AudioConverter Kit
for iOS

coded by : Refaz Nabak
email : muzafsuka@gmail.com

last updated : May 6, 2015

INTRODUCTION

Audio conversion is one of the most needed software utilities because there are lots of audio formats present in the market. Most vendors have their proprietary formats and there are also well known commercial and open source formats too. This project mainly focuses on audio codecs which are widely supported in Apple products such as **AAC** (Advanced Audio Codec), **ALAC** (Apple Lossless Audio Codec), **AIFF** (Audio Interchange File Format), **MP3** and **WAV**. You can convert any of these formats to one another by the help of the library. It's just easy as calling a function. Except the MP3 format there is built in hardware encoder in iPhone so that the encoding time is fast.

FEATURES

- Support most popular audio types (**AAC, AIFF, ALAC, WAV** and **MP3**)
- Support conversion for a specified **interval** of media
- **Easy** to use (just as easy as initializing an object and setting up the params)
- Fast application **integration** saves you time and money
- Quality proven **LAME** Encoding Engine under the hood (MP3)
- **Sample** GUI application gives you easy to integrate code snippets
- Works on **all** iOS (iPhone 6, iPhone 6+,iPhone 5/5S,iPhone 4/4S and iPads) devices

REQUIREMENTS

iOS **7.0** or above is recommended to compile and use **AudioConverter Kit**.

AudioConverter Kit is consist of two engines, which are AudioConverter and MP3Converter.

MP3Converter is specialized for converting audio files to MP3 format (MP3Converter.h) and AudioConverter is specialized for converting to other audio formats (AudioConverter.h).

AudioConverter engine additionally requires the following Apple frameworks:

- CoreMedia
- AVFoundation

MP3Converter engine uses **lame** library for MP3 encoding.

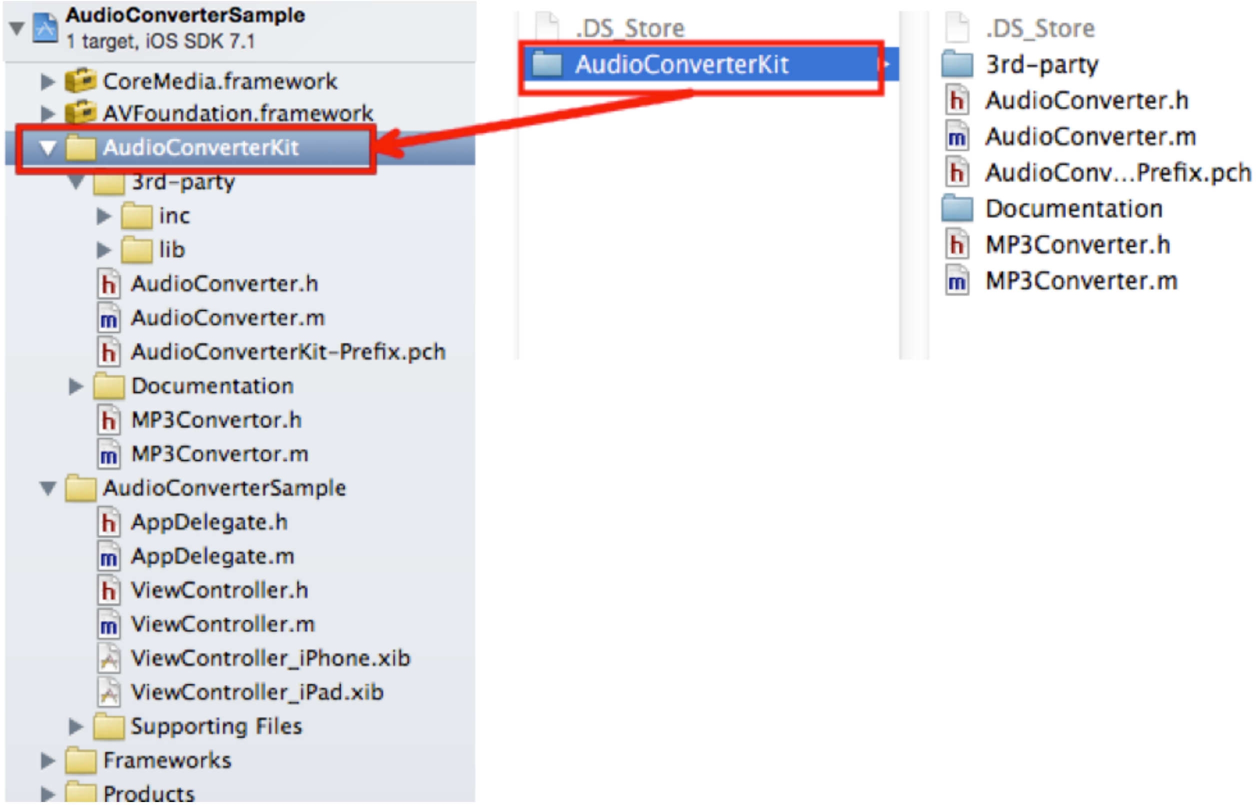
Therefore, required libraries mentioned above should be added to support related engine. You can add all of them (CoreMedia.framework + AVFoundation.framework + lame library) to support both engine.

INTEGRATION

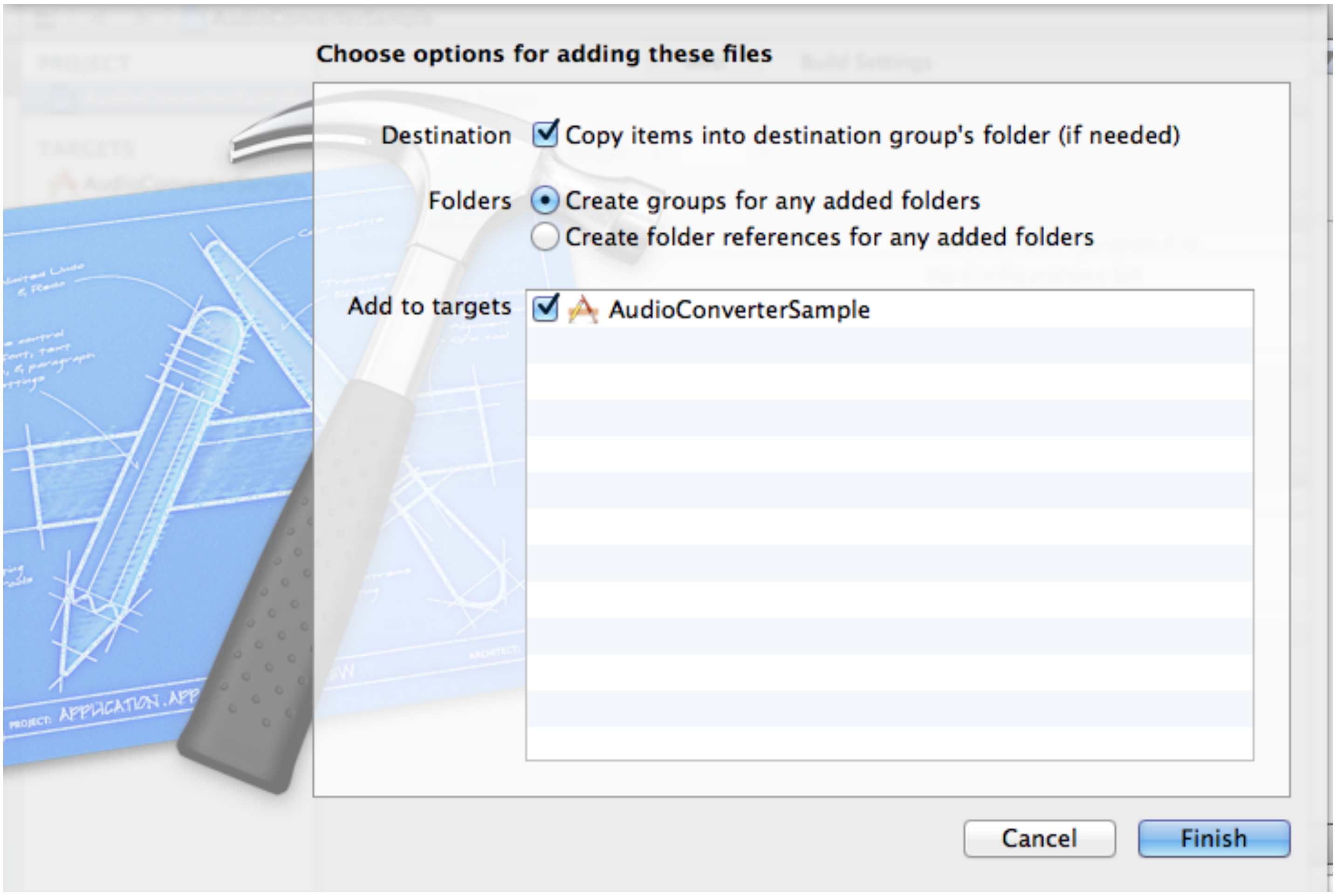
AudioConverter Kit is easy to integrate. Just copy the contents of the AudioConverterKit folder to your Xcode project. Full app license includes full source code, binary app license includes library binaries and headers only.

Integration of Full App License

Copy the contents of the AudioConverterKit folder to your Xcode project. Folder includes necessary source files and headers.



Be sure that you checked “Copy items into destination group's folder (if needed)” and “Add to Target” checkboxes.



Choose options for adding these files

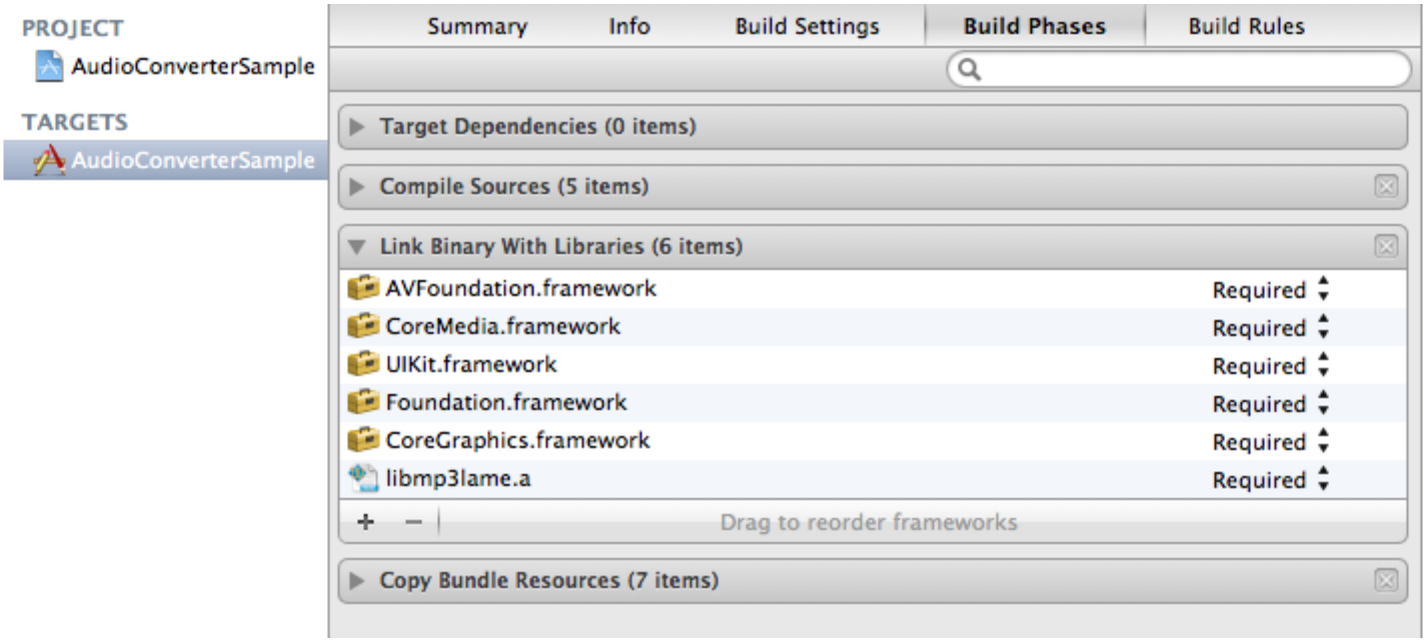
Destination ☑ Copy items into destination group's folder (if needed)

Folders ● Create groups for any added folders
○ Create folder references for any added folders

Add to targets ☑ AudioConverterSample

If you don’t need both engine (AudioConverter & MP3Converter), you can omit any of them from your project by simply deleting source files.

After adding necessary files, go to Build Phases tab in your project settings. Be sure you have AVFoundation.framework and CoreMedia.framework (for converting media files to **WAV, AIF, ALAC or AAC** format) , and/or libMp3Lame.a (for converting media files to **MP3** format) file included in the “Link Binary with Libraries” part.



Target Dependencies (0 items)

Compile Sources (5 items)

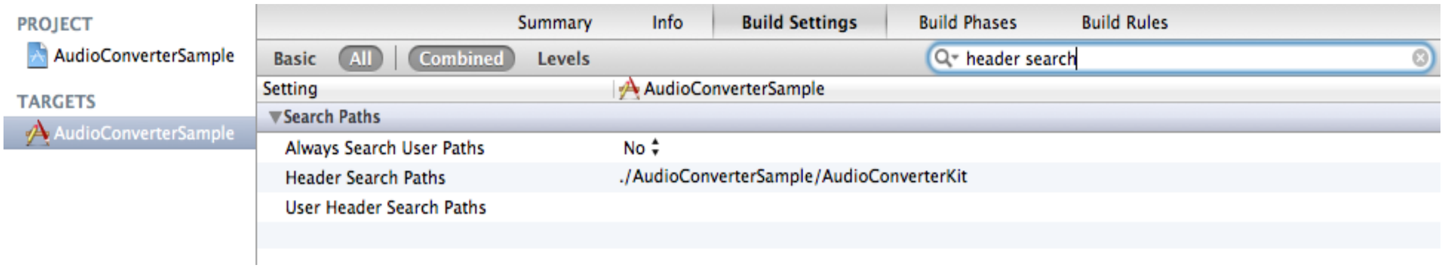
Link Binary With Libraries (6 items)

AVFoundation.framework	Required
CoreMedia.framework	Required
UIKit.framework	Required
Foundation.framework	Required
CoreGraphics.framework	Required
libmp3lame.a	Required

Copy Bundle Resources (7 items)

If you don’t need both engine (AudioConverter & MP3Converter), you can omit the unnecessary library(ies).

Then go to “Build Settings” and make sure that your header search path is pointing folder that headers files are in.

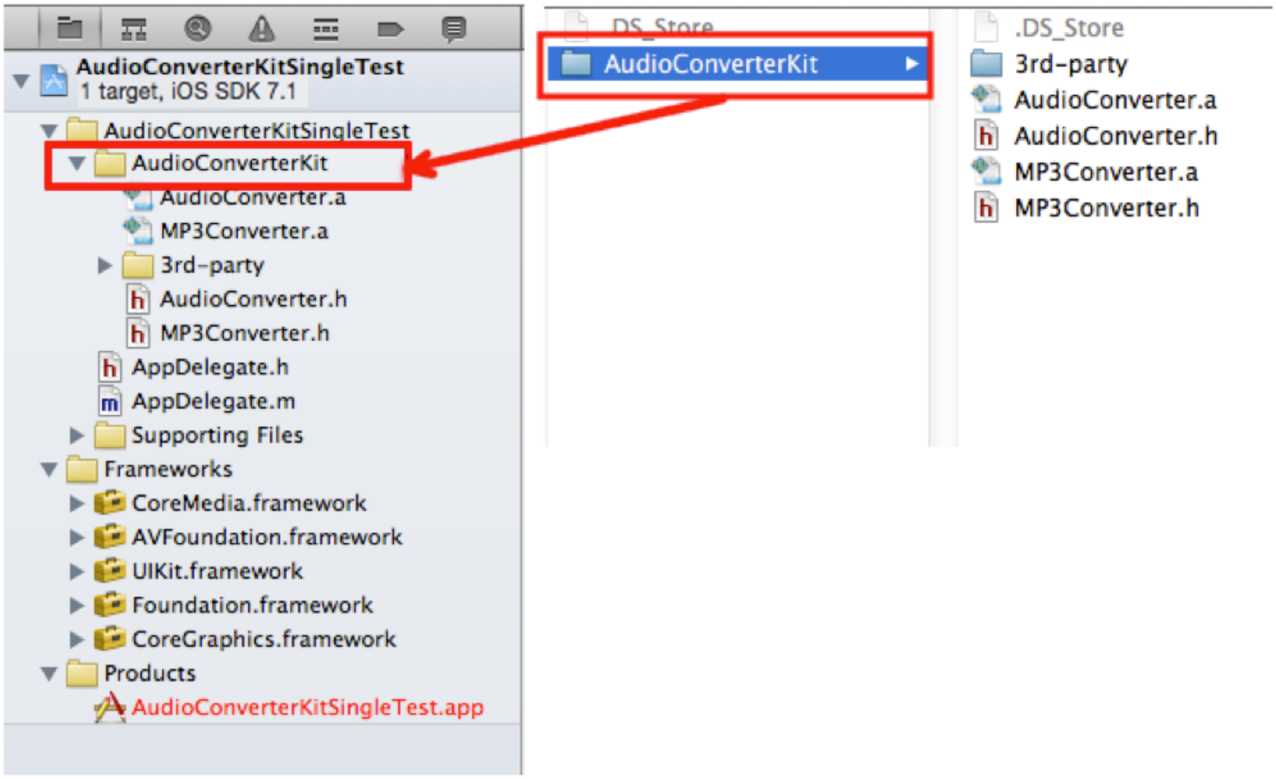


header search

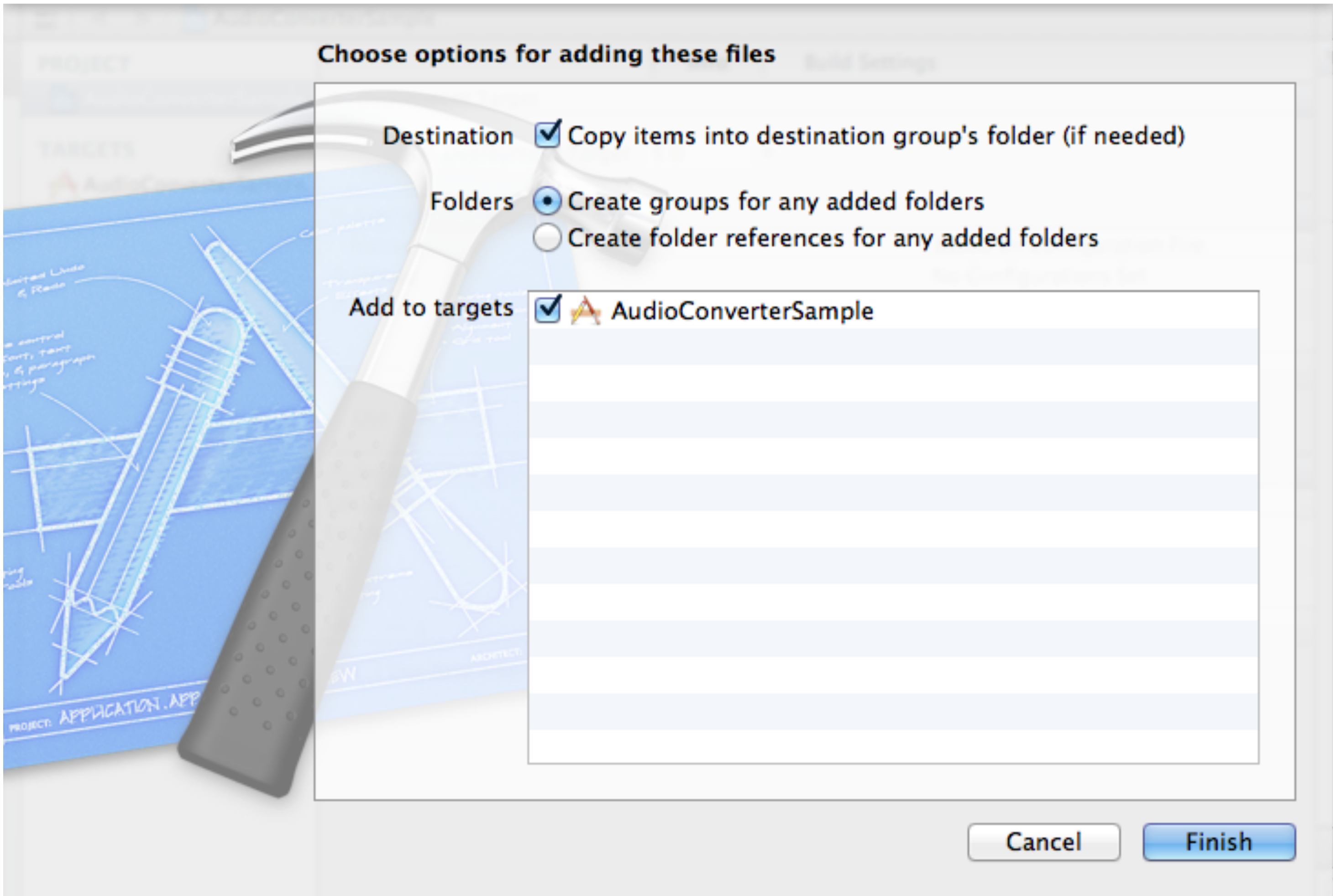
Setting	AudioConverterSample
Search Paths	
Always Search User Paths	No
Header Search Paths	./AudioConverterSample/AudioConverterKit
User Header Search Paths	

Integration of Binary App License

Copy the contents of the AudioConvertorKit folder to your Xcode project. Folder includes necessary library files (AudioConverter.a and MP3Converter.a) and headers.



Be sure that you checked “Copy items into destination group's folder (if needed)” and “Add to Target” checkboxes.



Choose options for adding these files

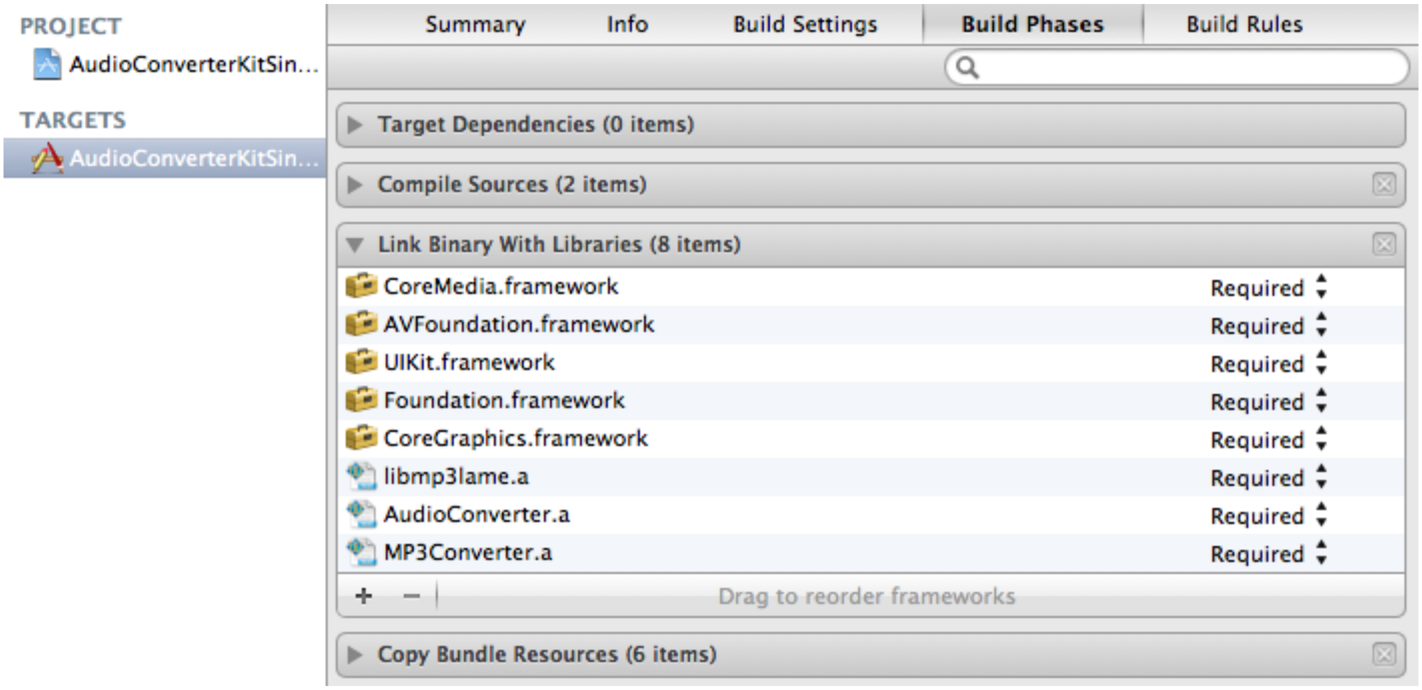
Destination ☑ Copy items into destination group's folder (if needed)

Folders ● Create groups for any added folders
○ Create folder references for any added folders

Add to targets ☑ AudioConverterSample

If you don’t need both engine (AudioConverter & MP3Converter), you can omit any of them from your project by simply deleting library and related header files.

After adding necessary files, go to Build Phases tab in your project settings. Be sure you have AVFoundation.framework and CoreMedia.framework (for converting media files to **WAV, AIF, ALAC or AAC** format) , and/or libMp3Lame.a (for converting media files to **MP3** format), and also (AudioConverter.a & MP3Converter.a) libraries included in the “Link Binary with Libraries” part.



Target Dependencies (0 items)

Compile Sources (2 items)

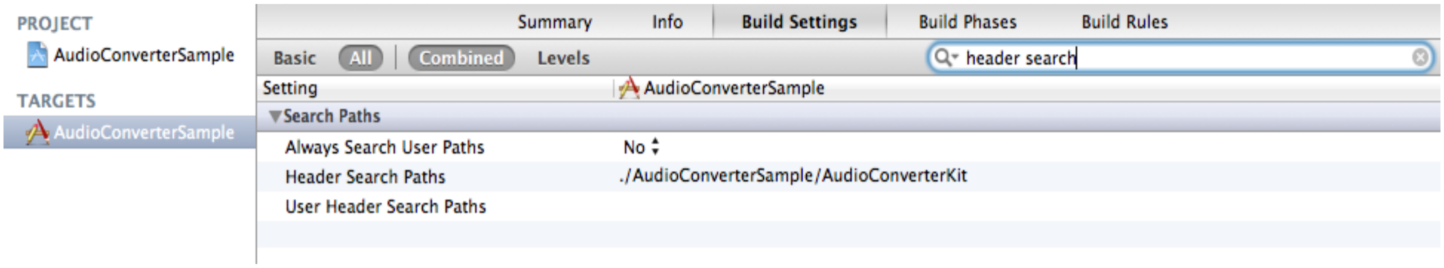
Link Binary With Libraries (8 items)

CoreMedia.framework	Required
AVFoundation.framework	Required
UIKit.framework	Required
Foundation.framework	Required
CoreGraphics.framework	Required
libmp3lame.a	Required
AudioConverter.a	Required
MP3Converter.a	Required

Copy Bundle Resources (6 items)

If you don't need both engine (AudioConverter & MP3Converter), you can omit the unnecessary library(ies).

Then go to “Build Settings” and make sure that your header search path is pointing folder that headers files are in.



header search

Setting	AudioConverterSample
▼Search Paths	
Always Search User Paths	No
Header Search Paths	./AudioConverterSample/AudioConverterKit
User Header Search Paths	

HOW TO USE

As mentioned above, AudioConverter Kit consists of two engines, which are

AudioConverter and MP3Converter.

We will briefly explain how to use these two engines:

Using **AudioConverter** :

In this mode it's so easy to convert source file to audio format other than MP3. You just select preferred output format and initialize AudioConverter. Output format definitions are as follows :

AUDIO_OUTPUT_TYPE_AAC
AUDIO_OUTPUT_TYPE_AIF
AUDIO_OUTPUT_TYPE_ALAC
AUDIO_OUTPUT_TYPE_WAV

You can set the delegate to retrieve information about the status of the conversion process.
You can also set conversionStartPoint and conversionLength parameters to select which portion of the input file you want to encode. This gives you great flexibility on encoding of relatively large audio files.

```
NSString *pathIn = [[NSBundle mainBundle] pathForResource:@"tokyo_mp3"
ofType:@"mp3"];
AudioConverter *audioConverter = [[AudioConverter alloc] init];
audioConverter.delegate = self;
audioConverter.conversionStartPoint = 0.0;
audioConverter.conversionLength = 40.0;
NSString *outputName = @"outputFilename.m4a";
[audioConverter convertAudioWithFilePath:pathIn outputName:outputName
ofType:AUDIO_OUTPUT_TYPE_AAC];
```

Mp3Converter can be used in 2 ways :

1) Simple usage (with built-in presets) :

In this mode it's so easy to convert source file to MP3 file. You just select preferred preset and initialize MP3Converter. Below the sample code and preset value matching table are given.

```
MP3Converter *mp3Converter = [[MP3Converter alloc] initWithPreset:PRESET_CD];
mp3Converter.delegate = self;
[mp3Converter initializeLame];
[mp3Converter convertMP3WithFilePath:pathIn outputName:pathOut];
```

2) Advanced usage (with manual parameter setting) :

Preset Parameter Values:

Preset / Parameter	**Bitrate Type**	**Bitrate**	**Encoding Engine Quality**	**Channel**	**VBR Quality**
PRESET_PHONE	CBR	16	Fastest	Mono	-
PRESET_VOICE	CBR	64	Fastest	Mono	-
PRESET_FM	CBR	112	Standard	Stereo	-
PRESET_TAPE	CBR	128	Standard	Stereo	-
PRESET_CD	VBR	-	High	Stereo	Fine
PRESET_STUDIO	CBR	256	Highest	Stereo	-
PRESET_HIFI	VBR	-	Highest	Stereo	Best

In this mode you initialize MP3Converter object with a BitrateType. Available bitrate types are CBR, ABR and VBR. Then you can change channel information, encoding engine quality, bitrate (for CBR and ABR) and VBRQuality (for VBR only) accordingly.

Available values for channel are: Stereo, JointStereo and Mono. Default value is Stereo, if you do not set individually.
Available values for encoding engine quality are Highest, High, Standard, Fast and Fastest. Default value is Standard if you do not set individually.

Available values for VBR Quality are BEST_QUALITY, FINE_QUALITY, MODERATE_QUALITY, LOW_QUALITY and LOWEST_QUALITY. Default value is MODERATE_QUALITY if you do not set individually.
Bitrate is an integer representing amount of data processed per second. Default value is 128. Maximum value is 320 according to MP3 standard.

Example for VBR:

```
mp3Converter = [[ MP3Converter alloc] initWithBitrateType:VBR];
mp3Converter.channel = Mono;
mp3Converter.encodingEngineQuality = Fast;
mp3Converter.vbrQuality = LOW_QUALITY;
mp3Converter.delegate = self;
[ mp3Converter initializeLame];
[ mp3Converter convertMP3WithFilePath:pathIn outputName:pathOut];
```

Example for CBR and ABR:

```
mp3Converter = [[MP3Converter alloc] initWithBitrateType:CBR];  // or ABR
mp3Converter.channel = JointStereo;
mp3Converter.encodingEngineQuality = High;
mp3Converter.bitrate = 192;
mp3Converter.delegate = self;
[mp3Converter initializeLame];
[mp3Converter convertMP3WithFilePath:pathIn outputName:pathOut];
```

LICENSING

AudioConverter Kit project includes third party library(ies) written below,

- LAME Project (http://lame.sourceforge.net/about.php)

LAME is licensed under LGPL license. You can find more info about licensing of LAME project here - http://lame.sourceforge.net/license.txt

- When encoding to AIF, AAC and WAV formats, AudiConverterKit uses apple's own encoders, please see about this topic for more info here - http://developer.apple.com/library/ios/#documentation/AudioVideo/Conceptual/MultimediaPG/UsingAudio/UsingAudio.html

MP3 licensing

For info about MP3 audio format licensing, please see below sites,

- http://en.wikipedia.org/wiki/MP3
- http://mp3licensing.com/

CHANGELOG

Version: 1.0 – Release Date: 04/20/2013

- First initial release

Version: 1.1 – Release Date: 12/14/2014

- 64 bit and iOS 8 Support
- Bug fixes

Version: 1.2 – Release Date: 05/06/2015

- Bug fixes