# My Project

Generated by Doxygen 1.8.13

# Contents

# Chapter 1

# Class Index

## 1.1 Class List

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

# Chapter 2

# Class Documentation

## 2.1 WAV Class Reference

### Public Member Functions

- bool compression (double cutOff, double scale)
- bool echo ()
- bool loPass ()
- bool normalize ()
- bool gain (double gain)
- bool loadData (std::string filePath)
- bool writeData (std::string filePath)
- wav_meta **getMetaData** () const
- uint8_t $*$ **getDataBytes** () const

### 2.1.1 Member Function Documentation

#### 2.1.1.1 compression()

```
bool WAV::compression (
             double cutOff,
             double scale )
```

The dynamic range of the signal is compressed using a non-linear map of input to output.

**Returns**

bool if successful

**Parameters**

| | |
|---|---|
| *cutOff* | percentage of max value in which all above that cutoff is applied a gain. e.g. if the max value of the data set is 36 and the cutOff is .66 all data points above 24 and below -24 will have a gain applied. |
| *scale* | the decimal to apply to the data sets beyond the cutOff. |

#### 2.1.1.2 echo()

```
bool WAV::echo ( )
```

Applies an echo effect to the data

**Returns**

bool if successful

#### 2.1.1.3 gain()

```
bool WAV::gain (
             double gain )
```

Applies a gain effect to the data

**Returns**

bool if successful

**Parameters**

| | |
|---|---|
| *gain* | decimal value to scale the data |

#### 2.1.1.4 loadData()

```
bool WAV::loadData (
             std::string filePath )
```

Loads the meta data into the struct and the data into the 8 bit int array

**Parameters**

| | |
|---|---|
| *filePath* | direct or relative path to the .wav file |

#### 2.1.1.5 loPass()

```
bool WAV::loPass ( )
```

Lo pass filter that removes all any high intesity values from the sample set.

**Returns**

bool if successful

#### 2.1.1.6 normalize()

```
bool WAV::normalize ( )
```

maximizes the amplitude of the final waveform

**Returns**

bool if successful

#### 2.1.1.7 writeData()

```
bool WAV::writeData (
            std::string filePath )
```

Writes the data from the struct and 8 bit int array to the provided file

**Parameters**

| | |
|---|---|
| *filePath* | direct or relative path to the .wav file to be made or changed |

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

- /home/jasmine/Desktop/CS-202-Semester-Project-F21/src/wav.h
- /home/jasmine/Desktop/CS-202-Semester-Project-F21/src/wav.cpp

## 2.2 WAV_HEADER Struct Reference

### Public Attributes

- uint8_t **RIFF** [4]
- uint32_t **chunkSize**
- uint8_t **WAVE** [4]
- uint8_t **fmt** [4]
- uint32_t **subchunk1Size**
- uint16_t **audioFormat**
- uint16_t **numOfChan**

- uint32_t **samplesPerSec**
- uint32_t **bytesPerSec**
- uint16_t **blockAlign**
- uint16_t **bitsPerSample**
- uint8_t **subchunk2ID** [4]
- uint32_t **subchunk2Size**

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

- /home/jasmine/Desktop/CS-202-Semester-Project-F21/src/wav.h

# Index