TWL-System NITRO-Composer

Bank Data Manual

2009/07/08

The content of this document is highly confidential and should be handled accordingly.

© 2004-2009 Nintendo
CONFIDENTIAL

TWL-06-0090-001-B
Released: September 11, 2009

Confidential

These coded instructions, statements, and computer programs contain proprietary information of Nintendo and/or its licensed developers and are protected by national and international copyright laws. They may not be disclosed to third parties or copied or duplicated in any form, in whole or in part, without the prior written consent of Nintendo.

Table of Contents

© 2004-2009 Nintendo
CONFIDENTIAL

Code

Tables

Figures

Revision History

Revision Date	Description
2009/07/08	Clarified the amount of memory consumed when individually loading wave data.
2008/05/30	Made revisions in line with the NITRO-System name change (from NITRO-System to TWL-System).
2008/04/08	Changed the format of the Revision History. Changed page headers.
2006/05/29	Corrected errors.
2005/06/06	Added a description of release disable. Added a description of original key setting using a value.
2005/03/28	Added a description of the Key Split and Drum Set Waveform Groups. Added a description of the NULL type instruments. Added a description of numeric value notation.
2005/01/31	Revised a description related to waveform data loop management due to the addition of a feature for individually loading waveform data. Added a description related to the notation of the UNC format in `@PATH`.
2004/08/10	Added explanation of omitting original key.
2004/07/20	Changed the overall structure. Added the grouping feature for waveform data using `@WGROUP`.
2004/06/01	Revised to reflect the ability to specify a label to each instrument. Revised the Decay/Release Table and Maximum Release Time Table to reflect changes in release and decay. Corrected misaligned heading numbers.
2004/04/12	Moved description of bank converter to Sound Tool Manual.
2004/04/01	Added description of waveform format. Created detailed specifications for envelope value.
2004/03/01	Initial version.

1 Overview

Bank data corresponds to the sound source data. It configures multiple waveform data sets into a more meaningful set of instruments. Each instrument is associated with a program number. The bank data receives the program number and note command from the sequencer and generates sounds.

To create the bank data, you use a text editor to create a file that defines the bank data based on the bank format. This file is called the *bank definition file*. In the bank definition file, you define instruments using waveform files.

This manual first describes the waveform data used with NITRO-Composer. Then it describes how to write the bank definition file using the waveform data sets.

© 2004-2009 Nintendo
CONFIDENTIAL

2 Waveform Data

This chapter provides basic information about creating waveform data.

2.1 Waveform Format

2.1.1 File Format

Two waveform file formats can be used: Audio Interchange File Format (AIFF) and Wave (Microsoft).

2.1.2 Number of Channels

There is one channel. Therefore, the sound is MONO.

2.1.3 Quantifying Bit Number

The converter automatically converts the quantifying bit number to the output format so there are no limits. However, if possible, matching the number with the output format is advisable. For example, use 8 bits if outputting to 8-bit PCM, and 16 bits if outputting to ADPCM.

2.1.4 Sampling Rate

The sampling rate is almost unlimited. A realistic sampling rate range is approximately from 4 kHz to 44.1 kHz. A higher sampling rate corresponds to better sound quality. However, a high sampling rate increases the frequency of data transfer and the processing load.

2.1.5 Original Key

The original key is not set in the waveform file but is described in the bank file. The original key will be ignored if the original key is set in the waveform file.

2.1.6 Loop

One type of loop is supported.

2.2 Waveform Editor

You can use any waveform editor application as long as it outputs waveform formats compatible with NITRO-Composer.

2.3 Registering Waveform Data

The waveform data is registered in the bank definition file shown in Chapter 3 Bank Definition File. The waveform data converts automatically when the bank file is converted.

3 Bank Definition File

3.1 Organization

This section explains Code 3-1.

Code 3-1 Bank Definition File

```
@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 Section

The file is organized into three major sections.

- The instrument section, which starts with `@INSTLIST`
- The drum set section, which starts with `@DRUM_SET`
- The key split section, which starts with `@KEY_SPLIT`

You do not need to create a section if you will not use that section.

The following sections discuss each file section in detail.

3.2 Comments

Comments start with a semicolon as shown in Code 3-2.

Code 3-2 Comment Line

```
; Sample
```

A line break specifies the end of a comment.

A comment can be written inline with statement text as long as it has a semicolon preceeding the comment text as shown here.

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

3.3 Labels

In the following explanation, a label refers to a string that indicates a specific item. In the following example, PRG ORGAN is a label.

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

Labels begin with an uppercase Roman letter. After the first letter, uppercase Roman letters, underscores, and numerals can be used. Shown below are examples of correctly named labels.

```
PRG_STRINGS
PRG_PIANO_01
PRG_DRUM_SET
```

Because labels refer to a specific item, the same label name cannot be used for another item.

3.3.1 Program Number List File

When a label corresponding to a program number is defined, the label that replaces the program number is output to the program number list file that has the `.spdl` file extension. By including this file in an include statement in the text sequence file, you can write program changes with labels instead of numbers.

For details, see the section on Preprocessor Directives in the *NITRO-Composer Sound Tools Manual*.

3.4 @PATH

When specifying a waveform file, a relative path from the directory where the bank definition file is located must be specified by default. However, by using `@PATH`, you can change the initial directory.

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

In the above example, `"aif/organ.b4.aiff"` is specified. This statement represents the same statement in the following example.

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

You can also designate a directory with `@PATH` in the following UNC format.

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

3.5 Numeric Value Notation

In the bank definition file where parameters such as envelope are specified, you can use the following notation in addition to entering numbers in decimal notation.

3.5.1 Binary and Hexadecimal Notation

Numeric values are commonly represented as decimals, but can also be represented in binary and hexadecimal notation.

When numeric values are represented in binary or hexadecimal, `0b` or `0x` must precede the numeric value characters, respectively. For example, the decimal notation number 12 would be described as follows.

```
0b1100
0xc
```

3.5.2 Bit Notation

Bit notation is effective when describing a numeric value for the state of a certain bit, such as a bit flag.

Bit notation describes which lower bits are set to 1. For example, when describing a value with the lower bits 1, 3, and 6 to 8 set to 1, write it as follows.

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

This corresponds to `0b111001010`. Note that the LSB is 0.

3.5.3 Mathematical Expressions

Numeric values can also be written as mathematical expressions. Binary, hexadecimal, and bit notations can be used in each term of the mathematical expression.

For example, the following notations are possible.

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

Table 3-1 shows the operators that can be used in an expression and the priority of the operators.

Table 3-1 Operators

Priority	Operator	Meaning
1	*	Multiplication
	/	Division
2	+	Addition
	-	Subtraction
3	>>	Right Shift
	<<	Left Shift
4	<	Left side of the expression is less than the right side
	<=	Left side is less than or equal to the right side
	>	Left side is greater than the right side
	>=	Left side is greater than or equal to the right side
5	==	Left side is equal to the right side
6	&	Bitwise AND
7	|	Bitwise OR

4 Instrument Section

The instrument section uses waveform files to define single instruments. The definitions start with `@INSTLIST`.

Code 4-1 Instrument Section

```
@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 Instruments

The tone for each program number is called an instrument. Executing a program change from the sequence data selects the instrument to use.

There are six main types of instruments.

- PCM type
- PSG type
- Noise type
- Drum Set type
- Key Split type
- NULL type

These instrument types are explained in the following sections.

4.2 PCM Type

The PCM type `t` uses PCM waveform data.

4.2.1 Format

The statement for the PCM type is as follows.

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

The general statement format is as follows.

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

Table 4-1 describes these elements.

Table 4-1 PCM Type Instrument Elements

Element	Description
prgNo	Program number
format	Data format
filename	Filename
originalKey	Original key (can be omitted)
attack	Envelope attack time
decay	Envelope decay time
sustain	Envelope sustain level
release	Envelope release time
pan	Pan (can be omitted)

The following sections describe each element in detail.

4.2.2 Program Number

Specify a program number. The same number cannot be defined twice. The value ranges from 0 to 32767, but only up to 127 can be specified with MIDI.

A label that corresponds to each program number can be defined as shown below.

```
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
```

The first statement specifies both a program number and a label. The program number is `1` and the label is `PRG_ORGAN`.

If only a label name is specified, the program number for this statement will be determined by incrementing the program number of the previous statement by one. Therefore, in the second statement in the above example, the program number is 2.

The third statement is assigned a program number only and the label name is undefined. Because this sequence archive is assigned only a program number, the sequence archive must be referred to using the program number.

4.2.3 Data Format

Specify the data format to convert the PCM waveform data to. Table 4-2 lists the data formats that can be specified.

Table 4-2 PCM Waveform Format

Format	Description
PCM16	Converts to 16-bit PCM data
PCM8	Converts to 8-bit PCM data
ADPCM	Converts to ADPCM data
SWAV	Does not convert

4.2.4 Filename

If the format specification is not SWAV, specify a mono AIFF or WAV file. If it is SWAV, specify the converted SWAV file.

The path is the relative path from the bank definition file. The initial directory can be changed using the `@PATH` command.

4.2.5 Original Key

Specify the original key of the waveform file. By properly setting the pitch of the waveform file, the file plays back at the correct pitch. However, by shifting the original key on purpose, pitch at playback can be adjusted. When omitted, the original key is set to `cn4`.

The original key can be denoted with a value instead of with key notation. A value of 60 indicates `cn4`.

4.2.6 Envelope

Specify the ADSR of the envelope. This value ranges from 0 to 127.

- The larger the value for attack time, the faster the attack.
- The larger the value for decay time, the faster the decay.
- The larger the value for sustain level, the larger the gain. The scale for the numeric value is the same as for the velocity.
- The larger the value for release time, the faster the release.

See section 8.1 Envelope Table for more details on attack, decay, and release.

4.2.6.1 Disabling release

Instead of specifying a value for the release time, you can specify the string `DISABLE`. An instrument with a disabled release will play to the end of the waveform without releasing, even after the specified note length has elapsed.

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

For example, by disabling release on instruments that do not need it, such as percussion instruments, you can play the waveform data until the end without worrying about sound length or note-off.

Note: If release is disabled on looping waveform data, the waveform data will play until the sequence stops.

4.2.7 Pan

Specify the pan value for each instrument. The value ranges from 0 (left) to 64 (center) to 127 (right). If omitted, the default value is 64 (center).

The effects of pan will be added to the settings on the track.

4.3 PSG Type

The PSG type generates sounds using PSG rectangular waves.

4.3.1 Format

The statement is as follows.

```
0 : PSG, DUTY_4_8, cn4, 127,127,127,123
```

The general statement format follows.

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

Table 4-3 describes these elements.

Table 4-3 PSG Type Instrument Elements

Element	Description
prgNo	Program number
duty	Duty ratio
originalKey	Original key (can be omitted)
attack	Envelope attack time
decay	Envelope decay time
sustain	Envelope sustain level
release	Envelope release time
pan	Pan (can be omitted)

The following sections describe each element in detail.

4.3.2 Program Number

Specify the program number in the same way as PCM Type instruments. See section 4.2.2 Program Number.

4.3.3 Duty Ratio

Specify the duty ratio of the PSG rectangular waveform. Table 4-4 lists the values that can be specified.

Table 4-4 PSG Rectangular Waveform Duty Ratio

Label	Description
DUTY_1_8	1/8 (12.5%) Duty
DUTY_2_8	2/8 (25.0%) Duty
DUTY_3_8	3/8 (37.5%) Duty
DUTY_4_8	4/8 (50.0%) Duty
DUTY_5_8	5/8 (62.5%) Duty
DUTY_6_8	6/8 (75.0%) Duty
DUTY_7_8	7/8 (87.5%) Duty

4.3.4 Original Key

Specify the original key. When set to `cn4`, the playback is done at the correct pitch. However, if you shift the original key, the pitch at playback is shifted. When omitted, the original key is set to `cn4`.

The original key can be denoted with a value instead of key notation. A value of 60 maps to `cn4`.

4.3.5 Envelope

Specify the envelop ADSR using the same method as PCM type instruments. See section 4.2.6 Envelope.

4.3.6 Pan

Specify the pan value for each instrument. The value ranges from 0 (left) to 64 (center) to 127 (right). When omitted, the value is set to 64 (center).

The effect of pan is combined with the track settings.

4.4 Noise Type

The Noise type generates sound using white noise.

4.4.1 Format

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

The statement is as shown below.

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

Table 4-5 describes these elements.

Table 4-5 Noise Type Instrument Elements

Element	Description
prgNo	Program number
originalKey	Original key (can be omitted)
attack	Envelope attack time
decay	Envelope decay time
sustain	Envelope sustain level
release	Envelope release time
pan	Pan (can be omitted)

The following sections describe each element in detail.

4.4.2 Program Number

Specify the program number using the same method used for PCM Type instruments. See section 4.2.2 Program Number.

4.4.3 Original Key

Specify the original key. When set to cn4, playback is at the correct pitch. However, if you shift the original key, the pitch at playback is shifted. When omitted, the original key is set to cn4.

The original key can be denoted with a value instead of key notation. A value of 60 maps to cn4.

4.4.4 Envelope

Specify the ADSR of the envelope using the same method used for PCM type instruments. See section 4.2.6 Envelope.

4.4.5 Pan

Specify the pan value for each instrument. The value ranges from 0 (left) to 64 (center) to 127 (right). When omitted, the default is 64 (center) is set.

The effect of the pan is combined with the track settings.

4.5 Drum Set Type

The Drum Set type generates sounds using drum sets. For details on drum sets, see Chapter 5 Drum Set.

4.5.1 Format

A drum set type instrument is specified as follows.

```
127 : DRUM_SET,   _DRUM_SET
```

The general statement format is shown below.

```
prgNo : DRUM_SET, label
```

Table 4-6 describes these elements.

Table 4-6 Drum Set Type Instrument Elements

Element	Description
prgNo	Program number
label	Key split label

The following sections describe each element in detail.

4.5.2 Program Number

Specify the program number using the same method used for PCM Type instruments. See section 4.2.2 Program Number.

4.5.3 Drum Set Label

Specify by label the drum set defined in the drum set section. For details, see section 5.3 Drum Set Label.

4.6 Key Split Type

The key split type generates sounds using key split. For details on key split, see Chapter 6 Key Split Section.

4.6.1 Format

The statement for a key split type instrument is as follows.

```
0 : KEY_SPLIT,   _INST_PIANO
```

The general statement format is shown below.

```
 prgNo : KEY_SPLIT, label
```

Table 4-7 describes these elements.

Table 4-7 Key Split Type Instrument Elements

Element	Description
prgNo	Program number
label	Drum set label

The following sections describe each element in detail.

4.6.2 Program Number

Specify the program number using the same method used in PCM Type instruments. See section 4.2.2 Program Number.

4.6.3 Key Split Label

Specify by label the key split defined in the key split section. For details, see section 6.3 Key Split Label.

4.7 NULL Type

The NULL type does not generate any sounds. It can be combined with key split to create an instrument that does not generate sounds on a certain split.

4.7.1 Format

A NULL type instrument is specified as follows.

```
0 : NULL
```

The general format statement is shown below.

```
prgNo : NULL
```

Table 4-8 describes this element.

Table 4-8 NULL Type Instrument Elements

Element	Description
prgNo	Program number

The following sections describe each element in detail.

4.7.2 Program Number

Specify the program number using the same method used for PCM Type instruments. See section 4.2.2 Program Number.

5 Drum Set

The drum set section defines the drum set. The section starts with @DRUM_SET.

Code 5-1 Drum Set Section

```
@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 Drum Set

A different instrument is assigned to each key. The pitch cannot be changed, but different instrument sounds can be generated without program changes. Use the drum set mainly for percussion instruments, where changes in pitch are unnecessary. A slight change in pitch is possible, but depends on the usage and is explained in detail in a subsequent section.

5.2 Format

The following statements correspond to a single drum set definition.

```
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
```

The format is specified as follows.

```
drumsetLabel =

   key : inst-definition

   key : inst-definition

   key : inst-definition
```

Table 5-1 describes these elements.

Table 5-1 Drum Set Definition Elements

Elements	Description
drumsetLabel	Drum set label
key	Key
inst-definition	Instrument definition

First, the drum set label is defined. In the next line, the instrument is assigned for each key.

The following sections describe each element in detail.

5.3 Drum Set Label

Specify the label name for the drum set to be defined. The label name specifies the instrument type of the drum set.

5.4 Key

Specify which key to assign the instrument. The same key cannot be defined twice.

5.5 Instrument Definition

Define an instrument for each key. The format for the instrument definition follows the format used for the instrument section. However, the key split or drum set type instrument cannot be specified.

If the original key specification is omitted, the key becomes the same as the assigned key, rather than cn4. In other words, for a PCM type instrument, the waveform data plays back at the waveform data pitch. For PSG and noise type instruments, the pitch is middle C.

5.6 Automatic Assignment to Undefined Keys

Undefined keys take the setting of the lowest key in the defined set.

In the following example, because keys cs2 through ds2 are undefined, these keys are set to cn2.

```
_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
```

The original key remains cn2, so the pitch for cn2 through ds2 can be changed.

Keys below the lowest key (cn2) and keys above the highest key (fn4) will play no sound.

6 Key Split Section

The key split section is a section that defines key splits. The section starts with @KEY_SPLIT.

Code 6-1 Key Split Section

```
@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 Key Split

Key split allows the assignment of a single waveform to each key range. Use key split if the range from a low to high sound cannot be included in one waveform.

Here, the key range is called Split. In other words, a waveform file is assigned to each split. With one key split, you can create up to eight splits.

6.2 Format

The following statements are an example of a key split definition.

```
_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
```

The general statement format is shown below.

```
keysplitLabel =

   key : inst-definition

   key : inst-definition

   key : inst-definition
```

Table 6-1 describes these elements.

Table 6-1 Key Split Definition Elements

Elements	Description
keysplitLabel	Key split label
Key	Key
inst-definition	Instrument definition

The key split label is defined, and in the following lines the split is defined.

The following sections describe each element in detail.

6.3 Key Split Label

This specifies the label name of the key split. The label name specified here is used when defining the key split type instrument.

6.4 Key

Specify the limit of the highest key for a split. Specify the highest key as 127 for the widest split range.

When the key split is defined as the following:

```
_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
```

the key range of each split is shown in Table 6-2.

Table 6-2 Key Split Range

Key Range			Waveform File
	～	gn2	hhcl.fs2.aiff
gs2	～	an2	Hhpd.gs2.aiff
as2	～		Hhop.as2.aiff

6.5 Instrument Definitions

Define an instrument for each split. The format of the instrument definition follows the instrument section format. However, the key split and drum set type instruments cannot be specified.

When the original key specification is omitted, the original key becomes the same as the lowest key in the key range instead of `cn4`.

7 Group Management of Waveform Data

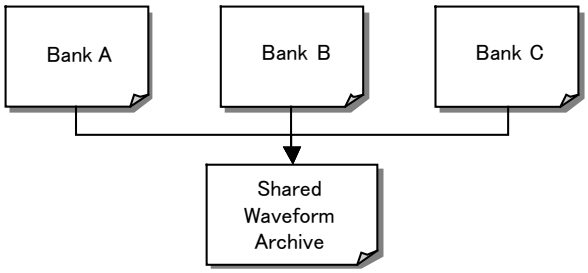
This chapter explains group management of waveform data, which permits more efficient use of waveform data memory.

7.1 Introduction

Consider a case where the same waveform data is used in multiple banks. Allowing one waveform data set to be used by multiple banks, would require less memory.

A method to share waveform data is to create a single waveform archive that multiple banks can use. You can accomplish this by writing the sound archive definition file.

Figure 7-1 Shared Waveform Archive



In addition, if the option flag in the sound archive definition is used to specify that each waveform data load one at a time, only necessary waveform data is extracted and loaded from the shared waveform archive. For example, waveform data in the shared waveform archive used only by Bank A does not need to be loaded when using Bank B.

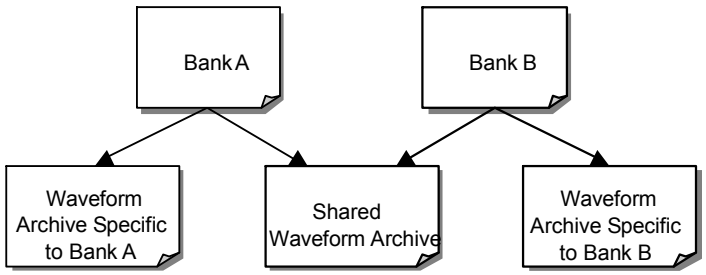
In most cases, this method will work. However, if waveform data is being loaded individually, memory overhead and loading time may become a problem. (Specifically, 32 bytes of memory are consumed for management purposes for every four wave files.) If grouping the banks that use the same waveform data is simple, group management of waveform data is more efficient than individually loading waveform data.

7.2 Grouping

To simplify the example, consider only two banks, Bank A and Bank B.

In this case, the waveform data that Bank A uses is waveform data exclusively used by Bank A and the waveform data shared with Bank B. This waveform data is divided into two waveform archives.

Figure 7-2 Waveform Data Grouping



By dividing the waveform archive into two, the minimum required waveform data can be loaded without individually loading waveform data sets.

Implementing these groups requires that the waveform data used in one bank must be grouped based on whether the waveform data is shared or exclusive. The following instructions explain grouping.

7.3 @WGROUP

`@WGROUP` is used to group waveform data in a bank.

```
@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
```

In the above example, `organ.b4.aiff` is set to Group 1, and the files are set to Group 0.

After a group number is specified with `@WGROUP`, all following waveform data sets are classified under that group number. If there is no specification, the waveform data is classified as Group 0. The group number can be specified as 0, 1, 2, or 3. The data can only be divided into a maximum of four groups.

7.3.1 Key Split and Drum Set Waveform Groups

The grouping of waveform data used in key split and drum is specified in units of waveform data set, and not in units of key split or drum set. Therefore, the group number used to define the instrument in `KEY_SPLIT` or `DRUM_SET` does not carry a meaning. See the example below.

```
@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
```

In this example, waveform data is grouped in Groups 2 and 3. No waveform data is grouped in Group 1.

7.4 Creating Associations with Multiple Waveform Archives

After dividing the data sets into groups, the waveform archive must be associated with each group. These specifications are made in the sound archive definition file.

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

In the above statement, the Group 0 waveform data is stored in waveform archive `WAVE_A`, and the Group 1 waveform data is stored in waveform archive `WAVE_COMMON`.

For example, if Bank B also uses `organ.b4.aiff` and Bank B is put into Group 1:

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

only `organ.b4.aiff` can be shared.

7.5 Loading a Waveform Archive

To use banks associated with multiple waveform archives, all archives associated with the bank must be loaded. However, you can load specific waveform archives in advance.

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

In the case above, if WAVE_COMMON is loaded in advance, only loading WAVE_A to use BANK_A is required. For BANK_B, only WAVE_B needs to be loaded.

8 Appendix

8.1 Envelope Tables

8.1.1 Attack Table

Table 8-1 shows the relationship between the attack value and the attack time (in milliseconds).

Table 8-1 Attack Table

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

Some entries have the same attack time even though the attack value is different. This difference means that, although the time to reach the peak level is the same, how the attack changes to this point differs slightly. (The larger the value, the faster the attack.)

8.1.2 Decay/Release Table

Table 8-2 shows the relationship between the decay or release values and the release speed decay. The release speed decay is measured in dB/msec.

Table 8-2 Decay/Release Table

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

© 2004-2009 Nintendo
CONFIDENTIAL

For decay, the sound continues to decay until it reaches the gain specified with sustain. For release, the sound generation stops if the gain that includes the influence of velocity and other factors becomes -72.3 dB or lower.

8.1.3 Maximum Release Time Table

The release is regulated by the release speed, but this concept may not be intuitive. Table 8-3 shows the relationship of release speed to release time (time until sound generation completely stops). Note, for example, that the volume parameters for velocity must be at a maximum and the actual time shorter to meet the conditions for this relationship to occur. Release time is measured in milliseconds.

Table 8-3 Maximum Release Time Table

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

Release	Time	Release	Time	Release	Time	Release	Time
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

© 2004-2009 Nintendo
CONFIDENTIAL

All company and product names in this document are the trademarks or registered trademarks of their respective companies.

© 2004-2009 Nintendo

The contents of this document cannot be duplicated, copied, reprinted, transferred, distributed, or loaned in whole or in part without the prior approval of Nintendo.