= にデシベルでの指定が出来るように追加

ただし、この場合も、FX2WAV に渡されるオプションは、/Xn の形式となります。

ARBULEA.TXT の制御コードの変更

\*Channel=2 から IOStyle=LOUT、もしくは、IOStyle=ROUT に変更しました。

どちらを記述するかは、実際に聴いてみて音が良かった方にしています。

もっとも、それほどキチンと聴いているわけではなく、最初の数秒聴いただけで判断していますし、その数秒で大差ないように聞こえた場合は、前のコードをそのまま持ち越しています。そのため、逆のチャンネルの方が良い音ってこともあるかも知れません。

なお、一つだけ、どちらのチャンネルにしてもプチノイズが消えない物がありましたので、これだけは FadeIn を 100m ミリ秒にしました。

それともう一点、ZeroAdjust を ON にしました。

MIRA.TXT の制御コードの変更

\*ZeroAdjust=ON の追加他

## 8.4.Ver.0.04 (1998/7/3)

### 8.4.1.FX2WAV

#### 全面書き換え

またか...(^; とりあえず、ユーザの方には何のメリットもないんですけど、あまりにも解りづらくなったソース(まあ、機能追加の都合上のも理由としてはありますが...)を各機能ごとに分離し、その大部分を書き換えました。

ただ、この再プログラミングに伴い、今回からは素直にテンポラリファイルを作るようになったのですが、それ故にディスク容量を結構圧迫してくれます。このテンポラリファイル、実は 32bit の浮動小数点数で書き込まれていて、そのサイズは元データの実に 8 倍にもなります。また、周波数の変換をする場合などは、それと同容量(出力先の周波数によって増減しますが)のファイルが作られますので、最大で元ファイルの 16 倍のテンポラリファイルが作られることとなります。また、処理ごとにこれらの巨大なファイルを逐一読みに行くということで、変換時間も若干長くなってしまったかも知れませんが(キャッシュディスクに入る程度のファイルサイズだったら処理はかえって早いかも知れませんが...)

これらのことはあらかじめご了承くださいませよう、お願いします。もし、こういった仕様がまずいようでしたら、以前のバージョン(Ver.0.02 など)をお使いください。

表示される内容がまた変わりました。

もう、お約束になってしまっていますね(^; 今回は、全面書き直してことで、この部分もその例にもれず書き直されました。よって、Ver.0.02 や Ver.0.03 とはまた若干違った表示をします....

#### ADPCM のデコードでリミッタをかける方法を若干変更

今までは、次のサンプル点でデコードする都合で 32767 (2047.9375) を超えた時点で強制的に超えた部分をカットしていたのですが、今回からは出力に関してはそのまま出力するようにしました(デコードに使う値はリミッタをかけた値を使います)

コンフィグファイルによりユーザデフォルト値の指定を可能に

デバッグしてて、毎回毎回 /s とか書くのが面倒で、追加しちやいました(^;

インダイレクトファイルに暫定対応

なんか、オプションがどんどん増えまくってって、もしかしたら、command.com のデフォルトの入力バッファサイズである 128 文字を超える場合があるかもってことで、追加しました。

一部オプションにシステムデフォルトに戻す場合のものも追加

上記、コンフィグファイルやインダイレクトファイルの追加で、そういったものの必要性に迫られたので....

ブロックの配置を考慮した変換範囲指定が可能に

Ver.0.03 での /O オプション、/TE オプションに加えて /TS オプションを追加しました。

## ADPCM のデータ解析順序の選択が可能に

あ、いえ、ご存じのとおり、FXのチップの大部分がPCエンジンで使われていたものとはほぼ同等のものとなっていますよね？じゃあってことで、もしかして、こっちにも使えないかなって思って、追加したのですが...

いや、いちおう、音は出ます。ただ、やっぱり、なんか変な音にしかならないのも事実で...(^;

というわけで、PCエンジンの音を変換させたかったら、NIFTYなどにある専用のツールを素直に使いましょう。

## ループ処理に対応

やっぱり、FXの音楽を変換するつもりなら避けては通れない機能だと思ひまして...。ただ、この機能、1バイト単位で正確にオフセットと変換終了位置を指定してやらないと、奇麗につながってくれません。

また、このループ処理が正しく行えるのは、『ファイアーウーマン 纏組』、『ファーストKiss 物語』のような単純なループについてのみです。部分的にループ処理をする必要がある『超神兵器ゼロイガー』やイマイチどこでループしているのか分からない『虚空漂流ニルゲンツ』なんかの場合はループできません。

それと、今回、ADPCMのデコードの時点で、ループ処理を行っているため、その後の処理もそのループ回数倍のディスク容量が必要となってしまっています。

これらの点については、一応ご注意ください。

## 周波数の変換に対応

とりあえず、導き出したいデータの近傍2点間を結ぶ直線から求める方法と、導き出したいデータの近傍3点を通る2次関数から求める方法とを実装しました。

直線補間に関しては、初めから入れるつもりではいたのですが...、2次関数による補間はハッキリ言って入れる予定はありませんでした。...元々はベーススプラインによる補間を考えていたんですねえ、これが...(^;

ところが、いざプログラムが出来て補間させると、とんでもなく酷い音...

で、仕方なしにいろいろいじったり計算したりしているうちに、今のような工夫も何も無いような形に落ち着いてしまいました...(せっかく作ったので残してあるだけです)。

で、それ以外の実装予定、というか、とにかく作ってみて音が良かったものを採用しようとし、その候補として挙げられていた補間方法が、3次スプラインと三角関数による補間だったので...

三角関数による補間...。これは、実は一番やろうとしていたものなのですが...。いえ、フーリエ変換すれば、高域フィルタや低域フィルタも一緒に実現出来ないかなって思って...。ところが、これがまた、理論が思い出せなくて...。フーリエ変換といえば、大学2年で習う(工業系大学の場合)応用数学の範囲で、基礎学問に過ぎないものなのですが...

でも...、全然思い出せない...(涙)。授業中寝ていたのがあだになっちゃいましたねえ...(^;

で、仕方なしに数学のお勉強を一からやり直して、なんとかプログラムにはしてみたのですが...(まだ、FFTは理解していませんので、DFTの定義式を真面目にプログラムしたものです)。これが最悪...。無茶苦茶時間がかかる...(涙)。多分2~3分のデータで丸1日くらいかかるのではないかと...。じゃあってことで、デバッグなどをせずに途中で諦めちゃったというわけです(^;

3次スプラインについては、まだ計算式を理解出来ていないので、試してもいない状況です。ただ、実装しているツールを使ってみると、すごく時間がかかっていましたので、これもダメかなあって思い始めているところです...

ということで、補間に関しては、もうこれ以上はやらないと思います。きちんとした補間をさせたい場合は、それ相応のツールを使って補間させてください<m(\_ \_)m>

## 量子化ビット数 8bit の出力に対応

やはり、ファイルサイズが小さくなることは良い事です(笑)。音もホワイトノイズがうるさく感じる以外はそう大差ありませんし、結構お薦めですね、これは...。  
まあ、サイズを小さくしたければ、CCITTのA-Lawか $\mu$ -LawあたりのADPCMに変換する方がよりベターですが...。あと、妙な音に変わっても良いつてのなら、MPEG Layer-3なんかが極度にサイズが小さくなって来て助かるかも?... (これらの変換はFX2WAVでは対応していません。必要とあらば、別途サウンドレコーダなどを用いて変換してください)。

ところで、どうでもいいことですが、こうやって機能を増やしていく際、オプションの表記がどうしてもダブってしまって困ってます(^; 今回、このオプションを /8 としたのも実はそのあたりに理由があって...

そもそも出力周波数の /of にあわせて /ob としようとしたのですが、これが、オフセットのブロック単位表記とバッティングしてしまって...(^^;

量子化ビット数を /bo、出力周波数を /fo としてしまっても、フェードアウトとバッティングしてしまうし...。ただ単に /b としようとしても、これがバランスとバッティングしてしまうし...。全部オプションの命名を考え直すともう少しマシになるかとは思いますが、そうなってくると、互換性が取れなくなってくるし...。ホント、悩みの種です(苦笑)。

## 8.4.2.MAKELST

### FX2WAV 用インダイレクトファイルの出力に対応

正直言って、LFN を使わなくてもコマンドラインのデフォルト値 128 文字の限界に近いものになるので、これは絶対に必要なものですね。とりあえず、対応させておきました。

### 出力先ファイルの省略が可能に

せっかくだから、省略可能になるようにしておきました。ただ、FX2WAV と違って、上書き確認はしてくれません。

### FX2WAV の追加機能に伴う制御コードの追加

```
*Bit=  
*ReadMode=  
*OutputFrequency=  
*InpMethod=  
*Offset/FX2WAV=  
*StartPoint/FX2WAV=  
*EndPoint/FX2WAV=  
*Loop=
```

等を追加しました。

### サンプルとして『ファースト Kiss 物語』のものを追加

もともと「ミュージックモードの無い」ゲームでも音楽が聴けるようになりたい、との一心からこのツールを作ったので(さすがにそのためだけにやり終わったゲームをプレイする気にもなれないし... )、ちょっと本来の目的から外れてしまっていますが...。

ただ、ループ曲用サンプルとしては、結構良いのではないかと...。

## 8.5.Ver.0.05(1998/7/14)

### 8.5.1.FX2WAV

加算出力のバグをフィックス

ソース上ではたった2文字のミスなのに...(涙)。

画面出力、ドキュメントなど一部修正

スペルミスとか、見つかったものについて、細々と直しておきました。

一部動画フォーマットの抽出に対応

とりあえず、Makelstのサンプルとして付属してあるものについては、オープニングくらいは抽出出来るように、ということで...。ただ、いずれの場合も、データやプログラムを細かく解析していったものでは無く、バイナリエディタで眺めていて「多分こうだろう」と思われるものとして追加しましたので、まともに対応出来ているかちょっと自身がありません。特にJPEGの内容なんてまるで分かっていませんし...(^^;

ま、おまけですから...。

...でも、それにしても、こうやって機能を追加していくごとに、ソースの冗長性が増えていっているのがちょっと...。MIX形式からの音声抽出をする場合なんかが、その典型的なわけですが...。また、オプション類も増えすぎちゃって、作者ですら把握し切れないくらいになっているのもナニだし...(^^;

ホント、本来なら作り直したいところですが、それもまた面倒ってことで、出来ないでいます...(/\_;)

### 8.5.2.MAKELST

サンプルファイルの出力周波数を44.1KHzに統一

Ver.0.04をリリースした後、色々を試してみたところ、どのデータも周波数を水増しした方が音が良いようなので...(^^;  
しかし...、なんで、水増ししなければ、FXより大幅に劣ってしまうんだろ?(涙)

サンプルファイルを別アーカイブとした

結構ファイル数が増えたので、別アーカイブとしました。

FX2WAVの追加機能に伴う制御コードの追加

いちおう、動画抽出関連で、

\*Format=

というオプションを追加しましたが...、実は、書式のチェックなどをせずに、“=”の後ろをそのままFX2WAVに渡しています。だから、FX2WAV側で訳の分からないエラーメッセージに悩まされることがあるかも知れません、もしかしたら...。

## 8.6.Ver.0.06(1998/11/16)

### 8.6.1.FX2WAV

#### 平滑化処理の追加

ザラザラした音をなんとかしたくって、移動平均法による平滑化処理を追加しました。でも...、音がこもるう...(涙)。確かにザラザラ感が少なくなる場合がありますが、それと引き換えに音がこもってしまうんですね。なんか、ガッカリ...。これだったら、ザラザラでも高域が出ている方がいいなあ。

まあ、でも、副産物的にホワイトノイズが減少するので、そういった用途のためだったら、それなりに使い道があるのかも？

それと、何やら色々用意されていますが、処理はいずれも指定範囲内で荷重平均を取っているだけです。重み関数が違うだけで...(^^;

唯一、手の込んだ(?)処理をしているのは、適応化平滑化法のみとなりますが...、実は、これ、本来の目的である、「歪んだ音を滑らかにしたい」ってのには、どうも有用ではないようです(苦笑)。

使えるのは、経験上、二次関数、三角波、指数関数を重み関数としたものくらいのもののみです(でも、どっちにしたって、グライコ処理には負けるかも...。ま、せっかく作ったので、残しておきますけど(^^; )

ちなみに、重み関数には、今の6つの他、ユーザ定義関数を入れようかと思っていましたが、指定がややこしくなるのでやめました。

#### 補間形式のオプションを変更

オプションが結構多くなって来ているので、平滑化処理のオプション指定と同様、オプションにサブオプションを指定する方法に変更しました(以前の /itd, /isl は残していません)

ちなみに、なんで、/itdが/inp=parabolaで、/islが/inp=linearなのかって言うと...。いや、「直線」とか「2次関数」とかって、英語でなんて言うのか知らなくて...(^^; で、以前は、考えるのが面倒になって、Straight LineとTwo Dimensionから、それぞれ /islと /itdって命名をしちゃったわけです。ところが、やっぱり、なんか、しっくり来なかったのも事実で...

で、今回は、「2次曲線」は「放物線」と言い換えられるかなあってことで、「2次関数による補間」に関してはparabolaとしたわけです。(「2乗」からsquareにした方がよかったかな?)

でも、「直線」に関しては、なんにも良い案が浮かばなくて...。私にとっては、Linearってのは、どうしても「線形」ってイメージがあるので、本当にこう書いてしまって良いのかわかりませんでした...。ま、とりあえず、Straight lineよりは、良いような気がしたので、こう置きちゃいました(^^;

## 入力ファイルフォーマットのデフォルトを自動解析とした

ファイルフォーマットを自動で解析し、ヘッダが見つからなかったらヘッダ無しADPCM形式と見立てる、とした方が指定がいくらか楽になるかと思ひまして...(^^;

## オーバーフロー時の処理を選択可能に

Ver.0.03以降、ADPCMの解析時にオーバーフローした場合、そのオーバーフローした値をそのまま残すように変更しましたが、これは、その残す場合と、Ver.0.02以前のリミッタをかける場合とを選択出来るようにしたものです。

ちなみに...、オーバーフロー時にその値を周囲の点から補間する、という方法も思いついたのですが...。実際作ってみると、その部分だけ、ポコポコと妙な音が出てしまい、とても使えるものではなかったので、削除しました。

## FFT(高速フーリエ変換)によるフィルタの追加

波形がどんなものになっているのかなって思って、GoldWaveというソフトを試用して、見てみたのですが...。酷いもんだ...(^^; なぜ、こうなるのかは結局分かりませんでした。とりあえず、同ソフトでハイパスフィルタをかけてみると、綺麗に中央に寄ってくれるではないですか!

ということで、急遽、FFTによるフィルタを作成しました(ようやく理解出来ました(^^;)。ちなみに、あんまり遅すぎるのもナニなので、N=4096としています。

でも、後でよくよく考えてみれば、中央に寄せるのが目的なら、何もFFTをかけなくても、区間平均をとって補正するだけでよかったですよ...(^^;

それから...、作っている最中、FFTを施すある区間とその次の区間との間の連続性が失われてしまって、高域のノイズがいっぱいになってしまい、非常に困りました。とりあえず、FFTを施す区間をある程度重ねながらかけるという手法で多少はマシになりましたが...。こういうのって、ホントは、どうやって対処すればいいんでしょうね?

まあ、なにはともあれ、これで、一部のソフトでMP3に変換した場合でも、そんなに変な音にはならなくなったので、OKかな? PCエンジンの音も上手くグライコをかければ十二分に聴ける音になるし...(^^-^^)

## 周波数変換の補間方法に最小二乗法によるものを追加

とりあえず、入れておきました。ただ...、サンプル点一点一点すべてについて連立方程式を解く(Gauss法)ので、ムチャクチャ遅いんですけど...(^^; 450MHzくらいのマシンだったら我慢出来るかなあ(^^;

## 周波数変換の補間方法に三角関数補間を追加

これも...(^^; これは、最小二乗法より酷くて(^^; サンプル点一点一点すべてについてフーリエ級数展開をしていますので、さらに遅いです(爆)。まあ、1GHzくらいのマシンだったら我慢出来るかも? でも...、音質は最小二乗法とかのほうがよいやうな...(^^;

それにしても、これについても、FFTが応用出来れば、もう少し実用的な速度になりそうなんですけど...、やり方が分からん...(^^;

## 周波数変換の補間方法にベーススプライン補間を追加

実は、Ver.0.04あたりからず~っと入ってはいたんですが、デバッグ用の隠しオプションとしていました。今回、FFTによるフィルタを追加したおかげで、少々音質の悪さは挽回出来るようになりまして、晴れて正式オプションとしたという...(^^;

## RIFF/WAVE フォーマットに 1 種類だけ対応

後からフィルタなどをかけたいってことが結構多かったため、WAVEファイルからの読み込みが出来るようにしました。

対応フォーマットは、リニア PCM の 8 ビット、もしくは 16 ビットのみです。

このフォーマットは 3 チャンネル以上にも対応出来るようですが、FX2WAV で読み出せるの 1 チャンネル、もしくは 2 チャンネルのものです。一応、そういったフォーマットが読み込まれたときも対処出来るよう、最初の 2 チャンネル分のみ読み込むようにプログラミングはしていますが、なにせ、3 チャンネル以上の WAV ファイルって私は持っていないもので、確認が出来ないでいます...。あ...、後、0 チャンネルってのがもしあったとして、そういったモノを読み込ませた場合、どうなるか分かりません(^^; それから...、サンプリング周波数については、多分、どれでも行けるのではないかと...

## RIFF/ADP フォーマットに対応

RIFF 形式の読み込みに対応したんで、せっかくなら、同じ RIFF 形式となっている FXGA 標準 ADPCM フォーマットの方も読み込めるようにしておこうかな、と思ひまして...

どの程度使えるかは、正直言って不明です...(START.ADP とか、そんなのしかないし...)

## 最適化オプションを指定してビルド

Enterprise 版にバージョンアップしたため、最適化をかけることができるようになりました。でも...、大して速くなってないかな？

## ドキュメントの PDF 化

いや、何でって言われても困りますけど...(^^;

とりあえず、図が記入出来るような形にしたいって...。HTML はファイルが増えちゃって嫌だし、Word97 だと、誰でも読めるというわけにはいかないでしょうし...

ということで、PDF での配布にしました。

ただ、実際の話、PDF で完璧な文書が作れるかと言うとそうでもないわけで...(ごによごによ...)。しかも、今回は、ファイルサイズを考慮して、より印刷物に近い出力が得られる PS ファイル経由 (Distiller Assistant 経由) ではなく、Acrobat Writer 経由としていますし...。そんな訳で、若干変なところ (フォントが重なるとか...) もあるかも知れませんが、ご了承を...

## 「参考文献」項目を追加

常識的に考えて、このツールを作るにあたり、参考にしたものは書いておく必要があるかな、と思ひまして...

とりあえず、ひと目見たといったレベルのものも含めて、思いつくかぎり書いておきました。

## 8.6.2.MAKELST

制御コードに \*inline= を追加

よくよく考えてみれば、FX2WAVをバージョンアップすることにMAKELSTもバージョンアップというのは面倒なので、そのままオプションを渡す制御コードを追加しました。ただ、これに伴い、スペース区切りによるデータ指定やコメントなどは出来なくなっています。ドキュメントにも元々そのようには書いていないつもりでしたが、一応ご注意ください。まあ、なににせよ、コメントは行頭に、データの区切りはTABで統一するようにしてください。

『ファースト Kiss 物語』の高音質版サンプル追加

Ver.0.06で追加された機能のうち、最小二乗法による補間とFFTによるフィルタ処理を使ったサンプルを別に用意しました。なにせ、これらって、ずいぶんと時間のかかる処理でして、デフォとするにはあまりにもナニなので...(^^;

『ミラクルム』のサンプル修正

途中でMAKELSTファイル内のサンプリング周波数を変えてしまったのか、いつの間にか、途中から変換開始位置に1~2秒のズレが生じてしていました。時間単位の場合、こういった問題を常にはらんでいることは分かり切っていることではあったのですが...(^^; とりあえず、バイト単位に直して、すべて書き換えておきましたm(..)m

## 8.7.Ver.0.07(1998/12/28)

### 8.7.1.FX2WAV

#### BAK ファイルを作成する手順を変更

入力ファイルが WAV ファイルのとき、最初にバックアップを作成して処理していましたが、よくよく考えれば、最後に WAV ファイルに落す時に元ファイルをリネームする方が時間がかからずに済むことに気付いたので、そのように変更しました。

#### BAK ファイルの上書き確認を止めた

さすがにここまで確認するというのは、しつこいかと思いましたが、確認は止めにしました。

#### 上書き確認に「すべて上書き」と「変換の中断」を追加

当然やっておくべきだったものではありませんが(^\_^; 今回からは対応しました。

なお、今回の「変換の中断」の追加により「上書きしない」選択肢は、そのファイルを変換しないで、次のファイルの変換作業に移る、という動作になりました(インダイレクトファイルでの指定の場合)。...こちらも当たり前のことですが、今までは面倒でやっていなかったことですm(..)m ...でも、ものの4~5分で追加作業が済んでしまったため、もっと早くやっておくべきだったと、自己嫌悪に陥りました(^\_^;

#### 出力チャンネル数のデフォルトを入力チャンネル数と同一にした

WAV、ADP、MIX、ARS など、ヘッダからチャンネル数を取得出来るフォーマットでそれがステレオとなっている場合でも、今までは /s などと明示しない限り、勝手にモノラル出力となっていました。

しかし、さすがにこれは不便なので、/m などと明示しない限り、入力チャンネル数のまま出力するように仕様変更しました。

#### ログファイルの出力に対応

いいかげん作っておかなければならないとは思っていましたが、なかなか出来ずにいました...。が、とりあえず、実装しました。

#### ループ処理の強化

毎ループごとに PCM 値とスケールレベルの初期化を行うかどうかを指定出来るようにしました。どうやら、『虚空漂流ニルゲンツ』は、これらを初期化しないでループさせると上手くいくようです。

#### ヘッダ無し PCM からの変換に対応

とりあえず、GMAKER スタータキットプラスに入っている、おなじみの WARNING.PCM (/f44100 /bs2 /s /fmt=pcm16, complement /hf) などからの変換が出来るようになります。

あと、色々パラメータを振れるようになっていきますから、その他のマシンでもあるいは使えるものがあるかも? (^\_^;

## FFT フィルタで、同一周波数が指定された時の処理を追加

バンドパスフィルタやグライコをかける時、周波数×増幅率の直線で近似して、近傍の周波数2点間の増幅率を求めているのですが、今までは、同一周波数が複数含まれていた場合、傾きが+、もしくは- となって、まともに動作しませんでした。  
こういった場合に例外処理を加えることで、同一周波数が複数ある場合にもまともに動作するようにしました(とはいうものの、こういった場合には大きなノイズが発生してしまいますが...(^^ゞ)

## グライコパラメータのソートを廃止

上の同一周波数対応により、ソート処理の動作が怪しくなりましたので、とりあえず、廃止しておきました。

## ノイズ重畳処理の追加

これは...(^^;  
もともと、あのFXのザラザラ感がADPCMの分解能の限界でADPCMデータに変換する前のデータを正確に記録できていないためなのでは? という考えから出てきたものでして...。で、どうせアテにできないデータなら、そこにノイズを重畳してやれば、マクロ的に見ればまともなデータになりはしないかな、という考えで追加したという...。でも...、余計に酷くなるだけでしたね(爆)。まあ、当たり前ですけど...(^^;  
ホント...、ほとんどやけくそでザラザラ対策をしています(^^;

## 飛び石補間の追加

これも、ノイズ重畳処理と同様に、ADPCMのデータに疑いを持って追加したものです。でも、やっぱり、巧くいかない(涙)。音を聴いた限り、移動平均よりも平滑化効果が緩い平滑化としか言えそうもありません(^^;  
ちなみに、「飛び石補間」というのは、どう言えば良いのか分からず、苦し紛れに私が勝手に作った造語ですので、他では使わないように...(^^;  
しかし...、これ以上何も思い付かないなあ...。ザラザラノイズ対策...。ホント、どうしましょ...(;\_;

## FFT フィルタルーチンの見直し

掛け算を少なくする方向で見直しを行いました。それにより、多少は速くなったようです。それと...、一部演算精度によるバグが紛れ込んでいましたので、それも直しておきました。

## 最小二乗法ルーチンの見直し

今まではpow関数を使って累乗を求めていましたが、よくよく考えてみれば、正の整数乗しかしないので、double型で累乗するpow関数はちょっと贅沢かと思ひまして...。で、pow関数を全廃してループで累乗を表現したわけですが...、今までの半分以下の時間で周波数変換が出来るようになってしまいました(^^; な...、なんだかなあって感じですが、とりあえず、使えるレベル(K6-2/333MHzの場合)になりましたんで、結果オーライかな?

## 三角関数補間ルーチンの見直し

今までは、フーリエ級数展開の定義式どおりに処理を行っていましたが、これだとピープ音みたいな妙なノイズが重畳されてしまいますので（サンプル数を増やしても改善しません。実際、N=50くらいまでサンプル数を上げて試してみましたが、ノイズは相変わらず乗っていました(;\_;)）最初に区間平均をとってゼロ補正し、補間した後にその平均値を加えるという方法に変えました。つまり、 $a_0$ の項が0となるようにあらかじめバイアスをかけ、後でそのバイアス分を戻すという方法となります。これにより、あの妙なノイズも出なくなり、処理に時間がかかるという欠点以外は、けっこう使える補間方法になりました(^\_^)v  
音的には、多項式補間に比べると、鮮烈な感じのする音と言えそうです。ただし、元データがザラついているものに関しては、そのザラつきがより強調されてしまいますので、時と場合を考えつつ使う必要があります。  
あと、アルゴリズムの若干の改良により、多少処理が速くなりましたので（といっても微々たるものだけど...(^\_^;)、サンプル数のデフォルトを15から25に引き上げました。

## ステレオ時のチャンネル合成処理を追加

アルバレアの乙女のデータは、ゲーム中はモノラルで鳴っているにもかかわらず、2ch分のデータがあることが非常に疑問だったのですが...。何のことはない、ただ単に両チャンネル加算するだけで良かったみたいです。今までいくらやっても上手く音が出なかったのは、波形に変化を加えた後だったからみたいです。  
ホント、これだったら、加算は最初にするようにプログラミングするべきでした...。  
...と、いうわけで(^\_^;、ほとんど『アルバレアの乙女』専用オプションともいえる、/Mixオプションを急遽作り、これに対処しました。

## 疑似ステレオ出力に対応

何となく作ってみました。  
例によって、処理はけっこういいかげんですので、過度な期待はしないで下さい(^\_^;、...いいかげん故に、モノによっては聞くに耐えられないものもあるかも？(^\_^;

## 無音部指定の追加

FX2WAVでは、FFTの処理をする時、ファイル先頭とファイル終端で、それぞれ他端のデータを補ってFFTをかけています。このため、ファイル終端が0以外の値で飽和していたら、IFFTをかけた時、大きなノイズとなって現れてしまいます。  
『虚空漂流ニルゲンツ』のデータなんか正にそれで（抜き出し方を間違っているのだろうか？(^\_^;)、暫定対策として追加しました。

## FFT演算精度の指定が可能に

疑似出力に対応したのはいいのですが、一部のデータでプチプチとレコードみたいなノイズが乗ってしまって困っていたのですが、その回避策として、このような物を追加しました。このオプションでFFTをかけるサンプル数を増やすことにより、それなりには効果はあげられ、また、全体的な音質も向上しますが、処理時間の方も大幅に増してしまいます(;\_;)というわけで、時と場合を考えつつ使うのが一番のようですね。

ドキュメント作成を Adobe PageMaker に

すぐに死にまくる上に、張り付けてある図とのリンクが何かと言えば簡単に途切れてしまう、Word97 に嫌気が差していたので、PageMaker を買って、そちらに乗り換えました。けど、これはこれでけっこう不満があるなあ...。OK ボタンが右に配置されていたり、OK が 'y' キーで代用出来ないなど、Windows ユーザのことをほとんど考えずに作ってあるし...。あと、間違っつけて索引の書式を一括変更出来ないとか、ワープロソフトでは当たり前のアウトライン機能が無い(もしかして、見落としている?)とか、大量文書を書くにはちょっと辛いものに仕上がっているしなあ...。どっちかっていうと、これはパンフレットとか、そういったものを書くのに適しているって感じですかねえ...。ちょっと、ガツカリ。ま、でも、PDF に落とすっていう点では、いいツールですね。目次とか、索引とか、ハイパーリンクとかを全自動で引き継いでくれますから。今までの Word97+PDF Writer だったら、Acrobat Exchange 上で一つ一つコツコツとリンクを張って目次を作っていましたが、その作業が無くなって、非常に助かりました(^\_^)

MIX/RTF 形式のバグを修正

MIX/RTF 形式にバグがありましたので、修正しました。具体的には、符号無しとしなければならぬところを、符号付きでコーディングしていました。また、無いはずのデータを読み込んで、その後のデータがムチャクチャになる、というも修正しました。

拡張グライコを追加

疑似ステレオ化の処理を追加した後に気付いたのですが、左右のチャンネルを個別にグライコをかけられ、位相も変化させられるようになると、もっと自由度が上がるかな、って思いました...

というわけで、追加しました。これにより、どの程度の補正が効くようになるのかは、正直言ってよく分かりませんが...(^\_^; (まだ、ほとんど試していない)

インダイレクトファイルにコメントを追加できるようにした

...とは言っても、**行頭に“;”がある場合**だけですけど...。とりあえず、動作チェックをしていて、特定行のみを変換させたい、といったことが多かったので、追加しました。

## 8.7.2. MAKELST

出力周波数を指定しない場合は /of オプションを書き出さないようにした

なんか、出力サンプリング周波数を指定しない場合、必ず 31600Hz が出力サンプリング周波数となる、間抜けな仕様となっていたみたいなので、修正しておきました。

サンプルファイルを同一ゲームに 2 種類用意することにした

Windows98 に代えてからなのか、以前のように、わざわざ 44.1KHz にリサンプルしなくても、良い音が出るようになりましたので、今までのサンプルの方をリサンプル無しとしました。変換時間も大幅に削減できることですし、ファイルサイズも 44.1KHz ほどは大きくなりませんしね(^\_^)

ただ、それだけだと、せっかく色々な機能をつけている意味がありませんので、それとは別に、長時間かけて変換するサンプルというものも用意しました(一部を除く)。こちらは、今までの近傍 3 点を通る 2 次関数で補間する方法から、近傍 4 点を通る 3 次関数で補間する方法に変えています。そして、FFT を用いて、データのズレを補正します。また、モノラルデータの場合は、全て疑似ステレオ化するものとしました。というわけで、こちらのサンプルの方は、処理に非常に膨大な時間を要し、さらに出来上がってくるファイルもそのサイズが非常に大きなものとなることをご注意ください。

## サンプルファイルをいくつか追加

『ブルーブレイカー ～剣よりも微笑みを～』、『アンジェリーク Special 2』、『アンジェリーク 天空の鎮魂歌』のものを追加しました。  
いずれも全データモノラル音声のものとなっていることからお分かりのように、疑似ステレオのサンプル的な意味合いでつけました。

## 8.8.Ver.0.08(1999/3/7)

### 8.8.1.FX2WAV

#### 位相シフト平滑化(仮称)の追加

なんでかよく分かりませんが、疑似ステレオ化を行うと、あれだけ除去に苦労していた(使えない機能がいっぱい増えていることでお分かりですよ? (^;)) ザラザラノイズがある程度軽減してくれていましたので、ステレオデータにも使用出来るよう、それとほぼ同一の処理を施すものを追加しました。  
多少癖のある音になりますが、これでとりあえずはザラザラノイズ対策も終結として良さそうかな?

#### インダイレクトファイルの一括処理ファイル数の制限を緩和

今までは、256 ファイルまでしかいっぺんに変換できませんでしたが、2048 ファイルまでに拡張しました。  
PCエンジンのボイスデータ抜き出しをされていて気付いた不具合ですが、本文中にもあるように、これは明らかな設計ミスです <m(\_\_)m> 　ただ、これを今から修正するのは、また手間がかかるため(新たなバグが発生する可能性も高いし(^;)), 制限を緩和するに留めておきました。2048 ファイル以上変換したい場合は、インダイレクトファイルを複数に分けて記述してください <m(\_\_)m>

#### 著作権情報の記載を可能に

抜きだし元の著作権情報が表示されるとけっこう格好いいかなってことで、色々と試行錯誤をし、LIST チャンクに記載されている著作権情報の付加、及び、WAV ファイルからの抜き出しで、LIST チャンクがあった場合の LIST チャンクの継承・修正に対応しておきました。ただし、一部よくわからなくて、暫定対処しているところもありますので、もしかしたら、上手く付加できない項目、上手く継承できない項目もあるかもしれませんが、その場合はご了承ください。

この対応により、WAV ファイルの詳細プロパティ (Windows98 の場合、Windows95 の場合こういった表示がされていたか覚えていないのでよく分かりません) や Windows Media Player で表示されるようになります。ただし、WAV 詳細プロパティの WAV アイコンの代わりに表示されるグラフィックと「表示名」項目については、DISP という別のチャンクに記載されており、とくにグラフィックのデータ形式 (DIB 形式らしいですが) がよく分からなかったため未対応としています (継承処理もしていません)。また、Windows Media Player の「ショー」欄、「作成者」欄、「規制」欄、「説明」欄も、どこをいじれば出てくるのものなのか分からず、対応していません。

#### MIX/RTF ファイル抽出の不具合を Fix

Ver.0.07 で、MIX/RTF 関連のバグを修正しましたが、それと同時にソース上で若干の変更をかけました。ところが、その変更が中途半端であったため、新たなバグが発生してしまっていたようです。  
ということで、それを修正しておきました。

#### 一部整数表示をカンマ区切り数値に変更

今まで、経過表示の処理済容量などで、桁数の非常に多い表示がなされて、数値をパッと目で判断しづらかったのですが、そういった一部の数値をカンマ区切りの数値で表現するように変更しました。

統計値情報にいくつか項目を追加

再生時間、データの間値（最大値と最小値の間値）、標準偏差を表示するようにしました。これを元に波形を処理することはできませんが...、とりあえず、「統計値」と書くにはあまりにも寂しかったので(^^;

## 8.8.2.MAKELST

『アンジェリーク Special2』用サンプルの記述ミス修正

Ver.0.07で追加した『アンジェリーク Special2』のサンプルファイルに記述ミスが見つかりましたので、それを修正しておきました。

位置指定を間違えてしまい、最後のファイル「なんだか悲しい曲.wav」の後ろに、音声データがくっついてしまっていたんです。しかも、モロにネタバレになるものが....

本当に申し訳ありません<m(\_\_)m> >まだプレイしていないにも拘らず、聴いてしまった方

『アンジェリーク 天空の鎮魂歌』用サンプルの記述ミス修正

こちら、位置指定を間違えてしまい、最後のファイル「面影.wav」の後ろに、SEデータがくっついてしまっていたので、修正しました。

すべてのサンプルファイルに著作権情報を付加

せっくなので、つけておきました。

ただ、この著作権情報って、一般的でないのか、サウンドレコーダでは表示されないし、他の形式（MP3やWAVファイルの別形式など）に変換したときにちゃんと継承されることも無いようですし、あまり役に立たないかもしれません(;\_;

いくつかのサンプルファイルで位相シフト平滑化（仮称）を行うように修正

『ミラクルム』のサンプルファイルで、抜きだし元がステレオ音声になっているものについては、位相シフト平滑化（仮称）を行うように修正しました。

いくつかの音声抜き出し用サンプルを追加

例えば、Windows起動時に「素敵な予感、微風にのせてあなただけに届けます」とか、アプリケーション起動時に「呼ばれて飛び出てじゃーんじゃじゃじゃ〜ん」とか、一般保護例外発生時に「逆恨みのどこが悪い〜！オレに教えてみるお〜っ!!」とか、色々と言ってくれと面白いかと思ひまして...(^^;

ということで、『ときめきメモリアル』、『ときめきメモリアル体験版』、『エメラルドドラゴン』、『Linda3』、『プライベート・アイ・ドル』、『天外魔境II』（いずれもPCエンジン版）から音声データを抜き出すサンプルファイルを新たにつけました。

ただし、これらのサンプルはファイル数が非常に多いため、変換時間、変換ファイルサイズともどのくらいになるのか見当もつきませんので（後者は計算すればすぐに分かりますが...(^^);）リサンプルを含め、何も加工せずに出力しています。必要とあらば、必要になったファイルのみ、手動でそういった加工を施してください。

また、これらのファイルは、とても手作業で扱える数ではないので、非常に単純なプログラムを作って音声データの位置を検索しています。このため、すべて手作業で行っているほかの音楽抜きだしファイルと比べて、ケアレスミスこそ無くなるものの、抜きだし漏れなどが発生している可能性があります。その点はご了承ください。

## 9. 参考文献

基礎解析学 改訂版 / 矢野健太郎・石原繁 共著 / 裳華房  
C言語によるはじめてのアルゴリズム入門 / 川西朝雄 著 / 技術評論社  
C言語による最新アルゴリズム事典 / 奥村晴彦 著 / 技術評論社  
Cによる科学技術計算 / 小池慎一 著 / CQ出版  
科学計測のための波形データ処理 / 南茂夫 編著 / CQ出版  
FORTRAN77 数値計算プログラミング 増補版 / 森正武 著 / 岩波書店  
デジタル信号処理の基礎 / 三上直樹 著 / CQ出版  
General RIFF File Background  
/ General RIFF description provided by Robert Shuler  
/ <http://www.wotsit.org/wmusic/wav.zip>  
Visual Studio 6.0 Enterprise 付属 msdn Library  
/(c)1992-1998 Microsoft Corporation. All rights reserved.  
GMAKER スタータキットプラス付属ドキュメント C6230.WRI  
/ デバイス : HuC6230 の解説 / (c)1995 HUDSON SOFT  
GMAKER スタータキット付属ツール MpConv2 のヘルプ MpConv2.hlp  
/ (c)1995 HUDSON SOFT

## 10. 免責等

このツールはフリーソフトとします。

このツールの転載、配布等に関しては、アーカイブ内容を改変しない限り自由とします。とくに連絡の必要もありません。

作者はサポート、バージョンアップなどの義務を負わないものとします。

このツールは作者が独自に調べて作成したものです。そのため、PC-FXのハードメーカーである、NECホームエレクトロニクスやハドソン、及びその他のサードパーティへの質問はしないでください。

CD-ROMから抜き出したデータ、及び、このツールによって変換されたデータは、フォーマットこそ異なるものの、その著作権は各ソフトを制作・製作・販売したメーカーにあります。そのため、個人での利用に留め、配布や2次利用などはしないでください。

このツールはMS-DOSの知識をある程度有している方を対象としています。あらかじめご了承下さい。

このドキュメント内に登場する製品名は、一般に各社の商標、あるいは登録商標です。

このツールにより生じたいかなる問題も作者はその責務を負わないものとします。ご使用になれる場合は、使用者個人の責任においてお使いください。

1999/3/7 こう F X

BIGLOBE([magaki@mvh.biglobe.ne.jp](mailto:magaki@mvh.biglobe.ne.jp)/SWN79986)

NIFTY-Serve([magaki.kohei@nifty.ne.jp](mailto:magaki.kohei@nifty.ne.jp)/YHQ00212)

# 索引

## 記号

/Bn 29  
/BSn 26  
/CF 29  
/dBn 28  
/DO 29  
/EdNSn 29  
/F 14  
/FFT=n 30  
/Fil=xxxx 21  
/Fin 29  
/FMT 14  
/FOn 29  
/HF 15, 24  
/IARL=xxxx 31  
/IART=xxxx 31  
/ICMS=xxxx 31  
/ICMT=xxxx 31  
/ICOP=xxxx 31  
/ICRD=xxxx 31  
/ICRP=xxxx 31  
/IDIM=xxxx 31  
/IDPI=xxx 31  
/IENG=xxxx 31  
/IGNR=xxxx 31  
/IKEY=xxxx 31  
/ILGT=xxxx 31  
/IMED=xxxx 31  
/INAM=xxxx 31  
/INP=xxxx 17  
/IPLT=xxxx 31  
/IPRD=xxxx 31  
/ISBJ=xxxx 31  
/ISFT=xxxx 31  
/ISHP=xxxx 31  
/ISL 17  
/ISRC=xxxx 31  
/ISRF=xxxx 31  
/ITCH=xxxx 31  
/ITD 17  
/L- 25, 28  
/LF 24  
/lf 15  
/Log 30  
/Log=Off 30  
/LP 14  
/LPn[,init] 24  
/M 25  
/MB 28  
/ME 28  
/MIXn 30  
/MM 28  
/NC 29  
/noise=aaaa[,bbbb][,cccc][,x] 16  
/NV 29  
/NZA 28  
/On 26  
/Ovf=xxxx 16  
/PO 29  
/R- 25, 28  
/S 25  
/SA 25  
/SL 25  
/SM=xxxx 18  
/SR 25  
/SS[=[n][,[m][,[x1,y1][,x2,y2][,...]]]] 25  
/SSInp[=Off] 16  
/StNSn 29  
/TE 14  
/TEn 26  
/TS 14  
/TSn 26  
/V 29  
/Xn 28  
/ZA 28  
\*Balance=x 36  
\*BlockSize=n 35  
\*Channel=n 35  
\*EndPoint/FX2WAV=n 35  
\*EndPoint=xxx 34  
\*FadeIn=n 36  
\*FadeOut=n 36  
\*Frequency=n 34  
\*Information=xxx 37  
\*Inline=xxx 37  
\*InpMethod=xxx 35  
\*Loop=n 35

\*MAXSeek=xxx 36  
\*Offset/FX2WAV=n 35  
\*Offset=n 34  
\*OutputFrequency=n 34  
\*ReadMode=xxx 35  
\*ReverseL=xxx 36  
\*ReverseR=xxx 36  
\*StartPoint/FX2WAV=n 35  
\*StartPoint=xxx 34  
\*Track=n 34  
\*Volume=x[dB] 36  
\*WriteConfirm=xxx 37  
\*ZeroAdjust=xxx 36

## A

Adaptive[n][,y] 20  
Add 16  
ADDOUT 35  
ADP 15  
ADPCM 解析順序の指定(MAKELST) 35  
ADPCM 後置 14  
ADPCM の解析順序の指定 24  
ADP 形式 14  
AGE[L|R]x1,y1,a1[,x2,y2,a2[,x3,y3,a3...]] 23  
ARBULEA 15  
ARS 形式 14  
AUTO 14

## B

BaseSpline 17  
BLOCK+ 34  
BLOCK- 34  
BOTH 36  
BP[k,l[,m,n]] 22  
BYTE 34

## C

CDREAD 8  
Complement 15  
CONFIRM 37  
Cosine[n][,y] 19

## D

DataInfoOnly 37

## E

EACH 36  
Exponential[n][,[a][,1]] 19

## F

FFT(高速フーリエ変換) 21  
FFT 演算精度の指定 30  
FX2WAV にそのまま渡す(MAKELST) 37  
FX2WAV ファイル内変換位置の指定(MAKELST) 35

## G

Gauss 法 17  
GEx1,y1[,x2,y2[,x3,y3...]] 22, 23

## H

HighFast 35  
HP[m[,n]] 21

## I

Init 24

## L

LAST[ffff[,aaaa[,ssss]]] 14  
Limit 16  
Linear 17  
LOUT 35  
LowFast 35  
LP[m[,n]] 21

## M

MAKELST 形式 12  
MANUAL 36  
MIX 形式 14  
MONORAL 35  
Mul 16  
M 系列信号 16

## N

NIRGENDS 15  
NotConfirm 37

## O

Off 18, 21

## P

Parabola 17  
Parabola[n],,1 20  
Parabola[n][,y] 18  
PCM 15  
PCM 値の初期化 24  
PhaseShift[l][,m] 20  
ProgressiveInfoOnly 37

## R

Rand 16  
ROUT 35  
RTF 形式 14

## S

SECTOR+ 34  
SECTOR- 34  
Signed 15  
Simple[[n] 18  
SLI 35  
STEREO 35  
Straight 16

## T

TDI 35  
Triangle[n][,y] 18  
Trigonometric[n] 17

## U

Unsigned 15

## V

VERBOSE 37

## W

WAV 形式 14  
WAV ファイルの限界値 16  
White 16

## ア行

アーカイブの場所 31  
アーティスト 31  
明るさ 31  
アドレスの丸め方(MAKELST) 34  
アドレス表記 11

アルバレアの乙女動画形式 15  
位相シフト平滑化 20  
インストール 10  
インダイレクトファイル 13  
上書き確認の指定 29  
上書き確認の指定(MAKELST) 37  
上書き確認をしない 29  
上書き確認をする 29  
エンジニア 31  
オーバーフロー処理 16  
大きさ 31  
オプションの優先度 32  
オプション文字列の指定(MAKELST) 37  
オフセット 26  
オフセットの指定(MAKELST) 34  
音量 28  
音量の指定 28  
音量の指定(MAKELST) 36  
音量バランス 29

## カ行

開始アドレスの丸め方の指定(MAKELST) 34  
解析順序 24  
下位 4bit 上位 4bit 24  
下位 4bit 上位 4bit(MAKELST) 35  
各チャンネルごとに最大値を取得 28  
各チャンネルごとに最大値を取得(MAKELST) 36  
拡張グライコモード 23  
加算により重畳 16  
加算モノラル出力 25  
片チャンネルあたりのバイト数 26  
画面表示の指定 29  
環境 9, 9  
キーワード 31  
疑似ステレオ出力 25  
疑似ホワイトノイズ 16  
技術者 31  
切り捨て 16  
グライコモード 22, 23  
グラフィックイコライザー 22, 23  
クロップ 31  
経過表示のみする 29  
限界値 16  
減算モノラル出力 25  
高域通過型フィルタ 21

高域と低域をカットする	22	ステレオ入力、ステレオ出力	25
高域をカットする	21	ステレオ入力、ステレオ出力(MAKELST)	35
高次関数への最小二乗フィット	17	すべて表示する	29
虚空漂流ニルゲンツ動画形式	15	正規化しない	28
コマンドライン	32	正規化の指定	28
コミッション	31	正規化の方法(MAKELST)	36
コメント	31	整数表記	11
コンフィグファイル	32	製品	31
<b>サ行</b>		セクタ単位	12
最小二乗法による補間	17	セクタ単位に切り上げる(MAKELST)	34
最大値の取得	28	セクタ単位に切り捨てる(MAKELST)	34
作成日	31	ゼロ補正の指定	28
差分	16	ゼロ補正の指定(MAKELST)	36
左右の音量バランス	29	ゼロ補正をしない	28
差をとる	25	ゼロ補正をする	28
三角関数による補間	17	先頭無音部時間	29
三角波	18	鮮明度	31
サンプリング周波数	16	ソース	31
サンプリング周波数の指定(MAKELST)	34	ソースの形式	31
サンプリング周波数の変換	17	増幅率	28
サンプリング周波数変換(MAKELST)	35	そのまま渡す(MAKELST)	37
時間単位	12	ソフトウェア	31
指数関数	19	<b>タ行</b>	
システムデフォルト	32	帯域通過型フィルタ	22
ジャンル	31	タイトル	31
終端無音部時間	29	単純移動平均法	18
周波数領域でのフィルタ	21	チャンネル合成	30
終了アドレスの丸め方(MAKELST)	34	チャンネル数の指定	25
16進整数	12	チャンネル数の指定(MAKELST)	35
10進整数	12	チャンネルを反転	25
出力サンプリング周波数	16	直接出力	16
出力サンプリング周波数の指定(MAKELST)	34	直線補間	17
出力の偏り	21	直線補間(MAKELST)	35
出力ファイル	13	著作権	31
出力ファイルの量子化ビット数の指定	24	データの反転	28
瞬時値	16	データの表示のみする	29
上位4bit 下位4bit	24	データ反転の指定(MAKELST)	36
上位4bit 下位4bit(MAKELST)	35	低域通過型フィルタ	21
乗算により重畳	16	低域をカットする	21
小数表記	11	適応化平滑化法	20
情報表示の指定(MAKELST)	37	デコードの開始位置	26
スケールレベルの初期化	24	デシベル	28
ステレオっぽい音	25	ドット/インチ	31
		飛び石補間	16
		トラックの指定(MAKELST)	34

## ナ行

なにも表示しない 29  
名前 31  
2次関数による補間 17  
2次関数補間(MAKELST) 35  
2次多項式 18  
2次多項式適合移動平均法 20  
入力サンプリング周波数 16  
入力ファイル 13  
入力ファイルフォーマット 14  
ニルゲンツ動画形式 15  
ノイズ重畳 16

## ハ行

バイト単位 12  
バイト単位のまま(MAKELST) 34  
ハイパスフィルタ 21  
バランス 29  
バランスの指定(MAKELST) 36  
パレット設定 31  
反転 28  
バンドパスフィルタ 22  
左チャンネル出力(MAKELST) 35  
左チャンネル反転 28  
左チャンネル反転(MAKELST) 36  
左チャンネルモノラル出力 25  
ビッグエンディアン 15, 24  
一つ前のデータとの差分 16  
表示しない 29  
表示の指定 29  
フーリエ級数展開 17  
フィルタの指定 21  
フィルタをかけない 21  
フェードアウトの時間 29  
フェードアウトの時間(MAKELST) 36  
フェードインの時間 29  
フェードインの時間(MAKELST) 36  
フェードイン・フェードアウト時間(MAKELST) 36  
ブロックサイズの指定 26  
ブロックサイズの指定(MAKELST) 35  
ブロック単位 12  
ブロック単位に切り上げる(MAKELST) 34  
ブロック単位に切り捨てる(MAKELST) 34  
ベーススプラインによる補間 17  
平滑化しない 18  
平滑化法の指定 18  
平均をとって平滑化 18

ヘッダ自動解析 14  
ヘッダ無しADPCM形式 15  
ヘッダ無しPCM形式 15  
変換開始位置の指定 26  
変換終了位置の指定 26  
変換処理中の画面表示の指定 29  
補間方法の指定 17  
補間方法の指定(MAKELST) 35  
ホワイトノイズ 16

## マ行

右チャンネル出力(MAKELST) 35  
右チャンネル反転 28  
右チャンネル反転(MAKELST) 36  
右チャンネルモノラル出力 25  
無音部の指定 29  
メディア 31  
モノラル入力、モノラル出力 25  
モノラル入力、モノラル出力(MAKELST) 35

## ヤ行

ユーザデフォルト 32  
余弦波 19  
読み飛ばし中 26

## ラ行

乱数によるノイズ 16  
リトルエンディアン 15, 24  
量子化ビット数 24  
両チャンネル加算モノラル出力 25  
両チャンネル加算モノラル出力(MAKELST) 35  
両チャンネルから最大値を取得 28  
両チャンネルから最大値を取得(MAKELST) 36  
ループ回数の指定 24  
ループ回数の指定(MAKELST) 35  
連立方程式 17  
ローパスフィルタ 21  
ログファイルの指定 30  
ログファイルを作成しない 30  
ログファイルを作成する 30

## ワ行

和をとる 25