

PHD/PBD Formats

© 2014 Sony Computer Entertainment Inc.
All Rights Reserved.
SCE Confidential

Table of Contents

1 Overview of PlayStation®Vita Sound Data.....	3
PHD / PBD and PEF Data.....	3
VAG Data.....	3
Basic Types of Sound Data	3
2 PHD Data Format	4
PHD Overall Structure	4
Basic Structure of an Attribute.....	5
Common Attribute.....	5
3 libsndp PHD Data	6
libsndp PHD Data Structure	6
Common Attribute.....	6
VAG Attribute	7
VAG Parameter	8
Tone Attribute	9
Tone Parameter.....	10
Program Attribute	11
Program Parameter.....	12
4 PBD Data Format	13
PBD Structure.....	13
5 PEF Data Format.....	14
PEF Structure	14
Effect Parameters.....	14
PEF Data for libsndp	14

1 Overview of PlayStation®Vita Sound Data

PHD / PBD and PEF Data

PHD / PBD is phoneme data used by the high-level sound library (libsndp).

PHD, which is also referred to as the header, contains data that links together attributes for each phoneme. PBD, which is also referred to as the body, contains phoneme data that links together individual sections of ADPCM (VAG) waveform data.

Together the PHD and PBD form a data structure that can be used to obtain the desired waveform data from the program number and note number, to be produced by the sound source during MIDI playback.

PEF, which is effect parameter data, is handled separately from PHD and PBD data.

A PHD data file has the extension ".phd", a PBD data file has the extension ".pbd", and a PEF data file has the extension ".pef".

VAG Data

VAG data is waveform data compressed using ADPCM and is the basis for PBD data. It consists of the waveform data and 48 bytes of information about the waveform data such as the sampling frequency.

A VAG data file has a ".vag" extension. For details, refer to the "VAG Converter 2 Tools User's Guide" document.

Basic Types of Sound Data

PHD / PBD and PEF data consists of the basic types shown below. The byte order is Little Endian.

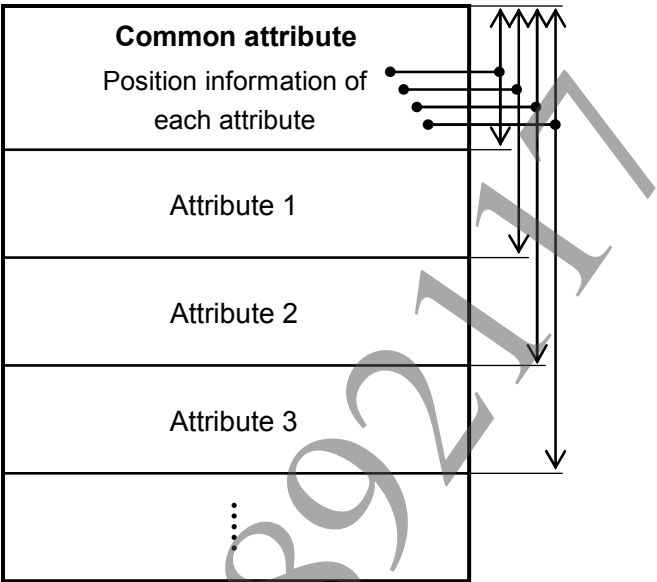
Type Name	Description
int	Signed 32-bit integer
unsigned char	Unsigned 8-bit integer
unsigned int	Unsigned 32-bit integer

2 PHD Data Format

PHD Overall Structure

PHD consists of a block structure in which attributes are linked, where an attribute is a single unit.

Figure 1 PHD Overall Structure



The "common attribute" must be placed at the beginning of the PHD. The common attribute is used to manage the entire PHD and includes the position of each attribute (offset from the beginning of the PHD) and version information.

Besides the common attribute, three types of attributes that are used by libsndp are defined, and attributes having a prescribed header and arbitrary data contents can also be defined.

PHD is an extremely flexible format since the block structure enables you to create individual data that can be easily maintained.

Basic Structure of an Attribute

An attribute has the following basic structure.

Field	Type	Description
ID	unsigned char[4]	Identification ID
attrSize	unsigned int	Attribute size (number of bytes in context)
data	Arbitrary	Context (arbitrary data)

The following identification IDs are reserved.

Identification ID	Use
'PPHD'	Common attribute
'PPVA'	For libsndp use (VAG attribute)
'PPTN'	For libsndp use (tone attribute)
'PPPG'	For libsndp use (program attribute)

Note that the attribute size is the number of bytes in the context part excluding the identification ID and attribute size, not the size of the entire attribute.

Common Attribute

The common attribute has the following structure.

Field	Type	Description
ID	unsigned char[4]	Identification ID ('PPHD')
attrSize	unsigned int	Attribute size
version	unsigned int	Version (High-order 16 bytes: Major version) (Low-order 16 bytes: Minor version)
attr_offset[0]	unsigned int	Offset to attribute 1 (Number of bytes from beginning of PHD)
attr_offset[1]	unsigned int	Offset to attribute 2 (Number of bytes from beginning of PHD)
...	...	(Continuation of offsets to subsequent attributes)

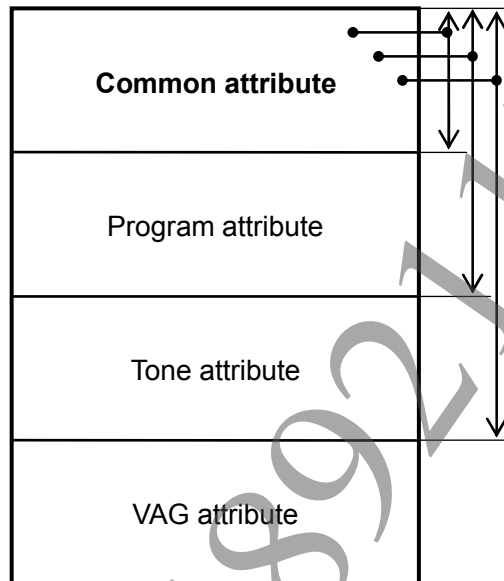
In PHD data that is to be used by libsndp, the attr_offset[0] to attr_offset[2] fields are reserved as position information for the program attribute, tone attribute, and VAG attribute, respectively.

3 libsndp PHD Data

libsndp PHD Data Structure

PHD data that is to be used by libsndp has a common attribute, program attribute, tone attribute, and VAG attribute as shown below.

Figure 2 PHD Structure Used by libsndp



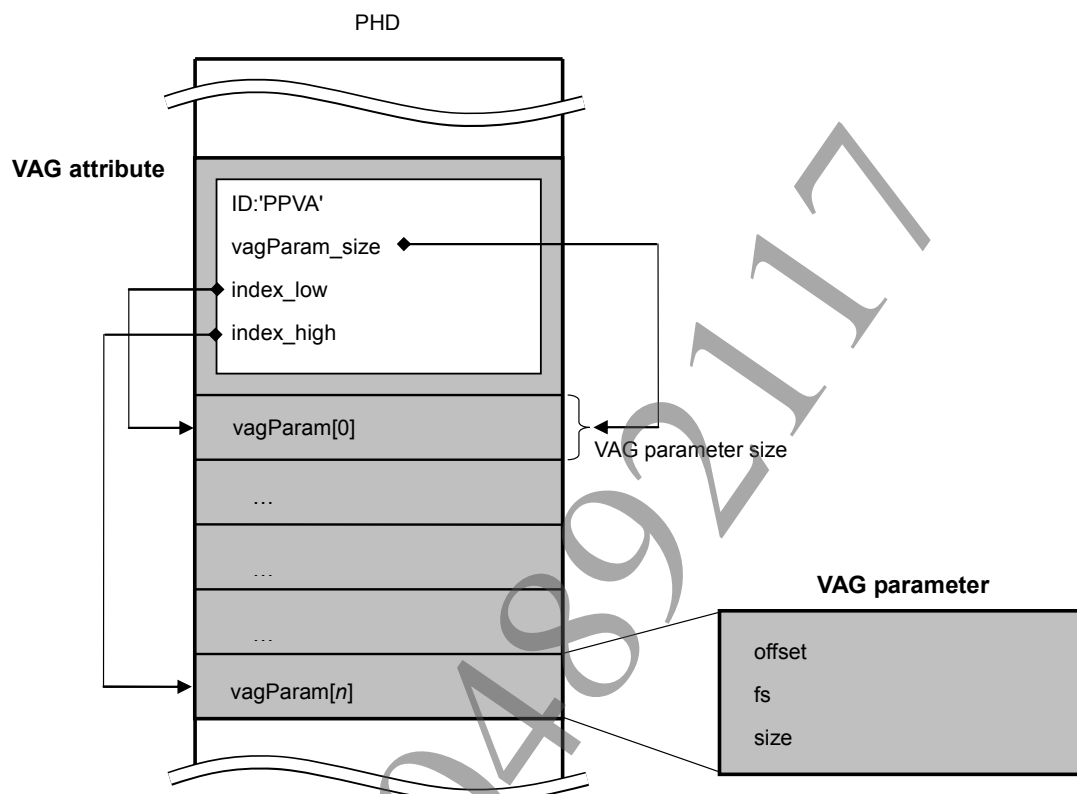
Common Attribute

Field	Type	Description
ID	unsigned char[4]	Identification ID ('PPHD')
attrSize	unsigned int	Attribute size (56)
version	unsigned int	Version (0x00010000)
reserve1	unsigned int	(Reserved area)
attr_offset[0]	unsigned int	Offset to program attribute (Number of bytes from beginning of PHD)
attr_offset[1]	unsigned int	Offset to tone attribute (Number of bytes from beginning of PHD)
attr_offset[2]	unsigned int	Offset to VAG attribute (Number of bytes from beginning of PHD)
reserve2[9]	unsigned int	(Arbitrary data)

VAG Attribute

The VAG attribute maintains basic information (VAG parameters) about each of the individual waveform data that are linked and maintained in the PBD. The VAG attribute is a structure that consists of a header followed by an array of VAG parameters as shown below.

Figure 3 VAG Attribute Structure



Field	Type	Description
ID	unsigned char[4]	'PPVA'
attrSize	unsigned int	Attribute size (Total number of bytes - 8)
vagParam_size	unsigned int	VAG parameter size sizeof(vagParamCtx)
reserve1	unsigned int	(Reserved area)
index_low	unsigned int	VAG parameter starting index
index_high	unsigned int	VAG parameter ending index
reserve2[2]	unsigned int	(Reserved area)
vagParam[0]	vagParamCtx	VAG parameter[index_low]
...
vagParam[index_high - index_low + 1]	vagParamCtx	VAG parameter[index_high]

vagParam is an array having the corresponding number of elements from the starting index to the ending index. Each element is a VAG parameter (vagParamCtx structure defined in sdphd.h). If there are unused indices among the indices from the starting index to the ending index, the elements corresponding to those unused indices are padded with 0xffffffff.

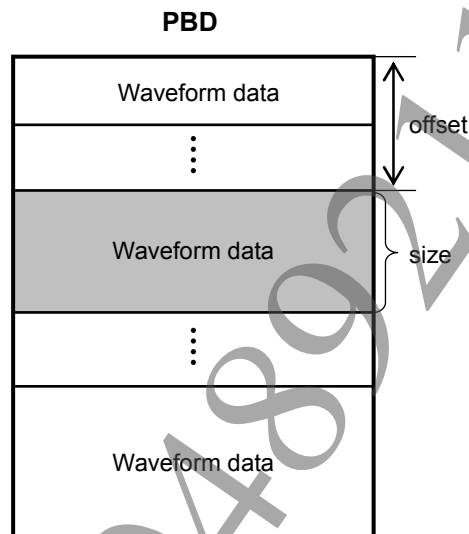
VAG Parameter

A VAG parameter, which corresponds to a single waveform data within the PBD, has the following structure.

Field	Type	Description
offset	unsigned int	Offset of waveform data from the beginning of the PBD
fs	unsigned int	Sampling frequency (Hz)
size	unsigned int	Waveform size (bytes)
reserve	unsigned int	(Reserved area)

The offset and size fields of a VAG parameter denote waveform data within the PBD as follows.

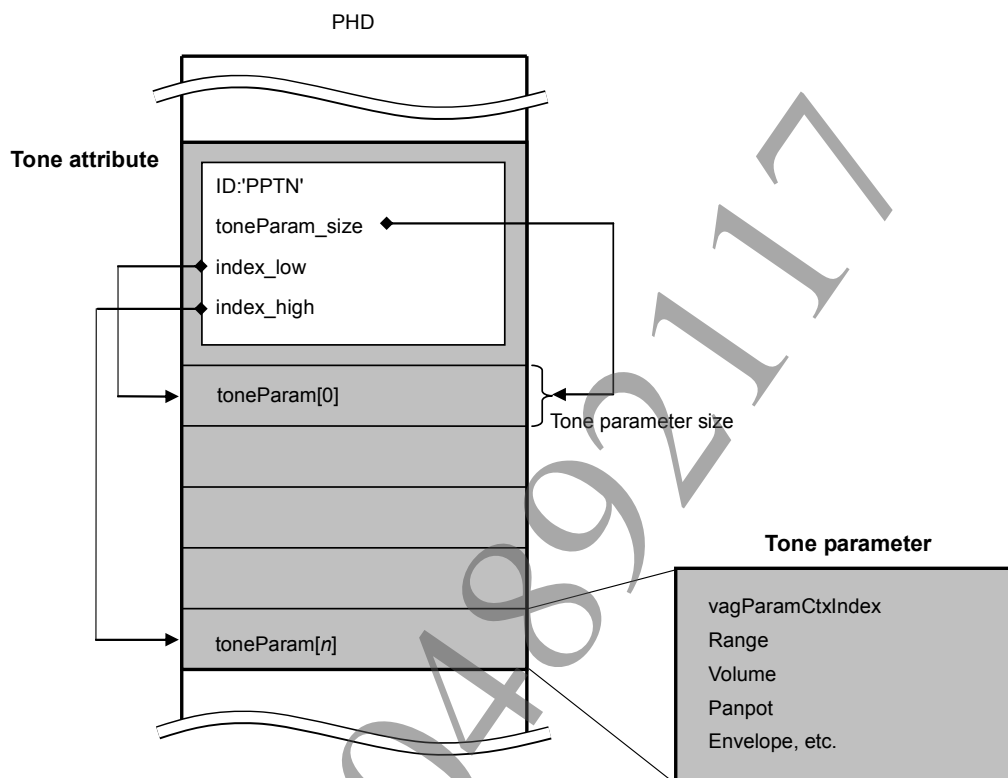
Figure 4 Specification of Waveform Data Within the PBD According to the VAG Parameter



Tone Attribute

The tone attribute consists of parameters such as volume, panpot, and envelope, which are used when producing sound for the waveform data, and parameters such as voice range and bend range that are required for MIDI playback. The tone attribute is a structure that consists of a header followed by an array of tone parameters as shown below.

Figure 5 Tone Attribute Structure



Field	Type	Description
ID	unsigned char[4]	'PPTN'
attrSize	unsigned int	Attribute size (Total number of bytes -8)
toneParam_size	unsigned int	Tone parameter size sizeof(toneParamCtx)
reserve1	unsigned int	(Reserved area)
index_low	unsigned int	toneParam starting index
index_high	unsigned int	toneParam ending index
reserve2[2]	unsigned int	(Reserved area)
toneParam[0]	toneParamCtx	Tone parameter[index_low]
...
toneParam[index_high - index_low + 1]	toneParamCtx	Tone parameter[index_high]

toneParam is an array having the corresponding number of elements from the starting index to the ending index. Each element is a tone parameter (toneParamCtx structure defined in sdphd.h). If there are unused indices among the indices from the starting index to the ending index, the elements corresponding to those unused indices are padded with 0xffffffff.

Tone Parameter

A tone parameter has an index that denotes a single VAG parameter as well as parameters that are used when producing sound.

Field	Type	Description	Value Range
vagParamCtxIndex	unsigned int	Index of VAG parameter used by this tone parameter (for a noise sound source, this is 0xffffffff)	
priority	unsigned int	Priority	0 (low) to 127 (high)
group	unsigned int	Group	0 to 127
noiseClock	unsigned int	Noise clock	0 to 60
note_low	unsigned int	Lowest sound producing note number	0 to 127
note_high	unsigned int	Highest sound producing note number	0 to 127
reserve1[2]	unsigned int	(Reserved area)	
volume	int	Volume	0 to 4096
panpot	unsigned int	Panpot	0 (left) to 64 to 127 (right)
sendVolume	int	Effect send volume	0 to 4096
sendPanpot	unsigned int	Effect send panpot	0 (left) to 64 to 127 (right)
centerNote	unsigned int	Note number when sampling	0 to 127
centerDetune	int	Detune when sampling	-99 to 99 (cents)
detune	int	Detune	-99 to 99 (cents)
reserve2	unsigned int	(Reserved area)	
bendRange_lower	unsigned int	Bend range-	0 to 24 (number of notes)
bendRange_upper	unsigned int	Bend range+	0 to 24 (number of notes)
portamentTime	unsigned int	Portament transition time	0
reserve3	unsigned int	(Reserved area)	
adsr1	unsigned int	ADSR1 (envelope)	
adsr2	unsigned int	ADSR2 (envelope)	
reserve4[2]	unsigned int	(Reserved area)	

vagParamCtxIndex is the index of the VAG parameter that is used by this tone parameter. In other words, it indicates the waveform data that is used by this tone parameter. When vagParamCtxIndex is TONE_NOISE (= 0xffffffff), it indicates that this tone parameter uses a noise sound source, and the value set for noiseClock becomes valid.

priority is the priority of the produced sound, and sound is produced by giving precedence to tone parameters having a higher value.

group is a function for performing exclusive sound production for sounds within a given group. When the group value is non-zero, the libsndp voice management system controls sound production so that sound will only be produced for a single tone among the tones that have that group value. For example, if there are two tones such as an open high hat tone and a closed high hat tone, but you only want to produce sound for one or the other, you can set both tones to have the same group value and sound will be produced for these tones exclusively (that is, only one or the other). If group is 0, the group function will be off.

note_low and note_high indicate the range. In other words, they indicate the lower and upper bounds of the note numbers for which tone sound production for this tone parameter is valid.

sendVolume and sendPanpot are the input volume to and the panpot for the digital reverb.

centerNote and centerDetune are the note number and detune used when sampling the waveform that is used by this tone parameter. Pitch information that is given to the sound source is calculated from these values and the note number that is specified when sound is produced.

detune is used to adjust the musical interval when sound is produced. Note that the detune units are cents, and a semitone is 100 cents.

bendRange_lower and bendRange_upper indicate the maximum width of the musical interval change for bend wheel or bend lever operations.

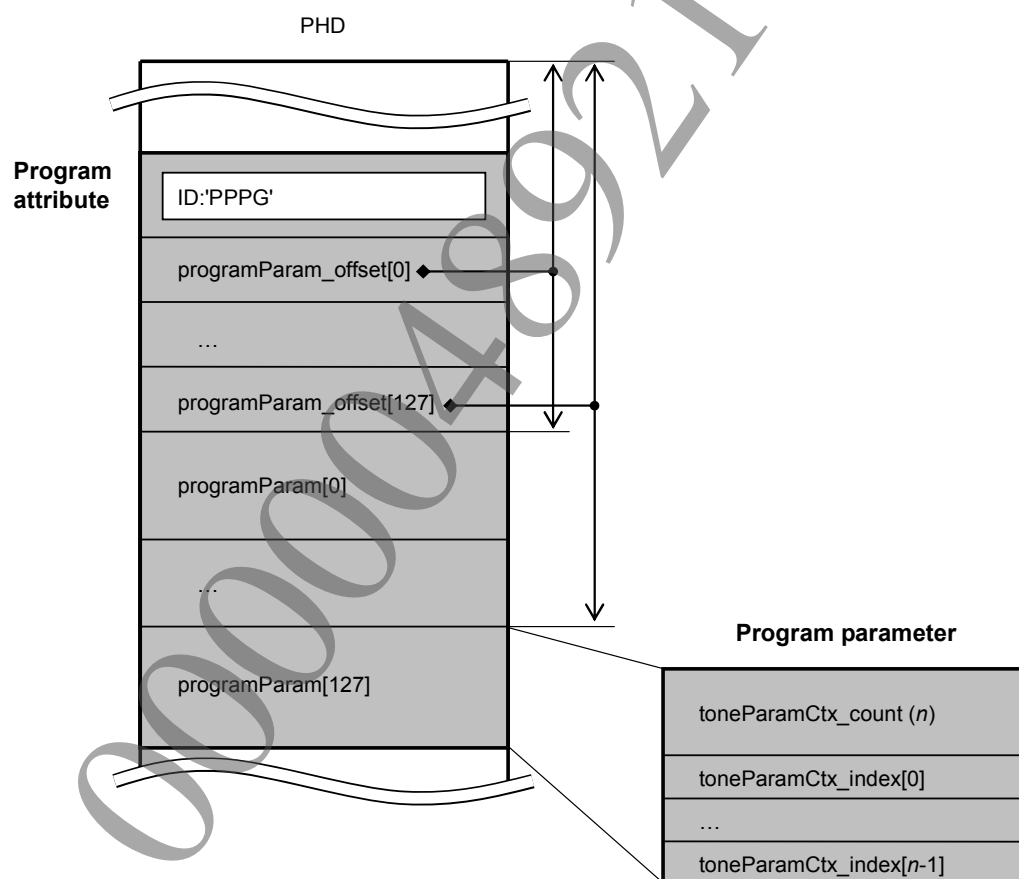
portamentTime is reserved for a future version upgrade. Currently, be sure that 0 is set.

adrs1 and adrs2 indicate the envelope. For details, refer to the "SAS Reference" document.

Program Attribute

The program attribute is data for associating a MIDI program number with a tone parameter. It is a structure consisting of a header, a table of offsets to program parameters, and program parameters of indefinite size.

Figure 6 Program Attribute Structure



Field	Type	Description
ID	unsigned char[4]	'PPPG'
attrSize	unsigned int	Attribute size (Total number of bytes -8)
reserve1[6]	unsigned int	(Reserved area)
programParam_offset[0]	unsigned int	Offset to program parameter of program number 0 (number of bytes from beginning of PHD)
...

Field	Type	Description
programParam_offset[127]	unsigned int	Offset to program parameter of program number 127 (number of bytes from beginning of PHD)
programParams		Program parameters (indefinite size, indefinite number, indefinite order)

programParam_offset has 128 elements. However, for an unused program number, the value of the element is 0xffffffff.

Program Parameter

A program parameter is data indicating the tone parameters that are used by an individual program.

Field	Type	Description
toneParamCtx_count	unsigned int	Number of tone parameters that are used by this program (n)
reserve[3]	unsigned int	(Reserved area)
toneParamCtxIndex[0]	unsigned int	Tone parameter index
...
toneParamCtxIndex[n-1]	unsigned int	Tone parameter index

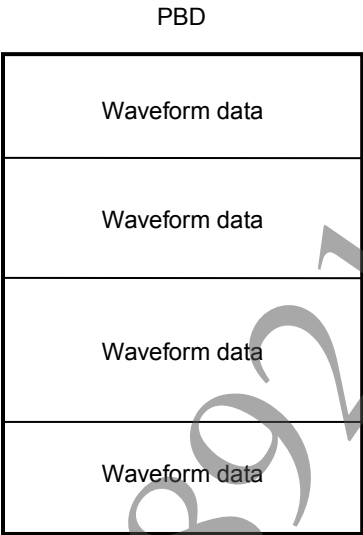
toneParamCtx_count is the number of tone parameters that are used by the program. The number of tone parameter indices that are used, which follow this entry, is equal to the value specified for this count.

4 PBD Data Format

PBD Structure

PBD is a file in which only the waveform parts within the VAG are linked together. It has a simple structure as follows.

Figure 7 PBD Structure

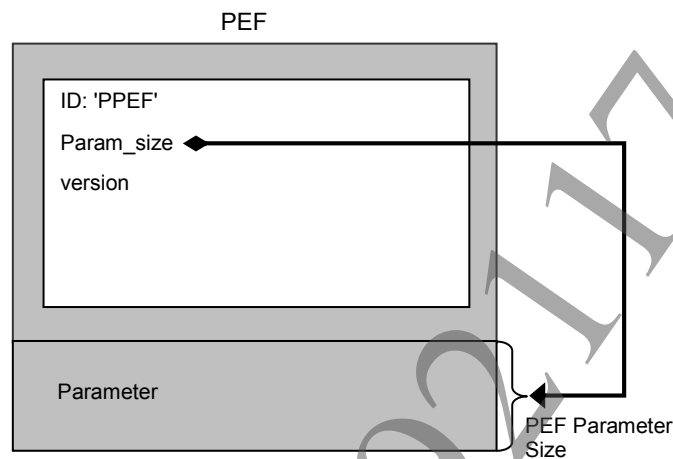


5 PEF Data Format

PEF Structure

PEF is effect parameter data. PEF, which is handled separately from PHD and PBD data, has the following structure.

Figure 8 Effect Structure



Effect Parameters

Field	Type	Description
ID	unsigned char[4]	'PPEF'
Param_size	unsigned int	PEF parameter size (sizeof(SceSsEffectParam))
version	Unsigned int	Data structure version (the high-order 16 bits and low-order 16 bits represent the major and minor versions, respectively)
Parameter	SceSsEffectParam	Parameters

PEF Data for libsndp

Parameter, which is of type `SceSsEffectParam`, is a structure for effect settings.

To pass PEF data as an argument of the libsndp function `sceSsSetEffectParam()`, extract and pass only the Parameter part from the PEF structure.