NGS Modules Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

Table of Contents

About DSP Effects

About DSP Effects

Introduction

This document describes the DSP functions, which are supplied with NGS.

Using DSP Effects

There are 2 methods within the API to modify parameters, they are outlined below:

Method 1: Lock / Unlock Parameters

The procedure for using the DSP effects follows a common pattern for each effect. Unless otherwise specified, the functions referred to here are defined in the *NGS Reference* manual:

- Call `sceNgsVoiceLockParams` to access a DSP effect Modules parameters within a voice. These parameters are stored within a structure type which is specific to the specific DSP module. During this "lock" period, NGS will continue to process audio data using the parameters stored previous to the lock command being issued. See point 3.
- User can now modify any parameter within the module parameter structure.
- Call `sceNgsVoiceUnlockParams` to allow any changes to the previously locked DSP effect Module to be processed.

Method 2: Block Parameter Setting

This method is supplied as a more efficient way of data-driving parameter settings. For example, from loaded parameter sets that have been developed offline. It also allows the user to setup multiple modules within a voice from a single command. Effectively, these structures are the same as those used in the lock/unlock method, but also contain an extra header with information on the module identifier and audio channel.

- Create / load your `ParamsBlock` structure(s). To setup multiple modules each data block should be contiguous within the same memory block.
- Call `sceNgsVoiceSetParamsBlock`.

Reference Materials

Use the following materials for information:

- *NGS Overview*
- *NGS Modules Overview*
- *NGS Reference*

Defines

Defines Table

Define	Value	Description
SCE_NGS_AT9_ID	(0x5CAA)	Specifies the ATRAC9™ Player ID.
SCE_NGS_COMPRESSOR_ID	(0x5CE1)	Specifies the Compressor ID.
SCE_NGS_DELAY_ID	(0x5CEB)	Specifies the Delay ID.
SCE_NGS_DISTORTION_ID	(0x5CE2)	Specifies the Distortion ID.
SCE_NGS_ENVELOPE_ID	(0x5CE3)	Specifies the Envelope ID.
SCE_NGS_FILTER_ID	(0x5CE4)	Specifies the Filter ID.
SCE_NGS_INPUT_MIXER_ID	(0x5CE0)	Specifies the Input Mixer ID.
SCE_NGS_MIXER_ID	(0x5CE9)	Specifies the Mixer ID.
SCE_NGS_OUTPUT_ID	(0x5CED)	Specifies the Output ID. The output module returns a user state block of PCM samples. The size of the block represents the number of channels (output's voice) * system granularity * sizeof(short).
SCE_NGS_PARAM_EQ_ID	(0x5CEC)	Specifies the Parametric EQ ID.
SCE_NGS_PAUSER_ID	(0x5CE5)	Specifies the Pauser ID.
SCE_NGS_PLAYER_ID	(0x5CE6)	Specifies the Player ID.
SCE_NGS_PITCHSHIFT_ID	(0x5CEA)	Specifies the Pitch Shift ID.
SCE_NGS_REVERB_ID	(0x5CE7)	Specifies the Reverb ID.
SCE_NGS_GENERATOR_ID	(0x5CE8)	Specifies the Signal Generator ID.

Parameter Structure Defines

Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_AT9_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_AT9_ID,1,1))	Specifies the ID of the ATRAC9™ Player's parameter structure. The ID of the ATRAC9™ Player's parameter structure is specified using the module identifier *SCE_NGS_AT9_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CAA).
SCE_NGS_COMPRESSOR_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_COMPRESSOR_ID,1,1))	Specifies the ID of the Compressor's parameter structure. The ID of the Compressor's parameter structure is specified using the module identifier *SCE_NGS_COMPRESSOR_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE1).
SCE_NGS_COMPRESSOR_PARAMS_STRUCT_ID_V2	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_COMPRESSOR_ID,1,2))	Specifies the ID of the Compressor's parameter structure. The ID is specified using the module identifier *SCE_NGS_COMPRESSOR_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01025CE1). Note that this uses the same parameter structure as SCE_NGS_COMPRESSOR_PARAMS_STRUCT_ID, though the behavior is modified (see the *NGS Modules Overview* and SceNgsCompressorParams structure).
SCE_NGS_DELAY_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_DELAY_ID,1,1))	Specifies the ID of the Delay's parameter structure. The ID of the Delay's parameter structure is specified using the module identifier *SCE_NGS_DELAY_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CEB).
SCE_NGS_DISTORTION_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_DISTORTION_ID,1, 1))	Specifies the ID of the Distortion's parameter structure. The ID of the Distortion's parameter structure is specified using the module identifier *SCE_NGS_DISTORTION_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE2).
SCE_NGS_ENVELOPE_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_ENVELOPE_ID,1, 1))	Specifies the ID of the Envelope's parameter structure. The ID of the Envelope's parameter structure is specified using the module identifier *SCE_NGS_ENVELOPE_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE3).

Define	Value	Description
SCE_NGS_FILTER_ PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_FILTER_ID,1,1))	Specifies the ID of the Filter's parameter structure. The ID of the Filter's parameter structure is specified using the module identifier *SCE_NGS_FILTER_ID*, index 1 and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE4).
SCE_NGS_FILTER_ PARAMS_COEFF_ STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_FILTER_ID,2,1))	Specifies the ID of the Filter's parameter coefficient structure. The ID of the Filter's parameter coefficient structure is specified using the module identifier *SCE_NGS_FILTER_ID*, index 2 and the SCE_NGS_MAKE_PARAMS_ID macro (0x02015CE4).
SCE_NGS_GENERATOR_ PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_GENERATOR_ID, 1,1))	Specifies the ID of the Signal Generator's parameter structure. The ID of the Signal Generator's parameter structure is specified using the module identifier *SCE_NGS_GENERATOR_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE8).
SCE_NGS_MIXER_ PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_MIXER_ID,1,1))	Specifies the ID of the Mixer's parameter structure. The ID of the Mixer's parameter structure is specified using the module identifier *SCE_NGS_MIXER_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE9).
SCE_NGS_PARAM_ EQ_COEFF_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_PARAM_EQ_ID, 2,1))	Specifies the ID of the Parametric EQ's parameter coefficient structure. The ID of the Parametric EQ's parameter coefficient structure is specified using the module identifier *SCE_NGS_PARAM_EQ_ID*, index 2 and the SCE_NGS_MAKE_PARAMS_ID macro (0x02015CEC).
SCE_NGS_PARAM_EQ_ STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_PARAM_EQ_ID, 1,1))	Specifies the ID of the Parametric EQ's parameter structure. The ID of the Parametric EQ's parameter structure is specified using the module identifier *SCE_NGS_PARAM_EQ_ID*, index 1 and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CEC).
SCE_NGS_PAUSER_ PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_PAUSER_ID,1,1))	Specifies the ID of the Pauser's parameter structure. The ID of the Pauser's parameter structure is specified using the module identifier *SCE_NGS_PAUSER_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE5).

Define	Value	Description
SCE_NGS_PITCHSHIFT_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_PITCHSHIFT_ID,1,1))	Specifies the ID of the Pitch Shift's parameter structure. The ID of the Pitch Shift's parameter structure is specified using the module identifier *SCE_NGS_PITCHSHIFT_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CEA).
SCE_NGS_PLAYER_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_PLAYER_ID,1,1))	Specifies the ID of the Player's parameter structure. The ID of the Player's parameter structure is specified using the module identifier *SCE_NGS_PLAYER_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE6).
SCE_NGS_REVERB_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_REVERB_ID,1,1))	Specifies the ID of the Reverb's parameter structure. The ID of the Reverb's parameter structure is specified using the module identifier *SCE_NGS_REVERB_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE7).
SCE_NGS_REVERB_PARAMS_STRUCT_ID_V2	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_REVERB_ID,1,2))	Specifies the ID of the Reverb's parameter structure. The ID is specified using the module identifier SCE_NGS_REVERB_ID and the SCE_NGS_MAKE_PARAMS_ID macro (0x01025CE7). Note that this uses the same parameter structure as SCE_NGS_REVERB_PARAMS_STRUCT_ID, though the behavior is modified (see the *NGS Modules Overview* and SceNgsReverbParams structure).

ATRAC9™ Player DSP Effect Module

ATRAC9™ Player Defines

ATRAC9™ Player Defines Table

Define	Value	Description
SCE_NGS_AT9_ID	(0x5CAA)	Specifies the ATRAC9™ Player ID.
SCE_NGS_AT9_MAX_BUFFERS	(4)	Maximum number of buffers to stream input data from.
SCE_NGS_AT9_LOOP_CONTINUOUS	(-1)	Continue looping the current stream input buffer.
SCE_NGS_AT9_NO_NEXT_BUFFER	(-1)	The currently playing buffer is the final buffer to play.
SCE_NGS_AT9_MAX_PCM_CHANNELS	(2)	Maximum number of audio channels available.
SCE_NGS_AT9_MAX_OCTAVE_CHANGE	(2)	Maximum octave range (+2 octaves).
SCE_NGS_AT9_LEFT_CHANNEL	(0)	Left audio channel (used for channel mapping).
SCE_NGS_AT9_RIGHT_CHANNEL	(1)	Right audio channel (used for channel mapping).
SCE_NGS_AT9_NULL_CHANNEL	(-1)	Null audio channel, the data will not be decoded or processed through the voice.
SCE_NGS_AT9_END_OF_DATA	(0)	Callback information: NGS has processed all input data.
SCE_NGS_AT9_SWAPPED_BUFFER	(1)	Callback information: NGS is finished processing a buffer of data and has started processing another buffer (or is looping its current buffer).
SCE_NGS_AT9_HEADER_ERROR	(2)	Header error.
SCE_NGS_AT9_DECODE_ERROR	(3)	Decode error.
SCE_NGS_AT9_LOOPED_BUFFER	(4)	Callback information: NGS has looped and restarted processing a buffer.

ATRAC9™ Player Parameter Structure Defines

ATRAC9™ Player Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_AT9_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_AT9_ID,1,1))	Specifies the ID of the ATRAC9™ Player's parameter structure. The ID of the ATRAC9™ Player's parameter structure is specified using the module identifier *SCE_NGS_AT9_ID* and the SCE_NGS_MAKE_ PARAMS_ID macro (0x01015CAA).

ATRAC9™ Player Structures

SceNgsAT9BufferParams

Structure required to initialize the NGS ATRAC9™ Player DSP effect Module.

Definition

```
#include <ngs/modules/at9_player.h>
typedef struct {
        const void *pBuffer;
        SceInt32 nNumBytes;
        SceInt16 nLoopCount;
        SceInt16 nNextBuff;
        SceInt16 nSamplesDiscardStart;
        SceInt16 nSamplesDiscardEnd;
} SceNgsAT9BufferParams;
```

Members

pBuffer	Pointer to input audio data.
nNumBytes	Size of buffer to play (in bytes).
nLoopCount	Number of times to repeat buffer.
nNextBuff	Index of next buffer to process when this buffer is finished.
nSamplesDiscardStart	Start position of the samples to discard.
nSamplesDiscardEnd	End position of the samples to discard.

Description

Structure required to initialize the NGS ATRAC9™ Player DSP effect Module.

See Also

SceNgsAT9SkipBufferInfo, sceNgsAT9GetSectionDetails(), "ATRAC9™ Player DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsAT9States

Structure used to return the user state of an ATRAC9™ Player.

Definition

```
#include <ngs/modules/at9_player.h>
typedef struct {
        SceInt32 nCurrentBytePositionInBuffer;
        SceInt32 nCurrentBuffer;
        SceInt32 nSamplesGeneratedSinceKeyOn;
        SceInt32 nBytesConsumedSinceKeyOn;
        SceInt32 nSamplesGeneratedTotal;
        SceInt32 nBytesConsumedTotal;
} SceNgsAT9States;
```

Members

nCurrentBytePositionInBuffer	Current byte position in the buffer.
nCurrentBuffer	Buffer of the ATRAC9™ player.
nSamplesGeneratedSinceKeyOn	Samples generated since key on.
nBytesConsumedSinceKeyOn	Bytes consumed since key on.
nSamplesGeneratedTotal	Number of total samples generated for the ATRAC9™ player.
nBytesConsumedTotal	Number of total bytes consumed for the ATRAC9™ player.

Description

Structure used to return the user state information from an ATRAC9™ player to the user.

See Also

`sceNgsVoiceGetStateData`, "ATRAC9™ Player DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsAT9Params

Structure required to initialize the NGS ATRAC9™ Player DSP effect Module.

Definition

```
#include <ngs/modules/at9_player.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceNgsAT9BufferParams buffs[SCE_NGS_AT9_MAX_BUFFERS];
        SceFloat32 fPlaybackFrequency;
        SceFloat32 fPlaybackScalar;
        SceInt32 nLeadInSamples;
        SceInt32 nLimitNumberOfSamplesPlayed;
        SceInt8 nChannels;
        SceInt8 nChannelMap[SCE_NGS_AT9_MAX_PCM_CHANNELS];
        SceInt8 reserved;
        SceInt32 configData;
} SceNgsAT9Params;
```

Members

desc	Default parameter descriptor.
buffs	Pointer to input buffer information.
fPlaybackFrequency	Playback frequency (in Hz).
fPlaybackScalar	Scaling value for playback rate, allows doppler to be easily implemented.
nLeadInSamples	Number of samples to offset into the first packet of output. Allows mid granularity control. The lead in samples unit is the number of output samples and is therefore independent on input sample rate and playback scalar value. Note that the lead in samples unit has no effect after playback has begun.
nLimitNumberOfSamplesPlayed	Optional setting to limit the number of output samples generated. Can be used for time control using the base rate. If set to 0, ignored.
nChannels	Number of audio channels in source PCM.
nChannelMap	Map for each of the output channels.
reserved	Not used.
configData	ATRAC9™ Configuration Data.

Description

Structure required to initialize the NGS ATRAC9™ Player DSP effect Module.

See Also

"ATRAC9™ Player DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsAT9ParamsBlock

Structure required to initialize the parameter block for the NGS ATRAC9™ Player DSP effect Module.

Definition

```
#include <ngs/modules/at9_player.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsAT9Params params;
} SceNgsAT9ParamsBlock;
```

Members

moduleInfo	The Module parameter header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the ATRAC9™ Player DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS ATRAC9™ Player DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"ATRAC9™ Player DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsAT9SkipBufferInfo

Structure used with SceNgsAT9BufferParams and sceNgsAT9GetSectionDetails() to initialize the NGS ATRAC9™ Player DSP effect Module.

Definition

```
#include <ngs/ngs_top.h>
typedef struct {
        SceInt32 nStartByteOffset;
        SceInt32 nNumBytes;
        SceInt16 nStartSkip;
        SceInt16 nEndSkip;
        SceInt32 nIsSuperPacket;
} SceNgsAT9SkipBufferInfo;
```

Members

nStartByteOffset	Byte offset for the ATRAC9™ data that should be passed as a pointer into SceNgsAT9BufferParams.pBuffer.
nNumBytes	Size in bytes for the section of the ATRAC9™ data that should be passed into SceNgsAT9BufferParams.nNumBytes.
nStartSkip	Number of samples to discard from the start of the decoded data that should be passed into SceNgsAT9BufferParams.nSamplesDiscardStart.
nEndSkip	Number of samples to discard from the end of the decoded data that should be passed into SceNgsAT9BufferParams.nSamplesDiscardEnd.
nIsSuperPacket	Reference value of either 0 (not a super packet) or 1 (a super packet).

Description

Structure used with SceNgsAT9BufferParams and sceNgsAT9GetSectionDetails() to initialize the NGS ATRAC9™ Player DSP effect Module.

See Also

SceNgsAT9BufferParams, sceNgsAT9GetSectionDetails(), "ATRAC9™ Player DSP Effect Module Overview" in the *NGS Modules Overview*.

Compressor DSP Effect Module

Compressor Defines

Compressor Defines Table

Define	Value	Description
SCE_NGS_COMPRESSOR_ID	(0x5CE1)	Specifies the Compressor ID.
SCE_NGS_COMPRESSOR_RMS_MODE	(0)	Specifies the Compressor will operate in RMS mode.
SCE_NGS_COMPRESSOR_PEAK_MODE	(1)	Specifies the Compressor will operate in Peak mode.
SCE_NGS_COMPRESSOR_STEREO_LINK_OFF	(0)	Compressor will compress left and right audio signal data using information from the left and right audio input signals.
SCE_NGS_COMPRESSOR_STEREO_LINK_ON	(1)	Compressor will compress left and right audio signal data using an average of the left and right audio input signals.
SCE_NGS_COMPRESSOR_VOICE_COMPRESSOR_MODULE	(1)	Compressor Module.
SCE_NGS_COMPRESSOR_SIDE_CHAIN_VOICE_INPUT_0_MODULE	(0)	Input 2 Module.
SCE_NGS_COMPRESSOR_SIDE_CHAIN_VOICE_INPUT_1_MODULE	(1)	Input 1 Module.
SCE_NGS_COMPRESSOR_SIDE_CHAIN_VOICE_COMPRESSOR_MODULE	(2)	Compressor Module.

Compressor Parameter Structure Defines

Compressor Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_COMPRESSOR_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_COMPRESSOR_ID,1,1))	Specifies the ID of the Compressor's parameter structure. The ID of the Compressor's parameter structure is specified using the module identifier *SCE_NGS_COMPRESSOR_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE1).
SCE_NGS_COMPRESSOR_PARAMS_STRUCT_ID_V2	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_COMPRESSOR_ID,1,2))	Specifies the ID of the Compressor's parameter structure. The ID is specified using the module identifier *SCE_NGS_COMPRESSOR_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01025CE1). Note that this uses the same parameter structure as SCE_NGS_COMPRESSOR_PARAMS_STRUCT_ID, though the behavior is modified (see the *NGS Modules Overview* and SceNgsCompressorParams structure).

Compressor Structures

SceNgsCompressorParams

Structure required to initialize the NGS Compressor DSP effect Module.

Definition

```
#include <ngs/modules/compressor.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceFloat32 fRatio;
        SceFloat32 fThreshold;
        SceFloat32 fAttack;
        SceFloat32 fRelease;
        SceFloat32 fMakeupGain;
        SceInt32 nStereoLink;
        SceInt32 nPeakMode;
        SceFloat32 fSoftKnee;
} SceNgsCompressorParams;
```

Members

desc	Default parameter descriptor.
fRatio	Amount the volume is reduced in relation to how far above the threshold the signal is. Value ranges from 0 to n where 1 = 1:1 ratio, 0.5 = 2:1 ratio.
fThreshold	Threshold above which the volume will automatically be reduced based on the *fRatio* and *fAttack* parameters.
fAttack	Rate (in seconds) at which volume is reduced when volume exceeds the specified threshold. Value ranges from 0 to n.
fRelease	Rate (in seconds) at which the volume returns when the volume no longer exceeds the specified threshold. Value ranges from 0 to n.
fMakeupGain	Volume control for the compressed audio signal (dB). Acts as a post compression gain.
nStereoLink	Flag to link the two channels of stereo together and use an average of the two input audio channels as the current volume.
nPeakMode	Flag that is set to select either PEAK or RMS algorithms when calculating the input signal.
fSoftKnee	Width (in dB) for soft knee compressor to operate over. Relative to the input signal and centered around the threshold. Set to 0 for hard knee compression.

Description

Structure required to initialize the NGS Compressor DSP effect Module.

This structure is required when accessing parameter data via the `sceNgsVoiceLockParams()` function.

Note that there are two parameter structure defines that may be used to modify the behavior when used as a side chain compressor. With `SCE_NGS_COMPRESSOR_PARAMS_STRUCT_ID`, if the side chain input is switched off (achieved by bypassing incoming modules) the compressor treats the side chain as having zero volume. Alternately, when specifying `SCE_NGS_COMPRESSOR_PARAMS_STRUCT_ID_V2` with the side chain input switched off, the compressor reverts to the behavior of a standard, non-side chained compressor (compressing the input signal based on its own dynamics).

See Also

"Compressor DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsCompressorParamsBlock

Structure required to initialize the parameter block for the NGS Compressor DSP effect Module.

Definition

```
#include <ngs/modules/compressor.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsCompressorParams params;
} SceNgsCompressorParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see SceNgsModuleParamHeader).
params	The parameter structure including the data used to initialize the Compressor DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Compressor DSP effect Module.

Using ParamsBlock is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your ParamsBlock structure, then call the sceNgsVoiceSetParamsBlock function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Compressor DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsCompressorStates

Structure used to return the user state of a NGS Compressor DSP effect Module.

Definition

```
#include <ngs/modules/compressor.h>
typedef struct {
        SceFloat32 fInputLevel[SCE_NGS_MAX_SYSTEM_CHANNELS];
        SceFloat32 fOutputLevel[SCE_NGS_MAX_SYSTEM_CHANNELS];
} SceNgsCompressorStates;
```

Members

fInputLevel	The input level calculated by the NGS Compressor DSP effect Module, expressed as a linear value.
fOutputLevel	The output level calculated by the NGS Compressor DSP effect Module, expressed as a linear value or the gain if in side-chain compression mode.

Description

Structure used to return the user state of an active NGS Compressor DSP effect Module.

See Also

`sceNgsVoiceGetStateData`, "Compressor DSP Effect Module Overview" in the *NGS Modules Overview*.

Delay DSP Effect Module

Delay Defines

Delay Defines Table

Define	Value	Description
SCE_NGS_DELAY_ID	(0x5CEB)	Specifies the Delay ID.
SCE_NGS_DELAY_MAX_TAPS	(4)	Specifies the maximum number of taps that the Delay DSP effect Module can process.
SCE_NGS_DELAY_FILTER_MODE_OFF	(0)	No filter effect.
SCE_NGS_DELAY_FILTER_MODE_LOWPASS_ONEPOLE	(1)	Low pass one-pole filter, cut-off frequency only.
SCE_NGS_DELAY_FILTER_MODE_HIGHPASS_ONEPOLE	(2)	High pass one-pole filter, cut-off frequency only.
SCE_NGS_DELAY_FILTER_MODE_ALLPASS	(3)	All pass filter. Alters phase of signal. Cut-off and bandwidth (Q).
SCE_NGS_DELAY_PRESET_DEFAULT	(0)	Preset default delay effect.
SCE_NGS_DELAY_PRESET_MONO_CHORUS	(1)	Preset mono chorus delay effect.
SCE_NGS_DELAY_PRESET_STEREO_CHORUS_LEFT	(2)	Preset stereo chorus left-side delay effect.
SCE_NGS_DELAY_PRESET_STEREO_CHORUS_RIGHT	(3)	Preset stereo chorus right-side delay effect.
SCE_NGS_DELAY_PRESET_MONO_FLANGER	(4)	Preset mono flanger delay effect.
SCE_NGS_DELAY_PRESET_STEREO_FLANGER_LEFT	(4)	Preset stereo flanger left-side delay effect.
SCE_NGS_DELAY_PRESET_STEREO_FLANGER_RIGHT	(5)	Preset stereo flanger right-side delay effect.
SCE_NGS_DELAY_BUSS_DELAY_MODULE	(1)	Delay Module.

Delay Parameter Structure Defines

Delay Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_DELAY_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_DELAY_ID,1,1))	Specifies the ID of the Delay's parameter structure. The ID of the Delay's parameter structure is specified using the module identifier *SCE_NGS_DELAY_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CEB).

Delay Structures

SceNgsDelayTap

Structure to set up Delay Tap information.

Definition

```
#include <ngs/modules/delay.h>
typedef struct {
        SceFloat32 fDelayMillisecs;
        SceFloat32 fVolume;
        SceFloat32 fFeedback;
        SceUInt32 eFilterMode;
        SceFloat32 fCutoff;
        SceFloat32 fPhaseOffsetDeg;
        SceFloat32 fModWidthMillisecs;
} SceNgsDelayTap;
```

Members

fDelayMillisecs	Delay time in msecs.
fVolume	Tap volume (linear).
fFeedback	Tap feedback (linear).
eFilterMode	Filter mode.
fCutoff	Cutoff in Hz for the filter.
fPhaseOffsetDeg	Phase offset in degrees.
fModWidthMillisecs	Modulation width in msecs (Modulation speed is set via the *fModRate* argument in SceNgsDelayParams. This would affect all delay taps).

Description

Structure required to set up Delay Tap information.

SceNgsDelayParams

Structure required to initialize the NGS Delay DSP effect Module.

Definition

```
#include <ngs/modules/delay.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceFloat32 fDryVol;
        SceFloat32 fModRate;
        SceNgsDelayTap taps[SCE_NGS_DELAY_MAX_TAPS];
} SceNgsDelayParams;
```

Members

desc	Default parameter descriptor.
fDryVol	Dry throughput volume (linear).
fModRate	Modulation rate in Hz.
taps	Array of Delay taps.

Description

Structure required to initialize the NGS Delay DSP effect Module.

See Also

"Delay DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsDelayParamsBlock

Structure required to initialize the parameter block for the NGS Delay DSP effect Module.

Definition

```
#include <ngs/modules/delay.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsDelayParams params;
} SceNgsDelayParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Delay DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Delay DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Delay DSP Effect Module Overview" in the *NGS Modules Overview*.

Distortion DSP Effect Module

Distortion Defines

Distortion Defines Tables

Define	Value	Description
SCE_NGS_DISTORTION_ID	(0x5CE2)	Specifies the Distortion ID.
SCE_NGS_DISTORTION_VOICE_DISTORTION_MODULE	(1)	Distortion Module.

Distortion Parameter Structure Defines

Distortion Parameter Structure Defines Tables

Define	Value	Description
SCE_NGS_DISTORTION_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_DISTORTION_ID, 1, 1))	Specifies the ID of the Distortion's parameter structure. The ID of the Distortion's parameter structure is specified using the module identifier *SCE_NGS_DISTORTION_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE2).

Distortion Structures

SceNgsDistortionParams

Structure required to initialize the NGS Polynomial Distortion DSP effect Module.

Definition

```
#include <ngs/modules/distortion.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceFloat32 fA;
        SceFloat32 fB;
        SceFloat32 fClip;
        SceFloat32 fGate;
        SceFloat32 fWetGain;
        SceFloat32 fDryGain;
} SceNgsDistortionParams;
```

Members

desc	Default parameter descriptor.
fA	A parameter for the distortion algorithm. Value ranges from 0 to 10.
fB	B parameter for the distortion algorithm. Value ranges from 0 to 10.
fClip	Limiter on the audio output of the polynomial stage. Value ranges from 0 to 4.
fGate	Noise gate on the audio output. Value ranges from 0 to 4.
fWetGain	Wet gain. Amplitude of the distorted signal. Value ranges from 0 to 4 (where 1 = original volume, 2 = twice volume).
fDryGain	Dry gain. Amount of original signal to mix with the distorted signal. Value ranges from 0 to 4 (where 1 = original volume, 2 = twice volume).

Description

Structure required to initialize the NGS Polynomial Distortion DSP effect Module.

See Also

"Polynomial Distortion DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsDistortionParamsBlock

Structure required to initialize the parameter block for the NGS Distortion DSP effect Module.

Definition

```
#include <ngs/modules/distortion.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsDistortionParams params;
} SceNgsDistortionParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Distortion DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Distortion DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Distortion DSP Effect Module Overview" in the *NGS Modules Overview*.

Envelope DSP Effect Module

Envelope Defines

Envelope Defines Table

Define	Value	Description
SCE_NGS_ENVELOPE_ID	(0x5CE3)	Specifies the Envelope ID.
SCE_NGS_ENVELOPE_MAX_POINTS	(4)	Maximum number of points within an Envelope.
SCE_NGS_ENVELOPE_NO_LOOP	(-1)	Specifies that the Envelope does not loop.
SCE_NGS_ENVELOPE_NO_LOOP_TERMINATE	(-2)	Specifies that the envelope does not loop and will terminate immediately upon completion of the final segment.
SCE_NGS_ENVELOPE_NO_LOOP_RELEASE	(-3)	Specifies that the envelope does not loop and will automatically switch to a release phase upon completion of the final segment.
SCE_NGS_ENVELOPE_LINEAR	(0)	Use linear interpolation calculation between two points.
SCE_NGS_ENVELOPE_CURVED	(1)	Use curved interpolation calculate between two points.

Envelope Parameter Structure Defines

Envelope Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_ENVELOPE_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_ENVELOPE_ID, 1, 1))	Specifies the ID of the Envelope's parameter structure. The ID of the Envelope's parameter structure is specified using the module identifier *SCE_NGS_ENVELOPE_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE3).

Envelope Structures

SceNgsEnvelopePoint

Structure required to initialize the NGS Envelope DSP effect Module.

Definition

```
#include <ngs/modules/envelope.h>
typedef struct {
        SceUInt32 uMsecsToNextPoint;
        SceFloat32 fAmplitude;
        SceUInt32 eCurveType;
} SceNgsEnvelopePoint;
```

Members

uMsecsToNextPoint	Delay time (in milliseconds) to interpolate between current and next envelope point.
fAmplitude	Amplitude of source envelope point (0=silence, 1= original volume, 2= twice original volume).
eCurveType	See Envelope defines.

Description

Structure required to initialize the NGS Envelope DSP effect Module.

This structure is part of the SceNgsEnvelopeParams structure. Valid Envelope curve types are:

- SCE_NGS_ENVELOPE_LINEAR (0)
- SCE_NGS_ENVELOPE_CURVED (1)

See Also

"Envelope DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsEnvelopeParams

Structure required to initialize the NGS Envelope DSP effect Module.

Definition

```
#include <ngs/modules/envelope.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceNgsEnvelopePoint envelopePoints[SCE_NGS_ENVELOPE_MAX_POINTS];
        SceUInt32 uReleaseMsecs;
        SceUInt32 uNumPoints;
        SceUInt32 uLoopStart;
        SceInt32 nLoopEnd;
} SceNgsEnvelopeParams;
```

Members

desc	Default parameter descriptor.
envelopePoints	Envelope point information (see SceNgsEnvelopePoint structure).
uReleaseMsecs	Release rate length (in msecs).
uNumPoints	Number of points within the Envelope (1-4).
uLoopStart	Loop starting point (0-2).
nLoopEnd	Loop ending point (1-3 and > *uLoopStart*).

Description

Structure required to initialize the NGS Envelope DSP effect Module.

This structure is required when accessing parameter data via the `sceNgsVoiceLockParams()` function.

See Also

"Envelope DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsEnvelopeStates

Structure used to return the user state of an Envelope.

Definition

```
#include <ngs/modules/envelope.h>
typedef struct {
        SceFloat32 fCurrentHeight;
        SceFloat32 fPosition;
        SceFloat32 fReleaseScale;
        SceInt32 nCurrentPoint;
        SceInt32 nReleasing;
} SceNgsEnvelopeStates;
```

Members

fCurrentHeight	The current amplitude that has been calculated by the envelope Module, based on the current processing position between two envelope points.
fPosition	Current processing point between two envelope points (previous and next). This point (X) is then used to calculate the current envelope amplitude (Y).
fReleaseScale	The volume multiplier that is applied to the current envelope amplitude and is scaled between 0.0f - 1.0f.
nCurrentPoint	The current envelope point (see SceNgsEnvelopePoint structure).
nReleasing	A flag that is set to 1 if the envelope is in the release / keyoff state; otherwise 0.

Description

Structure used to return the user state information from an Envelope to the user.

See Also

sceNgsVoiceGetStateData, "Envelope DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsEnvelopeParamsBlock

Structure required to initialize the parameter block for the NGS Envelope DSP effect Module.

Definition

```
#include <ngs/modules/envelope.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsEnvelopeParams params;
} SceNgsEnvelopeParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Envelope DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Envelope DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock()` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Envelope DSP Effect Module Overview" in the *NGS Modules Overview*.

Document serial number: 000004892117

Filter DSP Effect Module

Filter Defines

Filter Defines Table

Define	Value	Description
SCE_NGS_FILTER_ID	(0x5CE4)	Specifies the Filter ID.
SCE_NGS_FILTER_MODE_OFF	(0)	No effect.
SCE_NGS_FILTER_LOWPASS_RESONANT	(1)	Low pass two-pole resonant filter. Accepts cut-off and resonant Q.
SCE_NGS_FILTER_HIGHPASS_RESONANT	(2)	High pass two-pole resonant filter. Accepts cut-off and resonant Q.
SCE_NGS_FILTER_BANDPASS_PEAK	(3)	Band pass filter with tunable gain in decibels at the peak frequency.
SCE_NGS_FILTER_BANDPASS_ZERO	(4)	Band pass filter, maximum output is zero dB relative to input. Tunable bandwidth (Q).
SCE_NGS_FILTER_NOTCH	(5)	A notch filter, removes frequencies around cut-off frequency, bandwidth controlled by Q.
SCE_NGS_FILTER_PEAK	(6)	Peak/Notch EQ, frequency controls center frequency, Q controls bandwidth, gain controls dB gain at center frequency.
SCE_NGS_FILTER_HIGHSHELF	(7)	High shelving filter/EQ. Gain controls dB cut/boost above cut-off.
SCE_NGS_FILTER_LOWSHELF	(8)	Low shelving filter/EQ. Gain controls dB cut/boost below cut-off.
SCE_NGS_FILTER_LOWPASS_ONEPOLE	(9)	Low pass one-pole filter, cut-off frequency only.
SCE_NGS_FILTER_HIGHPASS_ONEPOLE	(10)	High pass one-pole filter, cut-off frequency only.
SCE_NGS_FILTER_ALLPASS	(11)	All pass filter. Alters phase of signal. Cut-off and bandwidth (Q).
SCE_NGS_FILTER_LOWPASS_RESONANT_NORMALIZED	(12)	Low pass resonant filter. In this mode, the input before the cut-off peak is reduced at higher resonance Q values.

Filter Parameter Structure Defines

Filter Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_FILTER_ RAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_FILTER_ID,1,1))	Specifies the ID of the Filter's parameter structure. The ID of the Filter's parameter structure is specified using the module identifier *SCE_NGS_FILTER_ID*, index 1 and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE4).
SCE_NGS_FILTER_ PARAMS_COEFF_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_FILTER_ID,2,1))	Specifies the ID of the Filter's parameter coefficient structure. The ID of the Filter's parameter coefficient structure is specified using the module identifier *SCE_NGS_FILTER_ID*, index 2 and the SCE_NGS_MAKE_PARAMS_ID macro (0x02015CE4).

Filter Structures

SceNgsFilterParams

Structure required to initialize the NGS Filter DSP effect Module.

Definition

```
#include <ngs/modules/filter.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceUInt32 eFilterMode;
        SceFloat32 fFrequency;
        SceFloat32 fResonance;
        SceFloat32 fGain;
} SceNgsFilterParams;
```

Members

desc	Default parameter descriptor.
eFilterMode	Selected filter mode (see SCE_NGS_FILTER_MODE).
fFrequency	Cut-off frequency of the filter.
fResonance	Amount of resonant Q for the filter. This parameter only affects certain filter modes. Typical ranges are 0.2 to 10.0.
fGain	Decibel gain or cut-off of the filter. This parameter only affects certain filter modes.

Description

Structure required to initialize the NGS Filter DSP effect Module.

See Also

"Filter DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsFilterParamsCoEff

Structure required to initialize the NGS Filter DSP effect Module.

Definition

```
#include <ngs/modules/filter.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceFloat32 fB0;
        SceFloat32 fB1;
        SceFloat32 fB2;
        SceFloat32 fA1;
        SceFloat32 fA2;
} SceNgsFilterParamsCoEff;
```

Members

desc	Default parameter descriptor.
fB0	Coefficient.
fB1	Coefficient.
fB2	Coefficient.
fA1	Coefficient.
fA2	Coefficient.

Description

Structure required to initialize the NGS Filter DSP effect Module. Allows you to pass coefficient information to the Filter, rather than specifying a Filter type, cutoff frequency and so on.

Coefficient formula is as follows:

```
y0 = x(0) * B0 + x(-1) * B1 + x(-2) * B2 - y(-1) * A1 - y(-2) * A2
```

See Also

"Filter DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsFilterParamsBlock

Structure required to initialize the parameter block for the NGS Filter DSP effect Module.

Definition

```
#include <ngs/modules/filter.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsFilterParams params;
} SceNgsFilterParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Filter DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Filter DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock()` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Filter DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsFilterCoEffParamsBlock

Structure required to initialize the parameter block for the NGS Filter DSP effect Module.

Definition

```
#include <ngs/modules/filter.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsFilterParamsCoEff params;
} SceNgsFilterCoEffParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Filter DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Filter DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock()` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Filter DSP Effect Module Overview" in the *NGS Modules Overview*.

Input Mixer DSP Effect Module

Input Mixer Defines

Input Mixer Defines Table

Define	Value	Description
SCE_NGS_INPUT_MIXER_ID	(0x5CE0)	Specifies the Input Mixer ID.

Mixer DSP Effect Module

Mixer Defines

Mixer Defines Table

Define	Value	Description
SCE_NGS_MIXER_ID	(0x5CE9)	Specifies the Mixer ID.
SCE_NGS_MIXER_MAX_PORTS	(2)	Maximum number of input ports.

Mixer Parameter Structure Defines

Mixer Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_MIXER_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_MIXER_ID,1,1))	Specifies the ID of the Mixer's parameter structure. The ID of the Mixer's parameter structure is specified using the module identifier *SCE_NGS_MIXER_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE9).

Mixer Structures

SceNgsMixerParams

Structure required to initialize the NGS Mixer DSP effect Module.

Definition

```
#include <ngs/modules/mixer.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceFloat32 fGainIn[SCE_NGS_MIXER_MAX_PORTS];
        SceUInt32 uPauseLeadIn;
        SceUInt32 uPauseLeadOut;
} SceNgsMixerParams;
```

Members

desc	Default parameter descriptor.
fGainIn	Multiplier for each input resulting in volume level change (0=silence, 1=original volume, 2= twice original volume).
uPauseLeadIn	Number of samples to process when fading audio in (when audio is resumed).
uPauseLeadOut	Number of samples to process when fading audio out (when audio is paused).

Description

Structure required to initialize the NGS Mixer DSP effect Module.

See Also

"Mixer DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsMixerParamsBlock

Structure required to initialize the parameter block for the NGS Mixer DSP effect Module.

Definition

```
#include <ngs/modules/mixer.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsMixerParams params;
} SceNgsMixerParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Mixer DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Mixer DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock()` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Mixer DSP Effect Module Overview" in the *NGS Modules Overview*.

Parametric EQ DSP Effect Module

Parametric EQ Defines

Parametric EQ Defines Table

Define	Value	Description
SCE_NGS_PARAM_EQ_ID	(0x5CEC)	Specifies the Parametric EQ ID.
SCE_NGS_MAX_EQ_FILTERS	(4)	Specifies the max number of Parametric EQ filters.

Parametric EQ Parameter Structure Defines

Parametric EQ Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_PARAM_EQ_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_PARAM_EQ_ID,1,1))	Specifies the ID of the Parametric EQ's parameter structure. The ID of the Parametric EQ's parameter structure is specified using the module identifier *SCE_NGS_PARAM_EQ_ID*, index 1 and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CEC).
SCE_NGS_PARAM_EQ_COEFF_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_PARAM_EQ_ID,2,1))	Specifies the ID of the Parametric EQ's parameter coefficient structure. The ID of the Parametric EQ's parameter coefficient structure is specified using the module identifier *SCE_NGS_PARAM_EQ_ID*, index 2 and the SCE_NGS_MAKE_PARAMS_ID macro (0x02015CEC).

Parametric EQ Structures

SceNgsParamEqFilter

Structure required to specify the initialization parameters used by the NGS Parametric EQ DSP effect Module.

Definition

```
#include <ngs/modules/para_eq.h>
typedef struct {
        SceUInt32 eFilterMode;
        SceFloat32 fFrequency;
        SceFloat32 fResonance;
        SceFloat32 fGain;
} SceNgsParamEqFilter;
```

Members

eFilterMode	Selected filter mode (see SCE_NGS_FILTER_MODE). Please see "Filter DSP Effect Module Overview" in the *NGS Modules Overview* for a list of filter modes.
fFrequency	Cut-off frequency of the filter.
fResonance	Amount of resonant Q for the filter. This parameter only affects certain filter modes. Typical ranges are 0.2 to 10.0.
fGain	Decibel gain or cut-off of the filter. This parameter only affects certain filter modes.

Description

Structure required to specify the initialization parameters used by the NGS Parametric EQ DSP effect Module. The parameters for initializing the Parametric EQ are specified in the SceNgsParamEqParams structure, which references the SceNgsParamEqFilter structure through its *eFilterMode* parameter.

See Also

SceNgsParamEqParams, "Parametric EQ DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsParamEqCoEff

Structure required to specify the coefficients used by the NGS Parametric EQ DSP effect Module.

Definition

```
#include <ngs/modules/para_eq.h>
typedef struct {
        SceFloat32 fB0;
        SceFloat32 fB1;
        SceFloat32 fB2;
        SceFloat32 fA1;
        SceFloat32 fA2;
} SceNgsParamEqCoEff;
```

Members

fB0	Coefficient.
fB1	Coefficient.
fB2	Coefficient.
fA1	Coefficient.
fA2	Coefficient.

Description

Structure required to specify the coefficients used by the NGS Parametric EQ DSP effect Module. Allows you to pass coefficient information to the Parametric EQ Module. The coefficient parameters for the Parametric EQ are specified in the SceNgsParamEqParamsCoEff structure, which references the SceNgsParamEqCoEff structure through its *FilterCoEff* parameter.

Coefficient formula is as follows:

y0 = x(0) * *fB0* + x(-1) * *fB1* + x(-2) * *fB2* - y(-1) * *fA1* - y(-2) * *fA2*

See Also

"Parametric EQ DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsParamEqParams

Structure required to initialize the NGS Parametric EQ DSP effect Module.

Definition

```
#include <ngs/modules/para_eq.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceNgsParamEqFilter filter[SCE_NGS_MAX_EQ_FILTERS];
} SceNgsParamEqParams;
```

Members

desc	Default parameter descriptor.
filter	Selected Parametric EQ filter mode. Refer to SCE_NGS_PARAM_EQ_MAX_EQ_FILTERS for the maximum number of filters allowed.

Description

Structure required to initialize the NGS Parametric EQ DSP effect Module.

See Also

SceNgsParamEqFilter, "Parametric EQ DSP Effect Module Overview" in the *NGS Modules Overview*, "Filter DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsParamEqParamsCoEff

Structure required to initialize the NGS Parametric EQ DSP effect Module.

Definition

```
#include <ngs/modules/para_eq.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceNgsParamEqCoEff filterCoEff[SCE_NGS_MAX_EQ_FILTERS];
} SceNgsParamEqParamsCoEff;
```

Members

desc	Default parameter descriptor.
filterCoEff	Selected Parametric EQ filter coefficient. Refer to SCE_NGS_PARAM_EQ_MAX_EQ_FILTERS for the maximum number of filters allowed.

Description

Structure required to initialize the NGS Parametric EQ DSP effect Module. Allows you to pass coefficient information to the Parametric EQ Module. To specify the coefficients used you must reference a SceNgsParamEqCoEff structure.

See Also

SceNgsParamEqCoEff, "Parametric EQ DSP Effect Module Overview" in the *NGS Modules Overview*, "Filter DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsParamEqParamsBlock

Structure required to initialize the parameter block for the NGS Parametric EQ DSP effect Module.

Definition

```
#include <ngs/modules/para_eq.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsParamEqParams params;
} SceNgsParamEqParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Parametric EQ DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Parametric EQ DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock()` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Parametric EQ DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsParamEqCoEffParamsBlock

Structure required to initialize the parameter block for the NGS Parametric EQ DSP effect Module.

Definition

```
#include <ngs/modules/para_eq.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsParamEqParamsCoEff params;
} SceNgsParamEqCoEffParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Parametric EQ DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Parametric EQ DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock()` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Parametric EQ DSP Effect Module Overview" in the *NGS Modules Overview*.

Pauser DSP Effect Module

Pauser Defines

Pauser Defines Table

Define	Value	Description
SCE_NGS_PAUSER_ID	(0x5CE5)	Specifies the Pauser ID.
SCE_NGS_PAUSER_VOICE_PAUSER_MODULE	(1)	Pauser Module.

Pauser Parameter Structure Defines

Pauser Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_PAUSER_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_PAUSER_ID,1,1))	Specifies the ID of the Pauser's parameter structure. The ID of the Pauser's parameter structure is specified using the module identifier *SCE_NGS_PAUSER_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE5).

Pauser Structures

SceNgsPauserParams

Structure required to initialize the NGS Pauser DSP effect Module.

Definition

```
#include <ngs/modules/pauser.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceUInt32 uPauseLeadIn;
        SceUInt32 uPauseLeadOut;
} SceNgsPauserParams;
```

Members

desc	Default parameter descriptor.
uPauseLeadIn	Number of samples to process when fading audio in (when audio is resumed).
uPauseLeadOut	Number of samples to process when fading audio out (when audio is paused).

Description

Structure required to initialize the NGS Pauser DSP effect Module.

See Also

"Pauser DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsPauserParamsBlock

Structure required to initialize the parameter block for the NGS Pauser DSP effect Module.

Definition

```
#include <ngs/modules/pauser.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsPauserParams params;
} SceNgsPauserParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Pauser DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Pauser DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock()` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Pauser DSP Effect Module Overview" in the *NGS Modules Overview*.

Player DSP Effect Module

Player Defines

Player Defines Table

Define	Value	Description
SCE_NGS_PLAYER_ID	(0x5CE6)	Specifies the Player ID.
SCE_NGS_PLAYER_MAX_BUFFERS	(4)	Maximum number of buffers to stream input data from.
SCE_NGS_PLAYER_LOOP_CONTINUOUS	(-1)	Continue looping the current stream input buffer.
SCE_NGS_PLAYER_NO_NEXT_BUFFER	(-1)	The currently playing buffer is the final buffer to play.
SCE_NGS_PLAYER_MAX_PCM_CHANNELS	(2)	Maximum number of audio channels available.
SCE_NGS_PLAYER_MAX_OCTAVE_CHANGE	(2)	Maximum octave range (+2 octaves).
SCE_NGS_PLAYER_STREAMING	(0)	Voice is a streaming Voice.
SCE_NGS_PLAYER_RESIDENT	(1)	Voice is a resident Voice.
SCE_NGS_PLAYER_LEFT_CHANNEL	(0)	Left audio channel (used for channel mapping).
SCE_NGS_PLAYER_RIGHT_CHANNEL	(1)	Right audio channel (used for channel mapping).
SCE_NGS_PLAYER_NULL_CHANNEL	(-1)	Null audio channel, the data will not be decoded or processed through the voice.
SCE_NGS_PLAYER_TYPE_PCM	(0)	PCM input data.
SCE_NGS_PLAYER_TYPE_ADPCM	(1)	ADPCM input data.
SCE_NGS_PLAYER_END_OF_DATA	(0)	Callback information: NGS has processed all input data.
SCE_NGS_PLAYER_SWAPPED_BUFFER	(1)	Callback information: NGS is finished processing a buffer of data and has started processing another buffer.
SCE_NGS_PLAYER_LOOPED_BUFFER	(2)	Callback information: NGS has looped and restarted processing a buffer.

Player Parameter Structure Defines

Player Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_PLAYER_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_PLAYER_ID,1,1))	Specifies the ID of the Player's parameter structure. The ID of the Player's parameter structure is specified using the module identifier *SCE_NGS_PLAYER_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE6).

Player Structures

SceNgsPlayerBufferParams

Structure required to initialize the NGS Player DSP effect Module.

Definition

```
#include <ngs/modules/pcm_player.h>
typedef struct {
        const void *pBuffer;
        SceInt32 nNumBytes;
        SceInt16 nLoopCount;
        SceInt16 nNextBuff;
} SceNgsPlayerBufferParams;
```

Members

pBuffer	Pointer to input audio data.
nNumBytes	Size of buffer to play (in bytes).
nLoopCount	Number of times to repeat buffer.
nNextBuff	Index of next buffer to process when this buffer is finished.

Description

Structure required to initialize the NGS Player DSP effect Module.

See Also

"Player DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsPlayerStates

Structure used to return the user state of a Player.

Definition

```
#include <ngs/modules/pcm_player.h>
typedef struct {
        SceInt32 nCurrentBytePositionInBuffer;
        SceInt32 nCurrentBuffer;
        SceInt32 nSamplesGeneratedSinceKeyOn;
        SceInt32 nBytesConsumedSinceKeyOn;
        SceInt32 nSamplesGeneratedTotal;
        SceInt32 nBytesConsumedTotal;
} SceNgsPlayerStates;
```

Members

nCurrentBytePositionInBuffer	Current read position within the audio buffer that is currently being processed.
nCurrentBuffer	Audio buffer number (0-3) that is currently being processed.
nSamplesGeneratedSinceKeyOn	Samples generated since key on.
nBytesConsumedSinceKeyOn	Bytes consumed since key on.
nSamplesGeneratedTotal	Number of total samples generated for the player.
nBytesConsumedTotal	Number of total bytes consumed for the player.

Description

Structure used to return the user state information from a Player to the user.

See Also

`sceNgsVoiceGetStateData`, "Player DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsPlayerParams

Structure required to initialize the NGS Player DSP effect Module.

Definition

```
#include <ngs/modules/pcm_player.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceNgsPlayerBufferParams buffs[SCE_NGS_PLAYER_MAX_BUFFERS];
        SceFloat32 fPlaybackFrequency;
        SceFloat32 fPlaybackScalar;
        SceInt32 nLeadInSamples;
        SceInt32 nLimitNumberOfSamplesPlayed;
        SceInt32 nStartByte;
        SceInt8 nChannels;
        SceInt8 nChannelMap[SCE_NGS_PLAYER_MAX_PCM_CHANNELS];
        SceInt8 nType;
        SceInt8 reserved;
        SceInt8 nStartBuffer;
        SceInt8 pad[2];
} SceNgsPlayerParams;
```

Members

desc	Default parameter descriptor.
buffs	Pointer to input buffer information.
fPlaybackFrequency	Playback frequency (in Hz).
fPlaybackScalar	Scaling value for playback rate, allows doppler to be easily implemented.
nLeadInSamples	Number of samples to wait when outputting audio data (allows for audio data to start within a single PCM output package, for sample accurate playback). The lead in samples unit is the number of output samples and is therefore independent on input sample rate and playback scalar value. Note that the lead in samples unit has no effect after playback has begun.
nLimitNumberOfSamplesPlayed	Optional setting to limit the number of output samples generated. Can be used for time control using the base rate. If set to 0, ignored.
nStartByte	Byte in buffer to start on, at a key on.
nChannels	Number of audio channels in input data.
nChannelMap	Map the output audio channels.
nType	Input data format.
reserved	Not used.
nStartBuffer	Buffer to start on, at a key on.
pad	Padding structure to align to 32-bit.

Description

Structure required to initialize the NGS Player DSP effect Module.

See Also

"Player DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsPlayerParamsBlock

Structure required to initialize the parameter block for the NGS Player DSP effect Module.

Definition

```
#include <ngs/modules/pcm_player.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsPlayerParams params;
} SceNgsPlayerParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Player DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Player DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock()` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Player DSP Effect Module Overview" in the *NGS Modules Overview*.

Pitch Shift DSP Effect Module

Pitch Shift Defines

Pitch Shift Defines Table

Define	Value	Description
SCE_NGS_PITCHSHIFT_ID	(0x5CEA)	Specifies the Pitch Shift ID.
SCE_NGS_PITCHSHIFT_VOICE_PITCHSHIFT_MODULE	(1)	Pitch Shift Module.

Pitch Shift Structure Defines

Pitch Shift Structure Defines Table

Define	Value	Description
SCE_NGS_PITCHSHIFT_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_PITCHSHIFT_ID,1,1))	Specifies the ID of the Pitch Shift's parameter structure. The ID of the Pitch Shift's parameter structure is specified using the module identifier *SCE_NGS_PITCHSHIFT_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CEA).

Pitch Shift Structures

SceNgsPitchShiftParams

Structure required to initialize the NGS Pitch Shift DSP effect Module.

Definition

```
#include <ngs/modules/pitch_shift.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceFloat32 fPitchOffsetInCents;
} SceNgsPitchShiftParams;
```

Members

desc	Default parameter descriptor.
fPitchOffsetInCents	Pitch shift (in cents, 1 semitone = 100 cents). 0.0 = no pitch shift, -1200.0 = 1 octave down, 1200.0 = 1 octave up.

Description

Structure required to initialize the NGS Pitch Shift DSP effect Module.

See Also

"Pitch Shift DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsPitchShiftParamsBlock

Structure required to initialize the parameter block for the NGS Pitch Shift DSP effect Module.

Definition

```
#include <ngs/modules/pitch_shift.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsPitchShiftParams params;
} SceNgsPitchShiftParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Pitch Shift DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Pitch Shift DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock()` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Pitch Shift DSP Effect Module Overview" in the *NGS Modules Overview*.

(I3DL2) Reverb DSP Effect Module

(I3DL2) Reverb Defines

(I3DL2) Reverb Defines Table

Define	Value	Description
SCE_NGS_REVERB_ID	(0x5CE7)	Specifies the Reverb ID.
SCE_NGS_REVERB_ROOM1_LEFT	0	Room template 1, left early reflections pattern.
SCE_NGS_REVERB_ROOM1_RIGHT	1	Room template 1, right early reflections pattern.
SCE_NGS_REVERB_ROOM2_LEFT	2	Room template 2, left early reflections pattern.
SCE_NGS_REVERB_ROOM2_RIGHT	3	Room template 2, right early reflections pattern.
SCE_NGS_REVERB_ROOM3_LEFT	4	Room template 3, left early reflections pattern.
SCE_NGS_REVERB_ROOM3_RIGHT	5	Room template 3, right early reflections pattern.
SCE_NGS_REVERB_PRESET_DEFAULT	(0)	Reverb Module default preset value.
SCE_NGS_REVERB_PRESET_GENERIC	(1)	Reverb Module generic preset value.
SCE_NGS_REVERB_PRESET_PADDED_CELL	(2)	Reverb Module padded cell preset value.
SCE_NGS_REVERB_PRESET_ROOM	(3)	Reverb Module room preset value.
SCE_NGS_REVERB_PRESET_BATHROOM	(4)	Reverb Module bathroom preset value.
SCE_NGS_REVERB_PRESET_LIVING_ROOM	(5)	Reverb Module living room preset value.
SCE_NGS_REVERB_PRESET_STONE_ROOM	(6)	Reverb Module stone room preset value.
SCE_NGS_REVERB_PRESET_AUDITORIUM	(7)	Reverb Module auditorium preset value.
SCE_NGS_REVERB_PRESET_CONCERT_HALL	(8)	Reverb Module hall preset value.
SCE_NGS_REVERB_PRESET_CAVE	(9)	Reverb Module cave preset value.
SCE_NGS_REVERB_PRESET_ARENA	(10)	Reverb Module arena preset value.
SCE_NGS_REVERB_PRESET_HANGAR	(11)	Reverb Module hangar preset value.
SCE_NGS_REVERB_PRESET_CARPETED_HALLWAY	(12)	Reverb Module carpeted hallway preset value.
SCE_NGS_REVERB_PRESET_HALLWAY	(13)	Reverb Module hallway preset value.
SCE_NGS_REVERB_PRESET_STONE_CORRIDOR	(14)	Reverb Module stone corridor preset value.
SCE_NGS_REVERB_PRESET_ALLEY	(15)	Reverb Module alley preset value.
SCE_NGS_REVERB_PRESET_FOREST	(16)	Reverb Module forest preset value.
SCE_NGS_REVERB_PRESET_CITY	(17)	Reverb Module city preset value.
SCE_NGS_REVERB_PRESET_MOUNTAINS	(18)	Reverb Module mountains preset value.
SCE_NGS_REVERB_PRESET_QUARRY	(19)	Reverb Module quarry preset value.
SCE_NGS_REVERB_PRESET_PLAIN	(20)	Reverb Module plain preset value.
SCE_NGS_REVERB_PRESET_PARKING_LOT	(21)	Reverb Module parking lot preset value.
SCE_NGS_REVERB_PRESET_SEWER_PIPE	(22)	Reverb Module sewer pipe preset value.
SCE_NGS_REVERB_PRESET_UNDERWATER	(23)	Reverb Module underwater preset value.
SCE_NGS_REVERB_PRESET_SMALL_ROOM	(24)	Reverb Module small room preset value.
SCE_NGS_REVERB_PRESET_MEDIUM_ROOM	(25)	Reverb Module medium room preset value.
SCE_NGS_REVERB_PRESET_LARGE_ROOM	(26)	Reverb Module large room preset value.
SCE_NGS_REVERB_PRESET_MEDIUM_HALL	(27)	Reverb Module medium hall preset value.
SCE_NGS_REVERB_PRESET_LARGE_HALL	(28)	Reverb Module large hall preset value.
SCE_NGS_REVERB_PRESET_PLATE	(29)	Reverb Module plate preset value.
SCE_NGS_REVERB_VOICE_REVERB_MODULE	(1)	Reverb Module.

(I3DL2) Reverb Parameter Structure Defines

(I3DL2) Reverb Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_REVERB_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_REVERB_ID,1,1))	Specifies the ID of the Reverb's parameter structure. The ID of the Reverb's parameter structure is specified using the module identifier *SCE_NGS_REVERB_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE7).
SCE_NGS_REVERB_PARAMS_STRUCT_ID_V2	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_REVERB_ID,1,2))	Specifies the ID of the Reverb's parameter structure. The ID is specified using the module identifier *SCE_NGS_REVERB_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01025CE7). Note that this uses the same parameter structure as SCE_NGS_REVERB_PARAMS_STRUCT_ID, though the behavior is modified (see the *NGS Modules Overview* and SceNgsReverbParams structure).

(I3DL2) Reverb Structures

SceNgsReverbParams

Structure required to initialize the NGS I3DL2 Reverb DSP effect Module.

Definition

```
#include <ngs/modules/reverb.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceFloat32 fRoom;
        SceFloat32 fRoomHF;
        SceFloat32 fDecayTime;
        SceFloat32 fDecayHFRatio;
        SceFloat32 fReflections;
        SceFloat32 fReflectionsDelay;
        SceFloat32 fReverb;
        SceFloat32 fReverbDelay;
        SceFloat32 fDiffusion;
        SceFloat32 fDensity;
        SceFloat32 fHFReference;
        SceInt32 eEarlyReflectionPattern[SCE_NGS_MAX_SYSTEM_CHANNELS];
        SceFloat32 fEarlyReflectionScalar;
        SceFloat32 fLFReference;
        SceFloat32 fRoomLF;
        SceFloat32 fDryMB;
} SceNgsReverbParams;
```

Members

desc	Default parameter descriptor.
fRoom	Controls the main output volume of the entire Reverb effect in millibels. Value ranges from -10,000 (Min) to 0 (Max).
fRoomHF	Controls the relative volume of high frequency sounds in millibels at the reference frequency (*fHFReference*). Value ranges from -10,000 (Min) to 0 (Max), where 0 is equal to no colouration.
fDecayTime	Decay time of the late reverberation in seconds. Value ranges from 0.1 (Min) to 20 (Max).
fDecayHFRatio	Ratio of high frequency decay in late reverb relative to low frequency decay at the reference frequency (*fHFReference*). Value ranges from 0.1 (Min) to 2.0 (Max).
fReflections	Volume of early reflections in millibels. Value ranges from -10,000 (Min) to 1,000 (Max).
fReflectionsDelay	Delay from input signal until first reflection in seconds. This is in effect a "pre-delay". Value ranges from 0.0 (Min) to 0.3 (Max).
fReverb	Volume of late reverberation in millibels. Value ranges from -10,000 (Min) to 2,000 (Max).
fReverbDelay	Delay from first reflection until late reverb in seconds. Note: all of the early reflections occur during this time. Value ranges from 0.0 (Min) to 0.1 (Max).
fDiffusion	Echo density in late reverberation, expressed as a percent. Value ranges from 0 (Min) to 100 (Max).
fDensity	Modal density in late reverberation, expressed as a percent. Value ranges from 0 (Min) to 100 (Max).

fHFReference	Reference high frequency in Hertz. Value ranges from 20 (Min) to 20,000 (Max).
eEarlyReflectionPattern	Enumerated pattern for the early reflections. There are several room reflections defined with both left and right variants; these give the ability to create stereo/multichannel reverbs by selecting different patterns per channel.
fEarlyReflectionScalar	Time scaling as a percent for the early reflections pattern. 0% is equal to instant reflections; for example, a single impulse. 100% is maximally spread reflections within the *fReverbDelay* time specified.
fLFReference	Reference low frequency in Hertz. This parameter is used in conjunction with the *fRoomLF* parameter to specify the amount of high pass filtering applied to the input signal. Value ranges from 20 (Min) to 20,000 (Max).
fRoomLF	Relative volume of low frequencies at the specified *fLFReference* frequency, in millibels. Value ranges from -10,000 (Min) to 0 (Max).
fDryMB	Dry volume (in millibels).

Description

Structure required to initialize the NGS I3DL2 Reverb DSP effect Module.

Note that there are two parameter structure defines that may be used interchangeably. `SCE_NGS_REVERB_PARAMS_STRUCT_ID` is provided for backwards compatibility only, and it is recommended that you use the new parameter structure define `SCE_NGS_REVERB_PARAMS_STRUCT_ID_V2`. The new ID fixes the following known issues within the reverb:

- A DC offset that may present in some conditions, based on the values of *fRoomLf* and *fLFReference*.
- An incorrect late reverb volume when `fRoom` was not set at `0mB`.

See Also

"I3DL2 Reverb DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsReverbParamsBlock

Structure required to initialize the parameter block for the NGS Reverb DSP effect Module.

Definition

```
#include <ngs/modules/reverb.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsReverbParams params;
} SceNgsReverbParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Reverb DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Reverb DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock()` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Reverb DSP Effect Module Overview" in the *NGS Modules Overview*.

Signal Generator DSP Effect Module

Signal Generator Defines

Signal Generator Defines Table

Define	Value	Description
SCE_NGS_GENERATOR_ID	(0x5CE8)	Specifies the Signal Generator ID.
SCE_NGS_GENERATOR_DISABLE	(0)	Signal Generator is disabled.
SCE_NGS_GENERATOR_ENABLE	(1)	Signal Generator is enabled.
SCE_NGS_GENERATOR_SINE	(0)	Sine waveform.
SCE_NGS_GENERATOR_TRIANGLE	(1)	Triangle waveform.
SCE_NGS_GENERATOR_SAW	(2)	Saw tooth waveform.
SCE_NGS_GENERATOR_NOISE	(3)	White noise waveform.
SCE_NGS_GENERATOR_NOISE_PSP	(4)	White noise (using the PlayStation®Portable noise algorithm) waveform.
SCE_NGS_GENERATOR_PULSE	(5)	Pulse waveform.

Signal Generator Parameter Structure Defines

Signal Generator Parameter Structure Defines Table

Define	Value	Description
SCE_NGS_GENERATOR_PARAMS_STRUCT_ID	(SCE_NGS_MAKE_PARAMS_ID(SCE_NGS_GENERATOR_ID,1,1))	Specifies the ID of the Signal Generator's parameter structure. The ID of the Signal Generator's parameter structure is specified using the module identifier *SCE_NGS_GENERATOR_ID* and the SCE_NGS_MAKE_PARAMS_ID macro (0x01015CE8).

Signal Generator Enumerations

SceNgsGeneratorSettings

Structure required to initialize the NGS Signal Generator DSP effect Module.

Definition

```
#include <ngs/modules/signal_generator.h>
typedef struct {
        SceUInt32 eGeneratorMode;
        SceInt32 nFrequency;
        SceFloat32 fAmplitude;
        SceFloat32 fPulseWidth;
        SceUInt32 uSampleOffset;
        SceUInt32 uPhaseAngle;
} SceNgsGeneratorSettings;
```

Members

eGeneratorMode	Waveform type to generate. Sine, Triangle, Saw, Noise and Pulse waveforms are available; for example, `SCE_NGS_GENERATOR_SINE`. See Signal Generator defines for more information.
nFrequency	Playback frequency of waveform in Hz.
fAmplitude	Amplitude of waveform. 1 = full amplitude (without clipping). 0.5 = half amplitude.
fPulseWidth	Pulse with (if pulse waveform is selected).
uSampleOffset	Offset through waveform to start generation (for example, allows sine waves to be generated as cosine waves).
uPhaseAngle	Start phase offset for sinewave generation (range is 0 to 360 specified as an angle in degrees).

Description

Structure required to initialize the NGS Signal Generator DSP effect Module.

See Also

"Signal Generator DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsGeneratorParams

Structure required to initialize the NGS Signal Generator DSP effect Module.

Definition

```
#include <ngs/modules/signal_generator.h>
typedef struct {
        SceNgsParamsDescriptor desc;
        SceNgsGeneratorSettings generator;
} SceNgsGeneratorParams;
```

Members

desc	Default parameter descriptor.
generator	Generator information (see SceNgsGeneratorSettings).

Description

Structure required to initialize the NGS Signal Generator DSP effect Module.

See Also

"Signal Generator DSP Effect Module Overview" in the *NGS Modules Overview*.

SceNgsGeneratorParamsBlock

Structure required to initialize the parameter block for the NGS Signal Generator DSP effect Module.

Definition

```
#include <ngs/modules/signal_generator.h>
typedef struct {
        SceNgsModuleParamHeader moduleInfo;
        SceNgsGeneratorParams params;
} SceNgsGeneratorParamsBlock;
```

Members

moduleInfo	The Module param header that defines the Module and channel that will be affected (see `SceNgsModuleParamHeader`).
params	The parameter structure including the data used to initialize the Signal Generator DSP effect Module.

Description

Structure required to initialize the parameter block for the NGS Signal Generator DSP effect Module.

Using `ParamsBlock` is the recommended method for changing a block load of data; for example, at initialization. This saves on having to run through a lot of unlock and lock procedures, therefore saving on processing time.

Firstly, create or load your `ParamsBlock` structure, then call the `sceNgsVoiceSetParamsBlock()` function to set the block. Note that to setup multiple Modules each data block should be contiguous within the same memory block.

For information on setting small amounts of data, see About DSP Effects.

See Also

"Signal Generator DSP Effect Module Overview" in the *NGS Modules Overview*.

Output Module

About Output Module

Introduction

The output module allows the user to retrieve PCM data from NGS. This can then be passed to an output library.

Using Output Module

The data is retrieved via the `sceNgsVoiceGetStateData()` function.

The data returned from the Output Module is a single buffer of `short` (16-bit) sample data. For stereo instances this is interleaved per sample.

The size of the buffer can be determined as: size in bytes = `sizeof(short)` * *system granularity* * *number of channels*.

Where, *system granularity* is the NGS system granularity (see `SceNgsSystemInitParams` structure in the *NGS Reference*), and *number of channels* is the number of channels in the voice (see `SceNgsRackDescription` in the *NGS Reference*).

Module Parameters

Module Parameter Ranges

Introduction

The following table summarizes the parameter ranges for each module.

Module/Structure	Members	Minimum	Maximum	Unit
AT9 Player				
SceNgsAT9BufferParams				
	pBuffer	0	max uint32	address
	nNumBytes	0	max int32	bytes
	nLoopCount	SCE_NGS_AT9_LOOP_CONTINUOUS (-1)	max int16	-
	nNextBuff	SCE_NGS_AT9_NO_NEXT_BUFFER (-1)	SCE_NGS_AT9_MAX_BUFFERS-1 (3)	index
	nSamplesDiscardStart	0	2048	samples
	nSamplesDiscardEnd	0	2048	samples
SceNgsAT9Params				
	fPlaybackFrequency	0	192000	Hz
	fPlaybackScalar	0	10	linear multiplier
	nLeadInSamples	0	max int32	samples
	nLimitNumberOfSamplesPlayed	0	max int32	samples
	nChannels	1	SCE_NGS_MAX_SYSTEM_CHANNELS (2)	-
	nChannelMap[0]	SCE_NGS_AT9_NULL_CHANNEL (-1)	SCE_NGS_MAX_SYSTEM_CHANNELS-1 (1)	index
	nChannelMap[1]	SCE_NGS_AT9_NULL_CHANNEL (-1)	SCE_NGS_MAX_SYSTEM_CHANNELS-1 (1)	index
	reserved	0	1	boolean
	configData	0	max uint32	-

Module/Structure	Members	Minimum	Maximum	Unit
Compressor				
SceNgsCompressorParams				
	fRatio	0	1	ratio
	fThreshold	-90	0	dB
	fAttack	0.001	10	seconds
	fRelease	0.001	20	seconds
	fMakeupGain	-90	24	dB
	nStereoLink	SCE_NGS_COMPRESSOR_STEREO_LINK_OFF (0)	SCE_NGS_COMPRESSOR_STEREO_LINK_ON (1)	mode
	nPeakMode	SCE_NGS_COMPRESSOR_RMS_MODE (0)	SCE_NGS_COMPRESSOR_PEAK_MODE (1)	mode
	fSoftKnee	0	10	dB
Delay				
SceNgsDelayTap				
	fDelay Millisecs	0	1000	milliseconds
	fVolume	-1	1	linear multiplier
	fFeedback	-0.95	0.95	linear multiplier
	eFilterMode	SCE_NGS_DELAY_FILTER_MODE_OFF (0)	SCE_NGS_DELAY_FILTER_MODE_ALLPASS (3)	mode
	fCutoff	20	23500	Hz
	fPhaseOffset Deg	-180	360	degrees
	fModWidth Millisecs	0	500	milliseconds
SceNgsDelayParams				
	fDryVol	0	1	linear multiplier
	fModRate	1	20	Hz
Distortion				
SceNgsDistortionParams				
	fA	0	10	-
	fB	0	10	-
	fClip	0	4	linear volume
	fGate	0	4	linear volume
	fWetGain	0	4	linear volume
	fDryGain	0	4	linear volume

Module/Structure	Members	Minimum	Maximum	Unit
Envelope				
SceNgsEnvelopePoint				
	uMsecsToNextPoint	0	max uint32	milliseconds
	fAmplitude	0	100	linear volume
	eCurveType	SCE_NGS_ENVELOPE_LINEAR (0)	SCE_NGS_ENVELOPE_CURVED (1)	mode
SceNgsEnvelopeParams				
	uReleaseMsecs	0	max uint32	milliseconds
	uNumPoints	0	SCE_NGS_ENVELOPE_MAX_POINTS (4)	-
	uLoopStart	0	SCE_NGS_ENVELOPE_MAX_POINTS-1 (3)	index
	nLoopEnd	SCE_NGS_ENVELOPE_NO_LOOP_RELEASE (-3)	SCE_NGS_ENVELOPE_MAX_POINTS-1 (3)	index
Filter				
SceNgsFilterParams				
	eFilterMode	SCE_NGS_FILTER_MODE_OFF (0)	SCE_NGS_FILTER_LOWPASS_RESONANT_NORMALIZED (12)	mode
	fFrequency	20	23500	Hz
	fResonance	0.5	10	Q factor
	fGain	-90	24	dB
SceNgsFilterParamsCoEff				
	fB0	-127	127	
	fB1	-127	127	
	fB2	-127	127	
	fA1	-127	127	
	fA2	-127	127	
Mixer				
SceNgsMixerParams				
	fGainIn[0]	0	8	linear multiplier
	fGainIn[1]	0	8	linear multiplier
	uPauseLeadIn	0	256	samples
	uPauseLeadOut	0	256	samples

Module/Structure	Members	Minimum	Maximum	Unit
Parametric EQ				
SceNgsParamEqFilter				
	eFilterMode	SCE_NGS_FILTER_MODE_OFF (0)	SCE_NGS_FILTER_LOWPASS_RESONANT_NORMALIZED (12)	mode
	fFrequency	20	23500	Hz
	fResonance	0.5	10	Q factor
	fGain	-90	24	dB
SceNgsParamEqCoEff				
	fB0	-127	127	
	fB1	-127	127	
	fB2	-127	127	
	fA1	-127	127	
	fA2	-127	127	
Pauser				
SceNgsPauserParams				
	uPauseLeadIn	0	256	samples
	uPauseLeadOut	0	256	samples
Player				
SceNgsPlayerBufferParams				
	pBuffer	0	max uint32	address
	nNumBytes	0	max int32	bytes
	nLoopCount	SCE_NGS_PLAYER_LOOP_CONTINUOUS (-1)	max int16	-
	nNextBuff	SCE_NGS_PLAYER_NO_NEXT_BUFFER (-1)	SCE_NGS_PLAYER_MAX_BUFFERS-1 (3)	index

Module/Structure	Members	Minimum	Maximum	Unit
Player (cont.)				
SceNgsPlayerParams				
	fPlayback Frequency	0	192000	Hz
	fPlayback Scalar	0	10	linear multiplier
	nLeadIn Samples	0	max `int32`	samples
	nLimit NumberOf SamplesPlayed	0	max `int32`	samples
	nStartByte	0	max `int32`	byte
	nChannels	1	`SCE_NGS_MAX_SYSTEM_CHANNELS` (2)	-
	nChannel Map`[0]`	`SCE_NGS_PLAYER_NULL_CHANNEL` (-1)	`SCE_NGS_MAX_SYSTEM_CHANNELS-1` (1)	index
	nChannel Map`[1]`	`SCE_NGS_PLAYER_NULL_CHANNEL` (-1)	`SCE_NGS_MAX_SYSTEM_CHANNELS-1` (1)	index
	nType	`SCE_NGS_PLAYER_TYPE_PCM` (0)	`SCE_NGS_PLAYER_TYPE_ADPCM` (1)	mode
	reserved	0	1	boolean
	nStartBuffer	0	`SCE_NGS_PLAYER_MAX_BUFFERS-1` (3)	index

Module/Structure	Members	Minimum	Maximum	Unit
Reverb				
SceNgsReverbParams				
	fRoom	-10000	0	milliBel
	fRoomHF	-10000	0	milliBel
	fDecayTime	0.1	20	seconds
	fDecayHFRatio	0.1	2	ratio
	fReflections	-10000	1000	milliBel
	fReflections Delay	0	0.3	seconds
	fReverb	-10000	2000	milliBel
	fReverbDelay	0	0.1	seconds
	fDiffusion	0	100	percent
	fDensity	0	100	percent
	fHFReference	20	20000	Hz
	eEarly Reflection Pattern[0]	SCE_NGS_REVERB_ROOM1_LEFT (0)	SCE_NGS_REVERB_ROOM3_RIGHT (5)	index
	eEarly Reflection Pattern[1]	SCE_NGS_REVERB_ROOM1_LEFT (0)	SCE_NGS_REVERB_ROOM3_RIGHT (5)	index
	fEarly Reflection Scalar	0	100	percent
	fLFReference	20	20000	Hz
	fRoomLF	-10000	0	milliBel
	fDryMB	-10000	0	milliBel
Signal Generator				
SceNgsGeneratorSettings				
	eGenerator Mode	SCE_NGS_GENERATOR_SINE (0)	SCE_NGS_GENERATOR_PULSE (5)	mode
	nFrequency	0	12000	Hz
	fAmplitude	0	1	linear multiplier
	fPulseWidth	0	1	normalized amount
	uSampleOffset	0	max uint32	samples
	uPhaseAngle	0	360	degrees