

# NITRO-Composer

## サウンドアーカイブマニュアル

Ver 1.7.0

任天堂株式会社発行

このドキュメントの内容は、機密情報であるため、厳重な取り扱い、管理を行ってください。

# 目次

1	概要 .....	7
2	構成 .....	7
2.1	サウンドアーカイブ定義ファイルのサンプル .....	7
2.2	セクション .....	8
2.2.1	波形アーカイブセクション .....	8
2.2.2	バンクデータセクション .....	9
2.2.3	プレイヤー情報セクション .....	9
2.2.4	シーケンスデータセクション .....	9
2.2.5	シーケンスアーカイブセクション .....	9
2.2.6	ストリームプレイヤー情報セクション .....	9
2.2.7	ストリームデータセクション .....	9
2.2.8	グループ情報セクション .....	9
2.3	コメント .....	9
2.4	ラベル .....	9
2.4.1	ラベルファイル出力 .....	10
2.4.1.1	サウンドラベルファイル .....	10
2.4.1.2	サウンドアーカイブラベルファイル .....	10
2.5	@PATH .....	10
2.6	数値表記 .....	10
2.6.1	2進数、16進数表記 .....	10
2.6.2	ビット表記 .....	11
2.6.3	数式 .....	11
3	シーケンスデータセクション .....	12
3.1	書式 .....	12
3.2	ラベル名 .....	12
3.3	ファイルタイプ .....	13
3.4	ファイル名 .....	13
3.5	バンク .....	13
3.6	ボリューム .....	13
3.7	発音プライオリティ .....	13
3.8	プレイヤープライオリティ .....	14
3.9	プレイヤー番号 .....	14
4	シーケンスアーカイブセクション .....	15
4.1	書式 .....	15
4.2	ラベル名 .....	15
4.3	ファイルタイプ .....	16
4.4	ファイル名 .....	16

5	バンクデータセクション	17
5.1	書式	17
5.2	ラベル名	17
5.3	ファイルタイプ	18
5.4	ファイル名	18
5.5	波形アーカイブ	18
5.5.1	波形データを使わない場合	18
5.5.2	@WGROUPを使っていない場合	19
5.5.3	@WGROUPを使っている場合	19
6	波形アーカイブセクション	20
6.1	書式	20
6.2	ラベル名	20
6.3	ファイルタイプ	21
6.3.1	波形リストファイルの自動生成	21
6.4	ファイル名	22
6.5	オプションフラグ	22
6.5.1	波形データの個別ロード	23
6.5.1.1	消費メモリとロード時間のオーバーヘッド	23
6.5.1.2	ロード指定の注意	23
7	ストリームデータセクション	24
7.1	書式	24
7.2	ラベル名	24
7.3	フォーマット	25
7.4	ファイル名	25
7.5	ボリューム	26
7.6	プレイヤープライオリティ	26
7.7	プレイヤー番号	26
7.8	オプションフラグ	26
8	プレイヤー情報セクション	27
8.1	書式	27
8.2	プレイヤー番号	27
8.3	シーケンス最大同時再生数	28
8.4	プレイヤーヒープサイズ	28
8.5	確保可能なチャンネルビットフラグ	28
9	ストリームプレイヤー情報セクション	29
9.1	書式	29
9.2	プレイヤー番号	29
9.3	プレイヤータイプ	30
9.4	チャンネル番号	30

10	グループ情報セクション .....	31
10.1	書式 .....	31
10.2	グループラベル .....	32
10.3	サウンドデータラベル .....	32
10.3.1	インデックス番号指定 .....	32
10.4	オプション .....	33
11	サウンドマップファイル .....	34
11.1	サウンドマップファイルとは .....	34
11.2	サウンドマップファイルの内容 .....	34
11.2.1	実データサイズと消費ヒープサイズ .....	35
11.2.2	グループ .....	35
11.2.3	ファイルID .....	35

## コード

コード 2-1	サウンドアーカイブ定義ファイル .....	7
コード 3-1	シーケンスデータセクション .....	12
コード 4-1	シーケンスアーカイブセクション .....	15
コード 5-1	バンクデータセクション .....	17
コード 6-1	波形アーカイブセクション .....	20
コード 7-1	ストリームデータセクション .....	24
コード 8-1	プレイヤー情報セクション .....	27
コード 9-1	ストリームプレイヤー情報セクション .....	29
コード 10-1	グループ情報セクション .....	31
コード 11-1	サウンドマップファイル .....	34
コード 11-2	グループ .....	35

## 表

表 2-1	演算子 .....	11
表 3-1	シーケンス定義要素 .....	12
表 3-2	シーケンスファイルタイプ一覧 .....	13
表 4-1	シーケンスアーカイブ定義要素 .....	15
表 4-2	シーケンスアーカイブファイルタイプ一覧 .....	16
表 5-1	バンク定義要素 .....	17
表 5-2	バンクファイルタイプ一覧 .....	18
表 6-1	波形アーカイブ定義要素 .....	20
表 6-2	波形アーカイブファイルタイプ一覧 .....	21
表 6-3	波形アーカイブオプションフラグー一覧 .....	22
表 7-1	シーケンス定義要素 .....	24
表 7-2	ストリームデータフォーマット一覧 .....	25
表 7-3	ストリームデータフォーマットの特徴 .....	25
表 7-4	ストリームオプションフラグー一覧 .....	26
表 8-1	プレイヤー情報定義要素 .....	27

表 9-1 ストリームプレイヤー情報定義要素 .....	29
表 9-2 ストリームプレイヤータイプ一覧 .....	30
表 10-1 グループ情報定義要素 .....	31
表 10-2 同時にグループに含められるデータ .....	32



図 5-1 波形データグループ分けの例 .....	19
図 6-1 波形リストファイルの自動生成 .....	22
図 6-2 共有波形アーカイブ .....	23
図 10-1 オプションの例 .....	33

## 改訂履歴

版	改訂日	改 訂 内 容	承認者	担当者
1.7.0	2005-03-28	1. 数値表記に関する説明追加 2. サウンドマップファイルフォーマット変更に伴う修正		井田
1.6.0	2005-01-31	1. 波形データの個別ロード機能に関する説明追加 2. @PATH における UNC 形式の表記に関する説明追加		井田
1.5.0	2004-12-06	1. ストリームオプションフラグに関する説明追加		井田
1.4.0	2004-10-12	1. プレイヤーの指定にラベルが使えるようになったことに伴う修正 2. ラベル名に関する説明修正		井田
1.3.0	2004-09-16	1. ラベルファイルに関する説明修正 2. *.sdl ファイルの呼称を「サウンドラベルファイル」に統一 3. バンクリストファイル(*.sbd1)の呼称をサウンドアーカイブラベルファイルに変更		井田
1.2.0	2004-09-02	1. バンクリストファイル出力に関する説明追加		井田
1.1.0	2004-08-10	1. ストリームデータセクションに関する説明追加 2. ストリームプレイヤー情報セクションに関する説明追加		井田
1.0.0	2004-07-20	1. 全体的な構成変更 2. バンクが複数の波形アーカイブと関連づけられるようになったことに伴う修正 3. サウンドマップファイルの説明追加		井田
0.4.0	2004-06-01	1. プレイヤー情報セクション、グループ情報セクションを追加し、全体の構成を見直す 2. ラベル名の代わりにインデックス番号を指定できるようになったことによる修正 3. 名称“データブロック”を“セクション”に変更		井田
0.3.0	2004-04-12	1. サウンドアーカイバの説明を「サウンドツールマニュアル」へ移動 2. 章構成を変更 3. ファイルタイプの説明に加筆		井田
0.2.0	2004-04-01	1. ディレクトリ構成変更に伴う修正 2. ファイルタイプの説明に注釈追加 3. 発音プライオリティ比較に関する誤りを訂正		井田
0.1.0	2004-03-01	初版		井田

# 1 概要

サウンドアーカイブは、NITRO-Composer で扱うファイルフォーマットの一つです。シーケンスデータや波形データなどの各種サウンドデータを1つのファイルにまとめたもので、最終的にはこのサウンドアーカイブだけを ROM に格納すれば良いようになります。

サウンドアーカイブを作成するには、あらかじめテキストフォーマットで、どのサウンドデータをまとめるのかを指定しておく必要があります。このファイルを「サウンドアーカイブ定義ファイル」と呼びます。ここでは、サウンドアーカイブ定義ファイルの書き方を説明します。

## 2 構成

### 2.1 サウンドアーカイブ定義ファイルのサンプル

まずは、下のサンプルをご覧ください。

#### コード 2-1 サウンドアーカイブ定義ファイル

```
;/=====
;
; NITRO-Composer sample
;
;/=====

;;;;;;;;;;;;;
;; Wave Archive

@WAVEARC
  @PATH "swar"
WAVE_SE      : AUTO, "se.swls"
WAVE_BGM     : AUTO, "bgm.swls"

;;;;;;;;;;;;;
;; Bank

@BANK
  @PATH "bnk"
BANK_SE      : TEXT, "se.bnk",  WAVE_SE
BANK_BGM     : TEXT, "bgm.bnk",  WAVE_BGM

;;;;;;;;;;;;;
;; Player

@PLAYER
  PLAYER_BGM   : 1, 8000
  PLAYER_SE    = 10 : 1
  PLAYER_VOICE : 1

;;;;;;;;;;;;;
;; Sequence
```

```
@SEQ
  @PATH "mid"
  SEQ_MARIOKART : SMF, "kart64_title.mid", BANK_BGM, 127, 64, 64, PLAYER_BGM

  ;;;;;;;;;;;;;;
  ;; Sequence Archive

@SEQARC
  @PATH "mus"
  SEQ_SE : TEXT, "se.mus"

  ;;;;;;;;;;;;;;
  ;; Stream Player

@STRM_PLAYER
  PLAYER_STRM : STEREO, 6, 7

  ;;;;;;;;;;;;;;
  ;; Stream

@STRM
  @PATH "strm"
  STRM_MARIOKART : PCM8, "kart_title.32.aiff", 127, 64, PLAYER_STRM
  STRM_FANFARE : PCM8, "fanfare.32.aiff", 127, 64, PLAYER_STRM

  ;;;;;;;;;;;;;;
  ;; Group

@GROUP

GROUP_STATIC :
  SEQ_SE
  BANK_SE
  BANK_BGM
```

## 2.2 セクション

全体的な構成としては、大きく8つのセクションに分かれます。

- @WAVEARC で始まる、波形アーカイブセクション
- @BANK で始まる、バンクデータセクション
- @PLAYER で始まる、プレイヤー情報セクション
- @SEQ で始まる、シーケンスデータセクション
- @SEQARC で始まる、シーケンスアーカイブセクション
- @STRM\_PLAYER で始まる、ストリームプレイヤー情報セクション
- @STRM で始まる、ストリームデータセクション
- @GROUP で始まる、グループ情報セクション

なお、サンプルでは全てのセクションが定義されていますが、必要のないセクションは、作成する必要はありません。

### 2.2.1 波形アーカイブセクション

波形アーカイブを登録します。6章で詳しく説明しています。



---

### 2.2.2 バンクデータセクション

---

バンクデータを登録します。波形アーカイブとの対応関係も指定します。5章で詳しく説明しています。

---

### 2.2.3 プレイヤー情報セクション

---

各プレイヤー毎に設定するパラメータを指定します。8章で詳しく説明しています。

---

### 2.2.4 シーケンスデータセクション

---

シーケンスデータを登録します。バンクとの対応関係や、プレイヤー番号などのパラメータも指定します。3章で詳しく説明しています。

---

### 2.2.5 シーケンスアーカイブセクション

---

シーケンスアーカイブを登録します。4章で詳しく説明しています。

---

### 2.2.6 ストリームプレイヤー情報セクション

---

各ストリームプレイヤー毎に設定するパラメータを指定します。9章で詳しく説明しています。

---

### 2.2.7 ストリームデータセクション

---

ストリームデータを登録します。7章で詳しく説明しています。

---

### 2.2.8 グループ情報セクション

---

サウンドデータを一度にロードするためのグループを定義します。10章で詳しく説明しています。

---

## 2.3 コメント

---

コメントは、セミコロン";"で始まります。

```
; Sample
```

コメントは改行まで有効です。次のように、行の最後にも書くこともできます。

```
BANK_SE      : TEXT, "se.bnk"      ;;; comment
```

---

## 2.4 ラベル

---

以下の説明でラベルという用語が使われます。ラベルとは、特定のものを指し示す文字列です。次の例では、"WAVE\_MK64"がラベルです。

```
WAVE_MK64    : AUTO, "mariokart64.swls"
```

ラベルは、大文字のアルファベットで始まり、2文字目以降は、大文字アルファベットに加え、アンダースコア"\_"と数字が使えます。次に示すものは正しいラベルの例です。

```
SEQ_TITLE
SEQ_MAP01
SEQ_GAME_OVER
```

ラベルは特定のものを指すものであるため、別のものに対して同じラベル名を指定することはできません。

## 2.4.1 ラベルファイル出力

サウンドアーカイブ定義ファイル中で定義したラベルは、サウンドラベルファイル(\*.sdl)及び、サウンドアーカイブラベルファイル(\*.sddl)に出力されます。

### 2.4.1.1 サウンドラベルファイル

サウンドラベルファイル(\*.sdl)は、プログラムのソースファイルから取り込むことで、プログラムからも同じラベルを使って、特定のデータを指定することができます。このファイルを、サウンドアーカイブ\*.sdat と合わせて、プログラマーに渡してください。なお、サウンドラベルファイル(\*.sdl)には、シーケンスアーカイブ中のシーケンスラベルも含まれています。

このファイルを、サウンドデータから取り込むことはできませんので、注意してください。サウンドデータからラベルを使いたい場合は、次に説明するサウンドアーカイブラベルファイル(\*.sddl)を取り込みます。

### 2.4.1.2 サウンドアーカイブラベルファイル

サウンドアーカイブラベルファイル(\*.sddl)は、テキスト形式のサウンドデータから取り込むことによって、サウンドアーカイブ定義ファイルで定義されたラベルを使用できるようになります。例えば、シーケンスアーカイブの@SEQ\_TABLE におけるバンク指定を、番号ではなくラベルで記述できるようになります。

このファイルを取り込むためには、`#include` コマンドを使います。詳しくは、「サウンドツールマニュアル」をご覧ください。

## 2.5 @PATH

ファイルを指定するとき、デフォルトではサウンドアーカイブ定義ファイルのあるディレクトリからの相対パスを指定する必要があります。ただし、@PATH を使うことで、この起点となるディレクトリを変更することができます。

```
@PATH "bnk"
BANK_SE      : TEXT, "se.bnk" , WAVE_SE
```

この例では、"bnk/se.bnk"を指定したことになります。つまり次の記述と同じ意味です。

```
BANK_SE      : TEXT, "bnk/se.bnk" , WAVE_SE
```

なお、@PATH で指定するディレクトリは、次のような UNC 形式の表記で指定することも可能です。

```
@PATH "//server-1/path/bnk"
```

## 2.6 数値表記

サウンドアーカイブ定義ファイル中で、ボリュームなどのパラメータを指定する箇所について、直接数値を打ち込む方法の他に、以下のような表記も使えます。

### 2.6.1 2進数、16進数表記

通常10進数で記述しますが、2進数または16進数で記述することもできます。

2進数または16進数で記述するときは、それぞれ先頭に、0b または 0x を付けます。たとえば、10進数の12は、次のように記述することができます。

```
0b1100
0xc
```

## 2.6.2 ビット表記

ビットフラグのような、あるビットが1か0かが意味を持つ数値を記述する場合、ビット表記が有効です。

ビット表記では、下位何ビットを1にするかを記述します。例えば、下位1, 3ビットと6ビットから8ビットを1とした値を記述する場合、次のようにします。

```
{ 1, 3, 6-8 }
```

これは、0b111001010 に相当します。最下位ビットは0ビットとなることに注意してください。

## 2.6.3 数式

数値は、数式で記述することもできます。数式の各項には、2進数表記、16進数表記、ビット表記を使うこともできます。

例えば、次のような表記が可能です。

```
2 * 4 + 0x10
( 1 << 4 ) + 3
{ 0, 2 } | { 4-6 }
```

数式で扱える演算子と、演算子の優先順位は、以下の通りです。

表 2-1 演算子

優先度	演算子	意味
1	*	乗算
	/	除算
2	+	加算
	-	減算
3	>>	右シフト
	<<	左シフト
4	<	左辺が右辺未満
	<=	左辺が右辺以下
	>	左辺が右辺より大きい
	>=	左辺が右辺以上
5	==	左辺と右辺が等しい
6	&	ビット毎の AND
7		ビット毎の OR

## 3 シーケンスデータセクション

シーケンスデータセクションは、シーケンスデータを登録するセクションです。@SEQ で始まります。

### コード 3-1 シーケンスデータセクション

```
;;;;;;;;;;;;;;  
;; Sequence  
  
@SEQ  
  @PATH "mid"  
SEQ_MARIOKART : SMF, "kart64_title.mid", BANK_BGM, 127, 64, 64, PLAYER_BGM
```

### 3.1 書式

次の1行が1つのシーケンスを表しています。

```
SEQ_MARIOKART : SMF, "kart64_title.mid", BANK_BGM, 127, 64, 64, PLAYER_BGM
```

書式は次のようになっています。

label : filetype, filename, bank, volume, channelPrio, playerPrio, playerNo
---

各要素の意味は次の通りです。

表 3-1 シーケンス定義要素

要素	説明
label	ラベル名またはインデックス番号
filetype	ファイルタイプ
filename	ファイル名
bank	バンクラベルまたは、バンクナンバー
volume	ボリューム
channelPrio	発音プライオリティ
playerPrio	プレイヤープライオリティ
playerNo	プレイヤーラベルまたは、プレイヤー番号

以下、各要素について詳しく説明します。

### 3.2 ラベル名

シーケンスを指定するためのラベル名を定義します。各シーケンスには、0から順番にインデックス番号が割り振られます。インデックス番号を直接指定したり、ラベル名を定義しないようにすることもできます。

```
SEQ_TITLE = 2 : SMF, "title.mid", BANK_BGM, 127, 64, 64, PLAYER_BGM
SEQ_MENU   : SMF, "menu.mid", BANK_BGM, 127, 64, 64, PLAYER_BGM
10         : SMF, "main.mid", BANK_BGM, 127, 64, 64, PLAYER_BGM
```

1行目は、インデックス番号を直接指定するタイプで、ラベル名は `SEQ_TITLE` で、インデックス番号 2 となります。

2行目の様にラベル名のみ指定すると、1つ前の行のインデックス番号に1を足した値が、インデックス番号として設定されます。この例の場合では、インデックス番号は3になります。

3行目は、インデックス番号のみの指定で、ラベル名は定義しません。従って、このシーケンスを指すときにもラベル名ではなく、インデックス番号で指定する必要があります。

### 3.3 ファイルタイプ

シーケンスデータファイルのファイルタイプを指定します。実際にサウンドアーカイブに格納されるファイルは、バイナリシーケンスファイルと呼ばれるファイルタイプのものですが、別のファイルタイプを指定しても、自動的にバイナリシーケンスファイルに変換する機構が用意されています。

指定できるファイルタイプは以下の通りです。

**表 3-2 シーケンスファイルタイプ一覧**

ファイルタイプ	説明
SMF	スタンダード MIDI ファイル
TEXT	テキストシーケンスファイル
BIN	バイナリシーケンスファイル

各種フォーマットに関する詳細については、「シーケンスデータマニュアル」をご覧ください。

### 3.4 ファイル名

シーケンスデータファイルのファイル名を指定します。パスはデフォルトでは、サウンドアーカイブ定義ファイルからの相対パスになりますが、`@PATH` コマンドを使うことで、起点となるディレクトリを変更することもできます。

なお、ファイルタイプと指定ファイルの実際のファイルタイプを合わせることに注意してください。

### 3.5 バンク

このシーケンスがどのバンクを使って再生するのかを、バンクラベルまたはバンクナンバーで指定します。

バンクは、バンクデータセクションで定義します。詳しくは「5章バンクデータセクション」を参照してください。

### 3.6 ボリューム

シーケンス全体にかかるボリュームを指定します。値の範囲は 0 ～ 127 です。

### 3.7 発音プライオリティ

シーケンス全体に対する発音プライオリティを指定します。

発音プライオリティとは、16 チャンネルという制限の中で、どの音を発音するかを決める優先度です。値の範囲は 0～127 で、値が大きいほど優先度は高くなります。

発音プライオリティについての詳細は、「サウンドシステムマニュアル」をご覧ください。

### 3.8 プレイヤープライオリティ

---

プレイヤープライオリティを指定します。

プレイヤープライオリティとは、シーケンス最大同時再生数の制限の中で、どのシーケンスを再生するかを決める優先度です。値の範囲は 0～127 で、値が大きいほど優先度は高くなります。

プレイヤープライオリティについての詳細は、「サウンドシステムマニュアル」をご覧ください。

### 3.9 プレイヤー番号

---

プレイヤー番号は、32 個あるプレイヤーの内、どのプレイヤーを使って再生するかを指定します。プレイヤー情報セクションが先に定義されている場合は、プレイヤーをラベルで指定できます。番号で指定する場合は、0～31 の値を指定します。

1つのプレイヤーで再生できるシーケンスの数は限定されていますので、これを考慮して、プレイヤー番号を指定する必要があります。

## 4 シーケンスアーカイブセクション

シーケンスアーカイブセクションは、シーケンスアーカイブを登録するセクションです。@SEQARC で始まります。

### コード 4-1 シーケンスアーカイブセクション

```
;;;;;;;;;;;;;;
;; Sequence Archive

@SEQARC
@PATH "mus"
SEQ_SE : TEXT, "se.mus"
```

### 4.1 書式

次の1行が1つのシーケンスアーカイブを表しています。

- SEQ\_SE : TEXT, "se.mus"

書式は次のようになっています。

label : filetype, filename
----------------------------

各要素の意味は次の通りです。

表 4-1 シーケンスアーカイブ定義要素

要素	説明
label	ラベル名またはインデックス番号
filetype	ファイルタイプ
filename	ファイル名

以下、各要素について詳しく説明します。

### 4.2 ラベル名

シーケンスアーカイブを指定するためのラベル名を定義します。各シーケンスアーカイブには、0から順番にインデックス番号が割り振られます。インデックス番号を直接指定したり、ラベル名を定義しないようにすることもできます。

```
SEQ_SE = 2      : TEXT, "se.mus"
SEQ_ENEMY_SE   : TEXT, "enemy_se.mus"
10             : TEXT, "common.mus"
```

1行目は、インデックス番号を直接指定するタイプで、ラベル名は SEQ\_SE で、インデックス番号 2 となります。

2行目の様にラベル名のみ指定すると、1つ前の行のインデックス番号に1を足した値が、インデックス番号として設定されます。この例の場合では、インデックス番号は3になります。

3行目は、インデックス番号のみの指定で、ラベル名は定義しません。従って、このシーケンスアーカイブを指すときにも

ラベル名ではなく、インデックス番号で指定する必要があります。

## 4.3 ファイルタイプ

シーケンスアーカイブのファイルタイプを指定します。実際にサウンドアーカイブに格納されるファイルは、バイナリシーケンスアーカイブファイルと呼ばれるファイルタイプのものですが、別のファイルタイプを指定しても、自動的にバイナリシーケンスアーカイブファイルに変換する機構が用意されています。

指定できるファイルタイプは以下の通りです。

表 4-2 シーケンスアーカイブファイルタイプ一覧

ファイルタイプ	説明
TEXT	テキストシーケンスアーカイブファイル
BIN	バイナリシーケンスアーカイブファイル

各種フォーマットに関する詳細については、「シーケンスデータマニュアル」をご覧ください。

## 4.4 ファイル名

シーケンスアーカイブのファイル名を指定します。パスはデフォルトでは、サウンドアーカイブ定義ファイルからの相対パスになりますが、@PATH コマンドを使うことで、起点となるディレクトリを変更することもできます。

なお、ファイルタイプと指定ファイルの実際のファイルタイプを合わせることに注意してください。



## 5 バンクデータセクション

バンクデータセクションは、バンクデータを登録するセクションです。@BANK で始まります。

### コード 5-1 バンクデータセクション

```
;;;;;;;;;;;;;;  
;; Bank  
  
@BANK  
@PATH "bnk"  
BANK_SE= 0      : TEXT, "se.bnk",    WAVE_SE  
BANK_BGM       : TEXT, "bgm.bnk",    WAVE_BGM
```

### 5.1 書式

次の1行が1つのシーケンスアーカイブを表しています。

```
BANK_BGM       : TEXT, "bgm.bnk",    WAVE_BGM
```

書式は次のようになっています。

label : filetype, filename, wavearc0, wavearc1, wavearc2, wavearc3
--

各要素の意味は次の通りです。

表 5-1 バンク定義要素

要素	説明
label	ラベル名またはインデックス番号
filetype	ファイルタイプ
filename	ファイル名
wavearc0	0 番波形アーカイブ(省略可)
wavearc1	1 番波形アーカイブ(省略可)
wavearc2	2 番波形アーカイブ(省略可)
wavearc3	3 番波形アーカイブ(省略可)

以下、各要素について詳しく説明します。

### 5.2 ラベル名

バンクを指定するためのラベル名を定義します。各バンクには、0から順番にインデックス番号が割り振られます。インデックス番号を直接指定したり、ラベル名を定義しないようにすることもできます。

```

BANK_SE = 2      : TEXT, "se.bnk", WAVE_SE
BANK_BGM       : TEXT, "bgm.bnk", WAVE_BGM
10             : TEXT, "common.bnk", WAVE_COMMON

```

1行目は、インデックス番号を直接指定するタイプで、ラベル名は **BANK\_SE** で、インデックス番号 **2** となります。

2行目の様にラベル名のみ指定すると、1つ前の行のインデックス番号に1を足した値が、インデックス番号として設定されます。この例の場合では、インデックス番号は3になります。

3行目は、インデックス番号のみの指定で、ラベル名は定義しません。従って、このバンクを指すときにもラベル名ではなく、インデックス番号で指定する必要があります。

## 5.3 ファイルタイプ

バンクファイルのファイルタイプを指定します。実際にサウンドアーカイブに格納されるファイルは、バイナリバンクファイルと呼ばれるファイルタイプのものですが、別のファイルタイプを指定しても、自動的にバイナリバンクファイルに変換する機構が用意されています。

指定できるファイルタイプは以下の通りです。

**表 5-2 バンクファイルタイプ一覧**

ファイルタイプ	説明
TEXT	テキストバンクファイル
BIN	バイナリバンクファイル

テキストバンクフォーマットに関する詳細については、「バンクデータマニュアル」をご覧ください。

バイナリバンクファイルを指定すると、後で説明する波形リストファイルの自動生成機能が使えなくなりますので注意してください。

## 5.4 ファイル名

バンクデータのファイル名を指定します。パスはデフォルトでは、サウンドアーカイブ定義ファイルからの相対パスになりますが、@PATH コマンドを使うことで、起点となるディレクトリを変更することもできます。

なお、ファイルタイプと指定ファイルの実際のファイルタイプを合わせることに注意してください。

## 5.5 波形アーカイブ

バンクが使用する波形データが格納された波形アーカイブを、ラベルまたはインデックス番号で指定します。

バンク定義ファイルで波形ファイルをどのように定義したかによって、波形アーカイブの指定方法が変わります。

### 5.5.1 波形データを使わない場合

バンク定義ファイル中で波形データを使っていない場合、波形アーカイブの指定は不要です。

```

BANK_SE      : TEXT, "se.bnk"

```

### 5.5.2 @WGROUP を使っていない場合

バンク定義ファイル中で、@WGROUP を使って波形グループの指定をしていない場合は、波形アーカイブを1つだけ指定します。

```
BANK_MK64      : TEXT, "mariokart64.bnk",      WAVE_MK64
```

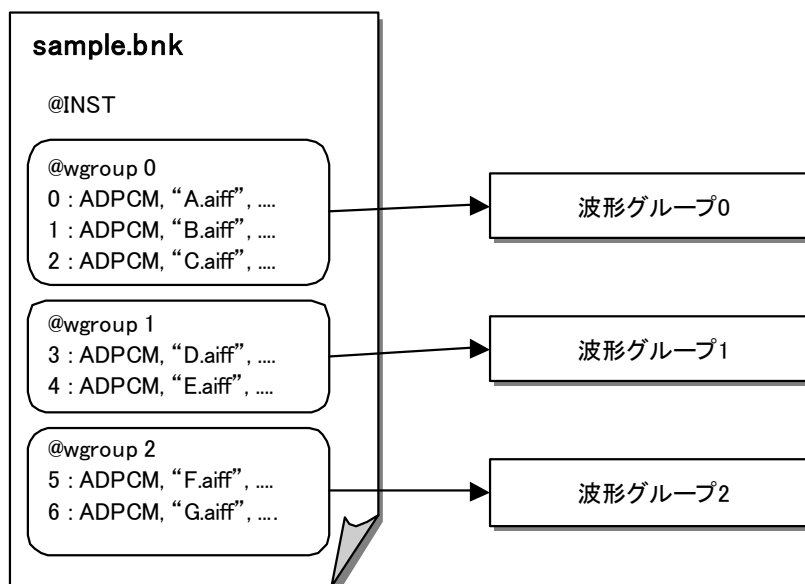
WAVE\_MK64 が波形アーカイブのラベルで、これは波形アーカイブセクションで定義されています。WAVE\_MK64 には、BANK\_MK64 が使用している波形データが全て含まれている必要があります。ただし、波形アーカイブの自動生成機能を使えば、あまり深く考える必要はありません。詳しくは、「6章波形アーカイブセクション」を参照してください。

### 5.5.3 @WGROUP を使っている場合

バンク定義ファイル中で@WGROUP を使い、最大4つまで波形データをグループ分けすることができます。グループ分けを行った場合には、それぞれのグループに対して、波形アーカイブを割り当てる必要があります。

例えば、あるバンクファイル中で、次のような波形データをグループ分けしたとします。

図 5-1 波形データグループ分けの例



このバンクの登録には、次のように3つの波形アーカイブを指定する必要があります。

```
BANK_SAMPLE    : TEXT, "sample.bnk",    WAVE_0, WAVE_1, WAVE_2
```

ここで、WAVE\_0, WAVE\_1, WAVE\_2 はそれぞれ、波形グループ 0、波形グループ 1、波形グループ 2 に対応しています。すなわち、WAVE\_0 には、"A.aiff" (正確には、"A.aiff"をコンバートした後のファイル)、"B.aiff", "C.aiff"が含まれている必要があります。ただし、波形アーカイブの自動生成機能を使えば、あまり深く考える必要はありません。詳しくは、「6章波形アーカイブセクション」を参照してください。

バンク定義ファイル中で波形データをグループ分けする方法について詳しくは、「バンクデータマニュアル」を参照してください。

## 6 波形アーカイブセクション

波形アーカイブセクションは、波形アーカイブを登録するセクションです。@WAVEARC で始まります。

### コード 6-1 波形アーカイブセクション

```
;;;;;;;;;;;;;
;; Wave Archive

@WAVEARC
@PATH "swar"
WAVE_SE      : AUTO, "se.swls"
WAVE_BGM     : AUTO, "bgm.swls"
```

### 6.1 書式

次の1行が1つの波形アーカイブを表しています。

```
WAVE_BGM      : AUTO, "bgm.swls"
```

書式は次のようになっています。

label : filetype, filename
----------------------------

各要素の意味は次の通りです。

**表 6-1 波形アーカイブ定義要素**

要素	説明
label	ラベル名またはインデックス番号
filetype	ファイルタイプ
filename	ファイル名
flags	オプションフラグ(省略可)

以下、各要素について詳しく説明します。

### 6.2 ラベル名

波形アーカイブを指定するためのラベル名を定義します。各波形アーカイブには、0から順番にインデックス番号が割り振られます。インデックス番号を直接指定したり、ラベル名を定義しないようにすることもできます。

```
WAVE_SE = 2    : AUTO, "se.swlsmus"
WAVE_BGM      : AUTO, "bgm.swls"
10           : AUTO, "common.swls"
```

1行目は、インデックス番号を直接指定するタイプで、ラベル名は WAVE\_SE で、インデックス番号 2 となります。

2行目の様にラベル名のみ指定すると、1つ前の行のインデックス番号に1を足した値が、インデックス番号として設定されます。この例の場合では、インデックス番号は3になります。

3行目は、インデックス番号のみの指定で、ラベル名は定義しません。従って、この波形アーカイブを指すときにもラベル名ではなく、インデックス番号で指定する必要があります。

## 6.3 ファイルタイプ

波形アーカイブのファイルタイプを指定します。実際にサウンドアーカイブに格納されるファイルは、波形アーカイブファイルと呼ばれるファイルタイプのものですが、別のファイルタイプを指定しても、自動的に波形アーカイブファイルに変換する機構が用意されています。

指定できるファイルタイプは以下の通りです。

表 6-2 波形アーカイブファイルタイプ一覧

ファイルタイプ	説明
AUTO	波形リストファイルの自動生成
TEXT	波形リストファイル
BIN	波形アーカイブファイル

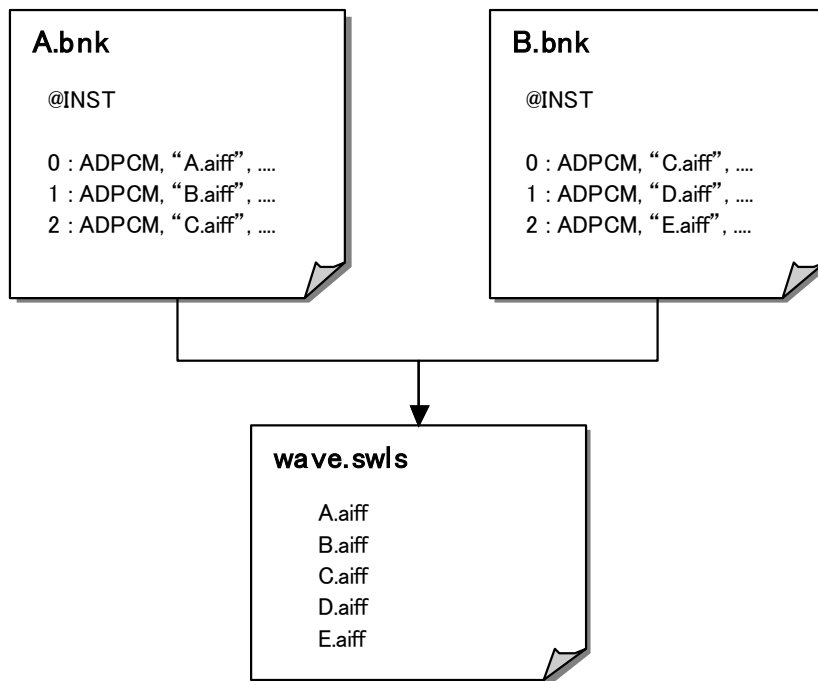
波形リストファイルは、単純に波形ファイルのパスを1行ずつ書き連ねたファイルです。そのリストを元に、複数の波形ファイルをまとめて、1つの波形アーカイブファイルが生成されます。

### 6.3.1 波形リストファイルの自動生成

上で述べたように、各バンクに対して、そのバンクが使用している波形ファイルが格納された波形アーカイブを指定する必要があります。ただし、バンクが必要とする波形ファイルが増えるたびに、波形リストファイルに波形ファイルを追加するのは面倒です。そのため、ファイルタイプに **AUTO** を指定しておくと、波形リストファイルが自動的に生成されるようになっています。

**AUTO** の場合、自分に対して関連づけられたバンクを探しだし、そのバンクで使われている波形ファイルの情報から、波形リストファイルを自動的に生成します。関連づけられたバンクが2つ以上あった場合には、各バンクで使われている全ての波形ファイルが含まれることになります。この時、別のバンクで同じ波形ファイルを使っていたら、1つの波形ファイルにまとめられます。

図 6-1 波形リストファイルの自動生成



正確には、"wave.swls"には、"A.aiff"などをコンバートした後のファイル名が記されます。

関連づけられたバンクのファイルタイプが **TEXT** で無い場合、波形ファイルの情報を取得できないため、波形リストファイルを自動生成できなくなりますので、注意してください。

## 6.4 ファイル名

波形アーカイブのファイル名を指定します。パスはデフォルトでは、サウンドアーカイブ定義ファイルからの相対パスになりますが、@PATH コマンドを使うことで、起点となるディレクトリを変更することもできます。

なお、ファイルタイプと指定ファイルの実際のファイルタイプを合わせることに注意してください。ただし、ファイルタイプが **AUTO** の場合は、適当に他のファイルと衝突しないファイル名を指定します。

## 6.5 オプションフラグ

波形アーカイブに関する補助的な設定を行うオプションフラグを指定します。指定しない場合は、省略できます。

```
WAVE_COMMON      : AUTO, "common.swls", s
```

オプションフラグには、以下のものを指定します。指定したオプションが有効になります。

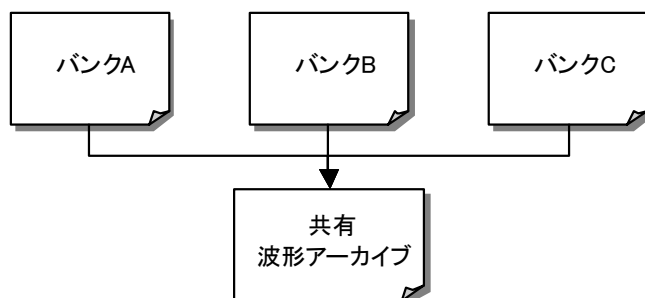
表 6-3 波形アーカイブオプションフラグ一覧

フラグ	説明
s	波形データをロードするとき、波形アーカイブをそのままロードせずに、波形アーカイブ中にある必要な波形データのみを、個別にロードします。(詳細は後述)

### 6.5.1 波形データの個別ロード

オプションフラグ's'をつけた波形アーカイブは、波形データをロードするとき、波形アーカイブをそのままロードせずに、波形アーカイブ中にある必要な波形データのみを、個別にロードします。これは、次のように1つの波形アーカイブを複数のバンクで共有しているときに、有効に働きます。

図 6-2 共有波形アーカイブ



この時、共有波形アーカイブには、バンクAのみで使用している波形データが含まれている場合があります。そのような場合、バンクBを使うときに必要のない波形データもロードすることになり、メモリ効率が悪くなります。

そこで、この共有波形アーカイブに、オプションフラグ's'をつけておけば、バンクBを使うときには、バンクBが使っていない波形データが、共有波形アーカイブに含まれていたとしても、それらはロードされません。

ただし、オプションフラグ's'をつける場合には、以下の点に注意する必要があります。

#### 6.5.1.1 消費メモリとロード時間のオーバーヘッド

波形データを個々に扱うので、一括で扱った場合と比べて、消費メモリとロード時間のオーバーヘッドが発生します。不要なデータがロードされないことによる消費メモリの削減とロード時間の短縮によって、ほとんどの場合無視できますが、不要なデータが無いもしくは少ない場合は、かえって効率が悪くなります。

そのような場合は、波形データのグループ管理の手法を検討してみてください。波形データのグループ管理について詳しくは、本ドキュメントの「5.5.3 @WGROUP を使っている場合」及び、「バンクデータマニュアル」を参照してください。

#### 6.5.1.2 ロード指定の注意

波形データの個別ロードは、バンクに必要なデータをロードしますので、バンクをロードするときに同時に波形データもロードするように指定する必要があります。

例えばオプションフラグで波形データをロードしないように指定していたり、波形アーカイブを直接指定してロードしていた場合、必要な波形データがロードされず、ただしく発音できないことになります。

## 7 ストリームデータセクション

ストリームデータセクションは、ストリームデータを登録するセクションです。@STRM で始まります。

### コード 7-1 ストリームデータセクション

```
;;;;;;;;;;;;;;  
;; Stream  
  
@STRM  
@PATH "strm"  
STRM_MARIOKART : PCM8, "kart_title.32.aiff", 127, 64, PLAYER_STRM  
STRM_FANFARE    : PCM8, "fanfare.32.aiff", 127, 64, PLAYER_STRM
```

### 7.1 書式

次の1行が1つのストリームを表しています。

```
STRM_MARIOKART : PCM8, "kart_title.32.aiff", 127, 64, PLAYER_STRM
```

書式は次のようになっています。

label : format, filename, volume, playerPrio, playerNo [ , flags ]
--

各要素の意味は次の通りです。

表 7-1 シーケンス定義要素

要素	説明
label	ラベル名またはインデックス番号
format	フォーマット
filename	ファイル名
volume	ボリューム
playerPrio	プレイヤープライオリティ
playerNo	プレイヤーラベルまたは、プレイヤー番号
flags	オプションフラグ(省略可)

以下、各要素について詳しく説明します。

### 7.2 ラベル名

ストリームを指定するためのラベル名を定義します。各ストリームには、0から順番にインデックス番号が割り振られます。インデックス番号を直接指定したり、ラベル名を定義しないようにすることもできます。



```

STRM_TITLE = 2 : PCM8, "title.aiff, 127, 64, PLAYER_STRM
STRM_MENU   : PCM8, "menu.aiff", 127, 64, PLAYER_STRM
10          : PCM8, "main.aiff", 127, 64, PLAYER_STRM

```

1行目は、インデックス番号を直接指定するタイプで、ラベル名は `STRM_TITLE` で、インデックス番号 2 となります。

2行目の様にラベル名のみ指定すると、1つ前の行のインデックス番号に1を足した値が、インデックス番号として設定されます。この例の場合では、インデックス番号は3になります。

3行目は、インデックス番号のみの指定で、ラベル名は定義しません。従って、このストリームを指すときにもラベル名ではなく、インデックス番号で指定する必要があります。

## 7.3 フォーマット

波形データをどのフォーマットに変換するかを指定します。指定できるフォーマットは以下の通りです。

**表 7-2 ストリームデータフォーマット一覧**

フォーマット	説明
PCM16	16bitPCM データに変換します。
PCM8	8bitPCM データに変換します。
ADPCM	ADPCM データに変換します。
STRM	変換しません。(変換済みファイルを指定します)

各フォーマットには、それぞれ一長一短があります。下の表はそれぞれのフォーマットの特徴を表したものです。

**表 7-3 ストリームデータフォーマットの特徴**

フォーマット	サイズ	音質	処理時間
PCM16	×	◎	△
PCM8	△	△	○
ADPCM	○	○	×

サイズは、データサイズの大きさでの比較です。サイズが大きいほど ROM 容量を消費します。ADPCM が一番サイズが小さく、サイズの比はおおよそ次のようになります。

PCM16 : PCM8 : ADPCM = 4 : 2 : 1

音質は、PCM16 が一番良いですが、ADPCM との違いはあまりありません。PCM8 は若干量子化ノイズが目立ちます。

処理時間は、ストリーム再生中に必要となる処理時間の比較です。ADPCM はデコード処理が必要なため、多くの処理時間を必要とします。PCM8 は PCM16 に比べて、ロードするデータサイズが少なくなるので、処理は比較的軽くて済みます。

## 7.4 ファイル名

ストリームデータファイルのファイル名を指定します。フォーマット指定が `STRM` 以外の場合、`AIFF` または `WAV` ファ

イルを指定します。STRM の場合はコンバート後の\*.strm ファイルを指定します。

波形ファイルは、ステレオ、モノラルどちらでも再生できますが、再生するストリームプレイヤーがステレオでないと、ステレオデータは再生できませんので注意してください。

パスはデフォルトでは、サウンドアーカイブ定義ファイルからの相対パスになりますが、@PATH コマンドを使うことで、起点となるディレクトリを変更することもできます。

## 7.5 ボリューム

ストリーム全体にかかるボリュームを指定します。値の範囲は 0 ～ 127 です。

## 7.6 プレイヤープライオリティ

プレイヤープライオリティを指定します。

プレイヤープライオリティとは、既に再生中のストリームプレイヤーに対し、新たにストリームを再生しようとしたときにどちらを優先して再生するかを表す優先度です。値の範囲は 0～127 で、値が大きいほど優先度は高くなります。値が等しい場合は、後から再生する方が優先されます。

## 7.7 プレイヤー番号

プレイヤー番号は、どのストリームプレイヤーを使って再生するのかを指定します。ストリームプレイヤー情報セクションが先に定義されている場合は、プレイヤーをラベルで指定できます。番号で指定する場合は、0～3 の値で指定します。

9章で説明するストリームプレイヤー情報セクションで定義したストリームプレイヤーのみ使用できます。また、1つのストリームプレイヤーで再生できるストリームは1つだけですので、これを考慮して、プレイヤー番号を指定する必要があります。

## 7.8 オプションフラグ

ストリームに関する補助的な設定を行うオプションフラグを指定します。指定しない場合は、省略できます。

```
STRM_MARIOKART : PCM8, "kart_title.32.aiff", 127, 64, PLAYER_STRM, s
```

オプションフラグには、以下のものを指定します。指定したオプションが有効になります。

表 7-4 ストリームオプションフラグ一覧

フラグ	説明
s	ステレオプレイヤーでモノラルデータを再生したとき、2チャンネル使って、より大きな音量で再生します。(デフォルトは1チャンネル)

## 8 プレイヤー情報セクション

プレイヤー情報セクションは、32 個あるプレイヤー毎にパラメータを設定するためのセクションです。@PLAYER で始まります。

### コード 8-1 プレイヤー情報セクション

```
;;;;;;;;;;  
;; Player  
  
@PLAYER  
  PLAYER_BGM      : 1, 8000  
  PLAYER_SE  = 10 : 1  
  PLAYER_VOICE    : 1
```

### 8.1 書式

次の1行が1つのプレイヤーに対する設定です。ただし、設定をしていないプレイヤーでも、シーケンスを再生することができます。この場合、デフォルト値が設定されます。

```
PLAYER_BGM : 1, 8000
```

書式は次のようになっています。

playerNo : seqMax, heapSize, chBitFlag
--

各要素の意味は次の通りです。

表 8-1 プレイヤー情報定義要素

要素	説明
playerNo	プレイヤー番号またはラベル
seqMax	シーケンス最大同時再生数(デフォルト1)
heapSize	プレイヤーヒープサイズ(省略可)
chBitFlag	確保可能なチャンネルビットフラグ(省略可)

以下、各要素について詳しく説明します。

### 8.2 プレイヤー番号

設定するプレイヤーのプレイヤー番号を、0～31 の値で指定します。また次のようにして、ラベルを定義することができます。

```
PLAYER_BGM = 1 : 1  
PLAYER_ENEMY : 4  
10          : 1
```

1行目は、プレイヤー番号 1 でラベル名が PLAYER\_BGM です。

2行目の様にラベル名のみ指定すると、1つ前の行のプレイヤー番号に1を足した値が、プレイヤー番号となります。こ

の例の場合では、プレイヤー番号は 2 になります。

3行目は、プレイヤー番号のみでラベルは定義しません。従って、このプレイヤーを指すときにもラベル名ではなく、インデックス番号で指定する必要があります。

## 8.3 シーケンス最大同時再生数

プレイヤー毎のシーケンス最大同時再生数を設定します。デフォルト値は1です。0～16 の範囲で設定できます。0 はそのプレイヤーでシーケンスを再生できないことを意味し、16 は同時再生数の制限が無いことを意味します。

シーケンス最大同時再生数について詳しくは、「サウンドシステムマニュアル」をご覧ください。

## 8.4 プレイヤーヒープサイズ

プレイヤーヒープのサイズを指定します。シーケンス最大同時再生数分のプレイヤーヒープが作成されます。

例えば、次のように指定した場合、

5 : 3, 1000

プレイヤー番号 5 のプレイヤーに対し、1000 バイトのプレイヤーヒープが 3 つ作成されます。3つまで同時に再生できるので、プレイヤーヒープも3つ分必要となるわけです。

プレイヤーヒープサイズの指定は省略できます。省略した場合は、プレイヤーヒープは作成されません。

プレイヤーヒープについて詳しくは、「サウンドシステムマニュアル」をご覧ください。

## 8.5 確保可能なチャンネルビットフラグ

設定するプレイヤーで再生するシーケンスが、確保できるチャンネルのビットフラグを指定します。0 を指定すると、全てのチャンネルから確保できるようになります。設定を省略すると、0 を指定したことと見なされます。

チャンネルビットフラグは、下位ビットから順にチャンネル番号1, 番号2, 、, 番号15を表します。チャンネル番号はニンテンドーDS ハードウェアのチャンネル番号です。ビットが立っているチャンネルは確保可能になります。

例えば、チャンネル0～チャンネル7までのみを使用したい場合は、

0 : 1, 0, 0x00ff

のようにします。

シーケンスとチャンネル制限について詳しくは、「サウンドシステムマニュアル」をご覧ください。

## 9 ストリームプレイヤー情報セクション

ストリームプレイヤー情報セクションは、ストリームプレイヤーを定義するためのセクションです。`@STRM_PLAYER` で始まります。

### コード 9-1 ストリームプレイヤー情報セクション

```
;;;;;;;;;;  
;; Stream Player  
  
@STRM_PLAYER  
PLAYER_STRM : STEREO, 6, 7
```

### 9.1 書式

次の1行が1つのストリームプレイヤーに対する設定です。設定したプレイヤーのみ使用できます。

```
PLAYER_STRM : STEREO, 6, 7
```

書式は次のようになっています。

<code>playerNo : type, chNo[, chNo]</code>
--

各要素の意味は次の通りです。

表 9-1 ストリームプレイヤー情報定義要素

要素	説明
playerNo	プレイヤー番号またはラベル
type	プレイヤータイプ
chNo	チャンネル番号

以下、各要素について詳しく説明します。

### 9.2 プレイヤー番号

設定するプレイヤーのプレイヤー番号を、0～3 の値で指定します。また次のようにして、ラベルで指定することができます。

```
PLAYER_BGM = 1 : STEREO 4, 5  
PLAYER_FANFARE : MONO, 6  
3 : MONO, 7
```

1行目は、プレイヤー番号 1 でラベル名が `PLAYER_BGM` です。

2行目の様にラベル名のみ指定すると、1つ前の行のプレイヤー番号に1を足した値が、プレイヤー番号として設定されます。この例の場合では、プレイヤー番号は 2 になります。

3行目は、プレイヤー番号のみでラベルは定義しません。従って、このプレイヤーを指すときにもラベル名ではなく、イン

デックス番号で指定する必要があります。

## 9.3 プレイヤータイプ

プレイヤーがステレオかモノラルかを指定します。

表 9-2 ストリームプレイヤータイプ一覧

プレイヤータイプ	説明
MONO	モノラルプレイヤー
STEREO	ステレオプレイヤー

ステレオプレイヤーは、ステレオデータもモノラルデータも再生できますが、ストリームバッファがモノラルプレイヤーの2倍必要になります。

モノラルプレイヤーは、モノラルデータしか再生できません。

## 9.4 チャンネル番号

ストリーム再生に使用するチャンネルのチャンネル番号を指定します。ステレオプレイヤーでは2つ、モノラルプレイヤーでは1つ指定します。

チャンネル番号は 0～15 の範囲で指定できますが、番号によってチャンネルの機能が異なりますので、他の用途との協調を考える必要があります。各チャンネル番号毎の機能については、「サウンドシステムマニュアル」をご覧ください。

## 10 グループ情報セクション

グループ情報セクションは、複数のサウンドデータを一度にロードできるようにするためのグループを定義するためのセクションです。@GROUP で始まります。

### コード 10-1 グループ情報セクション

```
;;;;;;;;;;  
;; Group  
  
@GROUP  
  
GROUP_STATIC:  
  SEQ_SE  
  BANK_SE  
  BANK_MK64
```

### 10.1 書式

次の記述が1つのグループに対する設定です。

```
GROUP_STATIC:  
  SEQ_SE  
  BANK_SE  
  BANK_MK64
```

書式は次のようになっています。

```
groupLabel :  
  
    dataLabel, option  
  
    dataLabel, option  
  
    dataLabel, option
```

各要素の意味は次の通りです。

表 10-1 グループ情報定義要素

要素	説明
groupLabel	グループラベル
dataLabel	サウンドデータラベル
option	オプション(省略可)

まず始めにグループラベルを定義し、その後の行で、そのグループに含めるサウンドデータを指定します。サウンドデータはいくつでも含めることができます。

以下、各要素について詳しく説明します。

## 10.2 グループラベル

これから定義するグループのラベル名を定義します。各グループには、0から順番にインデックス番号が割り振られます。インデックス番号を直接指定したり、ラベル名を定義しないようにすることもできます。

```
GROUP_COMMON = 2 :
GROUP_TITLE   :
10            :
```

1行目は、インデックス番号を直接指定するタイプで、ラベル名は `GROUP_COMMON` で、インデックス番号 `2` となります。

2行目の様にラベル名のみ指定すると、1つ前の行のインデックス番号に1を足した値が、インデックス番号として設定されます。この例の場合では、インデックス番号は3になります。

3行目は、インデックス番号のみの指定で、ラベル名は定義しません。従って、このグループを指すときにもラベル名ではなく、インデックス番号で指定する必要があります。

## 10.3 サウンドデータラベル

グループに含めるサウンドデータをラベルで指定します。サウンドデータとして、シーケンス、シーケンスアーカイブ、バンク、波形アーカイブを含めることができます。

シーケンスを指定すると、そのシーケンスが使うバンク及び波形アーカイブも同時にグループに含められます。また、バンクを指定すると、そのバンクが使う波形アーカイブも同時にグループに含められます。ただし、シーケンスアーカイブを指定しても、バンクなどは自動的に含められません。

表にまとめると次のようになります。

**表 10-2 同時にグループに含められるデータ**

	SEQ	BANK	WAVEARC	SEQARC
SEQ	○	○	○	
BANK		○	○	
WAVEARC			○	
SEQARC				○

左の列が指定したデータの種類の種類です。○がついているものが、同時にグループに含まれます。これらは、後で説明するオプションを使えば、もっと細かく制御できます。

なお、1つのグループ中に同じサウンドデータを2回以上登録しても、2重に含まれることはありません。

### 10.3.1 インデックス番号指定

ラベルの代わりに、インデックス番号を使って指定することもできます。データラベルの代わりに次のように書きます。

データタイプ[ インデックス番号 ]

データタイプには、"SEQ"、"SEQARC"、"BANK"、"WAVEARC"のいずれかを指定します。



```
GROUP_1 :  
  SEQ[ 2 ]  
  BANK[ 1 ]
```

## 10.4 オプション

オプションでは、どのデータを含めるのかを細かく指定します。オプションは省略可能です。

's'または'b'または'w'の文字の組み合わせで指定します。例えば、次のようになります。

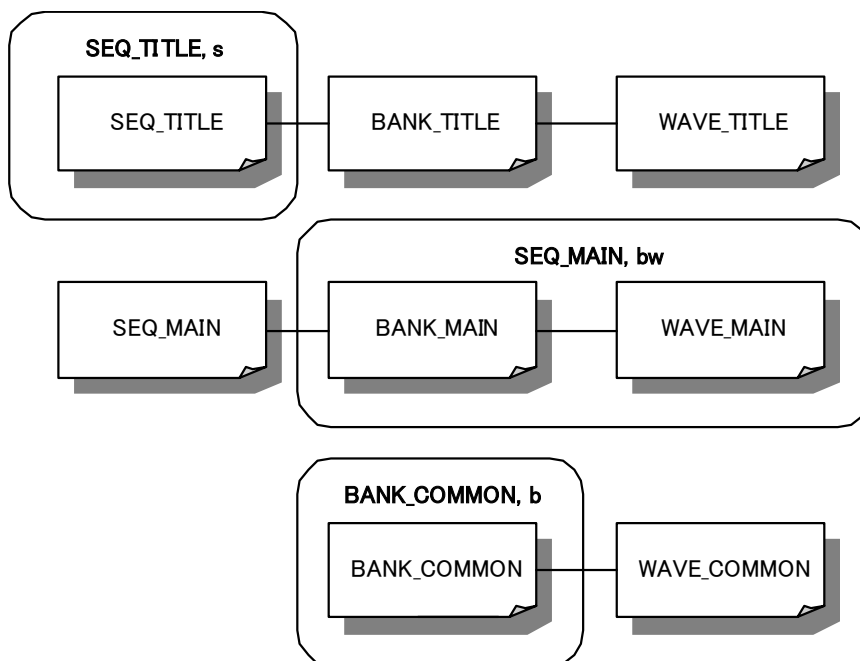
```
GROUP_2 :  
  SEQ_TITLE, s  
  SEQ_MAIN, bw  
  BANK_COMMON, b
```

's', 'b', 'w'はそれぞれ、シーケンス、バンク、波形アーカイブを表します。

SEQ\_TITLE だけでは、シーケンス、バンク、波形アーカイブ全てをグループに含めることになりますが、's'をオプション指定することで、シーケンスデータのみ含めることができます。また、オプション bw では、シーケンスは含めずに、バンク、波形アーカイブを含めることになります。

上の例を図で表すと、次のようになります。

図 10-1 オプションの例



シーケンスアーカイブでオプションを指定することはできません。また、バンクに対して、オプション s を指定するということもできません。

## 11 サウンドマップファイル

### 11.1 サウンドマップファイルとは

サウンドマップファイルとは、サウンドアーカイブにどのようなデータが格納されているのかといった情報が記載されたテキストファイルです。サウンドマップファイルを見ることで、グループのサイズや、個々のサウンドデータのサイズなどを簡単に確認できます。

このファイルは、サウンドアーカイブを作成したときに、自動的に作成されます。サウンドアーカイブ定義ファイルのファイル名が、sound\_data.sarc だった場合、sound\_data.smap というファイル名で出力されます。

### 11.2 サウンドマップファイルの内容

サンプルデータをコンバートして生成されたサウンドマップファイルは、次のようなものです。

コード 11-1 サウンドマップファイル

```
# GROUP:
# label          number    hsize  type    number    hsize
  GROUP_STATIC      0    208352
                        SEQARC      0      736
                        BANK       0    42592
                        BANK       1   165024

# SEQ:
# label          number  fileID  bnk  vol  cpr  ppr  ply    hsize    size name
  SEQ_MARIOKART      0      0    1  127   64   64   0    172032    6928
mid/kart64_title.sseq

# SEQARC:
# label          number  fileID    size name
  SEQ_SE          0      1        652
mus/se.ssar

# BANK:
# label          number  fileID  wa0  wa1  wa2  wa3    hsize    size name
  BANK_SE         0      2    0      42592    132
bnk/se.sbnk
  BANK_BGM        1      3    1    165024    1156
bnk/bgm.sbnk

# WAVEARC:
# label          number  fileID    size name
  WAVE_SE         0      4      42304
swar/se.swar
  WAVE_BGM        1      5    163684
swar/bgm.swar

# STRM:
# label          number  fileID  vol  pri  ply    size name
  STRM_MARIOKART      0      6  127   64   0    6073912
strm/kart_title.32.pcm8.strm
  STRM_FANFARE        1      7  127   64   0    132908
```

```

strm/fanfare.32.pcm8.strm

# FAT:
# fileID      offset      size name
  0 0x00000440    6928 mid/kart64_title.sseq
  1 0x00001f60     652 mus/se.ssar
  2 0x00002200     132 bnk/se.sbnk
  3 0x000022a0    1156 bnk/bgm.sbnk
  4 0x00002740   42304 swar/se.swar
  5 0x0000cc80  163684 swar/bgm.swar
  6 0x00034c00 6073912 strm/kart_title.32.pcm8.strm
  7 0x005ffa40  132908 strm/fanfare.32.pcm8.strm

```

上から順に、グループの情報、シーケンスの情報、シーケンスアーカイブの情報、、、と並んでいます。最後の FAT とあるものは、サウンドアーカイブデータファイル中で、個々のファイルがどのような順番で、どの位置に格納されているかを記載しています。

ほとんどの内容については、サウンドマップファイル中のコメントを見ればわかると思います。以下、わかりづらい点のみ説明します。

### 11.2.1 実データサイズと消費ヒープサイズ

サウンドマップファイル中、“size”と“hsize”の項目があります。“size”は実データサイズを表し、“hsize”は、消費ヒープサイズを表します。

実データサイズは、純粋なサウンドデータのサイズです。ファイルサイズそのものに相当します。

消費ヒープサイズは、そのデータをロードしたときに消費されるヒープのサイズです。例えば、バンクをロードしたとき、使用する波形アーカイブも同時にロードされますので、その分も換算した値になります。

### 11.2.2 グループ

グループに関しては、まず1行目にグループのサイズなどが記載されています。2行目以降に、そのグループがどのデータで構成されているかが書いてあります。

#### コード 11-2 グループ

```

# GROUP:
# label      number      hsize type      number      hsize
GROUP_STATIC      0      208352
                      SEQARC      0      736
                      BANK      0      42592
                      BANK      1      165024

```

上の例では、グループのサイズとしては、208352 バイトになります。また、このグループは、3 つの項目で構成されていて、各項目毎のサイズがわかります。(各サイズは、消費ヒープサイズです。)

なお、各項目毎のサイズの合計が、必ずしもグループサイズに等しくならないことに注意してください。グループサイズは、重複したデータを1つのデータとして計算します。従って、各バンクで共通の波形アーカイブを使用している場合など、合計値よりも小さくなります。

### 11.2.3 ファイル ID

所々に“fileID”と書いてある値は、サウンドアーカイブ内部で使用している識別番号で、サウンドアーカイブ中の全サウ

ンドデータに割り振られています。

ファイル ID が同じなら、それは同じサウンドデータを指していることになります。例えば、別々のシーケンスが同じシーケンスデータを使用している場合、それぞれのシーケンスが保持しているファイル ID は同じになります。

© 2004,2005 Nintendo

任天堂株式会社の許諾を得ることなく、本書に記載されている内容の一部あるいは全部を無断で複製・複写・転写・頒布・貸与することを禁じます。