

Overly Complicated Audio Engine

Generated by Doxygen 1.8.13

Mon Dec 23 2019 16:11:58

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	OCAE::Modifier::ADSR Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Member Enumeration Documentation	9
4.1.2.1	State	9
4.1.3	Constructor & Destructor Documentation	9
4.1.3.1	ADSR() [1/3]	9
4.1.3.2	ADSR() [2/3]	9
4.1.3.3	ADSR() [3/3]	10
4.1.4	Member Function Documentation	10
4.1.4.1	CreateMethodList()	10
4.1.4.2	IsBase()	11
4.1.4.3	operator=() [1/2]	11
4.1.4.4	operator=() [2/2]	11

4.1.4.5	Process()	12
4.1.4.6	Release()	12
4.2	OCAE::Modifier::BandPass Class Reference	12
4.2.1	Detailed Description	14
4.2.2	Constructor & Destructor Documentation	14
4.2.2.1	BandPass() [1/3]	14
4.2.2.2	BandPass() [2/3]	14
4.2.2.3	BandPass() [3/3]	15
4.2.3	Member Function Documentation	15
4.2.3.1	CreateMethodList()	15
4.2.3.2	GetFrequency()	16
4.2.3.3	GetQuality()	16
4.2.3.4	IsBase()	16
4.2.3.5	operator=() [1/2]	16
4.2.3.6	operator=() [2/2]	17
4.2.3.7	Process()	17
4.2.3.8	Reset()	18
4.2.3.9	SetFrequency()	18
4.2.3.10	SetQuality()	18
4.3	OCAE::Sound::Block Class Reference	18
4.3.1	Detailed Description	19
4.3.2	Constructor & Destructor Documentation	20
4.3.2.1	Block()	20
4.3.3	Member Function Documentation	20
4.3.3.1	GetGenerator() [1/2]	20
4.3.3.2	GetGenerator() [2/2]	20
4.3.3.3	GetModifier() [1/2]	21
4.3.3.4	GetModifier() [2/2]	21

4.3.3.5	PrimeInput()	21
4.3.3.6	Process()	22
4.4	OCAE::Modifier::Delay Class Reference	22
4.4.1	Detailed Description	23
4.4.2	Constructor & Destructor Documentation	23
4.4.2.1	Delay() [1/3]	23
4.4.2.2	Delay() [2/3]	24
4.4.2.3	Delay() [3/3]	24
4.4.3	Member Function Documentation	24
4.4.3.1	CreateMethodList()	24
4.4.3.2	GetDelay()	25
4.4.3.3	IsBase()	25
4.4.3.4	operator=() [1/2]	25
4.4.3.5	operator=() [2/2]	26
4.4.3.6	Process()	26
4.4.3.7	SetDelay()	26
4.5	OCAE::Core::Driver Class Reference	27
4.5.1	Detailed Description	28
4.5.2	Constructor & Destructor Documentation	28
4.5.2.1	Driver() [1/3]	28
4.5.2.2	Driver() [2/3]	29
4.5.2.3	~Driver()	29
4.5.2.4	Driver() [3/3]	29
4.5.3	Member Function Documentation	29
4.5.3.1	AddSound()	29
4.5.3.2	Create()	30
4.5.3.3	GetID()	30
4.5.3.4	GetOutputTrack()	31

4.5.3.5	<code>operator=()</code> [1/2]	31
4.5.3.6	<code>operator=()</code> [2/2]	31
4.5.3.7	<code>Process()</code>	32
4.5.3.8	<code>RemoveSound()</code>	32
4.5.3.9	<code>SetGain()</code>	32
4.6	OCAE::Modifier::Echo Class Reference	32
4.6.1	Detailed Description	34
4.6.2	Constructor & Destructor Documentation	34
4.6.2.1	<code>Echo()</code> [1/3]	34
4.6.2.2	<code>Echo()</code> [2/3]	34
4.6.2.3	<code>Echo()</code> [3/3]	35
4.6.3	Member Function Documentation	35
4.6.3.1	<code>CreateMethodList()</code>	35
4.6.3.2	<code>GetDecayRatio()</code>	35
4.6.3.3	<code>IsBase()</code>	36
4.6.3.4	<code>operator=()</code> [1/2]	36
4.6.3.5	<code>operator=()</code> [2/2]	36
4.6.3.6	<code>Process()</code>	37
4.6.3.7	<code>SetDecayRatio()</code>	37
4.7	OCAE::Modifier::EnvelopeFollower Class Reference	37
4.7.1	Detailed Description	39
4.7.2	Constructor & Destructor Documentation	39
4.7.2.1	<code>EnvelopeFollower()</code> [1/3]	39
4.7.2.2	<code>EnvelopeFollower()</code> [2/3]	39
4.7.2.3	<code>~EnvelopeFollower()</code>	40
4.7.2.4	<code>EnvelopeFollower()</code> [3/3]	40
4.7.3	Member Function Documentation	40
4.7.3.1	<code>CreateMethodList()</code>	40

4.7.3.2	IsBase()	41
4.7.3.3	operator=() [1/2]	41
4.7.3.4	operator=() [2/2]	41
4.7.3.5	Process()	42
4.8	OCAE::Modifier::Equalizer Class Reference	42
4.8.1	Detailed Description	43
4.8.2	Constructor & Destructor Documentation	44
4.8.2.1	Equalizer() [1/3]	44
4.8.2.2	Equalizer() [2/3]	44
4.8.2.3	~Equalizer()	44
4.8.2.4	Equalizer() [3/3]	44
4.8.3	Member Function Documentation	45
4.8.3.1	CreateMethodList()	45
4.8.3.2	GetGain()	45
4.8.3.3	IsBase()	46
4.8.3.4	operator=() [1/2]	46
4.8.3.5	operator=() [2/2]	46
4.8.3.6	Process()	47
4.8.3.7	SetGain()	47
4.9	OCAE::Modifier::Gain Class Reference	47
4.9.1	Detailed Description	49
4.9.2	Constructor & Destructor Documentation	49
4.9.2.1	Gain() [1/3]	49
4.9.2.2	Gain() [2/3]	49
4.9.2.3	~Gain()	50
4.9.2.4	Gain() [3/3]	50
4.9.3	Member Function Documentation	50
4.9.3.1	CreateMethodList()	50

4.9.3.2	GetGain()	51
4.9.3.3	IsBase()	51
4.9.3.4	operator=() [1/2]	51
4.9.3.5	operator=() [2/2]	52
4.9.3.6	Process()	52
4.9.3.7	SetGain()	52
4.10	OCAE::Generator::GeneratorBase Class Reference	53
4.10.1	Detailed Description	54
4.10.2	Constructor & Destructor Documentation	54
4.10.2.1	GeneratorBase() [1/3]	54
4.10.2.2	GeneratorBase() [2/3]	54
4.10.2.3	~GeneratorBase()	55
4.10.2.4	GeneratorBase() [3/3]	55
4.10.3	Member Function Documentation	55
4.10.3.1	CreateMethodList()	55
4.10.3.2	IsBase()	56
4.10.3.3	operator=() [1/2]	56
4.10.3.4	operator=() [2/2]	56
4.10.3.5	Process()	57
4.11	OCAE::Generator::GeneratorFactory Class Reference	57
4.11.1	Detailed Description	58
4.11.2	Member Function Documentation	58
4.11.2.1	CreateBase()	58
4.11.2.2	CreateNoise()	59
4.11.2.3	CreateSawtooth()	59
4.11.2.4	CreateSine()	59
4.11.2.5	CreateSquare()	60
4.11.2.6	CreateTriangle()	60

4.11.2.7	CreateWAV() [1/3]	60
4.11.2.8	CreateWAV() [2/3]	61
4.11.2.9	CreateWAV() [3/3]	61
4.12	OCAE::Modifier::GenericFilter Class Reference	61
4.12.1	Detailed Description	63
4.12.2	Constructor & Destructor Documentation	63
4.12.2.1	GenericFilter() [1/3]	63
4.12.2.2	GenericFilter() [2/3]	63
4.12.2.3	~GenericFilter()	64
4.12.2.4	GenericFilter() [3/3]	64
4.12.3	Member Function Documentation	64
4.12.3.1	CreateMethodList()	64
4.12.3.2	IsBase()	65
4.12.3.3	operator=() [1/2]	65
4.12.3.4	operator=() [2/2]	65
4.12.3.5	Process()	66
4.13	OCAE::Modifier::LowPass Class Reference	66
4.13.1	Detailed Description	68
4.13.2	Constructor & Destructor Documentation	68
4.13.2.1	LowPass() [1/3]	68
4.13.2.2	LowPass() [2/3]	68
4.13.2.3	~LowPass()	68
4.13.2.4	LowPass() [3/3]	69
4.13.3	Member Function Documentation	69
4.13.3.1	CreateMethodList()	69
4.13.3.2	IsBase()	69
4.13.3.3	operator=() [1/2]	69
4.13.3.4	operator=() [2/2]	70

4.13.3.5	Process()	70
4.13.3.6	Reset()	71
4.13.3.7	SetCutoff()	71
4.13.3.8	SetResonance()	71
4.14	OCAE::Tools::MethodTable Class Reference	71
4.14.1	Detailed Description	73
4.14.2	Constructor & Destructor Documentation	74
4.14.2.1	MethodTable() [1/2]	74
4.14.2.2	MethodTable() [2/2]	74
4.14.2.3	~MethodTable()	74
4.14.3	Member Function Documentation	74
4.14.3.1	CallMethod()	74
4.14.3.2	CreateMethodList()	75
4.14.3.3	RegisterMethod()	75
4.14.3.4	RegisterMethods()	76
4.15	OCAE::Modifier::ModifierBase Class Reference	76
4.15.1	Detailed Description	78
4.15.2	Constructor & Destructor Documentation	78
4.15.2.1	ModifierBase() [1/3]	78
4.15.2.2	ModifierBase() [2/3]	78
4.15.2.3	~ModifierBase()	79
4.15.2.4	ModifierBase() [3/3]	79
4.15.3	Member Function Documentation	79
4.15.3.1	CreateMethodList()	79
4.15.3.2	IsBase()	80
4.15.3.3	operator=() [1/2]	80
4.15.3.4	operator=() [2/2]	80
4.15.3.5	Process()	81

4.16	OCAE::Modifier::ModifierFactory Class Reference	81
4.16.1	Detailed Description	82
4.16.2	Constructor & Destructor Documentation	82
4.16.2.1	~ModifierFactory()	82
4.16.3	Member Function Documentation	83
4.16.3.1	CreateADSR()	83
4.16.3.2	CreateBandPass()	83
4.16.3.3	CreateBase()	84
4.16.3.4	CreateDelay()	84
4.16.3.5	CreateEcho()	84
4.16.3.6	CreateEnvelopeFollower()	85
4.16.3.7	CreateEqualizer()	85
4.16.3.8	CreateGain()	86
4.16.3.9	CreateGenericFilter()	86
4.16.3.10	CreateLowPass()	86
4.17	OCAE::Generator::Noise Class Reference	87
4.17.1	Detailed Description	88
4.17.2	Constructor & Destructor Documentation	88
4.17.2.1	Noise() [1/3]	88
4.17.2.2	Noise() [2/3]	89
4.17.2.3	~Noise()	89
4.17.2.4	Noise() [3/3]	89
4.17.3	Member Function Documentation	89
4.17.3.1	CreateMethodList()	89
4.17.3.2	IsBase()	90
4.17.3.3	operator=() [1/2]	90
4.17.3.4	operator=() [2/2]	90
4.17.3.5	Process()	91

4.18	OCAE::Tools::Resampler Class Reference	91
4.18.1	Detailed Description	92
4.18.2	Constructor & Destructor Documentation	92
4.18.2.1	Resampler()	92
4.18.3	Member Function Documentation	93
4.18.3.1	Process()	93
4.18.3.2	SetPlaybackSpeed()	93
4.19	OCAE::Generator::Sawtooth Class Reference	93
4.19.1	Detailed Description	94
4.19.2	Constructor & Destructor Documentation	95
4.19.2.1	Sawtooth() [1/3]	95
4.19.2.2	Sawtooth() [2/3]	95
4.19.2.3	~Sawtooth()	95
4.19.2.4	Sawtooth() [3/3]	95
4.19.3	Member Function Documentation	96
4.19.3.1	CreateMethodList()	96
4.19.3.2	GetFrequency()	96
4.19.3.3	IsBase()	97
4.19.3.4	operator=() [1/2]	97
4.19.3.5	operator=() [2/2]	97
4.19.3.6	Process()	98
4.19.3.7	SetFrequency()	98
4.20	OCAE::Generator::Sine Class Reference	98
4.20.1	Detailed Description	100
4.20.2	Constructor & Destructor Documentation	100
4.20.2.1	Sine() [1/3]	100
4.20.2.2	Sine() [2/3]	100
4.20.2.3	~Sine()	101

4.20.2.4	Sine() [3/3]	101
4.20.3	Member Function Documentation	101
4.20.3.1	CreateMethodList()	101
4.20.3.2	GetFrequency()	102
4.20.3.3	IsBase()	102
4.20.3.4	operator=() [1/2]	102
4.20.3.5	operator=() [2/2]	103
4.20.3.6	Process()	103
4.20.3.7	SetFrequency()	103
4.20.3.8	SetupWaveTable()	104
4.21	OCAE::Sound::Sound Class Reference	104
4.21.1	Detailed Description	106
4.21.2	Constructor & Destructor Documentation	106
4.21.2.1	Sound() [1/3]	106
4.21.2.2	Sound() [2/3]	106
4.21.2.3	Sound() [3/3]	106
4.21.3	Member Function Documentation	107
4.21.3.1	AddConnection()	107
4.21.3.2	GetInputBlock()	107
4.21.3.3	GetOutputBlock()	107
4.21.3.4	operator=() [1/2]	108
4.21.3.5	operator=() [2/2]	108
4.21.3.6	Pause()	108
4.21.3.7	PrepareGraph()	109
4.21.3.8	Process()	110
4.21.3.9	ProcessOrder()	110
4.21.3.10	Register()	110
4.21.3.11	RemoveConnection()	111

4.21.3.12 SetInputGain()	111
4.21.3.13 SetOutputGain()	111
4.21.3.14 Unpause()	112
4.21.3.15 Unregister()	112
4.22 OCAE::Sound::SoundFactory Class Reference	112
4.22.1 Detailed Description	113
4.22.2 Member Function Documentation	113
4.22.2.1 CreateBasicGenerator()	113
4.22.2.2 CreateBasicModifier()	113
4.22.2.3 CreateBlock() [1/4]	114
4.22.2.4 CreateBlock() [2/4]	114
4.22.2.5 CreateBlock() [3/4]	115
4.22.2.6 CreateBlock() [4/4]	115
4.22.2.7 CreateEmptySound()	116
4.23 OCAE::Generator::Square Class Reference	116
4.23.1 Detailed Description	117
4.23.2 Constructor & Destructor Documentation	118
4.23.2.1 Square() [1/3]	118
4.23.2.2 Square() [2/3]	118
4.23.2.3 ~Square()	118
4.23.2.4 Square() [3/3]	118
4.23.3 Member Function Documentation	119
4.23.3.1 CreateMethodList()	119
4.23.3.2 GetFrequency()	119
4.23.3.3 IsBase()	120
4.23.3.4 operator=() [1/2]	120
4.23.3.5 operator=() [2/2]	120
4.23.3.6 Process()	121

4.23.3.7	SetFrequency()	121
4.24	OCAE::Generator::Triangle Class Reference	121
4.24.1	Detailed Description	123
4.24.2	Constructor & Destructor Documentation	123
4.24.2.1	Triangle() [1/3]	123
4.24.2.2	Triangle() [2/3]	123
4.24.2.3	~Triangle()	124
4.24.2.4	Triangle() [3/3]	124
4.24.3	Member Function Documentation	124
4.24.3.1	CreateMethodList()	124
4.24.3.2	GetFrequency()	125
4.24.3.3	IsBase()	125
4.24.3.4	operator=() [1/2]	125
4.24.3.5	operator=() [2/2]	126
4.24.3.6	Process()	126
4.24.3.7	SetFrequency()	126
4.25	OCAE::Generator::WAV Class Reference	127
4.25.1	Detailed Description	128
4.25.2	Constructor & Destructor Documentation	128
4.25.2.1	WAV() [1/6]	128
4.25.2.2	WAV() [2/6]	128
4.25.2.3	~WAV()	129
4.25.2.4	WAV() [3/6]	129
4.25.2.5	WAV() [4/6]	129
4.25.2.6	WAV() [5/6]	129
4.25.2.7	WAV() [6/6]	130
4.25.3	Member Function Documentation	130
4.25.3.1	CreateMethodList()	130
4.25.3.2	IsBase()	131
4.25.3.3	LoadWAV()	131
4.25.3.4	operator=() [1/2]	131
4.25.3.5	operator=() [2/2]	132
4.25.3.6	ParseWAV()	132
4.25.3.7	Process()	132
4.25.3.8	ReadFile()	133
4.26	OCAE::Tools::WAVHeader Struct Reference	133
4.26.1	Detailed Description	134
4.26.2	Constructor & Destructor Documentation	134
4.26.2.1	WAVHeader()	134

5	File Documentation	135
5.1	ADSR.hpp File Reference	135
5.1.1	Detailed Description	135
5.2	BandPass.hpp File Reference	136
5.2.1	Detailed Description	136
5.3	Block.hpp File Reference	136
5.3.1	Detailed Description	137
5.4	Core.hpp File Reference	137
5.4.1	Detailed Description	137
5.5	Delay.hpp File Reference	138
5.5.1	Detailed Description	138
5.6	Driver.hpp File Reference	138
5.6.1	Detailed Description	139
5.6.2	Function Documentation	139
5.6.2.1	OCAE_TYPEDEF_SHARED()	139
5.7	Echo.hpp File Reference	139
5.7.1	Detailed Description	140
5.8	Engine.hpp File Reference	140
5.8.1	Detailed Description	141
5.9	Envelope.hpp File Reference	141
5.9.1	Detailed Description	141
5.10	Equalizer.hpp File Reference	142
5.10.1	Detailed Description	142
5.11	Gain.hpp File Reference	142
5.11.1	Detailed Description	143
5.11.2	Function Documentation	143
5.11.2.1	OCAE_TYPEDEF_SHARED()	143
5.12	GeneratorBase.hpp File Reference	144

5.12.1 Detailed Description	144
5.13 GeneratorFactory.hpp File Reference	144
5.13.1 Detailed Description	145
5.14 Generators.hpp File Reference	145
5.14.1 Detailed Description	146
5.15 GenericFilter.hpp File Reference	146
5.15.1 Detailed Description	146
5.16 Input.hpp File Reference	147
5.16.1 Detailed Description	147
5.16.2 Function Documentation	147
5.16.2.1 GetOption()	147
5.16.2.2 InitOptions()	148
5.17 LowPass.hpp File Reference	148
5.17.1 Detailed Description	148
5.18 Macro.hpp File Reference	149
5.18.1 Detailed Description	151
5.19 MethodTable.hpp File Reference	151
5.19.1 Detailed Description	151
5.20 ModifierBase.hpp File Reference	152
5.20.1 Detailed Description	152
5.21 ModifierFactory.hpp File Reference	152
5.21.1 Detailed Description	153
5.21.2 Function Documentation	154
5.21.2.1 OCAE_TYPEDEF_SHARED()	154
5.22 Modifiers.hpp File Reference	154
5.22.1 Detailed Description	154
5.23 Noise.hpp File Reference	155
5.23.1 Detailed Description	155

5.24 Resampler.hpp File Reference	155
5.24.1 Detailed Description	156
5.25 Sawtooth.hpp File Reference	156
5.25.1 Detailed Description	157
5.26 Sine.hpp File Reference	157
5.26.1 Detailed Description	157
5.27 Sound.hpp File Reference	158
5.27.1 Detailed Description	158
5.27.2 Function Documentation	158
5.27.2.1 OCAE_TYPEDEF_SHARED() [1/2]	159
5.27.2.2 OCAE_TYPEDEF_SHARED() [2/2]	159
5.28 SoundFactory.hpp File Reference	159
5.28.1 Detailed Description	159
5.29 Sounds.hpp File Reference	160
5.29.1 Detailed Description	160
5.30 Square.hpp File Reference	160
5.30.1 Detailed Description	161
5.31 Tools.hpp File Reference	161
5.31.1 Detailed Description	161
5.32 Triangle.hpp File Reference	162
5.32.1 Detailed Description	162
5.33 Types.hpp File Reference	162
5.33.1 Detailed Description	163
5.34 Util.hpp File Reference	163
5.34.1 Detailed Description	164
5.34.2 Function Documentation	164
5.34.2.1 Left() [1/2]	164
5.34.2.2 Left() [2/2]	165
5.34.2.3 Right() [1/2]	165
5.34.2.4 Right() [2/2]	166
5.35 WAV.hpp File Reference	166
5.35.1 Detailed Description	167
5.36 WAVHeader.hpp File Reference	167
5.36.1 Detailed Description	167
5.37 WAVWriter.hpp File Reference	168
5.37.1 Detailed Description	168
5.37.2 Function Documentation	168
5.37.2.1 WriteWAV()	168

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

OCAE::Sound::Block	18
OCAE::Core::Driver	27
OCAE::Generator::GeneratorFactory	57
OCAE::Tools::MethodTable	71
OCAE::Generator::GeneratorBase	53
OCAE::Generator::Noise	87
OCAE::Generator::Sawtooth	93
OCAE::Generator::Sine	98
OCAE::Generator::Square	116
OCAE::Generator::Triangle	121
OCAE::Generator::WAV	127
OCAE::Modifier::ModifierBase	76
OCAE::Modifier::ADSR	7
OCAE::Modifier::BandPass	12
OCAE::Modifier::Delay	22
OCAE::Modifier::Echo	32
OCAE::Modifier::EnvelopeFollower	37
OCAE::Modifier::Equalizer	42
OCAE::Modifier::Gain	47
OCAE::Modifier::GenericFilter	61
OCAE::Modifier::LowPass	66
OCAE::Modifier::ModifierFactory	81
OCAE::Tools::Resampler	91
OCAE::Sound::Sound	104
OCAE::Sound::SoundFactory	112
OCAE::Tools::WAVHeader	133

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

OCAE::Modifier::ADSR	
Attack - Decay - Sustain - Release filter	7
OCAE::Modifier::BandPass	
Bandpass filter	12
OCAE::Sound::Block	
This class defines a way of holding a Generator, Modifier and a method of combining the outputs of both of them to produce a single output sample	18
OCAE::Modifier::Delay	
Delay filter	22
OCAE::Core::Driver	
Handles the calculation of audio samples from different Sounds	27
OCAE::Modifier::Echo	
Echo IIR filter. Uses output sample for echoing instead of input, creating an infinite impulse response (IIR)	32
OCAE::Modifier::EnvelopeFollower	
Envelope follower filter. Calculates the gain of the input signal over time	37
OCAE::Modifier::Equalizer	
Equalizer filter	42
OCAE::Modifier::Gain	
Simple gain filter for amplifying the input signal. The gain value can be negative allowing for inverting the input signal	47
OCAE::Generator::GeneratorBase	
General base class for all generator (sounds) to inherit from. Any derived classes with extra methods that may need to be acquired can be accessed through their setup of the Tools::MethodTable	53
OCAE::Generator::GeneratorFactory	
Creates pointers to generators handled by std::shared_ptr to prevent memory leaks	57
OCAE::Modifier::GenericFilter	
Generic audio filter with simple poles	61
OCAE::Modifier::LowPass	
3rd Order Butterworth Low Pass filter with resonance	66

OCAE::Tools::MethodTable	
The purpose of this class is to create a simple interface for calling methods from an object of an unknown type	71
OCAE::Modifier::ModifierBase	
The base Modifier class that all modifiers should inherit from	76
OCAE::Modifier::ModifierFactory	
Factory class for constructing audio filters (Modifiers)	81
OCAE::Generator::Noise	
Generates white noise	87
OCAE::Tools::Resampler	
Class for taking audio data of one sampling rate and translating it to another sampling rate	91
OCAE::Generator::Sawtooth	
Generates a sawtooth sound	93
OCAE::Generator::Sine	
Generates sine data at the given frequency	98
OCAE::Sound::Sound	
Class for handling Generator and Modifier objects in a more abstract way in conjunction with a Driver	104
OCAE::Sound::SoundFactory	
Class containing functions that will generate Sound and Block objects from common inputs	112
OCAE::Generator::Square	
Generates square wave data at the given frequency	116
OCAE::Generator::Triangle	
Triangle wave generator	121
OCAE::Generator::WAV	
Plays audio from WAVE data	127
OCAE::Tools::WAVHeader	
A POD structure representing the structure of the header of a WAVE file	133

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

ADSR.hpp	135
BandPass.hpp	136
Block.hpp	136
Core.hpp	137
Delay.hpp	138
Driver.hpp	138
Echo.hpp	139
Engine.hpp	140
Envelope.hpp	141
Equalizer.hpp	142
Gain.hpp	142
GeneratorBase.hpp	144
GeneratorFactory.hpp	144
Generators.hpp	145
GenericFilter.hpp	146
Input.hpp	147
LowPass.hpp	148
Macro.hpp	149
MethodTable.hpp	151
ModifierBase.hpp	152
ModifierFactory.hpp	152
Modifiers.hpp	154
Noise.hpp	155
Resampler.hpp	155
Sawtooth.hpp	156
Sine.hpp	157
Sound.hpp	158
SoundFactory.hpp	159
Sounds.hpp	160
Square.hpp	160
Tools.hpp	161

Triangle.hpp	162
Types.hpp	162
Util.hpp	163
WAV.hpp	166
WAVHeader.hpp	167
WAVWriter.hpp	168

Chapter 4

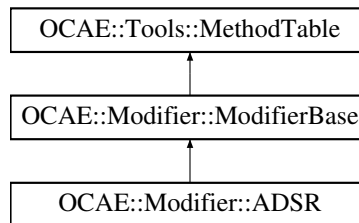
Class Documentation

4.1 OCAE::Modifier::ADSR Class Reference

Attack - Decay - Sustain - Release filter.

```
#include <ADSR.hpp>
```

Inheritance diagram for OCAE::Modifier::ADSR:



Public Member Functions

- **ADSR** (**ADSR** const &other)=delete
Copy constructor. Deleted.
- **ADSR** (**ADSR** &&other)=default
Default move constructor.
- virtual **~ADSR** ()=default
Default destructor.
- **ADSR** & **operator=** (**ADSR** const &rhs)=delete
Copy assignment operator. Deleted.
- **ADSR** & **operator=** (**ADSR** &&rhs)=default
Default move assignment operator.
- void **Release** (void)
Will set the phase to the release phase, regardless of what the current phase is.
- virtual **StereoData Process** (**StereoData** const &input)
Takes input sample and filters it, returning the result.
- virtual bool **IsBase** ()
*Returns boolean for if the object calling this function is a **ModifierBase** or not.*

Protected Member Functions

- [ADSR](#) (uint64_t attack, uint64_t decay, [Math_t](#) sustain, uint64_t release)
Constructor.
- virtual [Tools::MethodTable::MethodList_t](#) [CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Types

- enum [State](#) : int8_t {
 attack, decay, sustain, release,
 invalid = -1 }
Enum for tracking the current state of the [ADSR](#) envelope.

Private Attributes

- [Math_t m_Attack](#)
The rate of change in gain during the attack phase.
- [Math_t m_Decay](#)
The rate of change in gain during the decay phase.
- [Math_t m_Sustain](#)
The gain level during the sustain phase.
- [Math_t m_Release](#)
The rate of change in gain during the release phase.
- [State m_State](#)
The current phase of the envelope.
- [Math_t m_Gain](#)
The current gain value updated during filtering.

Friends

- class [ModifierFactory](#)
Add the factory as a friend so it can construct [ADSR](#) objects.

Additional Inherited Members

4.1.1 Detailed Description

Attack - Decay - Sustain - Release filter.

The most basic filter to create an envelope over a given signal. The filter uses only linear slopes for the attack, decay, and release phases. The filter will only continue to the release phase when the [ADSR::Release](#) method is called.

4.1.2 Member Enumeration Documentation

4.1.2.1 State

```
enum OCAE::Modifier::ADSR::State : int8_t [strong], [private]
```

Enum for tracking the current state of the [ADSR](#) envelope.

```
00048                                     : int8_t
00049     {
00050         attack,
00051         decay,
00052         sustain,
00053         release,
00054         invalid = -1,
00055     };
```

4.1.3 Constructor & Destructor Documentation

4.1.3.1 [ADSR\(\)](#) [1/3]

```
OCAE::Modifier::ADSR::ADSR (
    ADSR const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.1.3.2 [ADSR\(\)](#) [2/3]

```
OCAE::Modifier::ADSR::ADSR (
    ADSR && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.1.3.3 ADSR() [3/3]

```
OCAE::Modifier::ADSR::ADSR (
    uint64_t attack,
    uint64_t decay,
    Math_t sustain,
    uint64_t release ) [protected]
```

Constructor.

Parameters

<i>attack</i>	Time to increase gain from 0 to 1 in samples.
<i>decay</i>	Time to decrease gain from 0 to sustain in samples.
<i>sustain</i>	The gain level of the sustain phase.
<i>release</i>	Time to decrease from sustain to 0 in samples.

4.1.4 Member Function Documentation

4.1.4.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Modifier::ADSR::CreateMethodList ( ) [protected],
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.1.4.2 IsBase()

```
virtual bool OCAE::Modifier::ADSR::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

```
00153 { return false; };
```

4.1.4.3 operator=() [1/2]

```
ADSR& OCAE::Modifier::ADSR::operator= (
    ADSR const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.1.4.4 operator=() [2/2]

```
ADSR& OCAE::Modifier::ADSR::operator= (
    ADSR && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.1.4.5 Process()

```
virtual StereoData OCAE::Modifier::ADSR::Process (
    StereoData const & input ) [virtual]
```

Takes input sample and filters it, returning the result.

Parameters

<i>input</i>	The input sample.
--------------	-------------------

Returns

The filtered sample.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.1.4.6 Release()

```
void OCAE::Modifier::ADSR::Release (
    void )
```

Will set the phase to the release phase, regardless of what the current phase is.

The documentation for this class was generated from the following file:

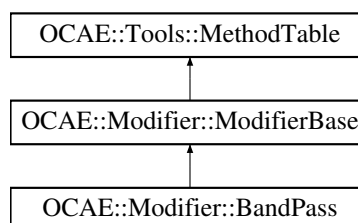
- [ADSR.hpp](#)

4.2 OCAE::Modifier::BandPass Class Reference

Bandpass filter.

```
#include <BandPass.hpp>
```

Inheritance diagram for OCAE::Modifier::BandPass:



Public Member Functions

- [BandPass](#) ([BandPass](#) const &other)=delete
Copy constructor. Deleted.
- [BandPass](#) ([BandPass](#) &&other)=default
Default move constructor.
- virtual [~BandPass](#) ()
Default destructor.
- [BandPass](#) & [operator=](#) ([BandPass](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [BandPass](#) & [operator=](#) ([BandPass](#) &&rhs)=default
Default move assignment operator.
- [Math_t](#) [GetFrequency](#) () const
Returns the central frequency of the filter.
- void [SetFrequency](#) ([Math_t](#) f)
Sets the central frequency of the filter.
- [Math_t](#) [GetQuality](#) () const
Returns the quality of the filter.
- void [SetQuality](#) ([Math_t](#) Q)
Sets the quality of the filter.
- virtual [StereoData](#) [Process](#) ([StereoData](#) const &input)
Takes input sample and filters it, returning the result.
- virtual bool [IsBase](#) ()
Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Protected Member Functions

- [BandPass](#) ([Math_t](#) f, [Math_t](#) Q=1)
Constructor.
- virtual [Tools::MethodTable::MethodList_t](#) [CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.
- void [Reset](#) (void)
Resets the filters values in response to a change in the object's parameters.

Private Attributes

- [Math_t](#) m_CentralFrequency
The central frequency.
- [Math_t](#) m_Quality
The quality.
- [Math_t](#) m_A0
The x_n and x_{n-2} coefficient.
- [Math_t](#) m_B1
The y_{n-1} coefficient.
- [Math_t](#) m_B2
The y_{n-2} coefficient.

- [StereoData m_X1](#)
The x_{n-1} sample.
- [StereoData m_X2](#)
The x_{n-2} sample.
- [StereoData m_Y1](#)
The y_{n-1} sample.
- [StereoData m_Y2](#)
The y_{n-2} sample.

Friends

- class [ModifierFactory](#)
Add the factory as a friend so it can construct [BandPass](#) objects.
- class [Equalizer](#)
Add the [Equalizer](#) filter as a friend so it can construct [BandPass](#) objects.

Additional Inherited Members

4.2.1 Detailed Description

Bandpass filter.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 [BandPass\(\)](#) [1/3]

```
OCAE::Modifier::BandPass::BandPass (
    BandPass const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.2.2.2 [BandPass\(\)](#) [2/3]

```
OCAE::Modifier::BandPass::BandPass (
```



```
BandPass && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.2.2.3 BandPass() [3/3]

```
OCAE::Modifier::BandPass::BandPass (
    Math_t f,
    Math_t Q = 1 ) [protected]
```

Constructor.

Parameters

<i>f</i>	The central frequency.
<i>Q</i>	The filter quality.

4.2.3 Member Function Documentation

4.2.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Modifier::BandPass::CreateMethodList ( ) [protected],
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.2.3.2 GetFrequency()

```
Math_t OCAE::Modifier::BandPass::GetFrequency ( ) const
```

Returns the central frequency of the filter.

Returns

The central frequency.

4.2.3.3 GetQuality()

```
Math_t OCAE::Modifier::BandPass::GetQuality ( ) const
```

Returns the quality of the filter.

Returns

The quality.

4.2.3.4 IsBase()

```
virtual bool OCAE::Modifier::BandPass::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

```
00179 { return false; };
```

4.2.3.5 operator=() [1/2]

```
BandPass& OCAE::Modifier::BandPass::operator= (
    BandPass const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.2.3.6 operator=() [2/2]

```
BandPass& OCAE::Modifier::BandPass::operator= (
    BandPass && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.2.3.7 Process()

```
virtual StereoData OCAE::Modifier::BandPass::Process (
    StereoData const & input ) [virtual]
```

Takes input sample and filters it, returning the result.

Parameters

<i>input</i>	The input sample.
--------------	-------------------

Returns

The filtered sample.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.2.3.8 Reset()

```
void OCAE::Modifier::BandPass::Reset (
    void ) [protected]
```

Resets the filters values in response to a change in the object's parameters.

4.2.3.9 SetFrequency()

```
void OCAE::Modifier::BandPass::SetFrequency (
    Math_t f )
```

Sets the central frequency of the filter.

Parameters

<i>f</i>	The new central frequency.
----------	----------------------------

4.2.3.10 SetQuality()

```
void OCAE::Modifier::BandPass::SetQuality (
    Math_t Q )
```

Sets the quality of the filter.

Parameters

<i>Q</i>	The new quality.
----------	------------------

The documentation for this class was generated from the following file:

- [BandPass.hpp](#)

4.3 OCAE::Sound::Block Class Reference

This class defines a way of holding a Generator, Modifier and a method of combining the outputs of both of them to produce a single output sample.

```
#include <Block.hpp>
```

Public Types

- using [Interaction_f](#) = std::function< [StereoData](#)([StereoData](#), [StereoData](#))>
Alias for a function that returns a sample, and takes in a generator sample as the first parameter and a modifier sample as the second parameter.
- using [GenBasePtr](#) = Generator::GeneratorBasePtr
Alias for GeneratorBasePtr.
- using [ModBasePtr](#) = Modifier::ModifierBasePtr
Alias for ModifierBasePtr.

Public Member Functions

- [Block](#) ([GenBasePtr](#) const &gen, [ModBasePtr](#) const &mod, [Interaction_f](#) const &interactor)
Block constructor.
- [GenBasePtr](#) & [GetGenerator](#) ()
Returns a reference to the managed generator.
- [ModBasePtr](#) & [GetModifier](#) ()
Returns a reference to the managed modifier.
- [GenBasePtr](#) const & [GetGenerator](#) () const
Returns a reference to the managed generator.
- [ModBasePtr](#) const & [GetModifier](#) () const
Returns a reference to the managed modifier.
- void [PrimeInput](#) ([StereoData](#) input)
Primes the input for the next Process loop.
- [StereoData](#) [Process](#) ()
Processes the managed objects.

Private Attributes

- [GenBasePtr](#) m_Generator
The generator managed by this Block.
- [ModBasePtr](#) m_Modifier
The modifier managed by this Block.
- [Interaction_f](#) m_Interaction
The interactor used by this Block.
- [StereoData](#) m_Input
The input sample.

4.3.1 Detailed Description

This class defines a way of holding a Generator, Modifier and a method of combining the outputs of both of them to produce a single output sample.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 Block()

```
OCAE::Sound::Block::Block (
    GenBasePtr const & gen,
    ModBasePtr const & mod,
    Interaction_f const & interactor )
```

[Block](#) constructor.

Parameters

<i>gen</i>	The generator used for the block.
<i>mod</i>	The modifier used for the block.
<i>interactor</i>	The function that defines how the output of the generator and the modifier are combined. The first argument is the sample from the generator, and the second argument is the sample from the modifier.

4.3.3 Member Function Documentation

4.3.3.1 GetGenerator() [1/2]

```
GenBasePtr& OCAE::Sound::Block::GetGenerator ( )
```

Returns a reference to the managed generator.

Returns

The managed generator.

4.3.3.2 GetGenerator() [2/2]

```
GenBasePtr const& OCAE::Sound::Block::GetGenerator ( ) const
```

Returns a reference to the managed generator.

Returns

The managed generator.

4.3.3.3 GetModifier() [1/2]

```
ModBasePtr& OCAE::Sound::Block::GetModifier ( )
```

Returns a reference to the managed modifier.

Returns

The managed modifier.

4.3.3.4 GetModifier() [2/2]

```
ModBasePtr const& OCAE::Sound::Block::GetModifier ( ) const
```

Returns a reference to the managed modifier.

Returns

The managed modifier.

4.3.3.5 PrimeInput()

```
void OCAE::Sound::Block::PrimeInput (
    StereoData input ) [inline]
```

Primes the input for the next Process loop.

Parameters

<i>input</i>	The input.
--------------	------------

References [OCAE::Left\(\)](#), [OCAE_TYPEDEF_SHARED](#), [Process\(\)](#), and [OCAE::Right\(\)](#).

```
00140     {
00141         Left(m_Input) += Left(input);
00142         Right(m_Input) += Right(input);
00143     };
```

4.3.3.6 Process()

```
StereoData OCAE::Sound::Block::Process ( )
```

Processes the managed objects.

Returns

The processed sample.

Referenced by [PrimeInput\(\)](#).

The documentation for this class was generated from the following file:

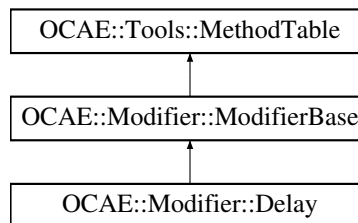
- [Block.hpp](#)

4.4 OCAE::Modifier::Delay Class Reference

[Delay](#) filter.

```
#include <Delay.hpp>
```

Inheritance diagram for OCAE::Modifier::Delay:



Public Member Functions

- [Delay](#) ([Delay](#) const &other)=delete
Copy constructor. Deleted.
- [Delay](#) ([Delay](#) &&other)=default
Default move constructor.
- [Delay](#) & [operator=](#) ([Delay](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [Delay](#) & [operator=](#) ([Delay](#) &&rhs)=default
Default move assignment operator.
- void [SetDelay](#) (uint64_t samples)
Sets a new delay length.
- uint64_t [GetDelay](#) () const
Gets the current delay length.
- virtual [StereoData Process](#) ([StereoData](#) const &input)
Takes input sample and filters it, returning the result.
- virtual bool [IsBase](#) ()
Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Protected Member Functions

- [Delay](#) (uint64_t samples)
Constructor.
- virtual [Tools::MethodTable::MethodList_t CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Attributes

- std::deque< [StereoData](#) > [m_Delay](#)
Delayed sample storage.

Friends

- class [ModifierFactory](#)
Add the factory as a friend so it can construct [Delay](#) objects.

Additional Inherited Members

4.4.1 Detailed Description

[Delay](#) filter.

The delay value is a whole number for simple whole sample calculations.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 [Delay\(\)](#) [1/3]

```
OCAE::Modifier::Delay::Delay (
    Delay const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.4.2.2 Delay() [2/3]

```
OCAE::Modifier::Delay::Delay (
    Delay && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.4.2.3 Delay() [3/3]

```
OCAE::Modifier::Delay::Delay (
    uint64_t samples ) [protected]
```

Constructor.

Parameters

<i>samples</i>	The delay amount in samples.
----------------	------------------------------

4.4.3 Member Function Documentation

4.4.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Modifier::Delay::CreateMethodList ( ) [protected],
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.4.3.2 GetDelay()

```
uint64_t OCAE::Modifier::Delay::GetDelay ( ) const
```

Gets the current delay length.

Returns

The delay length in samples.

4.4.3.3 IsBase()

```
virtual bool OCAE::Modifier::Delay::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

```
00143 { return false; };
```

4.4.3.4 operator=() [1/2]

```
Delay& OCAE::Modifier::Delay::operator= (
    Delay const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.4.3.5 operator=() [2/2]

```
Delay& OCAE::Modifier::Delay::operator= (
    Delay && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.4.3.6 Process()

```
virtual StereoData OCAE::Modifier::Delay::Process (
    StereoData const & input ) [virtual]
```

Takes input sample and filters it, returning the result.

Parameters

<i>input</i>	The input sample.
--------------	-------------------

Returns

The filtered sample.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.4.3.7 SetDelay()

```
void OCAE::Modifier::Delay::SetDelay (
    uint64_t samples )
```

Sets a new delay length.

If the new delay is larger than the previous delay, 0 samples are inserted to the front of the delayed sample list.

Parameters

<i>samples</i>	New delay length in samples.
----------------	------------------------------

The documentation for this class was generated from the following file:

- [Delay.hpp](#)

4.5 OCAE::Core::Driver Class Reference

Handles the calculation of audio samples from different Sounds.

```
#include <Driver.hpp>
```

Public Member Functions

- [Driver](#) ([Driver](#) const &other)=default
Default copy constructor.
- [Driver](#) ([Driver](#) &&other)=default
Default move constructor.
- [~Driver](#) ()
Destructor.
- [Driver](#) & [operator=](#) ([Driver](#) const &rhs)=default
Default copy-assignment operator.
- [Driver](#) & [operator=](#) ([Driver](#) &&rhs)=default
Default move-assignment operator.
- uint64_t [AddSound](#) ([Sound::SoundPtr](#) const &sound)
Adds the given sound to the internal list of tracked sounds.
- [Sound::SoundPtr](#) [RemoveSound](#) (uint64_t id)
Removes a sound from the [Driver](#)'s processing.
- void [SetGain](#) ([Math_t](#) gain=[OCAE_DEFAULT_GAIN](#))
Sets the gain to be used when summing all the audio values.
- [Track_t](#) const & [GetOutputTrack](#) () const
Returns the track used for writing audio output after it has been processed.
- void [Process](#) ()
Processes audio and returns a track of the calculated samples.

Static Public Member Functions

- static [DriverPtr](#) [Create](#) (uint64_t track_size, [Math_t](#) gain=[OCAE_DEFAULT_GAIN](#))
Constructs an audio driver object.

Private Member Functions

- [Driver](#) (uint64_t track_size, [Math_t](#) gain=[OCAE_DEFAULT_GAIN](#))
Constructs an audio driver object.

Static Private Member Functions

- static uint64_t [GetID](#) ()
Returns an ID value for use within the driver.

Private Attributes

- [Track_t](#) [m_OutputTrack](#)
The output track to store the results of processing.
- std::unordered_map< uint64_t, [Sound::SoundPtr](#) > [m_Sounds](#)
All the sounds this driver is responsible for.
- [Math_t](#) [m_Gain](#)
The output gain for the output samples.

Static Private Attributes

- static uint64_t [s_IDCounter](#)
ID counter for generating IDs.

4.5.1 Detailed Description

Handles the calculation of audio samples from different Sounds.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 [Driver\(\)](#) [1/3]

```
OCAE::Core::Driver::Driver (
    Driver const & other ) [default]
```

Default copy constructor.

Parameters

<i>other</i>	The object to copy.
--------------	---------------------

4.5.2.2 Driver() [2/3]

```
OCAE::Core::Driver::Driver (
    Driver && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The object to move.
--------------	---------------------

4.5.2.3 ~Driver()

```
OCAE::Core::Driver::~~Driver ( )
```

Destructor.

4.5.2.4 Driver() [3/3]

```
OCAE::Core::Driver::Driver (
    uint64_t track_size,
    Math_t gain = OCAE_DEFAULT_GAIN ) [private]
```

Constructs an audio driver object.

Parameters

<i>track_size</i>	The size of the output track in samples.
<i>gain</i>	The linear gain to be used when summing all audio values.

4.5.3 Member Function Documentation

4.5.3.1 AddSound()

```
uint64_t OCAE::Core::Driver::AddSound (
    Sound::SoundPtr const & sound )
```

Adds the given sound to the internal list of tracked sounds.

Parameters

<i>sound</i>	The sound to add.
--------------	-------------------

Returns

ID of the added sound.

4.5.3.2 Create()

```
static DriverPtr OCAE::Core::Driver::Create (
    uint64_t track_size,
    Math_t gain = OCAE_DEFAULT_GAIN ) [inline], [static]
```

Constructs an audio driver object.

Parameters

<i>track_size</i>	The size of the output track in samples.
<i>gain</i>	The linear gain to be used when summing all audio values.

Returns

The shared pointer holding the [Driver](#) object.

References [OCAE_DEFAULT_GAIN](#), and [OCAE_TYPEDEF_SHARED](#).

```
00185     {
00186         return DriverPtr(new Driver(track_size, gain));
00187     };
```

4.5.3.3 GetID()

```
static uint64_t OCAE::Core::Driver::GetID ( ) [static], [private]
```

Returns an ID value for use within the driver.

Returns

The generated ID value.

4.5.3.4 GetOutputTrack()

```
Track_t const& OCAE::Core::Driver::GetOutputTrack ( ) const
```

Returns the track used for writing audio output after it has been processed.

Returns

Track_t containing the output of the latest process call.

4.5.3.5 operator=() [1/2]

```
Driver& OCAE::Core::Driver::operator= (
    Driver const & rhs ) [default]
```

Default copy-assignment operator.

Parameters

<i>rhs</i>	The object to copy.
------------	---------------------

Returns

this.

4.5.3.6 operator=() [2/2]

```
Driver& OCAE::Core::Driver::operator= (
    Driver && rhs ) [default]
```

Default move-assignment operator.

Parameters

<i>rhs</i>	The object to move.
------------	---------------------

Returns

this.

4.5.3.7 Process()

```
void OCAE::Core::Driver::Process ( )
```

Processes audio and returns a track of the calculated samples.

4.5.3.8 RemoveSound()

```
Sound::SoundPtr OCAE::Core::Driver::RemoveSound (
    uint64_t id )
```

Removes a sound from the [Driver](#)'s processing.

Parameters

<i>id</i>	The ID of the sound to be removed.
-----------	------------------------------------

Returns

The sound that was removed.

4.5.3.9 SetGain()

```
void OCAE::Core::Driver::SetGain (
    Math_t gain = OCAE_DEFAULT_GAIN )
```

Sets the gain to be used when summing all the audio values.

Parameters

<i>gain</i>	The linear gain value to be set.
-------------	----------------------------------

The documentation for this class was generated from the following file:

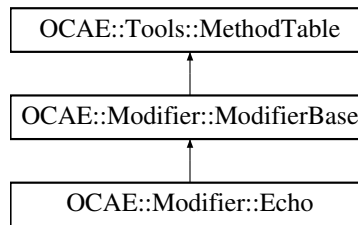
- [Driver.hpp](#)

4.6 OCAE::Modifier::Echo Class Reference

[Echo](#) IIR filter. Uses output sample for echoing instead of input, creating an infinite impulse response (IIR).

```
#include <Echo.hpp>
```

Inheritance diagram for OCAE::Modifier::Echo:



Public Member Functions

- [Echo](#) ([Echo](#) const &other)=delete
Copy constructor. Deleted.
- [Echo](#) ([Echo](#) &&other)=default
Default move constructor.
- virtual [~Echo](#) ()=default
Default destructor.
- [Echo](#) & [operator=](#) ([Echo](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [Echo](#) & [operator=](#) ([Echo](#) &&rhs)=default
Default move assignment operator.
- void [SetDecayRatio](#) ([Math_t](#) decay_ratio)
Sets the decay ratio of the echo samples.
- [Math_t](#) [GetDecayRatio](#) () const
Gets the decay ratio of the echo samples.
- virtual [StereoData Process](#) ([StereoData](#) const &input)
Takes input sample and filters it, returning the result.
- virtual bool [IsBase](#) ()
Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Protected Member Functions

- [Echo](#) (uint64_t sample_delay, [Math_t](#) decay_ratio)
Constructor.
- virtual [Tools::MethodTable::MethodList_t CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Attributes

- std::deque< [StereoData](#) > [m_Echo](#)
Filtered samples for continuous echo.
- [Math_t](#) [m_Ratio](#)
Decay ratio for the echo.

Friends

- class [ModifierFactory](#)
Add the factory as a friend so it can construct [Echo](#) objects.

Additional Inherited Members

4.6.1 Detailed Description

[Echo](#) IIR filter. Uses output sample for echoing instead of input, creating an infinite impulse response (IIR).

The delay value between echos is a whole number for simple whole sample calculations.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 [Echo\(\)](#) [1/3]

```
OCAE::Modifier::Echo::Echo (
    Echo const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.6.2.2 [Echo\(\)](#) [2/3]

```
OCAE::Modifier::Echo::Echo (
    Echo && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.6.2.3 Echo() [3/3]

```
OCAE::Modifier::Echo::Echo (
    uint64_t sample_delay,
    Math_t decay_ratio ) [protected]
```

Constructor.

Parameters

<i>sample_delay</i>	The delay in samples between the input signal and it's first echo.
<i>decay_ratio</i>	The decay ratio of the echo samples.

4.6.3 Member Function Documentation

4.6.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Modifier::Echo::CreateMethodList ( ) [protected],
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.6.3.2 GetDecayRatio()

```
Math_t OCAE::Modifier::Echo::GetDecayRatio ( ) const
```

Gets the decay ratio of the echo samples.

Returns

The decay ratio.

4.6.3.3 IsBase()

```
virtual bool OCAE::Modifier::Echo::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

```
00144 { return false; };
```

4.6.3.4 operator=() [1/2]

```
Echo& OCAE::Modifier::Echo::operator= (
    Echo const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.6.3.5 operator=() [2/2]

```
Echo& OCAE::Modifier::Echo::operator= (
    Echo && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.6.3.6 Process()

```
virtual StereoData OCAE::Modifier::Echo::Process (  
    StereoData const & input ) [virtual]
```

Takes input sample and filters it, returning the result.

Parameters

<i>input</i>	The input sample.
--------------	-------------------

Returns

The filtered sample.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.6.3.7 SetDecayRatio()

```
void OCAE::Modifier::Echo::SetDecayRatio (  
    Math_t decay_ratio )
```

Sets the decay ratio of the echo samples.

Parameters

<i>decay_ratio</i>	The new decay ratio.
--------------------	----------------------

The documentation for this class was generated from the following file:

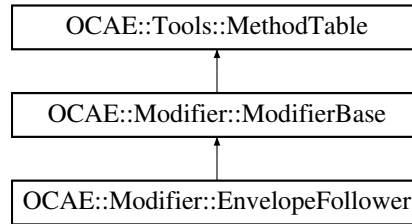
- [Echo.hpp](#)

4.7 OCAE::Modifier::EnvelopeFollower Class Reference

Envelope follower filter. Calculates the gain of the input signal over time.

```
#include <Envelope.hpp>
```

Inheritance diagram for `OCAE::Modifier::EnvelopeFollower`:



Public Member Functions

- [EnvelopeFollower](#) ([EnvelopeFollower](#) const &other)=delete
Copy constructor. Deleted.
- [EnvelopeFollower](#) ([EnvelopeFollower](#) &&other)=default
Default move constructor.
- virtual [~EnvelopeFollower](#) ()
Destructor.
- [EnvelopeFollower](#) & [operator=](#) ([EnvelopeFollower](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [EnvelopeFollower](#) & [operator=](#) ([EnvelopeFollower](#) &&rhs)=default
Default move assignment operator.
- virtual [StereoData Process](#) ([StereoData](#) const &input)
Takes input sample and filters it, returning the result.
- virtual bool [IsBase](#) ()
Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Protected Member Functions

- [EnvelopeFollower](#) ([Math_t](#) lower, [Math_t](#) upper)
Constructor. Creates a follower with upper and lower bounds to what frequencies it should follow.
- virtual [Tools::MethodTable::MethodList_t CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Attributes

- [Math_t m_AU](#)
Tracking variable.
- [Math_t m_BU](#)
Tracking variable.
- [Math_t m_AD](#)
Tracking variables.
- [Math_t m_BD](#)
Tracking variable.
- [StereoData m_X1](#)
Previous sample.
- [StereoData m_Y1](#)
Previous sample.

Friends

- class [ModifierFactory](#)

Add the factory as a friend so it can construct [EnvelopeFollower](#) objects.

Additional Inherited Members

4.7.1 Detailed Description

Envelope follower filter. Calculates the gain of the input signal over time.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 EnvelopeFollower() [1/3]

```
OCAE::Modifier::EnvelopeFollower::EnvelopeFollower (  
    EnvelopeFollower const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.7.2.2 EnvelopeFollower() [2/3]

```
OCAE::Modifier::EnvelopeFollower::EnvelopeFollower (  
    EnvelopeFollower && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.7.2.3 ~EnvelopeFollower()

```
virtual OCAE::Modifier::EnvelopeFollower::~EnvelopeFollower ( ) [virtual]
```

Deconstructor.

4.7.2.4 EnvelopeFollower() [3/3]

```
OCAE::Modifier::EnvelopeFollower::EnvelopeFollower (
    Math_t lower,
    Math_t upper ) [protected]
```

Constructor. Creates a follower with upper and lower bounds to what frequencies it should follow.

Parameters

<i>lower</i>	The lower bound of frequencies to follow.
<i>upper</i>	The upper bound of frequencies to follow.

4.7.3 Member Function Documentation

4.7.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Modifier::EnvelopeFollower::CreateMethodList ( )
[protected], [virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.7.3.2 IsBase()

```
virtual bool OCAE::Modifier::EnvelopeFollower::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Returns

True for this class, false for any derived class.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

```
00133 { return false; };
```

4.7.3.3 operator=() [1/2]

```
EnvelopeFollower& OCAE::Modifier::EnvelopeFollower::operator= (
    EnvelopeFollower const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.7.3.4 operator=() [2/2]

```
EnvelopeFollower& OCAE::Modifier::EnvelopeFollower::operator= (
    EnvelopeFollower && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.7.3.5 Process()

```
virtual StereoData OCAE::Modifier::EnvelopeFollower::Process (  
    StereoData const & input ) [virtual]
```

Takes input sample and filters it, returning the result.

Parameters

<i>input</i>	The input sample.
--------------	-------------------

Returns

The filtered sample.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

The documentation for this class was generated from the following file:

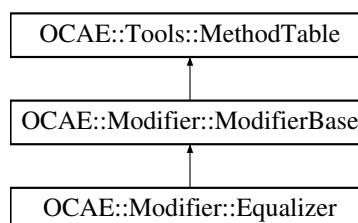
- [Envelope.hpp](#)

4.8 OCAE::Modifier::Equalizer Class Reference

[Equalizer](#) filter.

```
#include <Equalizer.hpp>
```

Inheritance diagram for OCAE::Modifier::Equalizer:



Public Member Functions

- [Equalizer](#) ([Equalizer](#) const &other)=delete
Copy constructor. Deleted.
- [Equalizer](#) ([Equalizer](#) &&other)=default
Default move constructor.
- virtual [~Equalizer](#) ()=default
Default destructor.
- [Equalizer](#) & [operator=](#) ([Equalizer](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [Equalizer](#) & [operator=](#) ([Equalizer](#) &&rhs)=default
Default move assignment operator.
- void [SetGain](#) (uint32_t band, [Math_t](#) gain)
Sets the gain for a given frequency band.
- [Math_t](#) [GetGain](#) (uint32_t band)
Gets the gain from a given frequency band.
- virtual [StereoData Process](#) ([StereoData](#) const &input)
Takes input sample and filters it, returning the result.
- virtual bool [IsBase](#) ()
Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Protected Member Functions

- [Equalizer](#) (uint32_t band_count, [Math_t](#) lower, [Math_t](#) upper)
Constructor.
- virtual [Tools::MethodTable::MethodList_t](#) [CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Attributes

- std::vector< [Math_t](#) > [m_BandGains](#)
List of gains for each frequency band.
- std::vector< [BandPassPtr](#) > [m_Bands](#)
List of band pass filters for each frequency band.

Friends

- class [ModifierFactory](#)
Add the factory as a friend so it can construct [Equalizer](#) objects.

Additional Inherited Members

4.8.1 Detailed Description

[Equalizer](#) filter.

This filter splits a given signal across bands, using [Modifier::BandPass](#) objects to do so, then amplifies each band by a given gain before combining the bands again for the final output.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 Equalizer() [1/3]

```
OCAE::Modifier::Equalizer::Equalizer (
    Equalizer const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.8.2.2 Equalizer() [2/3]

```
OCAE::Modifier::Equalizer::Equalizer (
    Equalizer && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.8.2.3 ~Equalizer()

```
virtual OCAE::Modifier::Equalizer::~~Equalizer ( ) [virtual], [default]
```

Default destructor.

4.8.2.4 Equalizer() [3/3]

```
OCAE::Modifier::Equalizer::Equalizer (
    uint32_t band_count,
    Math_t lower,
    Math_t upper ) [protected]
```

Constructor.

Parameters

<i>band_count</i>	The number of frequency bands for the equalizer.
<i>lower</i>	The lowest frequency of the lowest band pass filter (not the central frequency).
<i>upper</i>	The highest frequency of the highest band pass filter (not the central frequency).

4.8.3 Member Function Documentation

4.8.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList\_t OCAE::Modifier::Equalizer::CreateMethodList ( ) [protected],  
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.8.3.2 GetGain()

```
Math\_t OCAE::Modifier::Equalizer::GetGain (   
    uint32\_t band )
```

Gets the gain from a given frequency band.

Parameters

<i>band</i>	The frequency band to get the gain from.
-------------	--

Returns

4.8.3.3 IsBase()

```
virtual bool OCAE::Modifier::Equalizer::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

```
00155 { return false; };
```

4.8.3.4 operator=() [1/2]

```
Equalizer& OCAE::Modifier::Equalizer::operator= (
    Equalizer const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.8.3.5 operator=() [2/2]

```
Equalizer& OCAE::Modifier::Equalizer::operator= (
    Equalizer && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.8.3.6 Process()

```
virtual StereoData OCAE::Modifier::Equalizer::Process (  
    StereoData const & input ) [virtual]
```

Takes input sample and filters it, returning the result.

Parameters

<i>input</i>	The input sample.
--------------	-------------------

Returns

The filtered sample.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.8.3.7 SetGain()

```
void OCAE::Modifier::Equalizer::SetGain (  
    uint32_t band,  
    Math_t gain )
```

Sets the gain for a given frequency band.

Parameters

<i>band</i>	The frequency band to set the gain of.
<i>gain</i>	The new gain.

The documentation for this class was generated from the following file:

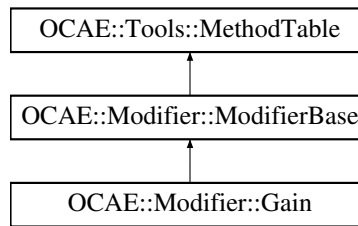
- [Equalizer.hpp](#)

4.9 OCAE::Modifier::Gain Class Reference

Simple gain filter for amplifying the input signal. The gain value can be negative allowing for inverting the input signal.

```
#include <Gain.hpp>
```

Inheritance diagram for OCAE::Modifier::Gain:



Public Member Functions

- [Gain](#) ([Gain](#) const &other)=delete
Copy constructor. Deleted.
- [Gain](#) ([Gain](#) &&other)=default
Default move constructor.
- virtual [~Gain](#) ()=default
Destructor.
- [Gain](#) & [operator=](#) ([Gain](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [Gain](#) & [operator=](#) ([Gain](#) &&rhs)=default
Default move assignment operator.
- void [SetGain](#) ([Math_t](#) gain)
Sets the gain for the filter.
- [Math_t](#) [GetGain](#) () const
Returns the current gain for the filter.
- virtual [StereoData Process](#) ([StereoData](#) const &input)
Takes input sample and filters it, returning the result.
- virtual bool [IsBase](#) ()
Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Protected Member Functions

- [Gain](#) ([Math_t](#) gain=[OCAE_DEFAULT_GAIN](#))
Constructor.
- virtual [Tools::MethodTable::MethodList_t](#) [CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Attributes

- [Math_t](#) [m_Gain](#)
The gain.

Friends

- class [ModifierFactory](#)

Add the factory as a friend so it can construct [Gain](#) objects.

Additional Inherited Members

4.9.1 Detailed Description

Simple gain filter for amplifying the input signal. The gain value can be negative allowing for inverting the input signal.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 Gain() [1/3]

```
OCAE::Modifier::Gain::Gain (
    Gain const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.9.2.2 Gain() [2/3]

```
OCAE::Modifier::Gain::Gain (
    Gain && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.9.2.3 ~Gain()

```
virtual OCAE::Modifier::Gain::~Gain ( ) [virtual], [default]
```

Destructor.

4.9.2.4 Gain() [3/3]

```
OCAE::Modifier::Gain::Gain (
    Math_t gain = OCAE_DEFAULT_GAIN ) [protected]
```

Constructor.

Parameters

<i>gain</i>	The gain to apply to the input data. Can be negative allowing for inverting the signal.
-------------	---

4.9.3 Member Function Documentation

4.9.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Modifier::Gain::CreateMethodList ( ) [protected],
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.9.3.2 GetGain()

```
Math_t OCAE::Modifier::Gain::GetGain ( ) const
```

Returns the current gain for the filter.

Returns

The gain of the filter.

4.9.3.3 IsBase()

```
virtual bool OCAE::Modifier::Gain::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

```
00141 { return false; };
```

4.9.3.4 operator=() [1/2]

```
Gain& OCAE::Modifier::Gain::operator= (
    Gain const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.9.3.5 operator=() [2/2]

```
Gain& OCAE::Modifier::Gain::operator= (
    Gain && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.9.3.6 Process()

```
virtual StereoData OCAE::Modifier::Gain::Process (
    StereoData const & input ) [virtual]
```

Takes input sample and filters it, returning the result.

Parameters

<i>input</i>	The input sample.
--------------	-------------------

Returns

The filtered sample.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.9.3.7 SetGain()

```
void OCAE::Modifier::Gain::SetGain (
    Math_t gain )
```

Sets the gain for the filter.

Parameters

<i>gain</i>	The new gain. Can be negative.
-------------	--------------------------------

The documentation for this class was generated from the following file:

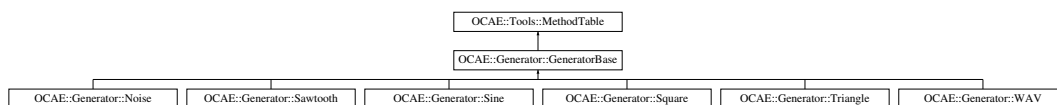
- [Gain.hpp](#)

4.10 OCAE::Generator::GeneratorBase Class Reference

General base class for all generator (sounds) to inherit from. Any derived classes with extra methods that may need to be acquired can be accessed through their setup of the [Tools::MethodTable](#).

```
#include <GeneratorBase.hpp>
```

Inheritance diagram for OCAE::Generator::GeneratorBase:



Public Member Functions

- [GeneratorBase](#) ([GeneratorBase](#) const &other)=delete
Copy constructor. Deleted.
- [GeneratorBase](#) ([GeneratorBase](#) &&other)=default
Default move constructor.
- virtual [~GeneratorBase](#) ()=default
Default destructor.
- [GeneratorBase](#) & [operator=](#) ([GeneratorBase](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [GeneratorBase](#) & [operator=](#) ([GeneratorBase](#) &&rhs)=default
Default move assignment operator.
- virtual [StereoData Process](#) (void)
Calculates the sample. For the base class this is simply 0.
- virtual bool [IsBase](#) ()
Returns boolean for if the object is a [GeneratorBase](#) or not.

Protected Member Functions

- [GeneratorBase](#) ()
Constructor.
- virtual [Tools::MethodTable::MethodList_t CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Friends

- class [GeneratorFactory](#)

Add the factory as a friend so it can construct [GeneratorBase](#) objects.

Additional Inherited Members

4.10.1 Detailed Description

General base class for all generator (sounds) to inherit from. Any derived classes with extra methods that may need to be acquired can be accessed through their setup of the [Tools::MethodTable](#).

4.10.2 Constructor & Destructor Documentation

4.10.2.1 [GeneratorBase\(\)](#) [1/3]

```
OCAE::Generator::GeneratorBase::GeneratorBase (
    GeneratorBase const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.10.2.2 [GeneratorBase\(\)](#) [2/3]

```
OCAE::Generator::GeneratorBase::GeneratorBase (
    GeneratorBase && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.10.2.3 ~GeneratorBase()

```
virtual OCAE::Generator::GeneratorBase::~~GeneratorBase ( ) [virtual], [default]
```

Default destructor.

4.10.2.4 GeneratorBase() [3/3]

```
OCAE::Generator::GeneratorBase::GeneratorBase ( ) [inline], [protected]
```

Constructor.

References [CreateMethodList\(\)](#), and [OCAE::Tools::MethodTable::RegisterMethods\(\)](#).

```
00135 : MethodTable() { RegisterMethods(CreateMethodList()); };
```

4.10.3 Member Function Documentation

4.10.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Generator::GeneratorBase::CreateMethodList ( )
[inline], [protected], [virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Implements [OCAE::Tools::MethodTable](#).

Reimplemented in [OCAE::Generator::WAV](#), [OCAE::Generator::Sine](#), [OCAE::Generator::Square](#), [OCAE::Generator::Sawtooth](#), [OCAE::Generator::Triangle](#), and [OCAE::Generator::Noise](#).

References [OCAE_TYPEDEF_SHARED](#).

Referenced by [GeneratorBase\(\)](#).

```
00147 { return {}; };
```

4.10.3.2 IsBase()

```
virtual bool OCAE::Generator::GeneratorBase::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object is a [GeneratorBase](#) or not.

Returns

True for this class, false for any derived class.

Reimplemented in [OCAE::Generator::Sine](#), [OCAE::Generator::Sawtooth](#), [OCAE::Generator::Triangle](#), [OCAE::Generator::WAV](#), [OCAE::Generator::Noise](#), and [OCAE::Generator::Square](#).

```
00122 { return true; };
```

4.10.3.3 operator=() [1/2]

```
GeneratorBase& OCAE::Generator::GeneratorBase::operator= (
    GeneratorBase const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.10.3.4 operator=() [2/2]

```
GeneratorBase& OCAE::Generator::GeneratorBase::operator= (
    GeneratorBase && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.10.3.5 Process()

```
virtual StereoData OCAE::Generator::GeneratorBase::Process (  
    void ) [inline], [virtual]
```

Calculates the sample. For the base class this is simply 0.

Returns

The stereo sample data.

Reimplemented in [OCAE::Generator::Sine](#), [OCAE::Generator::Sawtooth](#), [OCAE::Generator::Triangle](#), [OCAE::Generator::WAV](#), [OCAE::Generator::Noise](#), and [OCAE::Generator::Square](#).

```
00113 { return StereoData(0.f, 0.f); };
```

The documentation for this class was generated from the following file:

- [GeneratorBase.hpp](#)

4.11 OCAE::Generator::GeneratorFactory Class Reference

Creates pointers to generators handled by std::shared_ptr to prevent memory leaks.

```
#include <GeneratorFactory.hpp>
```

Public Member Functions

- [~GeneratorFactory](#) ()=delete

Deleted destructor, ensuring an instance of this class can never be created.

Static Public Member Functions

- static GeneratorBasePtr [CreateBase](#) ()
Creates a [GeneratorBase](#) object.
- static NoisePtr [CreateNoise](#) ()
Creates a [Noise](#) object.
- static SawtoothPtr [CreateSawtooth](#) (Math_t freq)
Creates a [Sawtooth](#) object.
- static SinePtr [CreateSine](#) (Math_t freq)
Creates a [Sine](#) object.
- static SquarePtr [CreateSquare](#) (Math_t freq)
Creates a [Square](#) object.
- static TrianglePtr [CreateTriangle](#) (Math_t freq)
Creates a [Triangle](#) object.
- static WAVPtr [CreateWAV](#) ()
Creates a [WAV](#) object with no [WAV](#) data.
- static WAVPtr [CreateWAV](#) (std::string const &filepath)
Creates a [WAV](#) object with a file name to open for reading.
- static WAVPtr [CreateWAV](#) (std::vector< char > const &wav_data)
Creates a [WAV](#) object with a vector containing the audio [WAV](#) data.

4.11.1 Detailed Description

Creates pointers to generators handled by std::shared_ptr to prevent memory leaks.

4.11.2 Member Function Documentation

4.11.2.1 CreateBase()

```
static GeneratorBasePtr OCABE::Generator::GeneratorFactory::CreateBase ( ) [static]
```

Creates a [GeneratorBase](#) object.

Returns

GeneratorBasePtr containing the created object.

4.11.2.2 CreateNoise()

```
static NoisePtr OCAE::Generator::GeneratorFactory::CreateNoise ( ) [static]
```

Creates a [Noise](#) object.

Returns

GeneratorBasePtr containing the created object.

4.11.2.3 CreateSawtooth()

```
static SawtoothPtr OCAE::Generator::GeneratorFactory::CreateSawtooth (
    Math_t freq ) [static]
```

Creates a [Sawtooth](#) object.

Parameters

<i>freq</i>	The frequency for the sawtooth.
-------------	---------------------------------

Returns

GeneratorBasePtr containing the created object.

4.11.2.4 CreateSine()

```
static SinePtr OCAE::Generator::GeneratorFactory::CreateSine (
    Math_t freq ) [static]
```

Creates a [Sine](#) object.

Parameters

<i>freq</i>	The frequency for the sine.
-------------	-----------------------------

Returns

GeneratorBasePtr containing the created object.

4.11.2.5 CreateSquare()

```
static SquarePtr OCAE::Generator::GeneratorFactory::CreateSquare (
    Math_t freq ) [static]
```

Creates a [Square](#) object.

Parameters

<i>freq</i>	The frequency for the square.
-------------	-------------------------------

Returns

GeneratorBasePtr containing the created object.

4.11.2.6 CreateTriangle()

```
static TrianglePtr OCAE::Generator::GeneratorFactory::CreateTriangle (
    Math_t freq ) [static]
```

Creates a [Triangle](#) object.

Parameters

<i>freq</i>	The frequency for the triangle.
-------------	---------------------------------

Returns

GeneratorBasePtr containing the created object.

4.11.2.7 CreateWAV() [1/3]

```
static WAVPtr OCAE::Generator::GeneratorFactory::CreateWAV ( ) [static]
```

Creates a [WAV](#) object with no [WAV](#) data.

Returns

GeneratorBasePtr containing the created object.

4.11.2.8 CreateWAV() [2/3]

```
static WAVPtr OCAE::Generator::GeneratorFactory::CreateWAV (
    std::string const & filepath ) [static]
```

Creates a [WAV](#) object with a file name to open for reading.

Parameters

<i>filepath</i>	The path to the file.
-----------------	-----------------------

Returns

GeneratorBasePtr containing the created object.

4.11.2.9 CreateWAV() [3/3]

```
static WAVPtr OCAE::Generator::GeneratorFactory::CreateWAV (
    std::vector< char > const & wav_data ) [static]
```

Creates a [WAV](#) object with a vector containing the audio [WAV](#) data.

Parameters

<i>wav_data</i>	Raw WAVE data in RIFF format.
-----------------	-------------------------------

Returns

GeneratorBasePtr containing the created object.

The documentation for this class was generated from the following file:

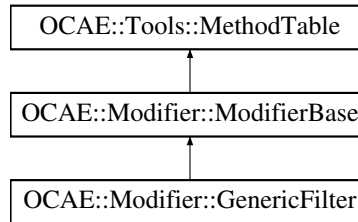
- [GeneratorFactory.hpp](#)

4.12 OCAE::Modifier::GenericFilter Class Reference

Generic audio filter with simple poles.

```
#include <GenericFilter.hpp>
```

Inheritance diagram for OCAE::Modifier::GenericFilter:



Public Types

- using [ZeroContainer](#) = std::vector< std::tuple< uint32_t, [Math_t](#) > >
Container used for coefficients of zeros of a filter.
- using [PoleContainer](#) = std::vector< std::tuple< uint32_t, [Math_t](#) > >
Container used for coefficients of poles of a filter.

Public Member Functions

- [GenericFilter](#) ([GenericFilter](#) const &other)=delete
Copy constructor. Deleted.
- [GenericFilter](#) ([GenericFilter](#) &&other)=default
Default move constructor.
- virtual ~[GenericFilter](#) ()=default
Destructor.
- [GenericFilter](#) & operator= ([GenericFilter](#) const &rhs)=delete
Assignment operator. Deleted.
- [GenericFilter](#) & operator= ([GenericFilter](#) &&rhs)=default
Default move assignment operator.
- virtual [StereoData Process](#) ([StereoData](#) const &input)
Takes input sample and filters it, returning the result.
- virtual bool [IsBase](#) ()
Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Protected Member Functions

- [GenericFilter](#) ([ZeroContainer](#) const &zeros, [PoleContainer](#) const &poles)
Constructor.
- virtual [Tools::MethodTable::MethodList_t CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Types

- using [SampleContainer](#) = std::deque< [StereoData](#) >
Container used for the previous outputs and inputs of the filter.

Private Attributes

- [ZeroContainer m_Zeros](#)
Vector of tuples, tuple of the x subscript and its coefficient.
- [PoleContainer m_Poles](#)
Vector of tuples, tuple of the y subscript and its coefficient.
- [SampleContainer m_Inputs](#)
Previous inputs to the filter.
- [SampleContainer m_Outputs](#)
Previous outputs to the filter.

Friends

- class [ModifierFactory](#)
Add the factory as a friend so it can construct [GenericFilter](#) objects.

Additional Inherited Members

4.12.1 Detailed Description

Generic audio filter with simple poles.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 [GenericFilter\(\)](#) [1/3]

```
OCAE::Modifier::GenericFilter::GenericFilter (
    GenericFilter const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.12.2.2 [GenericFilter\(\)](#) [2/3]

```
OCAE::Modifier::GenericFilter::GenericFilter (
    GenericFilter && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.12.2.3 ~GenericFilter()

```
virtual OCAE::Modifier::GenericFilter::~GenericFilter ( ) [virtual], [default]
```

Destructor.

4.12.2.4 GenericFilter() [3/3]

```
OCAE::Modifier::GenericFilter::GenericFilter (
    ZeroContainer const & zeros,
    PoleContainer const & poles ) [protected]
```

Constructor.

Parameters

<i>zeros</i>	Container a tuple of the x subscript and its coefficient. Expected to be ordered lowest to highest by subscript.
<i>poles</i>	Container of a tuple of the the y subscript and its coefficient. Expected to be ordered lowest to highest by subscript.

4.12.3 Member Function Documentation

4.12.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Modifier::GenericFilter::CreateMethodList ( ) [protected],
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.12.3.2 IsBase()

```
virtual bool OCAE::Modifier::GenericFilter::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

```
00143 { return false; };
```

4.12.3.3 operator=() [1/2]

```
GenericFilter& OCAE::Modifier::GenericFilter::operator= (
    GenericFilter const & rhs ) [deleted]
```

Assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to copy.
------------	---------------------

Returns

this.

4.12.3.4 operator=() [2/2]

```
GenericFilter& OCAE::Modifier::GenericFilter::operator= (
    GenericFilter && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.12.3.5 Process()

```
virtual StereoData OCAE::Modifier::GenericFilter::Process (
    StereoData const & input ) [virtual]
```

Takes input sample and filters it, returning the result.

Parameters

<i>input</i>	The input sample.
--------------	-------------------

Returns

The filtered sample.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

The documentation for this class was generated from the following file:

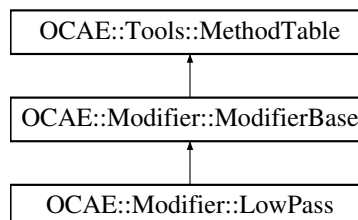
- [GenericFilter.hpp](#)

4.13 OCAE::Modifier::LowPass Class Reference

3rd Order Butterworth Low Pass filter with resonance.

```
#include <LowPass.hpp>
```

Inheritance diagram for OCAE::Modifier::LowPass:



Public Member Functions

- [LowPass](#) ([LowPass](#) const &other)=delete
Copy constructor. Deleted.
- [LowPass](#) ([LowPass](#) &&other)=default
Default move constructor.
- virtual [~LowPass](#) ()=default
Destructor.
- [LowPass](#) & [operator=](#) ([LowPass](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [LowPass](#) & [operator=](#) ([LowPass](#) &&rhs)=default
Default move assignment operator.
- void [SetCutoff](#) ([Math_t](#) cutoff)
Sets the cutoff frequency of the filter.
- void [SetResonance](#) ([Math_t](#) resonance)
Sets the resonance angle of the filter.
- virtual [StereoData Process](#) ([StereoData](#) const &input)
Takes input sample and filters it, returning the result.
- virtual bool [IsBase](#) ()
Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Protected Member Functions

- [LowPass](#) ([Math_t](#) cutoff, [Math_t](#) resonance)
Constructor.
- virtual [Tools::MethodTable::MethodList_t CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.
- void [Reset](#) ()
Resets the values of the object. Called during construction, [SetCutoff](#), and [SetResonance](#).

Private Attributes

- [Math_t m_Cutoff](#)
Cutoff frequency.
- [Math_t m_Resonance](#)
Resonance.
- [Math_t m_Coefficients](#) [4]
List of coefficients for the filter.
- [StereoData m_Outputs](#) [3]
Previous outputs for future calculations.

Friends

- class [ModifierFactory](#)
Add the factory as a friend so it can construct [LowPass](#) objects.

Additional Inherited Members

4.13.1 Detailed Description

3rd Order Butterworth Low Pass filter with resonance.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 LowPass() [1/3]

```
OCAE::Modifier::LowPass::LowPass (
    LowPass const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.13.2.2 LowPass() [2/3]

```
OCAE::Modifier::LowPass::LowPass (
    LowPass && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.13.2.3 ~LowPass()

```
virtual OCAE::Modifier::LowPass::~~LowPass ( ) [virtual], [default]
```

Destructor.

4.13.2.4 LowPass() [3/3]

```
OCAE::Modifier::LowPass::LowPass (
    Math_t cutoff,
    Math_t resonance ) [protected]
```

Constructor.

Parameters

<i>cutoff</i>	The cutoff frequency in Hz.
<i>resonance</i>	The resonance angle of the filter, value can be in range [0,1]. No safety checks are performed.

4.13.3 Member Function Documentation

4.13.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Modifier::LowPass::CreateMethodList ( ) [protected],
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.13.3.2 IsBase()

```
virtual bool OCAE::Modifier::LowPass::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

```
00147 { return false; };
```

4.13.3.3 operator=() [1/2]

```
LowPass& OCAE::Modifier::LowPass::operator= (
    LowPass const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.13.3.4 operator=() [2/2]

```
LowPass& OCAE::Modifier::LowPass::operator= (  
    LowPass && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.13.3.5 Process()

```
virtual StereoData OCAE::Modifier::LowPass::Process (  
    StereoData const & input ) [virtual]
```

Takes input sample and filters it, returning the result.

Parameters

<i>input</i>	The input sample.
--------------	-------------------

Returns

The filtered sample.

Reimplemented from [OCAE::Modifier::ModifierBase](#).

4.13.3.6 Reset()

```
void OCAE::Modifier::LowPass::Reset ( ) [protected]
```

Resets the values of the object. Called during construction, SetCutoff, and SetResonance.

4.13.3.7 SetCutoff()

```
void OCAE::Modifier::LowPass::SetCutoff (
    Math_t cutoff )
```

Sets the cutoff frequency of the filter.

Parameters

<i>cutoff</i>	The cutoff frequency.
---------------	-----------------------

4.13.3.8 SetResonance()

```
void OCAE::Modifier::LowPass::SetResonance (
    Math_t resonance )
```

Sets the resonance angle of the filter.

Parameters

<i>resonance</i>	The resonance angle, in range [0,1]. No safety checks are performed.
------------------	--

The documentation for this class was generated from the following file:

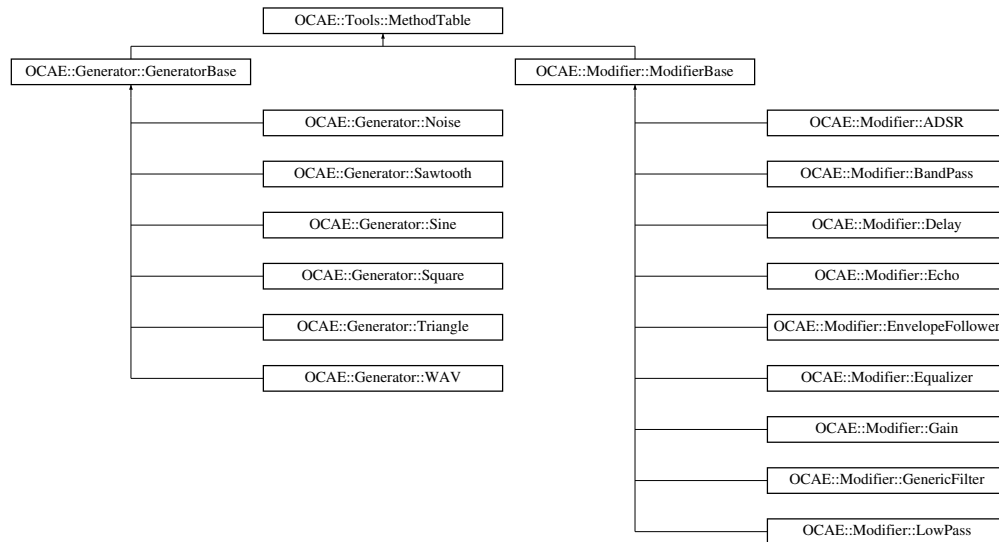
- [LowPass.hpp](#)

4.14 OCAE::Tools::MethodTable Class Reference

The purpose of this class is to create a simple interface for calling methods from an object of an unknown type.

```
#include <MethodTable.hpp>
```

Inheritance diagram for OCAE::Tools::MethodTable:



Public Types

- using `Void_fn` = `std::function< void(void *)>`
Alias for a void-returning function that takes a void pointer.
- using `MethodTable_t` = `std::unordered_map< MethodTable *, std::unordered_map< std::string, Void_fn > >`
Alias for the mapping of method names to the method.
- using `MethodList_t` = `std::vector< std::tuple< std::string, Void_fn > >`
Alias for the list of method names and their associated methods.

Public Member Functions

- `MethodTable ()`
Default constructor.
- `MethodTable (MethodList_t const &list)`
Constructor.
- `virtual ~MethodTable ()=default`
Default destructor.
- `template<typename... Args>`
`void CallMethod (std::string const &fn, Args &&... args)`
Calls a method.

Protected Member Functions

- `void RegisterMethod (std::string const &fn_name, Void_fn const &fn_obj)`
Registers a single method and its name within the internal method table.
- `void RegisterMethods (MethodList_t const &list)`
Registers a list of methods and their names within the internal method table.
- `virtual MethodList_t CreateMethodList ()=0`
Creates a vector containing the names of functions, and the callable functions themselves.

Static Protected Attributes

- static [MethodTable_t s_Table](#)
Object mapping a string to a function.

4.14.1 Detailed Description

The purpose of this class is to create a simple interface for calling methods from an object of an unknown type.

For example, within OCAE you have a Sine object currently represented by a GeneratorBasePtr object. To call the Sine method to set the frequency you would utilize this class in the following manner:

```
GeneratorBasePtr obj = CreateSine(440);

Math_t new_freq = 880;
obj->CallMethod("SetFrequency", OCAE_METHOD_PARAM(new_freq));
obj->CallMethod("GetFrequency", OCAE_METHOD_RET(new_freq));
```

Here, the [OCAE_METHOD_RET\(\)](#) and [OCAE_METHOD_PARAM\(\)](#) macros ensure that the values passed to the function will have the proper types, guaranteeing they are handled properly. See the macros' documentation and definition in [Macro.hpp](#) for more info.

It is recommended to construct the method table with the default constructor, and then set the methods for the class in a fashion like:

```
Foo:Foo() : MethodTable(), // ...
{
    RegisterMethods(CreateMethodList());

    // or

    RegisterMethod("method1", [this](void *){ method1(); });
    RegisterMethod("method2", [this](void * p){
        method2(
            std::get<0>({
                *reinterpret_cast<OCAE_METHOD_PARAM_T(int)>(p)
            })
        );
    });
    // ...
}

Tools::MethodTable::MethodList_t Foo::CreateMethodList()
{
    // Returns initializer list that constructs a MethodList_t
    return {
        std::make_tuple(
            std::string("method1"),
            Tools::MethodTable::Void_fn(
                [this](void *){ method1(); }
            )
        ),
        std::make_tuple(
            std::string("method2"),
            Tools::MethodTable::Void_fn(
                [this](void *){
                    method2(
                        std::get<0>({
                            *reinterpret_cast<OCAE_METHOD_PARAM_T(int)>(p)
                        })
                    );
                }
            )
        ),
        // ...
    };
}
```

Here, [OCAE_METHOD_PARAM_T\(\)](#) is a macro that helps ensure that the type being casted to is in the correct format.

The user creating the derived classes will need to ensure that it properly registers all the methods they want to be accessible through this class in the constructors of the derived classes, including grandchildren classes.

4.14.2 Constructor & Destructor Documentation

4.14.2.1 MethodTable() [1/2]

```
OCAE::Tools::MethodTable::MethodTable ( )
```

Default constructor.

4.14.2.2 MethodTable() [2/2]

```
OCAE::Tools::MethodTable::MethodTable (
    MethodList_t const & list )
```

Constructor.

Parameters

<i>list</i>	List of tuples for mapping a string to a function to initialize the internal method table.
-------------	--

4.14.2.3 ~MethodTable()

```
virtual OCAE::Tools::MethodTable::~~MethodTable ( ) [virtual], [default]
```

Default destructor.

4.14.3 Member Function Documentation

4.14.3.1 CallMethod()

```
template<typename... Args>
void OCAE::Tools::MethodTable::CallMethod (
    std::string const & fn,
    Args &&... args ) [inline]
```

Calls a method.

If the provided function name does not exist within the map an exception will be thrown by `std::unordered_map` and the user will need to handle it if desired.

If the method is to return a value, the first parameter must be a reference to a variable that will store the returned value.

Template Parameters

<i>Args</i>	The arguments' types of the given method.
-------------	---

Parameters

<i>fn</i>	The name of the method. If a function matching this name is registered with the table, an exception will be thrown by <code>std::unordered_map</code> and the user will need to handle it if desired.
<i>args</i>	The parameters for the method.

References [CreateMethodList\(\)](#), [RegisterMethod\(\)](#), and [RegisterMethods\(\)](#).

```

00186         {
00187             using tuple = std::tuple<Args...>;
00188             tuple params(std::forward<Args>(args)...);
00189             s_Table.at(this).at(fn)(reinterpret_cast<void*>(&params));
00190         }

```

4.14.3.2 CreateMethodList()

```
virtual MethodList_t OCAE::Tools::MethodTable::CreateMethodList ( ) [protected], [pure virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Implemented in [OCAE::Generator::WAV](#), [OCAE::Modifier::BandPass](#), [OCAE::Modifier::Equalizer](#), [OCAE::Modifier::A↔DSR](#), [OCAE::Modifier::LowPass](#), [OCAE::Modifier::GenericFilter](#), [OCAE::Modifier::Echo](#), [OCAE::Generator::Sine](#), [OCAE::Modifier::Delay](#), [OCAE::Modifier::Gain](#), [OCAE::Generator::Square](#), [OCAE::Generator::Sawtooth](#), [OCAE::Modifier↔::EnvelopeFollower](#), [OCAE::Modifier::ModifierBase](#), [OCAE::Generator::Triangle](#), [OCAE::Generator::GeneratorBase](#), and [OCAE::Generator::Noise](#).

Referenced by [CallMethod\(\)](#).

4.14.3.3 RegisterMethod()

```
void OCAE::Tools::MethodTable::RegisterMethod (
    std::string const & fn_name,
    Void_fn const & fn_obj ) [protected]
```

Registers a single method and its name within the internal method table.

Parameters

<i>fn_name</i>	The name of the function.
<i>fn_obj</i>	The callable function object.

Referenced by [CallMethod\(\)](#).

4.14.3.4 RegisterMethods()

```
void OCAE::Tools::MethodTable::RegisterMethods (
    MethodList_t const & list ) [protected]
```

Registers a list of methods and their names within the internal method table.

Parameters

<i>list</i>	A list of methods and names to be added.
-------------	--

Referenced by [CallMethod\(\)](#), and [OCAE::Generator::GeneratorBase::GeneratorBase\(\)](#).

The documentation for this class was generated from the following file:

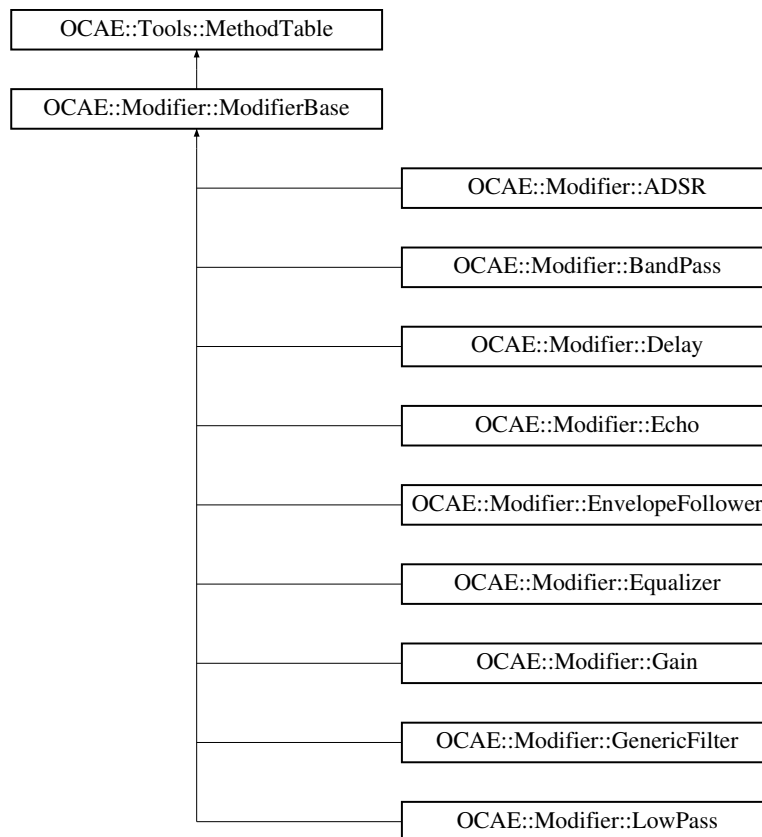
- [MethodTable.hpp](#)

4.15 OCAE::Modifier::ModifierBase Class Reference

The base Modifier class that all modifiers should inherit from.

```
#include <ModifierBase.hpp>
```

Inheritance diagram for OCAE::Modifier::ModifierBase:



Public Member Functions

- [ModifierBase](#) ([ModifierBase](#) const &other)=delete
Copy constructor. Deleted.
- [ModifierBase](#) ([ModifierBase](#) &&other)=default
Default move constructor.
- virtual [~ModifierBase](#) ()=default
Default destructor.
- [ModifierBase](#) & [operator=](#) ([ModifierBase](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [ModifierBase](#) & [operator=](#) ([ModifierBase](#) &&rhs)=default
Default move assignment operator.
- virtual [StereoData Process](#) ([StereoData](#) const &input)
Takes input sample and filters it, returning the result.
- virtual bool [IsBase](#) ()
Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Protected Member Functions

- [ModifierBase](#) ()
Default constructor.
- virtual [Tools::MethodTable::MethodList_t CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Friends

- class [ModifierFactory](#)
Add the factory as a friend so it can construct [ModifierBase](#) objects.

Additional Inherited Members

4.15.1 Detailed Description

The base Modifier class that all modifiers should inherit from.

There are a few functions that should be overridden by derived classes, but are also implemented here for default behavior: Process IsBase (This function will likely be removed in the future) CreateMethodList

See their individual documentation for more info.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 [ModifierBase\(\)](#) [1/3]

```
OCAE::Modifier::ModifierBase::ModifierBase (
    ModifierBase const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.15.2.2 [ModifierBase\(\)](#) [2/3]

```
OCAE::Modifier::ModifierBase::ModifierBase (
    ModifierBase && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.15.2.3 ~ModifierBase()

```
virtual OCAE::Modifier::ModifierBase::~~ModifierBase ( ) [virtual], [default]
```

Default destructor.

4.15.2.4 ModifierBase() [3/3]

```
OCAE::Modifier::ModifierBase::ModifierBase ( ) [inline], [protected]
```

Default constructor.

```
00153 : MethodTable() { RegisterMethods(CreateMethodList()); };
```

4.15.3 Member Function Documentation

4.15.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Modifier::ModifierBase::CreateMethodList ( ) [inline], [protected], [virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Implements [OCAE::Tools::MethodTable](#).

Reimplemented in [OCAE::Modifier::BandPass](#), [OCAE::Modifier::Equalizer](#), [OCAE::Modifier::ADSR](#), [OCAE::Modifier::LowPass](#), [OCAE::Modifier::GenericFilter](#), [OCAE::Modifier::Echo](#), [OCAE::Modifier::Delay](#), [OCAE::Modifier::Gain](#), and [OCAE::Modifier::EnvelopeFollower](#).

References [OCAE_TYPEDEF_SHARED](#).

```
00165 { return {}; };
```

4.15.3.2 IsBase()

```
virtual bool OCAE::Modifier::ModifierBase::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object calling this function is a [ModifierBase](#) or not.

Returns

True for this class, false for any derived class.

Reimplemented in [OCAE::Modifier::BandPass](#), [OCAE::Modifier::Equalizer](#), [OCAE::Modifier::ADSR](#), [OCAE::Modifier::LowPass](#), [OCAE::Modifier::Echo](#), [OCAE::Modifier::Delay](#), [OCAE::Modifier::GenericFilter](#), [OCAE::Modifier::Gain](#), and [OCAE::Modifier::EnvelopeFollower](#).

```
00140 { return true; };
```

4.15.3.3 operator=() [1/2]

```
ModifierBase& OCAE::Modifier::ModifierBase::operator= (
    ModifierBase const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.15.3.4 operator=() [2/2]

```
ModifierBase& OCAE::Modifier::ModifierBase::operator= (
    ModifierBase && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.15.3.5 Process()

```
virtual StereoData OCAE::Modifier::ModifierBase::Process (
    StereoData const & input ) [inline], [virtual]
```

Takes input sample and filters it, returning the result.

Parameters

<i>input</i>	The input sample.
--------------	-------------------

Returns

The filtered sample.

Reimplemented in [OCAE::Modifier::BandPass](#), [OCAE::Modifier::Equalizer](#), [OCAE::Modifier::ADSR](#), [OCAE::Modifier::LowPass](#), [OCAE::Modifier::Echo](#), [OCAE::Modifier::Delay](#), [OCAE::Modifier::GenericFilter](#), [OCAE::Modifier::Gain](#), and [OCAE::Modifier::EnvelopeFollower](#).

```
00130 { return input; };
```

The documentation for this class was generated from the following file:

- [ModifierBase.hpp](#)

4.16 OCAE::Modifier::ModifierFactory Class Reference

Factory class for constructing audio filters (Modifiers).

```
#include <ModifierFactory.hpp>
```

Public Types

- using [ZeroContainer](#) = [GenericFilter::ZeroContainer](#)
Container used for coefficients of zeros of a filter in [GenericFilter](#).
- using [PoleContainer](#) = [GenericFilter::PoleContainer](#)
Container used for coefficients of poles of a filter in [GenericFilter](#).

Public Member Functions

- `~ModifierFactory ()=delete`
Destructor. Deleted to ensure that an object can never be created.

Static Public Member Functions

- static `ModifierBasePtr CreateBase ()`
Creates an empty modifier which will simply forward any input it receives to its output.
- static `ADSRPtr CreateADSR (Math_t attack, Math_t decay, Math_t sustain, Math_t release)`
Creates a modifier for an [ADSR](#) envelope.
- static `BandPassPtr CreateBandPass (Math_t lower, Math_t upper)`
Creates a bandpass filter.
- static `DelayPtr CreateDelay (Math_t seconds)`
Creates a delay filter.
- static `EchoPtr CreateEcho (Math_t delay_seconds, Math_t decay_ratio)`
Creates an echo filter.
- static `EnvelopeFollowerPtr CreateEnvelopeFollower (Math_t lower=Math_t(20), Math_t upper=Math_t(20000))`
Creates an envelope follower filter.
- static `EqualizerPtr CreateEqualizer (uint32_t band_count=2, Math_t lower=20, Math_t upper=20000)`
Creates an equalizer filter.
- static `GainPtr CreateGain (Math_t gain=OCAE_DEFAULT_GAIN)`
Creates a gain filter.
- static `GenericFilterPtr CreateGenericFilter (ZeroContainer const &zeros, PoleContainer const &poles)`
Creates a generic filter.
- static `LowPassPtr CreateLowPass (Math_t cutoff, Math_t resonance=0)`
Creates a low pass filter.

4.16.1 Detailed Description

Factory class for constructing audio filters (Modifiers).

4.16.2 Constructor & Destructor Documentation

4.16.2.1 ~ModifierFactory()

```
OCAE::Modifier::ModifierFactory::~ModifierFactory ( ) [delete]
```

Destructor. Deleted to ensure that an object can never be created.

4.16.3 Member Function Documentation

4.16.3.1 CreateADSR()

```
static ADSRPtr OCAE::Modifier::ModifierFactory::CreateADSR (
    Math_t attack,
    Math_t decay,
    Math_t sustain,
    Math_t release ) [static]
```

Creates a modifier for an [ADSR](#) envelope.

Parameters

<i>attack</i>	The length of the attack phase in seconds.
<i>decay</i>	The length of the decay phase in seconds.
<i>sustain</i>	The sustain level in dB.
<i>release</i>	The length of the decay phase in seconds.

Returns

The generated modifier object.

4.16.3.2 CreateBandPass()

```
static BandPassPtr OCAE::Modifier::ModifierFactory::CreateBandPass (
    Math_t lower,
    Math_t upper ) [static]
```

Creates a bandpass filter.

Parameters

<i>lower</i>	The lower frequency of the band.
<i>upper</i>	The upper frequency of the band.

Returns

The generated modifier object.

4.16.3.3 CreateBase()

```
static ModifierBasePtr OCAE::Modifier::ModifierFactory::CreateBase ( ) [static]
```

Creates an empty modifier which will simply forward any input it receives to its output.

Returns

The generated modifier object.

4.16.3.4 CreateDelay()

```
static DelayPtr OCAE::Modifier::ModifierFactory::CreateDelay (
    Math_t seconds ) [static]
```

Creates a delay filter.

Parameters

<i>seconds</i>	The amount of time in seconds to delay for.
----------------	---

Returns

The generated modifier object.

4.16.3.5 CreateEcho()

```
static EchoPtr OCAE::Modifier::ModifierFactory::CreateEcho (
    Math_t delay_seconds,
    Math_t decay_ratio ) [static]
```

Creates an echo filter.

Parameters

<i>delay_seconds</i>	The amount of time between echos in seconds.
<i>decay_ratio</i>	The decay factor of the echo. Value should be in range of [0,1), if it's ≥ 1 or < 0 it will be clamped to the range.

Returns

The generated modifier object.

4.16.3.6 CreateEnvelopeFollower()

```
static EnvelopeFollowerPtr OCAE::Modifier::ModifierFactory::CreateEnvelopeFollower (
    Math_t lower = Math_t(20),
    Math_t upper = Math_t(20000) ) [static]
```

Creates an envelope follower filter.

Parameters

<i>lower</i>	The lower end of frequencies to follow. Defaults to 20Hz for normal human hearing range.
<i>upper</i>	The upper end of frequencies to follow. Defaults to 20kHz for normal human hearing range.

Returns

The generated modifier object.

4.16.3.7 CreateEqualizer()

```
static EqualizerPtr OCAE::Modifier::ModifierFactory::CreateEqualizer (
    uint32_t band_count = 2,
    Math_t lower = 20,
    Math_t upper = 20000 ) [static]
```

Creates an equalizer filter.

Parameters

<i>band_count</i>	The number of bands in the equalizer. Defaults to 2.
<i>lower</i>	The lowest frequency of the equalizer. Defaults to 20Hz.
<i>upper</i>	The highest frequency of the equalizer. Defaults to 20kHz.

Returns

The generated modifier object.

4.16.3.8 CreateGain()

```
static GainPtr OCAE::Modifier::ModifierFactory::CreateGain (
    Math_t gain = OCAE_DEFAULT_GAIN ) [static]
```

Creates a gain filter.

Parameters

<i>gain</i>	The gain to amplify the signal by. Value may be negative.
-------------	---

Returns

The generated modifier object.

4.16.3.9 CreateGenericFilter()

```
static GenericFilterPtr OCAE::Modifier::ModifierFactory::CreateGenericFilter (
    ZeroContainer const & zeros,
    PoleContainer const & poles ) [static]
```

Creates a generic filter.

Parameters

<i>zeros</i>	The list of coefficients for the zeros of the filter.
<i>poles</i>	The list of coefficients for the poles of the filter.

Returns

The generated modifier object.

4.16.3.10 CreateLowPass()

```
static LowPassPtr OCAE::Modifier::ModifierFactory::CreateLowPass (
    Math_t cutoff,
    Math_t resonance = 0 ) [static]
```

Creates a low pass filter.

Parameters

<i>cutoff</i>	The cutoff frequency of the filter.
<i>resonance</i>	The resonance of the filter at the cutoff frequency. Should be in the range of [0, 1], if the value is outside of this, it will be clamped to the range. Defaults to 0 for no resonance

Returns

The generated modifier object.

The documentation for this class was generated from the following file:

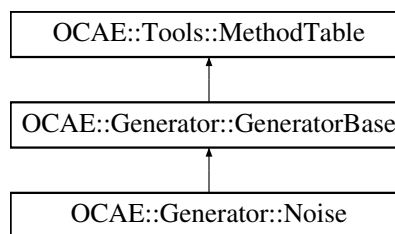
- [ModifierFactory.hpp](#)

4.17 OCAE::Generator::Noise Class Reference

Generates white noise.

```
#include <Noise.hpp>
```

Inheritance diagram for OCAE::Generator::Noise:



Public Member Functions

- [Noise](#) ([Noise](#) const &other)=delete
Copy constructor. Deleted.
- [Noise](#) ([Noise](#) &&other)=default
Default move constructor.
- virtual [~Noise](#) ()=default
Default destructor.
- [Noise](#) & [operator=](#) ([Noise](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [Noise](#) & [operator=](#) ([Noise](#) &&rhs)=default
Default move assignment operator.
- virtual [StereoData Process](#) (void)
Calculates the sample. For the base class this is simply 0.
- virtual bool [IsBase](#) ()
Returns boolean for if the object is a [GeneratorBase](#) or not.

Protected Member Functions

- [Noise](#) ()
Constructor.
- virtual [Tools::MethodTable::MethodList_t CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Attributes

- `std::uniform_real_distribution< float >` [m_Distribution](#)
Distribution for random value generation.
- `std::mt19937` [m_Engine](#)
Random value engine.

Friends

- class [GeneratorFactory](#)
Add the factory as a friend so it can construct [Noise](#) objects.

Additional Inherited Members

4.17.1 Detailed Description

Generates white noise.

4.17.2 Constructor & Destructor Documentation

4.17.2.1 [Noise](#)() [1/3]

```
OCAE::Generator::Noise::Noise (
    Noise const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.17.2.2 Noise() [2/3]

```
OCAE::Generator::Noise::Noise (
    Noise && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.17.2.3 ~Noise()

```
virtual OCAE::Generator::Noise::~Noise ( ) [virtual], [default]
```

Default destructor.

4.17.2.4 Noise() [3/3]

```
OCAE::Generator::Noise::Noise ( ) [protected]
```

Constructor.

Referenced by [lsBase\(\)](#).

4.17.3 Member Function Documentation

4.17.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Generator::Noise::CreateMethodList ( ) [inline],
[protected], [virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

References [OCAE_TYPEDEF_SHARED](#).

```
00147 { return {}; };
```

4.17.3.2 IsBase()

```
virtual bool OCAE::Generator::Noise::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object is a [GeneratorBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

References [Noise\(\)](#).

```
00122 { return false; };
```

4.17.3.3 operator=() [1/2]

```
Noise& OCAE::Generator::Noise::operator= (
    Noise const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.17.3.4 operator=() [2/2]

```
Noise& OCAE::Generator::Noise::operator= (
    Noise && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.17.3.5 Process()

```
virtual StereoData OCAE::Generator::Noise::Process (
    void ) [virtual]
```

Calculates the sample. For the base class this is simply 0.

Returns

The stereo sample data.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

The documentation for this class was generated from the following file:

- [Noise.hpp](#)

4.18 OCAE::Tools::Resampler Class Reference

Class for taking audio data of one sampling rate and translating it to another sampling rate.

```
#include <Resampler.hpp>
```

Public Member Functions

- [Resampler](#) (std::vector< [StereoData](#) > const &AudioData, int32_t SourceSampleRate, uint64_t LoopStart=0, uint64_t LoopEnd=0)
Constructor for the resampler. If the resampler is set up to loop, the range of the looping is [LoopStart, LoopEnd).
- void [SetPlaybackSpeed](#) ([Math_t](#) playback_speed=1.0)
Sets the playback speed. 1.0 is original playback speed.
- [StereoData Process](#) ()
Sends a single sample to [Core::Driver](#) for output to the OS.

Private Types

- using [Index_t](#) = [Math_t](#)
Type used for fractional indexing.

Private Attributes

- `std::vector< StereoData > m_Data`
The original audio data.
- `Index_t m_Index`
The index for tracking position within the audio data.
- `Math_t const m_IndexIncrement`
The value to increment the index by.
- `Math_t m_PlaybackSpeed`
The playback speed, allows speeding up and slowing down the data.
- `uint64_t m_LoopStart`
The start position of the loop in samples, if any.
- `uint64_t m_LoopEnd`
The end position of the loop in samples, if any.

4.18.1 Detailed Description

Class for taking audio data of one sampling rate and translating it to another sampling rate.

4.18.2 Constructor & Destructor Documentation

4.18.2.1 Resampler()

```
OCAE::Tools::Resampler::Resampler (
    std::vector< StereoData > const & AudioData,
    int32_t SourceSampleRate,
    uint64_t LoopStart = 0,
    uint64_t LoopEnd = 0 )
```

Constructor for the resampler. If the resampler is set up to loop, the range of the looping is [LoopStart, LoopEnd).

Parameters

<i>AudioData</i>	A const reference to the audio data.
<i>SourceSampleRate</i>	The sample rate of the source data.
<i>LoopStart</i>	The sample to start looping from. Defaults to 0.
<i>LoopEnd</i>	The sample at the loop point to loop back to LoopStart. Defaults to 0, which is interpreted as no looping.

4.18.3 Member Function Documentation

4.18.3.1 Process()

`StereoData` OCAE::Tools::Resampler::Process ()

Sends a single sample to [Core::Driver](#) for output to the OS.

Returns

The stereo sample data.

4.18.3.2 SetPlaybackSpeed()

`void` OCAE::Tools::Resampler::SetPlaybackSpeed (
 `Math_t` `playback_speed` = 1.0)

Sets the playback speed. 1.0 is original playback speed.

Parameters

<code>playback_speed</code>	The playback speed
-----------------------------	--------------------

The documentation for this class was generated from the following file:

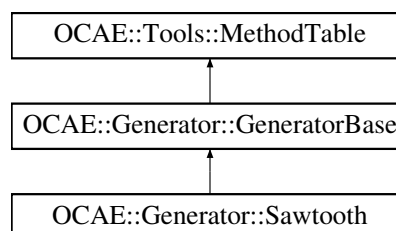
- [Resampler.hpp](#)

4.19 OCAE::Generator::Sawtooth Class Reference

Generates a sawtooth sound.

```
#include <Sawtooth.hpp>
```

Inheritance diagram for OCAE::Generator::Sawtooth:



Public Member Functions

- [Sawtooth](#) ([Sawtooth](#) const &other)=delete
Copy constructor. Deleted.
- [Sawtooth](#) ([Sawtooth](#) &&other)=default
Default move constructor.
- virtual [~Sawtooth](#) ()=default
Default destructor.
- [Sawtooth](#) & [operator=](#) ([Sawtooth](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [Sawtooth](#) & [operator=](#) ([Sawtooth](#) &&rhs)=default
Default move assignment operator.
- void [SetFrequency](#) ([Math_t](#) freq)
Sets a new frequency.
- [Math_t](#) [GetFrequency](#) () const
Gets the current frequency.
- virtual [StereoData Process](#) (void)
Processes and returns the next sample.
- virtual bool [IsBase](#) ()
Returns boolean for if the object is a [GeneratorBase](#) or not.

Private Member Functions

- [Sawtooth](#) ([Math_t](#) freq)
Constructor.
- virtual [Tools::MethodTable::MethodList_t](#) [CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Attributes

- [Math_t](#) [m_lrate](#)
Combination of the sampling rate and desired frequency.
- [Math_t](#) [m_inc](#)
Sample to sample increment value.

Friends

- class [GeneratorFactory](#)
Add the factory as a friend so it can construct [Sawtooth](#) objects.

Additional Inherited Members

4.19.1 Detailed Description

Generates a sawtooth sound.

4.19.2 Constructor & Destructor Documentation

4.19.2.1 Sawtooth() [1/3]

```
OCAE::Generator::Sawtooth::Sawtooth (
    Sawtooth const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

Referenced by [lsBase\(\)](#).

4.19.2.2 Sawtooth() [2/3]

```
OCAE::Generator::Sawtooth::Sawtooth (
    Sawtooth && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.19.2.3 ~Sawtooth()

```
virtual OCAE::Generator::Sawtooth::~~Sawtooth ( ) [virtual], [default]
```

Default destructor.

4.19.2.4 Sawtooth() [3/3]

```
OCAE::Generator::Sawtooth::Sawtooth (
    Math\_t freq ) [private]
```

Constructor.

Parameters

<i>freq</i>	The frequency for the generator.
-------------	----------------------------------

4.19.3 Member Function Documentation

4.19.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList\_t OCAE::Generator::Sawtooth::CreateMethodList ( ) [private],  
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

Referenced by [IsBase\(\)](#).

4.19.3.2 GetFrequency()

```
Math\_t OCAE::Generator::Sawtooth::GetFrequency ( ) const
```

Gets the current frequency.

Returns

The current frequency.

4.19.3.3 IsBase()

```
virtual bool OCAE::Generator::Sawtooth::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object is a [GeneratorBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

References [CreateMethodList\(\)](#), [OCAE_TYPEDEF_SHARED](#), and [Sawtooth\(\)](#).

```
00138 { return false; };
```

4.19.3.4 operator=() [1/2]

```
Sawtooth& OCAE::Generator::Sawtooth::operator= (
    Sawtooth const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.19.3.5 operator=() [2/2]

```
Sawtooth& OCAE::Generator::Sawtooth::operator= (
    Sawtooth && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.19.3.6 Process()

```
virtual StereoData OCAE::Generator::Sawtooth::Process (
    void ) [virtual]
```

Processes and returns the next sample.

Returns

The stereo sample data.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

4.19.3.7 SetFrequency()

```
void OCAE::Generator::Sawtooth::SetFrequency (
    Math_t freq )
```

Sets a new frequency.

Parameters

<i>freq</i>	The new frequency.
-------------	--------------------

The documentation for this class was generated from the following file:

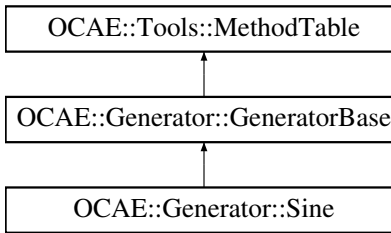
- [Sawtooth.hpp](#)

4.20 OCAE::Generator::Sine Class Reference

Generates sine data at the given frequency.

```
#include <Sine.hpp>
```

Inheritance diagram for OCAE::Generator::Sine:



Public Member Functions

- [Sine](#) ([Sine](#) const &other)=delete
Copy constructor. Deleted.
- [Sine](#) ([Sine](#) &&other)=default
Default move constructor.
- virtual [~Sine](#) ()=default
Destructor.
- [Sine](#) & [operator=](#) ([Sine](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [Sine](#) & [operator=](#) ([Sine](#) &&rhs)=default
Default move assignment operator.
- void [SetFrequency](#) ([Math_t](#) freq)
Sets the frequency to a new value.
- [Math_t](#) [GetFrequency](#) () const
Gets the current frequency.
- virtual [StereoData Process](#) (void)
Processes and returns the next sample.
- virtual bool [IsBase](#) ()
Returns boolean for if the object is a [GeneratorBase](#) or not.

Protected Member Functions

- [Sine](#) ([Math_t](#) freq)
Creates an object that outputs a simple sine wave without using inefficient functions like std::sin.
- virtual [Tools::MethodTable::MethodList_t](#) [CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Static Private Member Functions

- static int [SetupWaveTable](#) ()
Sets the default values for the wave table.

Private Attributes

- [Math_t](#) m_A
Value storing the non-integer index increment value.
- [Math_t](#) m_Index
The current index in the wave table to access.

Static Private Attributes

- static [Math_t s_Table](#) [[OCAE_SAMPLE_RATE](#)/10]
Wave table for efficiently calculating sine frequencies.
- static int [dummy](#)
Dummy int used to call `SetupWaveTable` at the beginning of the program.

Friends

- class [GeneratorFactory](#)
Add the factory as a friend so it can construct [Sine](#) objects.

Additional Inherited Members

4.20.1 Detailed Description

Generates sine data at the given frequency.

4.20.2 Constructor & Destructor Documentation

4.20.2.1 [Sine\(\)](#) [1/3]

```
OCAE::Generator::Sine::Sine (
    Sine const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

Referenced by [lsBase\(\)](#).

4.20.2.2 [Sine\(\)](#) [2/3]

```
OCAE::Generator::Sine::Sine (
    Sine && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.20.2.3 ~Sine()

```
virtual OCAE::Generator::Sine::~Sine ( ) [virtual], [default]
```

Destructor.

4.20.2.4 Sine() [3/3]

```
OCAE::Generator::Sine::Sine (
    Math_t freq ) [protected]
```

Creates an object that outputs a simple sine wave without using inefficient functions like std::sin.

Parameters

<i>freq</i>	The frequency for the sine-wave to output at.
-------------	---

4.20.3 Member Function Documentation

4.20.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Generator::Sine::CreateMethodList ( ) [protected],
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

Referenced by [IsBase\(\)](#).

4.20.3.2 GetFrequency()

```
Math_t OCAE::Generator::Sine::GetFrequency ( ) const
```

Gets the current frequency.

Returns

The frequency of the generator.

4.20.3.3 IsBase()

```
virtual bool OCAE::Generator::Sine::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object is a [GeneratorBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

References [CreateMethodList\(\)](#), [OCAE_TYPEDEF_SHARED](#), [SetupWaveTable\(\)](#), and [Sine\(\)](#).

```
00143 { return false; };
```

4.20.3.4 operator=() [1/2]

```
Sine& OCAE::Generator::Sine::operator= (
    Sine const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.20.3.5 operator=() [2/2]

```
Sine& OCAE::Generator::Sine::operator= (  
    Sine && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.20.3.6 Process()

```
virtual StereoData OCAE::Generator::Sine::Process (  
    void ) [virtual]
```

Processes and returns the next sample.

Returns

The stereo sample data.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

4.20.3.7 SetFrequency()

```
void OCAE::Generator::Sine::SetFrequency (  
    Math_t freq )
```

Sets the frequency to a new value.

Parameters

<i>freq</i>	The new frequency.
-------------	--------------------

4.20.3.8 SetupWaveTable()

```
static int OCAE::Generator::Sine::SetupWaveTable ( ) [static], [private]
```

Sets the default values for the wave table.

Returns

Dummy value to assign to the dummy static variable that allows this function to be called at the start of the program, guaranteeing the table is set up by the first time it is used.

Referenced by [IsBase\(\)](#).

The documentation for this class was generated from the following file:

- [Sine.hpp](#)

4.21 OCAE::Sound::Sound Class Reference

Class for handling Generator and Modifier objects in a more abstract way in conjunction with a Driver.

```
#include <Sound.hpp>
```

Public Types

- using [BlockList](#) = std::deque< BlockPtr >
Alias for a deque of BlockPtrs.
- using [Graph](#) = std::map< BlockPtr, [BlockList](#) >
Alias for the structure that represents the graph blocks that make up this [Sound](#).

Public Member Functions

- [Sound](#) ([Math_t](#) input_gain=[Math_t](#)(1.0), [Math_t](#) output_gain=[Math_t](#)(1.0))
Default constructor.
- [Sound](#) ([Sound](#) const &other)=delete
Deleted copy constructor.
- [Sound](#) ([Sound](#) &&other) noexcept
Move constructor. NOTE: The constructed sound will not be registered to a driver, even if the sound being moved is.
- [~Sound](#) ()=default
Default destructor.
- [Sound](#) & operator= ([Sound](#) const &rhs)=delete
Deleted copy assignment operator.
- [Sound](#) & operator= ([Sound](#) &&rhs) noexcept
Move assignment operator. NOTE: The moved sound will not change it's registration. If it needs to be registered to a different driver, you must handle that yourself.

- BlockPtr const & [GetInputBlock](#) () const
Returns a reference to the input block for use of adding a connection in the internal graph.
- BlockPtr const & [GetOutputBlock](#) () const
Returns a reference to the output block for use of adding a connection in the internal graph.
- void [SetInputGain](#) (Math_t gain)
Sets the gain for the input.
- void [SetOutputGain](#) (Math_t gain)
Sets the gain for the output.
- void [Pause](#) ()
Pauses the processing of this sound.
- void [Unpause](#) ()
Unpauses the processing of this sound.
- void [AddConnection](#) (BlockPtr const &from, BlockPtr const &to)
Adds a connection from the given blocks within the internal directed graph.
- void [RemoveConnection](#) (BlockPtr const &from, BlockPtr const &to)
Removes a connection from the given blocks within the internal directed graph.
- [StereoData Process](#) ([StereoData](#) input)
Processes audio configured in the internal graph, storing the output internally.

Static Public Member Functions

- static void [Register](#) (SoundPtr const &self, Core::DriverPtr const &driver)
Registers the given [Sound](#) object with the given Driver. If this [Sound](#) is already registered to a Driver, it will unregister itself before registering to the new Driver.
- static void [Unregister](#) (SoundPtr const &self)
Unregisters the given [Sound](#) object from it's registered Driver.

Private Member Functions

- void [ProcessOrder](#) ()
Processes the order in which the graph will be traversed.
- void [PrepareGraph](#) (BlockList const &list, BlockList &out)
Parses the given nodes of the graph to process the order the graph will be traversed.

Private Attributes

- [Graph m_Graph](#)
The graph of blocks.
- [BlockList m_ProcessOrder](#)
The order to process blocks in.
- BlockPtr [m_InputGain](#)
Input gain modifier.
- BlockPtr [m_OutputGain](#)
Output gain modifier.
- Core::DriverPtr [m_Driver](#)
Driver the [Sound](#) is registered with.
- uint64_t [m_ID](#)
The ID of this [Sound](#) within the Driver.
- bool [m_IsPaused](#)
Controls if the sound will Process.

4.21.1 Detailed Description

Class for handling Generator and Modifier objects in a more abstract way in conjunction with a Driver.

4.21.2 Constructor & Destructor Documentation

4.21.2.1 Sound() [1/3]

```
OCAE::Sound::Sound::Sound (
    Math_t input_gain = Math_t(1.0),
    Math_t output_gain = Math_t(1.0) )
```

Default constructor.

Parameters

<i>input_gain</i>	The gain for the input samples.
<i>output_gain</i>	The gain for the output samples.

4.21.2.2 Sound() [2/3]

```
OCAE::Sound::Sound::Sound (
    Sound const & other ) [delete]
```

Deleted copy constructor.

Parameters

<i>other</i>	The other sound being copied.
--------------	-------------------------------

4.21.2.3 Sound() [3/3]

```
OCAE::Sound::Sound::Sound (
    Sound && other ) [noexcept]
```

Move constructor. NOTE: The constructed sound will not be registered to a driver, even if the sound being moved is.

Parameters

<i>other</i>	The other sound being moved.
--------------	------------------------------

4.21.3 Member Function Documentation**4.21.3.1 AddConnection()**

```
void OCAE::Sound::Sound::AddConnection (
    BlockPtr const & from,
    BlockPtr const & to )
```

Adds a connection from the given blocks within the internal directed graph.

Parameters

<i>from</i>	The source of the connection.
<i>to</i>	The destination of the connection.

4.21.3.2 GetInputBlock()

```
BlockPtr const& OCAE::Sound::Sound::GetInputBlock ( ) const
```

Returns a reference to the input block for use of adding a connection in the internal graph.

Returns

The input block.

4.21.3.3 GetOutputBlock()

```
BlockPtr const& OCAE::Sound::Sound::GetOutputBlock ( ) const
```

Returns a reference to the output block for use of adding a connection in the internal graph.

Returns

The output block.

4.21.3.4 operator=() [1/2]

```
Sound& OCAE::Sound::Sound::operator= (
    Sound const & rhs ) [delete]
```

Deleted copy assignment operator.

Parameters

<i>rhs</i>	The sound being copied.
------------	-------------------------

Returns

this.

4.21.3.5 operator=() [2/2]

```
Sound& OCAE::Sound::Sound::operator= (
    Sound && rhs ) [noexcept]
```

Move assignment operator. NOTE: The moved sound will not change it's registration. If it needs to be registered to a different driver, you must handle that yourself.

Parameters

<i>rhs</i>	The sound being moved.
------------	------------------------

Returns

this.

4.21.3.6 Pause()

```
void OCAE::Sound::Sound::Pause ( )
```

Pauses the processing of this sound.

4.21.3.7 PrepareGraph()

```
void OCAE::Sound::Sound::PrepareGraph (
    BlockList const & list,
    BlockList & out ) [private]
```

Parses the given nodes of the graph to process the order the graph will be traversed.

Parameters

<i>list</i>	The ordered list to add nodes to.
<i>out</i>	The current list to parse.

4.21.3.8 Process()

```
StereoData OCAE::Sound::Sound::Process (
    StereoData input )
```

Processes audio configured in the internal graph, storing the output internally.

Parameters

<i>input</i>	The input for the Sound .
--------------	---

Returns

The output of the [Sound](#).

4.21.3.9 ProcessOrder()

```
void OCAE::Sound::Sound::ProcessOrder ( ) [private]
```

Processes the order in which the graph will be traversed.

4.21.3.10 Register()

```
static void OCAE::Sound::Sound::Register (
    SoundPtr const & self,
    Core::DriverPtr const & driver ) [static]
```

Registers the given [Sound](#) object with the given Driver. If this [Sound](#) is already registered to a Driver, it will unregister itself before registering to the new Driver.

Parameters

<i>self</i>	The Sound object to register to the given Driver.
<i>driver</i>	The Driver the given Sound object will be registered to.

4.21.3.11 RemoveConnection()

```
void OCAE::Sound::Sound::RemoveConnection (
    BlockPtr const & from,
    BlockPtr const & to )
```

Removes a connection from the given blocks within the internal directed graph.

Parameters

<i>from</i>	The source of the connection.
<i>to</i>	The destination of the connection.

4.21.3.12 SetInputGain()

```
void OCAE::Sound::Sound::SetInputGain (
    Math_t gain )
```

Sets the gain for the input.

Parameters

<i>gain</i>	The new gain.
-------------	---------------

4.21.3.13 SetOutputGain()

```
void OCAE::Sound::Sound::SetOutputGain (
    Math_t gain )
```

Sets the gain for the output.

Parameters

<i>gain</i>	The new gain.
-------------	---------------

4.21.3.14 Unpause()

```
void OCAE::Sound::Sound::Unpause ( )
```

Unpauses the processing of this sound.

4.21.3.15 Unregister()

```
static void OCAE::Sound::Sound::Unregister (
    SoundPtr const & self ) [static]
```

Unregisters the given [Sound](#) object from it's registered Driver.

Parameters

<i>self</i>	The Sound object to unregister.
-------------	---

The documentation for this class was generated from the following file:

- [Sound.hpp](#)

4.22 OCAE::Sound::SoundFactory Class Reference

Class containing functions that will generate [Sound](#) and [Block](#) objects from common inputs.

```
#include <SoundFactory.hpp>
```

Public Member Functions

- [~SoundFactory](#) ()=delete
Deleted destructor, ensuring an instance of this class can never be created.

Static Public Member Functions

- static SoundPtr [CreateEmptySound](#) ()
Creates a [Sound](#) object with no associated generators or modifiers.
- static SoundPtr [CreateBasicGenerator](#) (Generator::GeneratorBasePtr const &g)
Creates a [Sound](#) object from a given generator.
- static SoundPtr [CreateBasicModifier](#) (Modifier::ModifierBasePtr const &m)
Creates a [Sound](#) object from a given modifier. The modifier takes input from the input the [Sound](#) object is given.
- static BlockPtr [CreateBlock](#) (Generator::GeneratorBasePtr const &g)

Creates a [Block](#) object from a given generator.

- static BlockPtr [CreateBlock](#) (Modifier::ModifierBasePtr const &m)

Creates a [Block](#) object from a given modifier.

- static BlockPtr [CreateBlock](#) (Generator::GeneratorBasePtr const &g, Modifier::ModifierBasePtr const &m)

Creates a [Block](#) object from a given generator and modifier.

- static BlockPtr [CreateBlock](#) (Generator::GeneratorBasePtr const &g, Modifier::ModifierBasePtr const &m, [Block](#)↵
::Interaction_f const &interactor)

Creates a [Block](#) object from a given generator, modifier, and interactor.

4.22.1 Detailed Description

Class containing functions that will generate [Sound](#) and [Block](#) objects from common inputs.

4.22.2 Member Function Documentation

4.22.2.1 CreateBasicGenerator()

```
static SoundPtr OCAE::Sound::SoundFactory::CreateBasicGenerator (
    Generator::GeneratorBasePtr const & g ) [static]
```

Creates a [Sound](#) object from a given generator.

Parameters

<i>g</i>	The generator to be processed within this Sound object.
----------	---

Returns

The generated [Sound](#) object wrapped inside a std::shared_ptr.

4.22.2.2 CreateBasicModifier()

```
static SoundPtr OCAE::Sound::SoundFactory::CreateBasicModifier (
    Modifier::ModifierBasePtr const & m ) [static]
```

Creates a [Sound](#) object from a given modifier. The modifier takes input from the input the [Sound](#) object is given.

Parameters

<i>m</i>	The modifier to be processed within this Sound object.
----------	--

Returns

The generated [Sound](#) object wrapped inside a `std::shared_ptr`.

4.22.2.3 CreateBlock() [1/4]

```
static BlockPtr OCAE::Sound::SoundFactory::CreateBlock (
    Generator::GeneratorBasePtr const & g ) [static]
```

Creates a [Block](#) object from a given generator.

When processed, the output of the generator is forwarded to the output of the [Block](#).

Parameters

<i>g</i>	The generator to be held within the Block .
----------	---

Returns

The generated [Block](#) object wrapped inside a `std::shared_ptr`.

4.22.2.4 CreateBlock() [2/4]

```
static BlockPtr OCAE::Sound::SoundFactory::CreateBlock (
    Modifier::ModifierBasePtr const & m ) [static]
```

Creates a [Block](#) object from a given modifier.

When processed, the output of the modifier is forwarded to the output of the [Block](#).

Parameters

<i>m</i>	The modifier to be held within the Block .
----------	--

Returns

The generated [Block](#) object wrapped inside a `std::shared_ptr`.

4.22.2.5 CreateBlock() [3/4]

```
static BlockPtr OCAE::Sound::SoundFactory::CreateBlock (
    Generator::GeneratorBasePtr const & g,
    Modifier::ModifierBasePtr const & m ) [static]
```

Creates a [Block](#) object from a given generator and modifier.

When processed, the output of the generator and modifier are multiplied together and sent to the output of the [Block](#).

Parameters

<i>g</i>	The generator to be held within the Block .
<i>m</i>	The modifier to be held within the Block .

Returns

The generated [Block](#) object wrapped inside a `std::shared_ptr`.

4.22.2.6 CreateBlock() [4/4]

```
static BlockPtr OCAE::Sound::SoundFactory::CreateBlock (
    Generator::GeneratorBasePtr const & g,
    Modifier::ModifierBasePtr const & m,
    Block::Interaction_f const & interactor ) [static]
```

Creates a [Block](#) object from a given generator, modifier, and interactor.

When processed, the output of the generator and modifier are combined together using the given interactor and sent to the output of the [Block](#).

Parameters

<i>g</i>	The generator to be held within the Block .
<i>m</i>	The modifier to be held within the Block .
<i>interactor</i>	Function that will combine outputs from the generator and modifier when the Block is processed.

Returns

The generated [Block](#) object wrapped inside a `std::shared_ptr`.

4.22.2.7 CreateEmptySound()

```
static SoundPtr OCAE::Sound::SoundFactory::CreateEmptySound ( ) [static]
```

Creates a [Sound](#) object with no associated generators or modifiers.

Returns

The generated [Sound](#) object wrapped inside a `std::shared_ptr`.

The documentation for this class was generated from the following file:

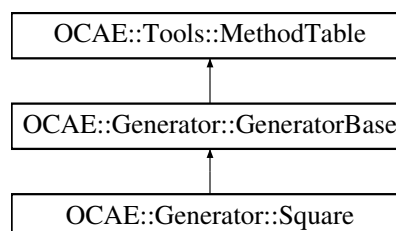
- [SoundFactory.hpp](#)

4.23 OCAE::Generator::Square Class Reference

Generates square wave data at the given frequency.

```
#include <Square.hpp>
```

Inheritance diagram for `OCAE::Generator::Square`:



Public Member Functions

- [Square](#) ([Square](#) const &other)=delete
Copy constructor. Deleted.
- [Square](#) ([Square](#) &&other)=default
Default move constructor.
- virtual [~Square](#) ()=default
Destructor.
- [Square](#) & [operator=](#) ([Square](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [Square](#) & [operator=](#) ([Square](#) &&rhs)=default
Default move assignment operator.
- virtual [StereoData Process](#) (void)
Sends a single sample to [Core::Driver](#) for output to the OS.
- virtual bool [IsBase](#) ()
Returns boolean for if the object is a [GeneratorBase](#) or not.
- void [SetFrequency](#) ([Math_t](#) freq)
Sets the frequency to a new value.
- [Math_t](#) [GetFrequency](#) () const
Gets the frequency.

Private Member Functions

- [Square](#) ([Math_t](#) freq)
Creates an object that outputs a simple square wave without using inefficient functions like std::sin.
- virtual [Tools::MethodTable::MethodList_t](#) [CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Attributes

- [Math_t](#) [m_Ind](#)
Current time value.
- [Math_t](#) [m_Inv](#)
Point of inversion.

Friends

- class [GeneratorFactory](#)
Add the factory as a friend so it can construct [Square](#) objects.

Additional Inherited Members

4.23.1 Detailed Description

Generates square wave data at the given frequency.

4.23.2 Constructor & Destructor Documentation

4.23.2.1 Square() [1/3]

```
OCAE::Generator::Square::Square (
    Square const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.23.2.2 Square() [2/3]

```
OCAE::Generator::Square::Square (
    Square && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.23.2.3 ~Square()

```
virtual OCAE::Generator::Square::~Square ( ) [virtual], [default]
```

Destructor.

4.23.2.4 Square() [3/3]

```
OCAE::Generator::Square::Square (
    Math_t freq ) [private]
```

Creates an object that outputs a simple square wave without using inefficient functions like std::sin.

Parameters

<i>freq</i>	The frequency for the square wav to output at.
-------------	--

4.23.3 Member Function Documentation

4.23.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList\_t OCAE::Generator::Square::CreateMethodList ( ) [private],  
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

4.23.3.2 GetFrequency()

```
Math\_t OCAE::Generator::Square::GetFrequency ( ) const
```

Gets the frequency.

Returns

The frequency.

Referenced by [lsBase\(\)](#).

4.23.3.3 IsBase()

```
virtual bool OCAE::Generator::Square::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object is a [GeneratorBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

References [GetFrequency\(\)](#), and [SetFrequency\(\)](#).

```
00120 { return false; };
```

4.23.3.4 operator=() [1/2]

```
Square& OCAE::Generator::Square::operator= (
    Square const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.23.3.5 operator=() [2/2]

```
Square& OCAE::Generator::Square::operator= (
    Square && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.23.3.6 Process()

```
virtual StereoData OCAE::Generator::Square::Process (
    void ) [virtual]
```

Sends a single sample to [Core::Driver](#) for output to the OS.

Returns

The stereo sample data.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

4.23.3.7 SetFrequency()

```
void OCAE::Generator::Square::SetFrequency (
    Math_t freq )
```

Sets the frequency to a new value.

Parameters

<i>freq</i>	The new frequency.
-------------	--------------------

Referenced by [lsBase\(\)](#).

The documentation for this class was generated from the following file:

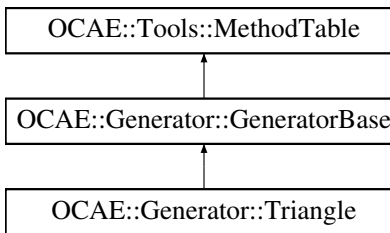
- [Square.hpp](#)

4.24 OCAE::Generator::Triangle Class Reference

[Triangle](#) wave generator.

```
#include <Triangle.hpp>
```

Inheritance diagram for OCAE::Generator::Triangle:



Public Member Functions

- [Triangle](#) ([Triangle](#) const &other)=delete
Copy constructor. Deleted.
- [Triangle](#) ([Triangle](#) &&other)=default
Default move constructor.
- virtual [~Triangle](#) ()=default
Default destructor.
- [Triangle](#) & [operator=](#) ([Triangle](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [Triangle](#) & [operator=](#) ([Triangle](#) &&rhs)=default
Default move assignment operator.
- void [SetFrequency](#) ([Math_t](#) freq)
Sets a new frequency for the generator.
- [Math_t](#) [GetFrequency](#) () const
Gets a frequency.
- virtual [StereoData Process](#) (void)
Calculates the sample. For the base class this is simply 0.
- virtual bool [IsBase](#) ()
Returns boolean for if the object is a [GeneratorBase](#) or not.

Private Member Functions

- [Triangle](#) ([Math_t](#) freq)
Constructor.
- virtual [Tools::MethodTable::MethodList_t](#) [CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Attributes

- [Math_t m_lrate](#)
Combination of the sampling rate and desired frequency.
- [Math_t m_inc](#)
Sample to sample increment value.

Friends

- class [GeneratorFactory](#)

Add the factory as a friend so it can construct [Triangle](#) objects.

Additional Inherited Members

4.24.1 Detailed Description

[Triangle](#) wave generator.

4.24.2 Constructor & Destructor Documentation

4.24.2.1 [Triangle\(\)](#) [1/3]

```
OCAE::Generator::Triangle::Triangle (  
    Triangle const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

Referenced by [lsBase\(\)](#).

4.24.2.2 [Triangle\(\)](#) [2/3]

```
OCAE::Generator::Triangle::Triangle (  
    Triangle && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.24.2.3 ~Triangle()

```
virtual OCAE::Generator::Triangle::~~Triangle ( ) [virtual], [default]
```

Default destructor.

4.24.2.4 Triangle() [3/3]

```
OCAE::Generator::Triangle::Triangle (
    Math_t freq ) [private]
```

Constructor.

Parameters

<i>freq</i>	The frequency for the generator.
-------------	----------------------------------

4.24.3 Member Function Documentation

4.24.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList_t OCAE::Generator::Triangle::CreateMethodList ( ) [private],
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

Referenced by [lsBase\(\)](#).

4.24.3.2 GetFrequency()

```
Math_t OCAE::Generator::Triangle::GetFrequency ( ) const
```

Gets a frequency.

Returns

The frequency.

4.24.3.3 IsBase()

```
virtual bool OCAE::Generator::Triangle::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object is a [GeneratorBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

References [CreateMethodList\(\)](#), [OCAE_TYPEDEF_SHARED](#), and [Triangle\(\)](#).

```
00138 { return false; };
```

4.24.3.4 operator=() [1/2]

```
Triangle& OCAE::Generator::Triangle::operator= (
    Triangle const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.24.3.5 operator=() [2/2]

```
Triangle& OCAE::Generator::Triangle::operator= (
    Triangle && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.24.3.6 Process()

```
virtual StereoData OCAE::Generator::Triangle::Process (
    void ) [virtual]
```

Calculates the sample. For the base class this is simply 0.

Returns

The stereo sample data.1

Reimplemented from [OCAE::Generator::GeneratorBase](#).

4.24.3.7 SetFrequency()

```
void OCAE::Generator::Triangle::SetFrequency (
    Math_t freq )
```

Sets a new frequency for the generator.

Parameters

<i>freq</i>	The new frequency.
-------------	--------------------

The documentation for this class was generated from the following file:

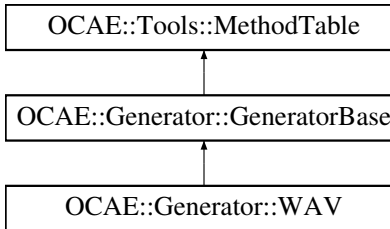
- [Triangle.hpp](#)

4.25 OCAE::Generator::WAV Class Reference

Plays audio from WAVE data.

```
#include <WAV.hpp>
```

Inheritance diagram for OCAE::Generator::WAV:



Public Member Functions

- [WAV](#) ([WAV](#) const &other)=delete
Copy constructor. Deleted.
- [WAV](#) ([WAV](#) &&other)=default
Default move constructor.
- virtual [~WAV](#) ()=default
Default destructor.
- [WAV](#) & [operator=](#) ([WAV](#) const &rhs)=delete
Copy assignment operator. Deleted.
- [WAV](#) & [operator=](#) ([WAV](#) &&rhs)=default
Default move assignment operator.
- virtual [StereoData Process](#) (void)
Sends a single sample to [Core::Driver](#) for output to the OS.
- virtual bool [IsBase](#) ()
Returns boolean for if the object is a [GeneratorBase](#) or not.
- void [ReadFile](#) (std::string const &path)
Reads a file from the disk and parses it for the [WAV](#) data.
- void [LoadWAV](#) (std::vector< char > const &wav_data)
Loads the supplied [WAV](#) data and sets up the object to play the audio data.

Protected Member Functions

- [WAV](#) ()
Default constructor. If no data is provided in calling [WAV::ReadFile](#), then [WAV::Process](#) will only output 0 data.
- [WAV](#) (std::string const &path)
Path to a [WAV](#) file.
- [WAV](#) (std::vector< char > const &wav_data)
std::vector with the contents of a [WAV](#) file.
- [WAV](#) (int argc)
*Integer argc parameter passed into main. Uses the functions in Input.*pp to access the command-line parameters.*
- void [ParseWAV](#) (char const *array, int size)
Parses WAVE data from the given raw data.
- virtual [Tools::MethodTable::MethodList_t CreateMethodList](#) ()
Creates a vector containing the names of functions, and the callable functions themselves.

Private Attributes

- Tools::ResamplerPtr [m_Resampler](#)
Resampler used for resampling input [WAV](#) data to the OCAE's sampling rate.

Friends

- class [GeneratorFactory](#)
Add the factory as a friend so it can construct [GeneratorBase](#) objects.

Additional Inherited Members

4.25.1 Detailed Description

Plays audio from WAVE data.

Supported formats: 8-bit, 16-bit, and 24-bit audio.

4.25.2 Constructor & Destructor Documentation

4.25.2.1 [WAV\(\)](#) [1/6]

```
OCAE::Generator::WAV::WAV (
    WAV const & other ) [delete]
```

Copy constructor. Deleted.

Parameters

<i>other</i>	The other object to be copied.
--------------	--------------------------------

4.25.2.2 [WAV\(\)](#) [2/6]

```
OCAE::Generator::WAV::WAV (
    WAV && other ) [default]
```

Default move constructor.

Parameters

<i>other</i>	The other object to be moved.
--------------	-------------------------------

4.25.2.3 ~WAV()

```
virtual OCAE::Generator::WAV::~~WAV ( ) [virtual], [default]
```

Default destructor.

4.25.2.4 WAV() [3/6]

```
OCAE::Generator::WAV::WAV ( ) [protected]
```

Default constructor. If no data is provided in calling [WAV::ReadFile](#), then [WAV::Process](#) will only output 0 data.

4.25.2.5 WAV() [4/6]

```
OCAE::Generator::WAV::WAV (
    std::string const & path ) [protected]
```

Path to a [WAV](#) file.

Parameters

<i>path</i>	The path.
-------------	-----------

4.25.2.6 WAV() [5/6]

```
OCAE::Generator::WAV::WAV (
    std::vector< char > const & wav_data ) [protected]
```

std::vector with the contents of a [WAV](#) file.

Parameters

<i>wav_data</i>	The WAV data
-----------------	------------------------------

4.25.2.7 WAV() [6/6]

```
OCAE::Generator::WAV::WAV (  
    int argc ) [protected]
```

Integer argc parameter passed into main. Uses the functions in Input.*pp to access the command-line parameters.

Parameters

<i>argc</i>	Parameter passed into main.
-------------	-----------------------------

4.25.3 Member Function Documentation

4.25.3.1 CreateMethodList()

```
virtual Tools::MethodTable::MethodList\_t OCAE::Generator::WAV::CreateMethodList ( ) [protected],  
[virtual]
```

Creates a vector containing the names of functions, and the callable functions themselves.

See [Tools::MethodTable](#) documentation on more info about this system.

Returns

The vector containing callable functions and their names as strings.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

4.25.3.2 IsBase()

```
virtual bool OCAE::Generator::WAV::IsBase ( ) [inline], [virtual]
```

Returns boolean for if the object is a [GeneratorBase](#) or not.

Returns

False.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

References [LoadWAV\(\)](#), and [ReadFile\(\)](#).

```
00126 { return false; };
```

4.25.3.3 LoadWAV()

```
void OCAE::Generator::WAV::LoadWAV (
    std::vector< char > const & wav_data )
```

Loads the supplied [WAV](#) data and sets up the object to play the audio data.

Parameters

<i>wav_data</i>	The WAV data
-----------------	------------------------------

Referenced by [IsBase\(\)](#).

4.25.3.4 operator=() [1/2]

```
WAV& OCAE::Generator::WAV::operator= (
    WAV const & rhs ) [delete]
```

Copy assignment operator. Deleted.

Parameters

<i>rhs</i>	The object to be copied.
------------	--------------------------

Returns

this.

4.25.3.5 operator=() [2/2]

```
WAV& OCAE::Generator::WAV::operator= (
    WAV && rhs ) [default]
```

Default move assignment operator.

Parameters

<i>rhs</i>	The object to be moved.
------------	-------------------------

Returns

this.

4.25.3.6 ParseWAV()

```
void OCAE::Generator::WAV::ParseWAV (
    char const * array,
    int size ) [protected]
```

Parses WAVE data from the given raw data.

NOTE: The data in the array should be the fully RIFF-structured data.

Parameters

<i>array</i>	The raw WAVE data to be parsed.
<i>size</i>	The size of the WAVE data.

4.25.3.7 Process()

```
virtual StereoData OCAE::Generator::WAV::Process (
    void ) [virtual]
```

Sends a single sample to [Core::Driver](#) for output to the OS.

Returns

The stereo sample data.

Reimplemented from [OCAE::Generator::GeneratorBase](#).

4.25.3.8 ReadFile()

```
void OCAE::Generator::WAV::ReadFile (
    std::string const & path )
```

Reads a file from the disk and parses it for the [WAV](#) data.

Parameters

<i>path</i>	The path to the file.
-------------	-----------------------

Referenced by [IsBase\(\)](#).

The documentation for this class was generated from the following file:

- [WAV.hpp](#)

4.26 OCAE::Tools::WAVHeader Struct Reference

A POD structure representing the structure of the header of a WAVE file.

```
#include <WAVHeader.hpp>
```

Public Member Functions

- [WAVHeader](#) (uint16_t af=1, uint16_t cc=2, uint32_t R=[OCAE_SAMPLE_RATE](#), uint16_t bps=16)
Constructor for a WAVE header, with default values for standard 16-bit audio data.
- [~WAVHeader](#) ()=default
Default destructor.

Public Attributes

- uint16_t [AudioFormat](#)
Offset 00 = 1.
- uint16_t [ChannelCount](#)
Offset 02 = 1 or 2.
- uint32_t [SamplingRate](#)
Offset 04 = (ex. 44.1kHz, 48kHz, 96kHz, 192kHz)
- uint32_t [BytesPerSecond](#)
*Offset 08 = SamplingRate * BytesPerSample.*
- uint16_t [BytesPerSample](#)
*Offset 12 = BitsPerSample/8 * ChannelCount.*
- uint16_t [BitsPerSample](#)
Offset 14 = 8 or 16.

4.26.1 Detailed Description

A POD structure representing the structure of the header of a WAVE file.

4.26.2 Constructor & Destructor Documentation

4.26.2.1 WAVHeader()

```
OCAE::Tools::WAVHeader::WAVHeader (
    uint16_t af = 1,
    uint16_t cc = 2,
    uint32_t R = OCAE_SAMPLE_RATE,
    uint16_t bps = 16 )
```

Constructor for a WAVE header, with default values for standard 16-bit audio data.

Parameters

<i>af</i>	The audio format, should generally be left at 1.
<i>cc</i>	The channel count. OCAE uses two-channel audio.
<i>R</i>	The sampling rate. OCAE uses OCAE_SAMPLE_RATE (probably defined as 48kHz).
<i>bps</i>	Bits per audio sample. We are using 16-bit audio as it is all of the quality you should need.

The documentation for this struct was generated from the following file:

- [WAVHeader.hpp](#)

Chapter 5

File Documentation

5.1 ADSR.hpp File Reference

```
#include "../Engine.hpp"
#include "ModifierBase.hpp"
```

Classes

- class [OCAE::Modifier::ADSR](#)
Attack - Decay - Sustain - Release filter.

Functions

- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (ADSR)
Alias for a `std::shared_ptr` instantiated with the [ADSR](#) class.

5.1.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.2 BandPass.hpp File Reference

```
#include "../Engine.hpp"
#include "ModifierBase.hpp"
```

Classes

- class [OCAE::Modifier::BandPass](#)
Bandpass filter.

Functions

- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (BandPass)
Alias for a `std::shared_ptr` instantiated with the [BandPass](#) class.

5.2.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.3 Block.hpp File Reference

```
#include <memory>
#include <type_traits>
#include <vector>
#include "../Engine.hpp"
#include "../Generators/GeneratorBase.hpp"
#include "../Modifiers/ModifierBase.hpp"
```

Classes

- class [OCAE::Sound::Block](#)

This class defines a way of holding a Generator, Modifier and a method of combining the outputs of both of them to produce a single output sample.

Functions

- [OCAE::Sound::OCAE_TYPEDEF_SHARED](#) (Block)

Alias for `std::shared_ptr` instantiated with [Block](#).

5.3.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.4 Core.hpp File Reference

```
#include "Engine.hpp"
#include "Core/Driver.hpp"
```

5.4.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.5 Delay.hpp File Reference

```
#include "../Engine.hpp"
#include <deque>
#include "ModifierBase.hpp"
```

Classes

- class [OCAE::Modifier::Delay](#)
Delay filter.

Functions

- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (Delay)
Alias for a `std::shared_ptr` instantiated with the [Delay](#) class.

5.5.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.6 Driver.hpp File Reference

```
#include <functional>
#include <unordered_map>
#include <memory>
#include "../Engine.hpp"
#include "../Sounds/Sound.hpp"
```

Classes

- class [OCAE::Core::Driver](#)
Handles the calculation of audio samples from different Sounds.

Functions

- [OCAE::Core::OCAE_TYPEDEF_SHARED](#) (Driver)
Typedef for a `std::shared_ptr` instantiated with the [Driver](#) class.

5.6.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.6.2 Function Documentation

5.6.2.1 OCAE_TYPEDEF_SHARED()

```
OCAE::Core::OCAE_TYPEDEF_SHARED (
    Driver )
```

Typedef for a `std::shared_ptr` instantiated with the `Driver` class.

Forwarded alias of `std::shared_ptr` instantiated with `Driver`.

5.7 Echo.hpp File Reference

```
#include "../Engine.hpp"
#include <deque>
#include "ModifierBase.hpp"
```

Classes

- class [OCAE::Modifier::Echo](#)

[Echo](#) IIR filter. Uses output sample for echoing instead of input, creating an infinite impulse response (IIR).

Functions

- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (Echo)

Alias for a `std::shared_ptr` instantiated with the [Echo](#) class.

5.7.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.8 Engine.hpp File Reference

```
#include "Macro.hpp"
#include "Types.hpp"
#include "Util.hpp"
#include "Core.hpp"
#include "Generators.hpp"
#include "Modifiers.hpp"
#include "Sounds.hpp"
#include "Tools.hpp"
```

5.8.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.9 Envelope.hpp File Reference

```
#include "../Engine.hpp"
#include "ModifierBase.hpp"
```

Classes

- class [OCAE::Modifier::EnvelopeFollower](#)
Envelope follower filter. Calculates the gain of the input signal over time.

Functions

- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (EnvelopeFollower)
Alias for a `std::shared_ptr` instantiated with the [ModifierBase](#) class.

5.9.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.10 Equalizer.hpp File Reference

```
#include <vector>
#include "../Engine.hpp"
#include "BandPass.hpp"
#include "ModifierBase.hpp"
```

Classes

- class [OCAE::Modifier::Equalizer](#)
Equalizer filter.

Functions

- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (Equalizer)
Alias for a `std::shared_ptr` instantiated with the [Equalizer](#) class.

5.10.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.11 Gain.hpp File Reference

```
#include "../Engine.hpp"
#include "ModifierBase.hpp"
```


Classes

- class [OCAE::Modifier::Gain](#)

Simple gain filter for amplifying the input signal. The gain value can be negative allowing for inverting the input signal.

Functions

- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (Gain)

Alias for a `std::shared_ptr` instantiated with the [Gain](#) class.

5.11.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.11.2 Function Documentation

5.11.2.1 `OCAE_TYPEDEF_SHARED()`

```
OCAE::Modifier::OCAE_TYPEDEF_SHARED (
    Gain )
```

Alias for a `std::shared_ptr` instantiated with the `Gain` class.

Forwarded alias of `std::shared_ptr` instantiated with `Gain`.

5.12 GeneratorBase.hpp File Reference

```
#include <functional>
#include <unordered_map>
#include <string>
#include "../Engine.hpp"
#include "../Tools/MethodTable.hpp"
#include "GeneratorFactory.hpp"
```

Classes

- class [OCAE::Generator::GeneratorBase](#)

General base class for all generator (sounds) to inherit from. Any derived classes with extra methods that may need to be acquired can be accessed through their setup of the [Tools::MethodTable](#).

Functions

- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (GeneratorBase)

Alias for a `std::shared_ptr` instantiated with the [GeneratorBase](#) class.

5.12.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.13 GeneratorFactory.hpp File Reference

```
#include "../Engine.hpp"
#include <string>
#include <vector>
```

Classes

- class [OCAE::Generator::GeneratorFactory](#)
Creates pointers to generators handled by `std::shared_ptr` to prevent memory leaks.

Functions

- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (GeneratorBase)
Alias for a `std::shared_ptr` instantiated with the [GeneratorBase](#) class.
- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (Noise)
Alias for a `std::shared_ptr` instantiated with the [Noise](#) class.
- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (Sawtooth)
Alias for a `std::shared_ptr` instantiated with the [Sawtooth](#) class.
- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (Sine)
Alias for a `std::shared_ptr` instantiated with the [Sine](#) class.
- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (Square)
Alias for a `std::shared_ptr` instantiated with the [Square](#) class.
- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (Triangle)
Alias for a `std::shared_ptr` instantiated with the [Triangle](#) class.
- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (WAV)
Alias for a `std::shared_ptr` instantiated with the [WAV](#) class.

5.13.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.14 Generators.hpp File Reference

```
#include "Engine.hpp"
#include "Generators/GeneratorFactory.hpp"
#include "Generators/GeneratorBase.hpp"
#include "Generators/Noise.hpp"
#include "Generators/Sawtooth.hpp"
#include "Generators/Sine.hpp"
#include "Generators/Square.hpp"
#include "Generators/Triangle.hpp"
#include "Generators/WAV.hpp"
```

5.14.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.15 GenericFilter.hpp File Reference

```
#include <tuple>
#include <vector>
#include <deque>
#include "../Engine.hpp"
#include "ModifierBase.hpp"
```

Classes

- class [OCAE::Modifier::GenericFilter](#)
Generic audio filter with simple poles.

Functions

- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (GenericFilter)
Alias for a `std::shared_ptr` instantiated with the [GenericFilter](#) class.

5.15.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.16 Input.hpp File Reference

```
#include <string>
#include <vector>
```

Functions

- void [OCAE::Tools::InitOptions](#) (int argc, char *argv[])
Creates a container to hold the command-line options passed into main.
- std::string const & [OCAE::Tools::GetOption](#) (int index)
Returns a const reference to string at the given index.

5.16.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.16.2 Function Documentation

5.16.2.1 GetOption()

```
std::string const& OCAE::Tools::GetOption (
    int index )
```

Returns a const reference to string at the given index.

Parameters

<i>index</i>	The argument index to retrieve.
--------------	---------------------------------

Returns

The string at the given index.

5.16.2.2 InitOptions()

```
void OCAE::Tools::InitOptions (
    int argc,
    char * argv[] )
```

Creates a container to hold the command-line options passed into main.

Parameters

<i>argc</i>	The number of arguments.
<i>argv</i>	Pointer to the array of arguments.

5.17 LowPass.hpp File Reference

```
#include "../Engine.hpp"
#include "ModifierBase.hpp"
```

Classes

- class [OCAE::Modifier::LowPass](#)
3rd Order Butterworth Low Pass filter with resonance.

Functions

- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (LowPass)
Alias for a `std::shared_ptr` instantiated with the [LowPass](#) class.

5.17.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.18 Macro.hpp File Reference

```
#include <fstream>
#include <cmath>
#include <type_traits>
#include <memory>
```

Macros

- `#define OCAE_WRITE_WAV(file, samples) auto _r = Tools::WriteWAV(samples); std::ofstream(file, std::ios_base::binary).write(reinterpret_cast<char*>(_r.data()), std::streamsize(_r.size()))`
Writes an `Track_t` to a file in WAVE format.
- `#define OCAE_SIZEOF_ARRAY(a) (sizeof(a)/sizeof(*a))`
Computes the size of a raw array.
- `#define OCAE_SAMPLE_RATE 48000`
The sample rate OCAE runs at (probably 48kHz)
- `#define OCAE_INC_RATE (1.0/double(OCAE_SAMPLE_RATE))`
Inverse of the sample rate.
- `#define OCAE_DEFAULT_GAIN Math_t(0.5)`
Default amplification of the engine.
- `#define OCAE_MAX_BUFFER (OCAE_SAMPLE_RATE/1000)`
Macro for the maximum buffer size to allow for high performant audio, which is currently defined as 1ms.
- `#define OCAE_EPSILON (1.0/double(1 << 24))`
Macro for the value at which we call the difference between two 64-bit floating point values effectively zero.
- `#define OCAE_EPSILON_F (1.0/float(1 << 16))`
Macro for the value at which we call the difference between two 32-bit floating point values effectively zero.
- `#define OCAE_PI std::acos(-1.0)`
It's uhh, it's Pi, the mathematical constant.
- `#define OCAE_PI2 (2*OCAE_PI)`
*2 * Pi, I hope I don't have to explain further*

- `#define OCAE_LOG_10 std::log(10.0)`
Logarithm of 10, for easy conversion of unknown bases to base 10.
- `#define OCAE_SQRT_HALF std::sqrt(0.5)`
sqrt(0.5) for easy use
- `#define OCAE_DB_TO_LINEAR(dB) std::pow(10.0, dB/20.0)`
Converts logarithmic decibels to linear gain.
- `#define OCAE_LINEAR_TO_DB(g) (20.0*std::log(g)/OCAE_LOG_10)`
Converts linear gain to logarithmic decibels.
- `#define OCAE_MONO_TO_STEREO(x) StereoData(SampleType(Math_t(x)*OCAE_SQRT_HALF),SampleType(Math_t(x)*OCAE_SQRT_HALF))`
Converts monophonic audio sample to stereophonic.
- `#define OCAE_STEREO_TO_MONO(x) SampleType(Math_t(std::get<0>(x) + std::get<1>(x))/OCAE_SQRT_HALF)`
Converts stereophonic audio sample to monophonic.
- `#define OCAE_METHOD_RET_T(t) std::add_lvalue_reference_t<std::remove_const_t<t>>`
Turns the given type into a reference.
- `#define OCAE_METHOD_RET(v) OCAE_METHOD_RET_T(decltype(v))(v)`
Casts the passed object to be a lvalue reference.
- `#define OCAE_METHOD_PARAM_T(t) std::add_lvalue_reference_t<t const>`
Turns the given type into a const reference.
- `#define OCAE_METHOD_PARAM(v) OCAE_METHOD_PARAM_T(decltype(v))(v)`
Casts the passed object to the plain type.
- `#define OCAE_TYPEDEF_SHARED(type) using type##Ptr = std::shared_ptr<type>`
Creates an alias for std::shared_ptr instantiated with the given type.
- `#define OCAE_TO_STR(p) #p`
Creates string from "p". E.g. OCAE_TO_STR(HEAP_SIZE) creates the string "HEAP_SIZE".
- `#define OCAE_PRINT(p) OCAE_TO_STR(p)`
Creates string from what "p" defines. E.g. PRINT(HEAP_SIZE) creates the string "1024" if HEAP_SIZE is defined to 1024
- `#define OCAE_DO_PRAGMA(x)`
Do platform-specific pragma command.
- `#define OCAE_TODO(x)`
Print the to-do message.
- `#define OCAE_UNREFERENCED_PARAMETER(P) (void)(P)`
Clears unused parameter warning.
- `#define OCAE_PUSH_WARNINGS()`
Push warnings.
- `#define OCAE_MSVC_DISABLE_WARNING(x)`
Disable given VC++ warning.
- `#define OCAE_CLANG_DISABLE_WARNING(x)`
Disable given clang warning.
- `#define OCAE_GCC_DISABLE_WARNING(x)`
Disable given gcc warning.
- `#define OCAE_POP_WARNINGS()`
POP_WARNINGS.

5.18.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.19 MethodTable.hpp File Reference

```
#include <string>
#include <tuple>
#include <type_traits>
#include <unordered_map>
#include <utility>
#include <vector>
#include "../Engine.hpp"
```

Classes

- class [OCAE::Tools::MethodTable](#)

The purpose of this class is to create a simple interface for calling methods from an object of an unknown type.

5.19.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.20 ModifierBase.hpp File Reference

```
#include <cstring>
#include <type_traits>
#include "../Engine.hpp"
#include "../Tools/MethodTable.hpp"
```

Classes

- class [OCAE::Modifier::ModifierBase](#)
The base Modifier class that all modifiers should inherit from.

Functions

- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (ModifierBase)
Alias for a `std::shared_ptr` instantiated with the [ModifierBase](#) class.

5.20.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: OCAE

Copyright

Copyright © 2018 Chyler

5.21 ModifierFactory.hpp File Reference

```
#include "../Engine.hpp"
#include "GenericFilter.hpp"
```

Classes

- class [OCAE::Modifier::ModifierFactory](#)
Factory class for constructing audio filters (Modifiers).

Functions

- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (ModifierBase)
Alias for a `std::shared_ptr` instantiated with the [ModifierBase](#) class.
- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (ADSR)
Alias for a `std::shared_ptr` instantiated with the [ADSR](#) class.
- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (BandPass)
Alias for a `std::shared_ptr` instantiated with the [BandPass](#) class.
- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (Delay)
Alias for a `std::shared_ptr` instantiated with the [Delay](#) class.
- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (Echo)
Alias for a `std::shared_ptr` instantiated with the [Echo](#) class.
- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (EnvelopeFollower)
Alias for a `std::shared_ptr` instantiated with the [ModifierBase](#) class.
- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (Equalizer)
Alias for a `std::shared_ptr` instantiated with the [Equalizer](#) class.
- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (Gain)
Alias for a `std::shared_ptr` instantiated with the [Gain](#) class.
- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (GenericFilter)
Alias for a `std::shared_ptr` instantiated with the [GenericFilter](#) class.
- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (LowPass)
Alias for a `std::shared_ptr` instantiated with the [LowPass](#) class.

5.21.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.21.2 Function Documentation

5.21.2.1 OCAE_TYPEDEF_SHARED()

```
OCAE::Modifier::OCAE_TYPEDEF_SHARED (
    Gain )
```

Alias for a `std::shared_ptr` instantiated with the `Gain` class.

Forwarded alias of `std::shared_ptr` instantiated with `Gain`.

5.22 Modifiers.hpp File Reference

```
#include "Modifiers/ModifierBase.hpp"
#include "Modifiers/ModifierFactory.hpp"
#include "Modifiers/ADSR.hpp"
#include "Modifiers/BandPass.hpp"
#include "Modifiers/Delay.hpp"
#include "Modifiers/Echo.hpp"
#include "Modifiers/Envelope.hpp"
#include "Modifiers/Equalizer.hpp"
#include "Modifiers/Gain.hpp"
#include "Modifiers/GenericFilter.hpp"
#include "Modifiers/LowPass.hpp"
```

5.22.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.23 Noise.hpp File Reference

```
#include "../Engine.hpp"
#include <random>
#include "GeneratorBase.hpp"
```

Classes

- class [OCAE::Generator::Noise](#)
Generates white noise.

Functions

- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (Noise)
Alias for a `std::shared_ptr` instantiated with the [Noise](#) class.

5.23.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.24 Resampler.hpp File Reference

```
#include <memory>
#include <vector>
#include "../Engine.hpp"
```

Classes

- class [OCAE::Tools::Resampler](#)

Class for taking audio data of one sampling rate and translating it to another sampling rate.

Functions

- [OCAE::Tools::OCAE_TYPEDEF_SHARED](#) (Resampler)

Alias for a `std::shared_ptr` instantiated with the [Resampler](#) class.

5.24.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.25 Sawtooth.hpp File Reference

```
#include "../Engine.hpp"
#include "GeneratorBase.hpp"
```

Classes

- class [OCAE::Generator::Sawtooth](#)

Generates a sawtooth sound.

Functions

- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (Sawtooth)

Alias for a `std::shared_ptr` instantiated with the [Sawtooth](#) class.

5.25.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.26 Sine.hpp File Reference

```
#include "../Engine.hpp"
#include "GeneratorBase.hpp"
```

Classes

- class [OCAE::Generator::Sine](#)
Generates sine data at the given frequency.

Functions

- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (Sine)
Alias for a `std::shared_ptr` instantiated with the [Sine](#) class.

5.26.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.27 Sound.hpp File Reference

```
#include <deque>
#include <map>
#include <memory>
#include "../Engine.hpp"
#include "Block.hpp"
```

Classes

- class [OCAE::Sound::Sound](#)

Class for handling Generator and Modifier objects in a more abstract way in conjunction with a Driver.

Functions

- [OCAE::Sound::OCAE_TYPEDEF_SHARED](#) (Sound)
Forwarded alias of `std::shared_ptr` instantiated with [Sound](#).
- [OCAE::Core::OCAE_TYPEDEF_SHARED](#) (Driver)
Typedef for a `std::shared_ptr` instantiated with the [Driver](#) class.
- [OCAE::Modifier::OCAE_TYPEDEF_SHARED](#) (Gain)
Alias for a `std::shared_ptr` instantiated with the [Gain](#) class.

5.27.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.27.2 Function Documentation

5.27.2.1 OCAE_TYPEDEF_SHARED() [1/2]

```
OCAE::Core::OCAE_TYPEDEF_SHARED (
    Driver )
```

Typedef for a std::shared_ptr instantiated with the Driver class.

Forwarded alias of std::shared_ptr instantiated with Driver.

5.27.2.2 OCAE_TYPEDEF_SHARED() [2/2]

```
OCAE::Modifier::OCAE_TYPEDEF_SHARED (
    Gain )
```

Alias for a std::shared_ptr instantiated with the Gain class.

Forwarded alias of std::shared_ptr instantiated with Gain.

5.28 SoundFactory.hpp File Reference

```
#include "../Engine.hpp"
#include "../Modifiers/ModifierBase.hpp"
#include "../Generators/GeneratorBase.hpp"
#include "Sound.hpp"
```

Classes

- class [OCAE::Sound::SoundFactory](#)

Class containing functions that will generate [Sound](#) and [Block](#) objects from common inputs.

5.28.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.29 Sounds.hpp File Reference

```
#include "Sounds/Sound.hpp"  
#include "Sounds/SoundFactory.hpp"  
#include "Sounds/Block.hpp"
```

5.29.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.30 Square.hpp File Reference

```
#include "../Engine.hpp"  
#include "GeneratorBase.hpp"
```

Classes

- class [OCAE::Generator::Square](#)
Generates square wave data at the given frequency.

Functions

- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (Square)
Alias for a `std::shared_ptr` instantiated with the [Square](#) class.

5.30.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.31 Tools.hpp File Reference

```
#include "Tools/Input.hpp"
#include "Tools/MethodTable.hpp"
#include "Tools/Resampler.hpp"
#include "Tools/WAVHeader.hpp"
#include "Tools/WAVWriter.hpp"
```

5.31.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.32 Triangle.hpp File Reference

```
#include "../Engine.hpp"  
#include "GeneratorBase.hpp"
```

Classes

- class [OCAE::Generator::Triangle](#)
[Triangle](#) wave generator.

Functions

- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (Triangle)
Alias for a `std::shared_ptr` instantiated with the [Triangle](#) class.

5.32.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.33 Types.hpp File Reference

```
#include <cstdint>  
#include <functional>  
#include <memory>  
#include <utility>
```

Typedefs

- using `OCAE::Math_t` = double
Define the type used for mathematics operations.
- using `OCAE::SampleType` = float
Define the type used for sample types.
- using `OCAE::StereoData` = std::pair< SampleType, SampleType >
Define the type used for stereo audio data.
- using `OCAE::Track_t` = std::vector< StereoData >
Define the type used for stereo audio tracks.

5.33.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.34 Util.hpp File Reference

```
#include "Engine.hpp"
```

Functions

- constexpr SampleType & `OCAE::Left` (StereoData &s)
Returns the left audio sample from a stereo data pair.
- constexpr SampleType const & `OCAE::Left` (StereoData const &s)
Returns the left audio sample from a stereo data pair.
- constexpr SampleType & `OCAE::Right` (StereoData &s)
Returns the right audio sample from a stereo data pair.
- constexpr SampleType const & `OCAE::Right` (StereoData const &s)
Returns the right audio sample from a stereo data pair.

5.34.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.34.2 Function Documentation

5.34.2.1 `Left()` [1/2]

```
constexpr SampleType& OCAE::Left (
    StereoData & s )
```

Returns the left audio sample from a stereo data pair.

Parameters

<code>s</code>	The stereo audio sample.
----------------	--------------------------

Returns

The left audio sample.

References [OCAE::Left\(\)](#).

Referenced by [OCAE::Left\(\)](#), and [OCAE::Sound::Block::PrimeInput\(\)](#).

```
00040      {
00041          return s.first; //std::get<0>(s);
00042      }
```

5.34.2.2 Left() [2/2]

```
constexpr SampleType const& OCAE::Left (
    StereoData const & s )
```

Returns the left audio sample from a stereo data pair.

Parameters

s	The stereo audio sample.
---	--------------------------

Returns

The left audio sample.

References [OCAE::Left\(\)](#).

```
00055     {
00056         return s.first; //std::get<0>(s);
00057     }
```

5.34.2.3 Right() [1/2]

```
constexpr SampleType& OCAE::Right (
    StereoData & s )
```

Returns the right audio sample from a stereo data pair.

Parameters

s	The stereo audio sample.
---	--------------------------

Returns

The right audio sample.

References [OCAE::Right\(\)](#).

Referenced by [OCAE::Sound::Block::PrimeInput\(\)](#), and [OCAE::Right\(\)](#).

```
00070     {
00071         return s.second; //std::get<1>(s);
00072     }
```

5.34.2.4 Right() [2/2]

```
constexpr SampleType const& OCAE::Right (
    StereoData const & s )
```

Returns the right audio sample from a stereo data pair.

Parameters

s	The stereo audio sample.
----------	--------------------------

Returns

The right audio sample.

References [OCAE::Right\(\)](#).

```
00085     {
00086         return s.second; //std::get<1>(s);
00087     }
```

5.35 WAV.hpp File Reference

```
#include <string>
#include <memory>
#include <vector>
#include "../Engine.hpp"
#include "GeneratorBase.hpp"
#include "../Tools/MethodTable.hpp"
#include "../Tools/Resampler.hpp"
```

Classes

- class [OCAE::Generator::WAV](#)
Plays audio from WAVE data.

Functions

- [OCAE::Generator::OCAE_TYPEDEF_SHARED](#) (WAV)
Alias for a `std::shared_ptr` instantiated with the [WAV](#) class.

5.35.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.36 WAVHeader.hpp File Reference

```
#include "../Engine.hpp"
```

Classes

- struct [OCAE::Tools::WAVHeader](#)

A POD structure representing the structure of the header of a WAVE file.

5.36.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.37 WAVWriter.hpp File Reference

```
#include <RIFF-Util/RIFF.hpp>
#include "../Engine.hpp"
```

Functions

- `RIFF::vector_t OCAE::Tools::WriteWAV` (`Track_t` const &audio)
To be used in tandem with the recording system built into [Core::Driver](#).

5.37.1 Detailed Description

Author

Chyler Morrison

Email: contact@chyler.info

Project: Audio Engine

Copyright

Copyright © 2019 Chyler Morrison

5.37.2 Function Documentation

5.37.2.1 WriteWAV()

```
RIFF::vector_t OCAE::Tools::WriteWAV (  
    Track\_t const & audio )
```

To be used in tandem with the recording system built into `Core::Driver`.

Parameters

<i>audio</i>	The audio to be formatted into WAVE (RIFF) data.
--------------	--

Returns

The formatted data.

Index

- ~Driver
 - OCAE::Core::Driver, [29](#)
- ~EnvelopeFollower
 - OCAE::Modifier::EnvelopeFollower, [39](#)
- ~Equalizer
 - OCAE::Modifier::Equalizer, [44](#)
- ~Gain
 - OCAE::Modifier::Gain, [49](#)
- ~GeneratorBase
 - OCAE::Generator::GeneratorBase, [54](#)
- ~GenericFilter
 - OCAE::Modifier::GenericFilter, [64](#)
- ~LowPass
 - OCAE::Modifier::LowPass, [68](#)
- ~MethodTable
 - OCAE::Tools::MethodTable, [74](#)
- ~ModifierBase
 - OCAE::Modifier::ModifierBase, [79](#)
- ~ModifierFactory
 - OCAE::Modifier::ModifierFactory, [82](#)
- ~Noise
 - OCAE::Generator::Noise, [89](#)
- ~Sawtooth
 - OCAE::Generator::Sawtooth, [95](#)
- ~Sine
 - OCAE::Generator::Sine, [101](#)
- ~Square
 - OCAE::Generator::Square, [118](#)
- ~Triangle
 - OCAE::Generator::Triangle, [123](#)
- ~WAV
 - OCAE::Generator::WAV, [129](#)

ADSR.hpp, [135](#)

ADSR

- OCAE::Modifier::ADSR, [9](#), [10](#)

AddConnection

- OCAE::Sound::Sound, [107](#)

AddSound

- OCAE::Core::Driver, [29](#)

BandPass

- OCAE::Modifier::BandPass, [14](#), [15](#)

BandPass.hpp, [136](#)

Block

- OCAE::Sound::Block, [20](#)

Block.hpp, [136](#)

CallMethod

- OCAE::Tools::MethodTable, [74](#)

Core.hpp, [137](#)

Create

- OCAE::Core::Driver, [30](#)

CreateADSR

- OCAE::Modifier::ModifierFactory, [83](#)

CreateBandPass

- OCAE::Modifier::ModifierFactory, [83](#)

CreateBase

- OCAE::Generator::GeneratorFactory, [58](#)
- OCAE::Modifier::ModifierFactory, [83](#)

CreateBasicGenerator

- OCAE::Sound::SoundFactory, [113](#)

CreateBasicModifier

- OCAE::Sound::SoundFactory, [113](#)

CreateBlock

- OCAE::Sound::SoundFactory, [114](#), [115](#)

CreateDelay

- OCAE::Modifier::ModifierFactory, [84](#)

CreateEcho

- OCAE::Modifier::ModifierFactory, [84](#)

CreateEmptySound

- OCAE::Sound::SoundFactory, [116](#)

CreateEnvelopeFollower

- OCAE::Modifier::ModifierFactory, [85](#)

CreateEqualizer

- OCAE::Modifier::ModifierFactory, [85](#)

CreateGain

- OCAE::Modifier::ModifierFactory, [85](#)

CreateGenericFilter

- OCAE::Modifier::ModifierFactory, [86](#)

CreateLowPass

- OCAE::Modifier::ModifierFactory, [86](#)

CreateMethodList

- OCAE::Generator::GeneratorBase, [55](#)
- OCAE::Generator::Noise, [89](#)
- OCAE::Generator::Sawtooth, [96](#)
- OCAE::Generator::Sine, [101](#)
- OCAE::Generator::Square, [119](#)
- OCAE::Generator::Triangle, [124](#)
- OCAE::Generator::WAV, [130](#)
- OCAE::Modifier::ADSR, [10](#)
- OCAE::Modifier::BandPass, [15](#)

- OCAE::Modifier::Delay, 24
- OCAE::Modifier::Echo, 35
- OCAE::Modifier::EnvelopeFollower, 40
- OCAE::Modifier::Equalizer, 45
- OCAE::Modifier::Gain, 50
- OCAE::Modifier::GenericFilter, 64
- OCAE::Modifier::LowPass, 69
- OCAE::Modifier::ModifierBase, 79
- OCAE::Tools::MethodTable, 75
- CreateNoise
 - OCAE::Generator::GeneratorFactory, 58
- CreateSawtooth
 - OCAE::Generator::GeneratorFactory, 59
- CreateSine
 - OCAE::Generator::GeneratorFactory, 59
- CreateSquare
 - OCAE::Generator::GeneratorFactory, 59
- CreateTriangle
 - OCAE::Generator::GeneratorFactory, 60
- CreateWAV
 - OCAE::Generator::GeneratorFactory, 60, 61
- Delay
 - OCAE::Modifier::Delay, 23, 24
- Delay.hpp, 138
- Driver
 - OCAE::Core::Driver, 28, 29
- Driver.hpp, 138
 - OCAE_TYPEDEF_SHARED, 139
- Echo
 - OCAE::Modifier::Echo, 34
- Echo.hpp, 139
- Engine.hpp, 140
- Envelope.hpp, 141
- EnvelopeFollower
 - OCAE::Modifier::EnvelopeFollower, 39, 40
- Equalizer
 - OCAE::Modifier::Equalizer, 44
- Equalizer.hpp, 142
- Gain
 - OCAE::Modifier::Gain, 49, 50
- Gain.hpp, 142
 - OCAE_TYPEDEF_SHARED, 143
- GeneratorBase
 - OCAE::Generator::GeneratorBase, 54, 55
- GeneratorBase.hpp, 144
- GeneratorFactory.hpp, 144
- Generators.hpp, 145
- GenericFilter
 - OCAE::Modifier::GenericFilter, 63, 64
- GenericFilter.hpp, 146
- GetDecayRatio
 - OCAE::Modifier::Echo, 35
- GetDelay
 - OCAE::Modifier::Delay, 24
- GetFrequency
 - OCAE::Generator::Sawtooth, 96
 - OCAE::Generator::Sine, 101
 - OCAE::Generator::Square, 119
 - OCAE::Generator::Triangle, 124
 - OCAE::Modifier::BandPass, 15
- GetGain
 - OCAE::Modifier::Equalizer, 45
 - OCAE::Modifier::Gain, 50
- GetGenerator
 - OCAE::Sound::Block, 20
- GetID
 - OCAE::Core::Driver, 30
- GetInputBlock
 - OCAE::Sound::Sound, 107
- GetModifier
 - OCAE::Sound::Block, 20, 21
- GetOption
 - Input.hpp, 147
- GetOutputBlock
 - OCAE::Sound::Sound, 107
- GetOutputTrack
 - OCAE::Core::Driver, 30
- GetQuality
 - OCAE::Modifier::BandPass, 16
- InitOptions
 - Input.hpp, 148
- Input.hpp, 147
 - GetOption, 147
 - InitOptions, 148
- IsBase
 - OCAE::Generator::GeneratorBase, 55
 - OCAE::Generator::Noise, 89
 - OCAE::Generator::Sawtooth, 96
 - OCAE::Generator::Sine, 102
 - OCAE::Generator::Square, 119
 - OCAE::Generator::Triangle, 125
 - OCAE::Generator::WAV, 130
 - OCAE::Modifier::ADSR, 10
 - OCAE::Modifier::BandPass, 16
 - OCAE::Modifier::Delay, 25
 - OCAE::Modifier::Echo, 35
 - OCAE::Modifier::EnvelopeFollower, 40
 - OCAE::Modifier::Equalizer, 45
 - OCAE::Modifier::Gain, 51
 - OCAE::Modifier::GenericFilter, 65
 - OCAE::Modifier::LowPass, 69
 - OCAE::Modifier::ModifierBase, 79
- Left
 - Util.hpp, 164
- LoadWAV

- OCAE::Generator::WAV, 131
- LowPass
 - OCAE::Modifier::LowPass, 68
- LowPass.hpp, 148
- Macro.hpp, 149
- MethodTable
 - OCAE::Tools::MethodTable, 74
- MethodTable.hpp, 151
- ModifierBase
 - OCAE::Modifier::ModifierBase, 78, 79
- ModifierBase.hpp, 152
- ModifierFactory.hpp, 152
 - OCAE_TYPEDEF_SHARED, 154
- Modifiers.hpp, 154
- Noise
 - OCAE::Generator::Noise, 88, 89
- Noise.hpp, 155
- OCAE::Core::Driver, 27
 - ~Driver, 29
 - AddSound, 29
 - Create, 30
 - Driver, 28, 29
 - GetID, 30
 - GetOutputTrack, 30
 - operator=, 31
 - Process, 31
 - RemoveSound, 32
 - SetGain, 32
- OCAE::Generator::GeneratorBase, 53
 - ~GeneratorBase, 54
 - CreateMethodList, 55
 - GeneratorBase, 54, 55
 - IsBase, 55
 - operator=, 56
 - Process, 57
- OCAE::Generator::GeneratorFactory, 57
 - CreateBase, 58
 - CreateNoise, 58
 - CreateSawtooth, 59
 - CreateSine, 59
 - CreateSquare, 59
 - CreateTriangle, 60
 - CreateWAV, 60, 61
- OCAE::Generator::Noise, 87
 - ~Noise, 89
 - CreateMethodList, 89
 - IsBase, 89
 - Noise, 88, 89
 - operator=, 90
 - Process, 91
- OCAE::Generator::Sawtooth, 93
 - ~Sawtooth, 95
- CreateMethodList, 96
- GetFrequency, 96
- IsBase, 96
- operator=, 97
- Process, 98
- Sawtooth, 95
- SetFrequency, 98
- OCAE::Generator::Sine, 98
 - ~Sine, 101
 - CreateMethodList, 101
 - GetFrequency, 101
 - IsBase, 102
 - operator=, 102
 - Process, 103
 - SetFrequency, 103
 - SetupWaveTable, 103
 - Sine, 100, 101
- OCAE::Generator::Square, 116
 - ~Square, 118
 - CreateMethodList, 119
 - GetFrequency, 119
 - IsBase, 119
 - operator=, 120
 - Process, 121
 - SetFrequency, 121
 - Square, 118
- OCAE::Generator::Triangle, 121
 - ~Triangle, 123
 - CreateMethodList, 124
 - GetFrequency, 124
 - IsBase, 125
 - operator=, 125
 - Process, 126
 - SetFrequency, 126
 - Triangle, 123, 124
- OCAE::Generator::WAV, 127
 - ~WAV, 129
 - CreateMethodList, 130
 - IsBase, 130
 - LoadWAV, 131
 - operator=, 131, 132
 - ParseWAV, 132
 - Process, 132
 - ReadFile, 133
 - WAV, 128–130
- OCAE::Modifier::ADSR, 7
 - ADSR, 9, 10
 - CreateMethodList, 10
 - IsBase, 10
 - operator=, 11
 - Process, 12
 - Release, 12
 - State, 9
- OCAE::Modifier::BandPass, 12

- BandPass, [14](#), [15](#)
- CreateMethodList, [15](#)
- GetFrequency, [15](#)
- GetQuality, [16](#)
- IsBase, [16](#)
- operator=, [16](#), [17](#)
- Process, [17](#)
- Reset, [17](#)
- SetFrequency, [18](#)
- SetQuality, [18](#)
- OCAE::Modifier::Delay, [22](#)
 - CreateMethodList, [24](#)
 - Delay, [23](#), [24](#)
 - GetDelay, [24](#)
 - IsBase, [25](#)
 - operator=, [25](#)
 - Process, [26](#)
 - SetDelay, [26](#)
- OCAE::Modifier::Echo, [32](#)
 - CreateMethodList, [35](#)
 - Echo, [34](#)
 - GetDecayRatio, [35](#)
 - IsBase, [35](#)
 - operator=, [36](#)
 - Process, [37](#)
 - SetDecayRatio, [37](#)
- OCAE::Modifier::EnvelopeFollower, [37](#)
 - ~EnvelopeFollower, [39](#)
 - CreateMethodList, [40](#)
 - EnvelopeFollower, [39](#), [40](#)
 - IsBase, [40](#)
 - operator=, [41](#)
 - Process, [42](#)
- OCAE::Modifier::Equalizer, [42](#)
 - ~Equalizer, [44](#)
 - CreateMethodList, [45](#)
 - Equalizer, [44](#)
 - GetGain, [45](#)
 - IsBase, [45](#)
 - operator=, [46](#)
 - Process, [47](#)
 - SetGain, [47](#)
- OCAE::Modifier::Gain, [47](#)
 - ~Gain, [49](#)
 - CreateMethodList, [50](#)
 - Gain, [49](#), [50](#)
 - GetGain, [50](#)
 - IsBase, [51](#)
 - operator=, [51](#)
 - Process, [52](#)
 - SetGain, [52](#)
- OCAE::Modifier::GenericFilter, [61](#)
 - ~GenericFilter, [64](#)
 - CreateMethodList, [64](#)
 - GenericFilter, [63](#), [64](#)
 - IsBase, [65](#)
 - operator=, [65](#)
 - Process, [66](#)
- OCAE::Modifier::LowPass, [66](#)
 - ~LowPass, [68](#)
 - CreateMethodList, [69](#)
 - IsBase, [69](#)
 - LowPass, [68](#)
 - operator=, [69](#), [70](#)
 - Process, [70](#)
 - Reset, [70](#)
 - SetCutoff, [71](#)
 - SetResonance, [71](#)
- OCAE::Modifier::ModifierBase, [76](#)
 - ~ModifierBase, [79](#)
 - CreateMethodList, [79](#)
 - IsBase, [79](#)
 - ModifierBase, [78](#), [79](#)
 - operator=, [80](#)
 - Process, [81](#)
- OCAE::Modifier::ModifierFactory, [81](#)
 - ~ModifierFactory, [82](#)
 - CreateADSR, [83](#)
 - CreateBandPass, [83](#)
 - CreateBase, [83](#)
 - CreateDelay, [84](#)
 - CreateEcho, [84](#)
 - CreateEnvelopeFollower, [85](#)
 - CreateEqualizer, [85](#)
 - CreateGain, [85](#)
 - CreateGenericFilter, [86](#)
 - CreateLowPass, [86](#)
- OCAE::Sound::Block, [18](#)
 - Block, [20](#)
 - GetGenerator, [20](#)
 - GetModifier, [20](#), [21](#)
 - PrimeInput, [21](#)
 - Process, [21](#)
- OCAE::Sound::Sound, [104](#)
 - AddConnection, [107](#)
 - GetInputBlock, [107](#)
 - GetOutputBlock, [107](#)
 - operator=, [107](#), [108](#)
 - Pause, [108](#)
 - PrepareGraph, [108](#)
 - Process, [110](#)
 - ProcessOrder, [110](#)
 - Register, [110](#)
 - RemoveConnection, [111](#)
 - SetInputGain, [111](#)
 - SetOutputGain, [111](#)
 - Sound, [106](#)
 - Unpause, [111](#)

- Unregister, 112
- OCAE::Sound::SoundFactory, 112
 - CreateBasicGenerator, 113
 - CreateBasicModifier, 113
 - CreateBlock, 114, 115
 - CreateEmptySound, 116
- OCAE::Tools::MethodTable, 71
 - ~MethodTable, 74
 - CallMethod, 74
 - CreateMethodList, 75
 - MethodTable, 74
 - RegisterMethod, 75
 - RegisterMethods, 76
- OCAE::Tools::Resampler, 91
 - Process, 93
 - Resampler, 92
 - SetPlaybackSpeed, 93
- OCAE::Tools::WAVHeader, 133
 - WAVHeader, 134
- OCAE_TYPEDEF_SHARED
 - Driver.hpp, 139
 - Gain.hpp, 143
 - ModifierFactory.hpp, 154
 - Sound.hpp, 158, 159
- operator=
 - OCAE::Core::Driver, 31
 - OCAE::Generator::GeneratorBase, 56
 - OCAE::Generator::Noise, 90
 - OCAE::Generator::Sawtooth, 97
 - OCAE::Generator::Sine, 102
 - OCAE::Generator::Square, 120
 - OCAE::Generator::Triangle, 125
 - OCAE::Generator::WAV, 131, 132
 - OCAE::Modifier::ADSR, 11
 - OCAE::Modifier::BandPass, 16, 17
 - OCAE::Modifier::Delay, 25
 - OCAE::Modifier::Echo, 36
 - OCAE::Modifier::EnvelopeFollower, 41
 - OCAE::Modifier::Equalizer, 46
 - OCAE::Modifier::Gain, 51
 - OCAE::Modifier::GenericFilter, 65
 - OCAE::Modifier::LowPass, 69, 70
 - OCAE::Modifier::ModifierBase, 80
 - OCAE::Sound::Sound, 107, 108
- ParseWAV
 - OCAE::Generator::WAV, 132
- Pause
 - OCAE::Sound::Sound, 108
- PrepareGraph
 - OCAE::Sound::Sound, 108
- PrimeInput
 - OCAE::Sound::Block, 21
- Process
 - OCAE::Core::Driver, 31
 - OCAE::Generator::GeneratorBase, 57
 - OCAE::Generator::Noise, 91
 - OCAE::Generator::Sawtooth, 98
 - OCAE::Generator::Sine, 103
 - OCAE::Generator::Square, 121
 - OCAE::Generator::Triangle, 126
 - OCAE::Generator::WAV, 132
 - OCAE::Modifier::ADSR, 12
 - OCAE::Modifier::BandPass, 17
 - OCAE::Modifier::Delay, 26
 - OCAE::Modifier::Echo, 37
 - OCAE::Modifier::EnvelopeFollower, 42
 - OCAE::Modifier::Equalizer, 47
 - OCAE::Modifier::Gain, 52
 - OCAE::Modifier::GenericFilter, 66
 - OCAE::Modifier::LowPass, 70
 - OCAE::Modifier::ModifierBase, 81
 - OCAE::Sound::Block, 21
 - OCAE::Sound::Sound, 110
 - OCAE::Tools::Resampler, 93
- ProcessOrder
 - OCAE::Sound::Sound, 110
- ReadFile
 - OCAE::Generator::WAV, 133
- Register
 - OCAE::Sound::Sound, 110
- RegisterMethod
 - OCAE::Tools::MethodTable, 75
- RegisterMethods
 - OCAE::Tools::MethodTable, 76
- Release
 - OCAE::Modifier::ADSR, 12
- RemoveConnection
 - OCAE::Sound::Sound, 111
- RemoveSound
 - OCAE::Core::Driver, 32
- Resampler
 - OCAE::Tools::Resampler, 92
- Resampler.hpp, 155
- Reset
 - OCAE::Modifier::BandPass, 17
 - OCAE::Modifier::LowPass, 70
- Right
 - Util.hpp, 165
- Sawtooth
 - OCAE::Generator::Sawtooth, 95
- Sawtooth.hpp, 156
- SetCutoff
 - OCAE::Modifier::LowPass, 71
- SetDecayRatio
 - OCAE::Modifier::Echo, 37
- SetDelay

- OCAE::Modifier::Delay, 26
- SetFrequency
 - OCAE::Generator::Sawtooth, 98
 - OCAE::Generator::Sine, 103
 - OCAE::Generator::Square, 121
 - OCAE::Generator::Triangle, 126
 - OCAE::Modifier::BandPass, 18
- SetGain
 - OCAE::Core::Driver, 32
 - OCAE::Modifier::Equalizer, 47
 - OCAE::Modifier::Gain, 52
- SetInputGain
 - OCAE::Sound::Sound, 111
- SetOutputGain
 - OCAE::Sound::Sound, 111
- SetPlaybackSpeed
 - OCAE::Tools::Resampler, 93
- SetQuality
 - OCAE::Modifier::BandPass, 18
- SetResonance
 - OCAE::Modifier::LowPass, 71
- SetupWaveTable
 - OCAE::Generator::Sine, 103
- Sine
 - OCAE::Generator::Sine, 100, 101
- Sine.hpp, 157
- Sound
 - OCAE::Sound::Sound, 106
- Sound.hpp, 158
 - OCAE_TYPEDEF_SHARED, 158, 159
- SoundFactory.hpp, 159
- Sounds.hpp, 160
- Square
 - OCAE::Generator::Square, 118
- Square.hpp, 160
- State
 - OCAE::Modifier::ADSR, 9
- Tools.hpp, 161
- Triangle
 - OCAE::Generator::Triangle, 123, 124
- Triangle.hpp, 162
- Types.hpp, 162
- Unpause
 - OCAE::Sound::Sound, 111
- Unregister
 - OCAE::Sound::Sound, 112
- Util.hpp, 163
 - Left, 164
 - Right, 165
- WAV.hpp, 166
- WAVHeader
 - OCAE::Tools::WAVHeader, 134
- WAVHeader.hpp, 167
- WAVWriter.hpp, 168
 - WriteWAV, 168
- WAV
 - OCAE::Generator::WAV, 128–130
- WriteWAV
 - WAVWriter.hpp, 168