

# NITRO-Composer バンクデータマニユアル

2008-04-08

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 .....	6
2	波形データ .....	7
2.1	波形フォーマット .....	7
2.1.1	ファイル形式 .....	7
2.1.2	チャンネル数 .....	7
2.1.3	量子化ビット数 .....	7
2.1.4	サンプリングレート .....	7
2.1.5	オリジナルキー .....	7
2.1.6	ループ .....	7
2.2	波形エディタ .....	7
2.3	波形データの登録 .....	7
3	バンク定義ファイル .....	8
3.1	構成 .....	8
3.1.1	セクション .....	9
3.2	コメント .....	9
3.3	ラベル .....	9
3.3.1	プログラムナンバリストファイル .....	9
3.4	@PATH .....	9
3.5	数値表記 .....	10
3.5.1	2進数、16進数表記 .....	10
3.5.2	ビット表記 .....	10
3.5.3	数式 .....	10
4	インストセクション .....	12
4.1	インスト .....	12
4.2	PCMタイプ .....	12
4.2.1	書式 .....	12
4.2.2	プログラムナンバ .....	13
4.2.3	フォーマット .....	13
4.2.4	ファイル名 .....	14
4.2.5	オリジナルキー .....	14
4.2.6	エンベロープ .....	14
4.2.6.1	リリースの無効化 .....	14
4.2.7	パン .....	14
4.3	PSGタイプ .....	14
4.3.1	書式 .....	15
4.3.2	プログラムナンバ .....	15
4.3.3	デューティ比 .....	15
4.3.4	オリジナルキー .....	16
4.3.5	エンベロープ .....	16

4.3.6	パン .....	16
4.4	ノイズタイプ .....	16
4.4.1	書式 .....	16
4.4.2	プログラムナンバ .....	17
4.4.3	オリジナルキー .....	17
4.4.4	エンベロープ .....	17
4.4.5	パン .....	17
4.5	ドラムセットタイプ .....	17
4.5.1	書式 .....	17
4.5.2	プログラムナンバ .....	17
4.5.3	ドラムセットラベル .....	18
4.6	キースプリットタイプ .....	18
4.6.1	書式 .....	18
4.6.2	プログラムナンバ .....	18
4.6.3	キースプリットラベル .....	18
4.7	NULLタイプ .....	18
4.7.1	書式 .....	18
4.7.2	プログラムナンバ .....	19
5	ドラムセットセクション .....	20
5.1	ドラムセット .....	20
5.2	書式 .....	20
5.3	ドラムセットラベル .....	21
5.4	キー .....	21
5.5	インスト定義 .....	21
5.6	未定義キーへの自動割付 .....	21
6	キースプリットセクション .....	22
6.1	キースプリット .....	22
6.2	書式 .....	22
6.3	キースプリットラベル .....	23
6.4	キー .....	23
6.5	インスト定義 .....	23
7	波形データのグループ管理 .....	25
7.1	イントロダクション .....	25
7.2	グループ分け .....	25
7.3	@WGROUP .....	26
7.3.1	キースプリットとドラムセットの波形グループ .....	26
7.4	複数波形アーカイブとの関連づけ .....	27
7.5	波形アーカイブのロード .....	27
8	付録 .....	28

8.1	エンベロープテーブル .....	28
8.1.1	アタックテーブル .....	28
8.1.2	ディケイ／リリーステーブル .....	29
8.1.3	最大リリースタイムテーブル .....	30

## コード

コード 3-1	バンク定義ファイル .....	8
コード 4-1	インストセクション .....	12
コード 5-1	ドラムセットセクション .....	20
コード 6-1	キースプリットセクション .....	22

## 表

表 3-1	演算子 .....	10
表 4-1	PCMタイプインスト要素 .....	13
表 4-2	PCM波形フォーマット .....	13
表 4-3	PSGタイプインスト要素 .....	15
表 4-4	PSG矩形波デューティ比 .....	15
表 4-5	ノイズタイプインスト要素 .....	16
表 4-6	ドラムセットタイプインスト要素 .....	17
表 4-7	キースプリットタイプインスト要素 .....	18
表 4-8	NULLタイプインスト要素 .....	19
表 5-1	ドラムセット定義要素 .....	21
表 6-1	キースプリット定義要素 .....	23
表 6-2	スプリットの範囲 .....	23
表 8-1	アタックテーブル .....	28
表 8-2	ディケイ／リリーステーブル .....	29
表 8-3	最大リリースタイムテーブル .....	30

## 改訂履歴

改訂日	改訂内容
2008-04-08	1. 改訂履歴の書式を変更 2. ページのヘッダを修正
2006-05-29	1. 誤記修正
2005-06-06	1. リリースの無効化に関する説明追加 2. オリジナルキーの数値指定に関する説明追加
2005-03-28	1. キースプリットとドラムセットの波形グループに関する説明追加 2. NULL タイプインストの説明追加 3. 数値表記に関する説明追加
2005-01-31	1. 波形データ個別ロード機能追加に伴い、波形データのグループ管理に関する説明を修正 2. @PATH における UNC 形式の表記に関する説明追加
2004-08-10	1. オリジナルキー省略時の説明補足
2004-07-20	1. 全体的な構成の変更 2. @WGROUP による波形データのグループ分け機能追加
2004-06-01	1. インストゥルメント毎にラベルを指定できるようになった事に伴い修正 2. リリース・ディケイの調整のため、ディケイ・リリーステーブルの及び、最大リリースタイムテーブル修正 3. 見出し番号がずれていたのを修正
2004-04-12	1. バンクコンバータの説明をサウンドツールマニュアルへ移動
2004-04-01	1. 波形フォーマットに関する説明追加 2. エンベロープ値に関する詳細仕様記載
2004-03-01	初版

# 1 概要

バンクデータは、音源データに相当するデータです。複数の波形データを、より意味をもった複数の楽器として構成します。各楽器はプログラムナンバと関連づけられます。バンクデータは、シーケンサからのプログラムナンバとノートコマンドを受け取って、発音を行います。

バンクデータを作成するには、テキストエディタを使いバンクフォーマットに従ってバンクデータの定義を行います。このファイルをバンク定義ファイルと呼びます。バンク定義ファイルでは、波形ファイルを使った楽器の定義などを行います。

ここでは、まず始めに **NITRO-Composer** で扱う波形データについて説明します。続いて、それらの波形データを用いて、テキスト形式のバンク定義ファイルを記述する方法について説明します。

## 2 波形データ

ここでは、波形データを作成するに当たって、必要な基礎知識について説明します。

### 2.1 波形フォーマット

---

#### 2.1.1 ファイル形式

---

扱える波形ファイルのフォーマットは、「Audio Interchange File Format(AIFF)」「Wave(Microsoft)」の2種類です。

#### 2.1.2 チャンネル数

---

チャンネル数は1、すなわちモノラルでなければなりません。

#### 2.1.3 量子化ビット数

---

量子化ビット数は、コンバータで自動的に出力形式に変換されますので制限はありませんが、できるだけ出力形式に揃えておくことが望ましいです。例えば、8bitPCM で出力する場合は8ビットに、ADPCM で出力する場合には16ビットにします。

#### 2.1.4 サンプリングレート

---

サンプリングレートはほぼ無制限で、現実的な範囲である4kHz～44.1kHz程度は使用可能です。サンプリングレートが高いほど音質は良くなりますが、データ転送の頻度が多くなり処理負荷増大の原因になりますので、注意してください。

#### 2.1.5 オリジナルキー

---

オリジナルキーは、波形ファイルで設定せずに、バンクファイルに記述します。波形ファイルに設定しても無視されます。

#### 2.1.6 ループ

---

1種類のループに対応しています。

### 2.2 波形エディタ

---

使用する波形エディタソフトは、上記の波形フォーマットを出力できるものなら、どのようなものでも構いません。

### 2.3 波形データの登録

---

波形データは、下で説明するバンク定義ファイルに記述する形で登録します。コンバートはバンクファイルのコンバート時に、自動的に行われます。

## 3 バンク定義ファイル

### 3.1 構成

下記のサンプルを参考に説明していきます。

#### コード 3-1 バンク定義ファイル

```
@PATH "../aif"

@INSTLIST
  0 : KEY_SPLIT, _SLAPBASS
  1 : ADPCM, "organ.b4.aiff", bn4, 127,127,127,123
  2 : ADPCM, "organ.b4.aiff", bn4, 127,127,127,123
  3 : KEY_SPLIT, _ORGAN_DET01
  4 : KEY_SPLIT, _ORGAN_DET02
  5 : ADPCM, "strings.ds5.aiff", ds5, 127,127,127,100
  6 : ADPCM, "starbell.e6.aiff", en6, 127,127,127,100
  7 : KEY_SPLIT, _PIANO

 11 : KEY_SPLIT, _HIHAT
 12 : ADPCM, "cabasa.aiff", fs4, 127,127,127,120

127 : DRUM_SET, _DRUM_SET

@DRUM_SET

_DRUM_SET =
cn2 : ADPCM, "bassdrum.aiff", , , 127,127,127,120,64
en2 : ADPCM, "snare.aiff", , , 127,127,127,120,64
cs3 : ADPCM, "cym_crash.aiff", cn3, 127,127,127,120,34
en3 : ADPCM, "cym_crash.aiff", , , 127,127,127,120,94
gn3 : ADPCM, "cym_splash.aiff", , , 127,127,127,120,54
fn4 : ADPCM, "timbale_h.aiff", , , 127,127,127,120,55

@KEY_SPLIT

_SLAPBASS =
bn3 : ADPCM, "bass_slpthm.e2.aiff", en2, 127,127,127,123
127 : ADPCM, "bass_pull.e2.aiff", en4, 127,127,127,123

_ORGAN_DET01 =
ds4 : ADPCM, "organ_det.g3.aiff", gn3, 127,127,127,125
127 : ADPCM, "organ_det.g4.aiff", gn4, 127,127,127,125

_ORGAN_DET02 =
ds4 : ADPCM, "organ_det.g3.aiff", gn3, 127,127,127,125
127 : ADPCM, "organ_det.g4.aiff", gn4, 127,127,127,125

_PIANO =
bn5 : ADPCM, "piano.g5.aiff", gn5, 127,127,127,100
127 : ADPCM, "piano.g6.aiff", gn6, 127,127,127,100

_HIHAT =
gn2 : ADPCM, "hhcl.fs2.aiff", fs2, 127,127,127,123
```



```
an2 : ADPCM, "hhpd.gs2.aiff", gs2, 127,127,127,123
127 : ADPCM, "hhop.as2.aiff", as2, 127,127,127,123
```

### 3.1.1 セクション

---

全体的な構成としては、大きく3つのセクションに分かれます。

- @INSTLIST で始まる、インストセクション
- @DRUM\_SET で始まる、ドラムセットセクション
- @KEY\_SPLIT で始まる、キースプリットセクション

必要のないセクションは、作成する必要はありません。

各セクションについて詳しくは、あとで説明します。

## 3.2 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。

```
1 : ADPCM, "organ.b4.aiff", bn4, 127,127,127,123 ;;; comment
```

## 3.3 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、"PRG\_ORGAN"がラベルです。

```
PRG_ORGAN : ADPCM, "organ.b4.aiff", bn4, 127,127,127,123
```

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。

```
PRG_STRINGS
PRG_PIANO_01
PRG_DRUM_SET
```

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。

### 3.3.1 プログラムナンバリストファイル

---

プログラムナンバに対応したラベルを定義すると、拡張子\*.spdl のプログラムナンバリストファイルに、ラベルをプログラムナンバに置換する情報が出力されます。このファイルをテキストシーケンスファイルでインクルードすることにより、プログラムチェンジを番号ではなく、ラベルで記述することができるようになります。

詳しくは、「サウンドツールマニュアル」の「プリプロセッサ命令」を参照してください。

## 3.4 @PATH

---

波形ファイルを指定するとき、デフォルトではバンク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 "aif"
1 : ADPCM, "organ.b4.aiff", bn4, 127,127,127,123
```

この例では、"aif/organ.b4.aiff"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。

```
1 : ADPCM, "aif/organ.b4.aiff", bn4, 127,127,127,123
```

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/aif"
```

## 3.5 数値表記

バンク定義ファイル中で、エンベロープ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。

### 3.5.1 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
```

### 3.5.2 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。

### 3.5.3 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 3-1 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

## 4 インストセクション

インストセクションは、波形ファイルなどを使って、1つの楽器を定義するセクションです。@INSTLIST で始まります。

### コード 4-1 インストセクション

```
@INSTLIST
0 : KEY_SPLIT, _SLAPBASS
1 : ADPCM, "organ.b4.aiff", bn4, 127,127,127,123
2 : ADPCM, "organ.b4.aiff", bn4, 127,127,127,123
3 : KEY_SPLIT, _ORGAN_DET01
4 : KEY_SPLIT, _ORGAN_DET02
5 : ADPCM, "strings.ds5.aiff", ds5, 127,127,127,100
6 : ADPCM, "starbell.e6.aiff", en6, 127,127,127,100
7 : KEY_SPLIT, _PIANO

11 : KEY_SPLIT, _HIHAT
12 : ADPCM, "cabasa.aiff", fs4, 127,127,127,120

127 : DRUM_SET, _DRUM_SET
```

### 4.1 インスト

ここでは、1つのプログラムナンバに対応した音色または音色のセットをインスト(楽器)と呼びます。シーケンスデータからプログラムチェンジを実行するということは、使用するインストを選択するということを意味します。

インストの種類は大きく分けて、6 種類あります。

- PCM タイプ
- PSG タイプ
- ノイズタイプ
- ドラムセットタイプ
- キースプリットタイプ
- NULL タイプ

以下、それぞれについてくわしく説明していきます。

### 4.2 PCMタイプ

PCMタイプは、PCM波形データを使って発音するインストです。

#### 4.2.1 書式

次のように指定します。

```
1 : ADPCM, "organ.b4.aiff", bn4, 127,127,127,123
```

書式は次のようになっています。

```
prgNo : format, filename, originalKey, attack, decay, sustain, release, pan
```

各要素の意味は次の通りです。

**表 4-1 PCM タイプインスト要素**

要素	説明
prgNo	プログラムナンバ
format	フォーマット
filename	ファイル名
originalKey	オリジナルキー(省略可)
attack	エンベロープのアタックタイム
decay	エンベロープのディケイタイム
sustain	エンベロープのサステインレベル
release	エンベロープのリリースタイム
pan	パン(省略可)

以下、各要素について詳しく説明します。

## 4.2.2 プログラムナンバ

プログラムナンバを指定します。同じ番号を2重定義することはできません。値の範囲は、0 ～ 32767 までですが、MIDI からは 127 までしか指定できません。

また、下記のようにして各プログラムナンバに対応するラベルを定義することもできます。

```
PRG_ORGAN = 1 : ADPCM, "organ.b4.aiff",    bn4, 127,127,127,123
PRG_STRINGS : ADPCM, "strings.ds5.aiff", ds5, 127,127,127,100
5           : ADPCM, "organ.b4.aiff",    bn4, 127,127,127,123
```

1行目は、プログラムナンバとラベルの両方を指定するタイプで、プログラムナンバ 1、ラベル PRG\_ORGAN となります。

2行目の様にラベル名のみ指定すると、1つ前の行のプログラムナンバに1を足した値が、プログラムナンバとして設定されます。この例の場合では、プログラムナンバは 2 になります。

3行目は、プログラムナンバのみの指定で、ラベル名は指定しません。

## 4.2.3 フォーマット

PCM 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 4-2 PCM 波形フォーマット**

フォーマット	説明
PCM16	16bit PCM データに変換します。
PCM8	8bit PCM データに変換します。
ADPCM	ADPCM データに変換します。

SWAV	変換しません。
------	---------

#### 4.2.4 ファイル名

フォーマットの指定が SWAV 以外の場合、モノラル AIFF または WAV ファイルを指定します。SWAV の場合は、コンバート後の\*.swav ファイルを指定します。

パスは、バンク定義ファイルからの相対パスですが、@PATH コマンドで起点となるディレクトリを変更することもできます。

#### 4.2.5 オリジナルキー

波形ファイルのオリジナルキーを指定します。波形ファイルの音程を正しく設定することで、再生時にも正しい音程で再生することができます。ただし、故意にオリジナルキーをずらすことで、再生時の音程をずらすこともできます。省略時は cn4 に設定されます。

なお、オリジナルキーには、キー表記の他に数値でも指定することができます。数値の 60 は、cn4 を表します。

#### 4.2.6 エンベロープ

エンベロープの ADSR の指定を行います。値の範囲は全て 0～127 です。

- アタックタイムは、値が大きいほど立ち上がりが早くなります。
- ディケイタイムは、値が大きいほど減衰が早くなります。
- サステインレベルは、値が大きいほどゲインが大きくなります。数値のスケールはベロシティと同じです。
- リリースタイムは、値が大きいほど減衰が早くなります。

アタック、ディケイ、リリースに関する詳細な仕様は、「8.1エンベロープテーブル」を参照してください。

##### 4.2.6.1 リリースの無効化

リリースタイムだけは、数値の代わりに"DISABLE"という文字列を指定できます。リリースが DISABLE であるインストは、指定したノート長が終わったとしてもリリースされずに波形データの終端まで発音されます。

```
0 : ADPCM, "bassdrum.aiff" , cn4, 127,127,127, DISABLE
```

例えばパーカッションなどのリリースが不要なインストはリリースを無効化しておくことで、発音長やノートオフのタイミングを気にすることなく、波形データの最後まで発音させることができます。

ただし、ループする波形データに対してリリースを無効化すると、シーケンスを止めるまで発音が続くことに注意してください。

#### 4.2.7 パン

インスト毎のパンの値を指定します。値の範囲は 0(左)～64(中央)～127(右)です。省略時は 64(中央)になります。

パンの影響は、トラックで設定するものなどと足し合わされます。

### 4.3 PSGタイプ

PSG タイプは、PSG 矩形波を使って発音するインストです。

### 4.3.1 書式

次のように指定します。

0 : PSG, DUTY\_4\_8, cn4, 127,127,127,123

書式は次のようになっています。

prgNo : PSG, duty, originalKey, attack, decay, sustain, release, pan

各要素の意味は次の通りです。

表 4-3 PSG タイプインスト要素

要素	説明
prgNo	プログラムナンバ
duty	デューティ比
originalKey	オリジナルキー(省略可)
attack	エンベロープのアタックタイム
decay	エンベロープのディケイタイム
sustain	エンベロープのサステインレベル
release	エンベロープのリリースタイム
pan	パン(省略可)

以下、各要素について詳しく説明します。

### 4.3.2 プログラムナンバ

プログラムナンバを指定します。指定方法は、PCMタイプのインストと同じです。「4.2.2プログラムナンバ」をご覧ください。

### 4.3.3 デューティ比

PSG 矩形波のデューティ比を指定します。指定できる値は下の通りです。

表 4-4 PSG 矩形波デューティ比

ラベル	説明
DUTY_1_8	1/8(12.5%) デューティ
DUTY_2_8	2/8(25.0%) デューティ
DUTY_3_8	3/8(37.5%) デューティ
DUTY_4_8	4/8(50.0%) デューティ
DUTY_5_8	5/8(62.5%) デューティ
DUTY_6_8	6/8(75.0%) デューティ

DUTY_7_8	7/8(87.5%) デューティ
----------	------------------

#### 4.3.4 オリジナルキー

オリジナルキーを指定します。cn4 の時、正しい音程で再生することができます。ただし、故意にオリジナルキーをずらすことで、再生時の音程をずらすこともできます。省略時は cn4 に設定されます。

なお、オリジナルキーには、キー表記の他に数値でも指定することができます。数値の 60 は、cn4 を表します。

#### 4.3.5 エンベロープ

エンベロープのADSRの指定を行います。指定方法は、PCMタイプのインストと同じです。「4.2.6エンベロープ」をご覧ください。

#### 4.3.6 パン

インスト毎のパンの値を指定します。値の範囲は 0(左)～64(中央)～127(右)です。省略時は 64(中央)になります。

パンの影響は、トラックで設定するものなどと足し合わされます。

### 4.4 ノイズタイプ

ノイズタイプは、ホワイトノイズ使って発音するインストです。

#### 4.4.1 書式

次のように指定します。

0 : NOISE, cn4, 127,127,127,123

書式は次のようになっています。

prgNo : NOISE, originalKey, attack, decay, sustain, release, pan
--

各要素の意味は次の通りです。

表 4-5 ノイズタイプインスト要素

要素	説明
prgNo	プログラムナンバ
originalKey	オリジナルキー(省略可)
attack	エンベロープのアタックタイム
decay	エンベロープのディケイタイム
sustain	エンベロープのサステインレベル
release	エンベロープのリリースタイム
pan	パン(省略可)

以下、各要素について詳しく説明します。



#### 4.4.2 プログラムナンバ

プログラムナンバを指定します。指定方法は、PCMタイプのインストと同じです。「4.2.2プログラムナンバ」をご覧ください。

#### 4.4.3 オリジナルキー

オリジナルキーを指定します。cn4 の時、正しい音程で再生することができます。ただし、故意にオリジナルキーをずらすことで、再生時の音程をずらすこともできます。省略時は cn4 に設定されます。

なお、オリジナルキーには、キー表記の他に数値でも指定することができます。数値の 60 は、cn4 を表します。

#### 4.4.4 エンベロープ

エンベロープのADSRの指定を行います。指定方法は、PCMタイプのインストと同じです。「4.2.6エンベロープ」をご覧ください。

#### 4.4.5 パン

インスト毎のパンの値を指定します。値の範囲は 0(左)～64(中央)～127(右)です。省略時は 64(中央)になります。

パンの影響は、トラックで設定するものなどと足し合わされます。

### 4.5 ドラムセットタイプ

ドラムセットタイプは、ドラムセット使って発音するインストです。ドラムセットについて詳しくは、5章で説明します。

#### 4.5.1 書式

ドラムセットタイプのインストは、次のように指定します。

```
127 : DRUM_SET, _DRUM_SET
```

書式は次のようになっています。

prgNo : DRUM_SET, label
-------------------------

各要素の意味は次の通りです。

表 4-6 ドラムセットタイプインスト要素

要素	説明
prgNo	プログラムナンバ
label	ドラムセットラベル

以下、各要素について詳しく説明します。

#### 4.5.2 プログラムナンバ

プログラムナンバを指定します。指定方法は、PCMタイプのインストと同じです。「4.2.2プログラムナンバ」をご覧ください。

### 4.5.3 ドラムセットラベル

ドラムセットセクションで定義したドラムセットをラベルで指定します。詳しくは、「5ドラムセットセクション」で説明します。

## 4.6 キースプリットタイプ

キースプリットタイプは、キースプリット使って発音するインストです。キースプリットについて詳しくは、6章で説明します。

### 4.6.1 書式

キースプリットタイプのインストは、次のように指定します。

0 : KEY\_SPLIT, \_SLAPBASS

書式は次のようになっています。

prgNo : KEY_SPLIT, label
--------------------------

各要素の意味は次の通りです。

表 4-7 キースプリットタイプインスト要素

要素	説明
prgNo	プログラムナンバ
label	キースプリットラベル

以下、各要素について詳しく説明します。

### 4.6.2 プログラムナンバ

プログラムナンバを指定します。指定方法は、PCMタイプのインストと同じです。「4.2.2プログラムナンバ」をご覧ください。

### 4.6.3 キースプリットラベル

キースプリットセクションで定義したキースプリットをラベルで指定します。詳しくは、「6キースプリットセクション」で説明します。

## 4.7 NULLタイプ

NULL タイプは、何も発音しないタイプのインストです。例えばキースプリットと組み合わせて、あるスプリットのみ発音しないインストを作成することができます。

### 4.7.1 書式

NULL タイプのインストは、次のように指定します。

0 : NULL

書式は次のようになっています。

prgNo : NULL
--------------

各要素の意味は次の通りです。

**表 4-8 NULL タイプインスタ要素**

要素	説明
prgNo	プログラムナンバ

以下、各要素について詳しく説明します。

#### 4.7.2 プログラムナンバ

プログラムナンバを指定します。指定方法は、PCMタイプのインスタと同じです。「4.2.2プログラムナンバ」をご覧ください。

## 5 ドラムセットセクション

ドラムセットセクションは、ドラムセットを定義するセクションです。@DRUM\_SET で始まります。

### コード 5-1 ドラムセットセクション

```
@DRUM_SET

_DRUM_SET =
cn2 : ADPCM, "bassdrum.aiff" , , 127,127,127,120,64
en2 : ADPCM, "snare.aiff" , , 127,127,127,120,64
cs3 : ADPCM, "cym_crash.aiff" ,cn3, 127,127,127,120,34
en3 : ADPCM, "cym_crash.aiff" , , 127,127,127,120,94
gn3 : ADPCM, "cym_splash.aiff" , , 127,127,127,120,54
fn4 : ADPCM, "timbale_h.aiff" , , 127,127,127,120,55
```

### 5.1 ドラムセット

ドラムセットとは、1つ1つのキーに別の楽器を割り当てたもので、音程を変化させられない代わりに、プログラムチェンジをしなくても別の音色を出せるようにしたものです。主にパーカッションのような音程変化の不要な楽器で使います。ただし、使い方によって多少、音程を変化させることも可能です。詳しくは後で説明します。

### 5.2 書式

次の記述が1つのドラムセットの定義に相当します。

```
_DRUM_SET =
cn2 : ADPCM, "bassdrum.aiff" , , 127,127,127,120,64
en2 : ADPCM, "snare.aiff" , , 127,127,127,120,64
cs3 : ADPCM, "cym_crash.aiff" ,cn3, 127,127,127,120,34
en3 : ADPCM, "cym_crash.aiff" , , 127,127,127,120,94
gn3 : ADPCM, "cym_splash.aiff" , , 127,127,127,120,54
fn4 : ADPCM, "timbale_h.aiff" , , 127,127,127,120,55
```

書式は次のようになっています。

```
drumsetLabel =

    key : inst-definition

    key : inst-definition

    key : inst-definition
```

各要素の意味は次の通りです。

表 5-1 ドラムセット定義要素

要素	説明
drumsetLabel	ドラムセットラベル
key	キー
inst-definition	インスト定義

まず始めに、ドラムセットラベルを定義し、その後の行で、個々のキーに対してインストを割り付けます。

以下、各要素について詳しく説明します。

## 5.3 ドラムセットラベル

定義するドラムセットのラベル名を指定します。ここで指定したラベル名を、ドラムセットタイプのインスト定義の際に、使います。

## 5.4 キー

どのキーに対してインストを割り付けるのかを指定します。同じキーへの2重定義はできません。

## 5.5 インスト定義

個々のキーに対するインスト定義を行います。インスト定義の書式は、インストセクションでの書式に倣います。ただし、キースプリットタイプのインストや、ドラムセットタイプのインストを指定することはできません。

また、オリジナルキーの指定を省略した場合は、"cn4"ではなく、割り当てキーと同じキーに設定されます。すなわち、PCM タイプのインストの場合、波形データそのままの音程で鳴ることになります。PSG タイプやノイズタイプのインストの場合は、中央のドの音程で鳴ります。

## 5.6 未定義キーへの自動割付

定義されていないキーに対しては、それより低い音程のキーに対する設定がそのまま引き継がれます。

例えば次の例では、cs2 ～ ds2 まで未定義ですが、これらには cn2 の設定が引き継がれます。

```
_DRUM_SET =
cn2 : ADPCM, "bassdrum.aiff" , , 127,127,127,120,64
en2 : ADPCM, "snare.aiff" , , 127,127,127,120,64
cs3 : ADPCM, "cym_crash.aiff" ,cn3, 127,127,127,120,34
en3 : ADPCM, "cym_crash.aiff" , , 127,127,127,120,94
gn3 : ADPCM, "cym_splash.aiff" , , 127,127,127,120,54
fn4 : ADPCM, "timbale_h.aiff" , , 127,127,127,120,55
```

この時、オリジナルキーは cn2 のままなので、cn2 ～ ds2 まで音程を変化させて鳴らすことができます。

なお、cn2 より低いキーや、一番高い fn4 より高いキーでは、音は鳴りません。

## 6 キースプリットセクション

キースプリットセクションは、キースプリットを定義するセクションです。@KEY\_SPLIT で始まります。

### コード 6-1 キースプリットセクション

```
@KEY_SPLIT

_SLAPBASS =
bn3 : ADPCM, "bass_slpthm.e2.aiff", en2, 127,127,127,123
127 : ADPCM, "bass_pull.e2.aiff", en4, 127,127,127,123

_ORGAN_DET01 =
ds4 : ADPCM, "organ_det.g3.aiff", gn3, 127,127,127,125
127 : ADPCM, "organ_det.g4.aiff", gn4, 127,127,127,125

_ORGAN_DET02 =
ds4 : ADPCM, "organ_det.g3.aiff", gn3, 127,127,127,125
127 : ADPCM, "organ_det.g4.aiff", gn4, 127,127,127,125

_PIANO =
bn5 : ADPCM, "piano.g5.aiff", gn5, 127,127,127,100
127 : ADPCM, "piano.g6.aiff", gn6, 127,127,127,100

_HIHAT =
gn2 : ADPCM, "hhcl.fs2.aiff", fs2, 127,127,127,123
an2 : ADPCM, "hhpd.gs2.aiff", gs2, 127,127,127,123
127 : ADPCM, "hhop.as2.aiff", as2, 127,127,127,123
```

### 6.1 キースプリット

キースプリットとは、あるキーの範囲毎に別々の波形を割り当てることができるものです。1つの波形では、低音から高音までカバーできないようなときに使います。

ここで、あるキーの範囲をスプリットと呼びます。すなわち、スプリット毎に波形ファイルを割り当てることになります。1つのキースプリットに対し、8つまでスプリットを作成することができます。

### 6.2 書式

次の記述が、1つのキースプリットの定義に相当します。

```
_HIHAT =
gn2 : ADPCM, "hhcl.fs2.aiff", fs2, 127,127,127,123
an2 : ADPCM, "hhpd.gs2.aiff", gs2, 127,127,127,123
127 : ADPCM, "hhop.as2.aiff", as2, 127,127,127,123
```

書式は次のようになっています。

```

keysplitLabel =
    key : inst-definition
    key : inst-definition
    key : inst-definition

```

各要素の意味は次の通りです。

**表 6-1 キースプリット定義要素**

要素	説明
keysplitLabel	キースプリットラベル
key	キー
inst-definition	インスタ定義

まず始めに、キースプリットラベルを定義し、その後の行で、スプリットを定義しています。

以下、各要素について詳しく説明します。

## 6.3 キースプリットラベル

定義するキースプリットのラベル名を指定します。ここで指定したラベル名を、キースプリットタイプのインスタ定義の際に、使用します。

## 6.4 キー

スプリットのキーの上限を指定します。一番広域のスプリットの上限キーには、127と記述します。

次のようなキースプリットを定義すると、

```

_HIHAT =
gn2 : ADPCM, "hhcl.fs2.aiff", fs2, 127,127,127,123
an2 : ADPCM, "hhpd.gs2.aiff", gs2, 127,127,127,123
127 : ADPCM, "hhop.as2.aiff", as2, 127,127,127,123

```

各スプリットのキーの範囲は、次のようになります。

**表 6-2 スプリットの範囲**

キーの範囲	波形ファイル
～ gn2	hhcl.fs2.aiff
gs2 ～ an2	hhpd.gs2.aiff
as2 ～	hhop.as2.aiff

## 6.5 インスタ定義

個々のスプリットに対するインスタ定義を行います。インスタ定義の書式は、インスタクションでの書式に倣います。ただ

し、キースプリットタイプのインストや、ドラムセットタイプのインストを指定することはできません。

また、オリジナルキーの指定を省略した場合は、"**cn4**"ではなく、キー範囲の下限キーと同じキーに設定されます。



## 7 波形データのグループ管理

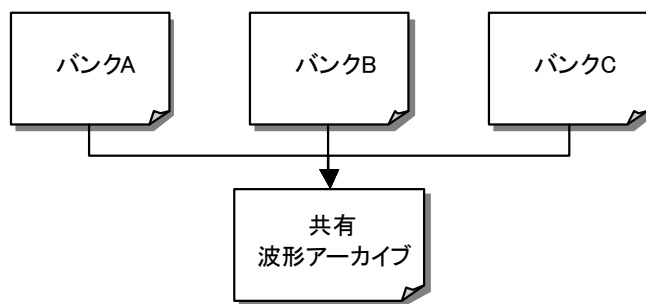
ここでは、波形データのメモリをより効率的に扱えるようにするための、波形データのグループ管理について説明します。

### 7.1 イン트로ダクション

まず、複数のバンク中で同じ波形データを使っていた場合を考えます。もし、1つの波形データを別々のバンクが使用できれば、より少ないメモリですませることができます。

まず思いつく方法は、複数のバンクで1つの波形アーカイブを使うようにすることです。これは、サウンドアーカイブ定義ファイルの記述で簡単に行うことができます。

図 7-1 共有波形アーカイブ



さらに、この共有波形アーカイブに対して、サウンドアーカイブ定義ファイルでのオプションフラグを使って、波形データを個別にロードするように指定しておけば、必要な波形データのみを抽出してロードします。たとえばバンクAでしか使わない波形データが共有波形アーカイブに入っていたとしても、バンクBを使うときにそれをロードする必要はありません。

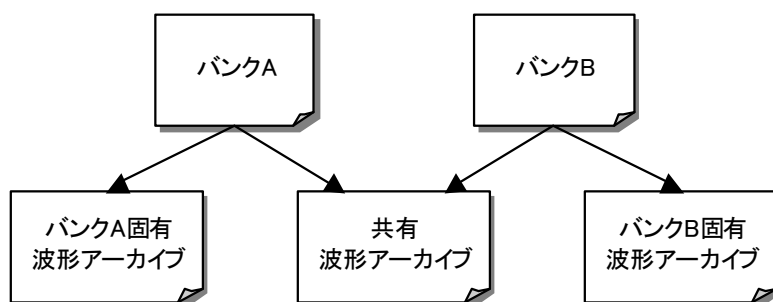
これでほとんどの場合は解決です。ただし、波形データの個別ロードを使うと、消費メモリやロード時間でのオーバーヘッドが発生します。どのバンクがどの波形データを使用しているかといった分類が、ある程度単純な場合は、波形データの個別ロードを使わずに、波形データのグループ管理の機能を使うことで、効率化が図れます。

### 7.2 グループ分け

単純化のためにバンクAとバンクBの2つだけで考えます。

この場合、バンクAが使う波形データは、バンクBと共有の波形データと、バンクAのみの固有波形データの2つに分類できます。これをそのまま、2つの波形アーカイブに分割します。

図 7-2 波形データのグループ分け



このように波形アーカイブを2つに分けることで、波形データの個別ロードを使わなくても、必要最低限の波形データをロードできるようになります。

このような使い方をするためには、あるバンク中で使用している波形データが、共有波形データなのか、固有波形データなのかをグループ分けする必要があります。以下では、このグループ分けの方法を説明します。

## 7.3 @WGROUP

あるバンク中で、波形データをグループ分けするためには、@WGROUPを使います。

```

@INSTLIST
0 : KEY_SPLIT, _SLAPBASS

@WGROUP 1
1 : ADPCM, "organ.b4.aiff", bn4, 127,127,127,123
2 : ADPCM, "organ.b4.aiff", bn4, 127,127,127,123

@WGROUP 0
3 : KEY_SPLIT, _ORGAN_DET01
4 : KEY_SPLIT, _ORGAN_DET02
5 : ADPCM, "strings.ds5.aiff", ds5, 127,127,127,100
6 : ADPCM, "starbell.e6.aiff", en6, 127,127,127,100
7 : KEY_SPLIT, _PIANO

11 : KEY_SPLIT, _HIHAT
12 : ADPCM, "cabasa.aiff", fs4, 127,127,127,120

127 : DRUM_SET, _DRUM_SET
  
```

上の例では、"organ.b4.aiff"をグループ 1 に、それ以外をグループ 0 に設定しています。

@WGROUP でグループ番号を指定すると、それ以降に指定した波形データを指定したグループ番号に分類されます。指定していない場合は、グループ0に分類されます。グループ番号は、0,1,2,3 のどれかを指定します。すなわち、最大4つのグループに分けることができます。

### 7.3.1 キースプリットとドラムセットの波形グループ

キースプリットとドラムセットで使う波形データのグループ指定は、キースプリット単位もしくは、ドラムセット単位で行うのではなく、あくまで波形データ単位で行います。従って、KEY\_SPLIT や DRUM\_SET のインストを定義した時点でのグループ番号は意味を持ちません。

次の例を見てください。

```

@INSTLIST
@WGROUP 1
  3 : KEY_SPLIT, _ORGAN_DET01
  4 : KEY_SPLIT, _ORGAN_DET02

@KEY_SPLIT
@WGROUP 2
_ORGAN_DET01 =
ds4 : ADPCM, "organ_det.g3.aiff", gn3, 127,127,127,125
127 : ADPCM, "organ_det.g4.aiff", gn4, 127,127,127,125

@WGROUP 3
_ORGAN_DET02 =
ds4 : ADPCM, "organ_det.g3.aiff", gn3, 127,127,127,125
127 : ADPCM, "organ_det.g4.aiff", gn4, 127,127,127,125

```

この例では、グループ2とグループ3に分類されます。グループ1にはどの波形データも分類されません。

## 7.4 複数波形アーカイブとの関連づけ

グループ分けしたら、それぞれのグループに対して波形アーカイブを関連づける必要があります。その指定は、サウンドアーカイブ定義ファイルで行います。

```
BANK_A : TEXT, "A.bnk", WAVE_A, WAVE_COMMON
```

上の例では、グループ0の波形データは波形アーカイブ WAVE\_A に、グループ 1 の波形データは波形アーカイブ WAVE\_COMMON に格納されることになります。

例えば、別のバンク B でも、"organ.b4.aiff"を使っていて、それをグループ1にグループ分けしていれば、

```

BANK_A : TEXT, "A.bnk", WAVE_A, WAVE_COMMON
BANK_B : TEXT, "B.bnk", WAVE_B, WAVE_COMMON

```

のようにして、"organ.b4.aiff"の部分だけ、共有化することができます。

## 7.5 波形アーカイブのロード

複数の波形アーカイブと関連づけられたバンクを使用するためには、関連づけられた全ての波形アーカイブもロードする必要があります。ただし、特定の波形アーカイブだけ、前もってロードしておくことは可能です。

```

BANK_A : TEXT, "A.bnk", WAVE_A, WAVE_COMMON
BANK_B : TEXT, "B.bnk", WAVE_B, WAVE_COMMON

```

上の場合、WAVE\_COMMON を前もってロードしておけば、BANK\_A を使うときには、WAVE\_A だけロードすれば良いことになります。また、BANK\_B を使うときにも、WAVE\_B のロードだけですみます。

## 8 付録

### 8.1 エンベロープテーブル

#### 8.1.1 アタックテーブル

アタックの値と、アタックタイムとの対応関係を表した表です。アタックタイムの単位は、msec です。

表 8-1 アタックテーブル

attack	time	attack	time	attack	time	attack	time
0	8606.1	32	369.9	64	182.3	96	114.6
1	4756.3	33	359.5	65	182.3	97	114.6
2	3339.3	34	349.0	66	177.1	98	114.6
3	2594.4	35	338.6	67	177.1	99	109.4
4	2130.7	36	328.2	68	171.9	100	109.4
5	1807.7	37	323.0	69	171.9	101	109.4
6	1573.3	38	312.6	70	166.7	102	109.4
7	1401.4	39	307.4	71	166.7	103	109.4
8	1255.5	40	297.0	72	161.5	104	104.2
9	1140.9	41	291.7	73	161.5	105	104.2
10	1047.1	42	286.5	74	156.3	106	104.2
11	963.8	43	276.1	75	156.3	107	104.2
12	896.0	44	270.9	76	151.1	108	99.0
13	838.7	45	265.7	77	151.1	109	93.8
14	786.6	46	260.5	78	145.9	110	88.6
15	745.0	47	255.3	79	145.9	111	83.4
16	703.3	48	250.1	80	145.9	112	78.2
17	666.8	49	244.9	81	145.9	113	72.9
18	630.4	50	239.6	82	140.7	114	67.7
19	599.1	51	234.4	83	140.7	115	62.5
20	578.3	52	229.2	84	140.7	116	57.3
21	547.0	53	224.0	85	130.2	117	52.1
22	526.2	54	218.8	86	130.2	118	46.9
23	505.3	55	213.6	87	130.2	119	41.7
24	484.5	56	213.6	88	125.0	120	36.5
25	468.9	57	208.4	89	125.0	121	31.3
26	448.0	58	203.2	90	125.0	122	26.1
27	437.6	59	203.2	91	125.0	123	20.8
28	416.8	60	198.0	92	119.8	124	15.6
29	406.4	61	198.0	93	119.8	125	10.4
30	395.9	62	192.8	94	119.8	126	10.4
31	385.5	63	192.8	95	114.6	127	0.0

アタックの値が違っていても、アタックタイムが同じ箇所がありますが、ピークレベルに達する時間が同じと言うだけで、それまでの立ち上がりの変化の仕方が、若干異なります。(値が大きいほど早い)

## 8.1.2 ディケイ／リリーステーブル

ディケイ及びリリースの値と、減衰速度の対応関係を表した表です。減衰速度の単位は、dB/msec です。

表 8-2 ディケイ／リリーステーブル

release	speed	release	speed	release	speed	release	speed
0	-0.0002	32	-0.0098	64	-0.0186	96	-0.0385
1	-0.0005	33	-0.0101	65	-0.0189	97	-0.0398
2	-0.0008	34	-0.0104	66	-0.0192	98	-0.0412
3	-0.0011	35	-0.0107	67	-0.0196	99	-0.0427
4	-0.0014	36	-0.0110	68	-0.0199	100	-0.0444
5	-0.0017	37	-0.0113	69	-0.0202	101	-0.0462
6	-0.0020	38	-0.0116	70	-0.0206	102	-0.0481
7	-0.0023	39	-0.0119	71	-0.0210	103	-0.0502
8	-0.0026	40	-0.0122	72	-0.0214	104	-0.0524
9	-0.0029	41	-0.0125	73	-0.0218	105	-0.0549
10	-0.0032	42	-0.0128	74	-0.0222	106	-0.0577
11	-0.0035	43	-0.0131	75	-0.0226	107	-0.0607
12	-0.0038	44	-0.0134	76	-0.0231	108	-0.0641
13	-0.0041	45	-0.0137	77	-0.0235	109	-0.0679
14	-0.0044	46	-0.0140	78	-0.0240	110	-0.0721
15	-0.0047	47	-0.0143	79	-0.0245	111	-0.0769
16	-0.0050	48	-0.0146	80	-0.0251	112	-0.0824
17	-0.0053	49	-0.0149	81	-0.0256	113	-0.0888
18	-0.0056	50	-0.0152	82	-0.0262	114	-0.0962
19	-0.0059	51	-0.0154	83	-0.0268	115	-0.1049
20	-0.0062	52	-0.0156	84	-0.0275	116	-0.1154
21	-0.0065	53	-0.0158	85	-0.0281	117	-0.1282
22	-0.0068	54	-0.0160	86	-0.0288	118	-0.1442
23	-0.0071	55	-0.0163	87	-0.0296	119	-0.1648
24	-0.0074	56	-0.0165	88	-0.0304	120	-0.1923
25	-0.0077	57	-0.0167	89	-0.0312	121	-0.2308
26	-0.0080	58	-0.0170	90	-0.0321	122	-0.2885
27	-0.0083	59	-0.0172	91	-0.0330	123	-0.3846
28	-0.0086	60	-0.0175	92	-0.0339	124	-0.5769
29	-0.0089	61	-0.0178	93	-0.0350	125	-1.1538
30	-0.0092	62	-0.0180	94	-0.0361	126	-2.2897
31	-0.0095	63	-0.0183	95	-0.0372	127	-9.8460

ディケイの場合、サステインで指定したゲインに到達するまで減衰し続けます。リリースの場合は、ベロシティなどの影響を含めた後のゲインが、-72.3dB 以下になると発音を停止します。

### 8.1.3 最大リリースタイムテーブル

リリースは減衰速度で規定されていますが、直感的にわかりづらいので、リリースタイム(発音が完全に停止するまでの時間)との対応関係を表した表を載せます。ただし、ペロシティなどのボリュームパラメータが全て最大という条件であり、実際にはこれより短い時間になることに注意してください。なお、リリースタイムの単位は、msec です。

表 8-3 最大リリースタイムテーブル

release	time	release	time	release	time	release	time
0	481,228.8	32	7,399.6	64	3,884.4	96	1,877.2
1	160,409.6	33	7,181.2	65	3,822.0	97	1,814.8
2	96,241.6	34	6,973.2	66	3,759.6	98	1,752.4
3	68,744.0	35	6,775.6	67	3,692.0	99	1,690.0
4	53,466.4	36	6,588.4	68	3,629.6	100	1,627.6
5	43,747.6	37	6,411.6	69	3,567.2	101	1,565.2
6	37,013.6	38	6,245.2	70	3,504.8	102	1,502.8
7	32,078.8	39	6,089.2	71	3,442.4	103	1,440.4
8	28,303.6	40	5,938.4	72	3,380.0	104	1,378.0
9	25,324.0	41	5,792.8	73	3,317.6	105	1,315.6
10	22,911.2	42	5,657.6	74	3,255.2	106	1,253.2
11	20,919.6	43	5,527.6	75	3,192.8	107	1,185.6
12	19,245.2	44	5,402.8	76	3,130.4	108	1,123.2
13	17,820.4	45	5,283.2	77	3,068.0	109	1,060.8
14	16,593.2	46	5,174.0	78	3,005.6	110	998.4
15	15,522.0	47	5,064.8	79	2,943.2	111	936.0
16	14,580.8	48	4,960.8	80	2,880.8	112	873.6
17	13,748.8	49	4,856.8	81	2,818.4	113	811.2
18	13,005.2	50	4,758.0	82	2,756.0	114	748.8
19	12,334.4	51	4,695.6	83	2,693.6	115	686.4
20	11,736.4	52	4,633.2	84	2,631.2	116	624.0
21	11,190.4	53	4,570.8	85	2,568.8	117	561.6
22	10,691.2	54	4,508.4	86	2,506.4	118	499.2
23	10,238.8	55	4,446.0	87	2,438.8	119	436.8
24	9,817.6	56	4,383.6	88	2,376.4	120	374.4
25	9,432.8	57	4,321.2	89	2,314.0	121	312.0
26	9,079.2	58	4,258.8	90	2,251.6	122	249.6
27	8,746.4	59	4,196.4	91	2,189.2	123	187.2
28	8,439.6	60	4,134.0	92	2,126.8	124	124.8
29	8,153.6	61	4,071.6	93	2,064.4	125	62.4
30	7,888.4	62	4,009.2	94	2,002.0	126	31.2
31	7,633.6	63	3,946.8	95	1,939.6	127	5.2

© 2004-2008 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。