

iOS 程式設計

音訊應用

Brian Wang 王昱景

brian.wang.frontline@gmail.com

AV Foundation Programming Guide

Introduction

- AV Foundation is **one of several frameworks** that you can use to **play and create time-based audiovisual media**.
- It provides an Objective-C interface you use to work on a detailed level with time-based audiovisual data.

- For example, you can use it to examine, create, edit, or reencode media files.
- You can also get input streams from devices and manipulate video during realtime capture and playback.

Media Player

UIKit

Audio-only
classes

AV Foundation

Core Audio

Core Media

Core Animation

- You should typically use the highest-level abstraction available that allows you to perform the tasks you want. For example, in iOS:
 - If you simply want to play movies, you can use the Media Player Framework (**MPMoviePlayerController** or **MPMoviePlayerViewController**), or for web-based media you could use a **UIWebView** object.
 - To record video when you need only minimal control over format, use the UIKit framework (**UINavigationController**).

- Note, however, that some of the primitive data structures that you use in AV Foundation - including time-related data structures and opaque objects to carry and describe media data - are declared in the **Core Media framework**.

- AV Foundation is available in iOS 4 and later, and OS X 10.7 and later.
- This document describes AV Foundation as introduced in iOS 4.0.

- To learn about changes and additions to the framework in subsequent versions, you should also read the appropriate release notes:
- **AV Foundation Release Notes** describe changes made for iOS 5.
- **AV Foundation Release Notes (iOS 4.3)** describe changes made for iOS 4.3 and included in OS X 10.7.

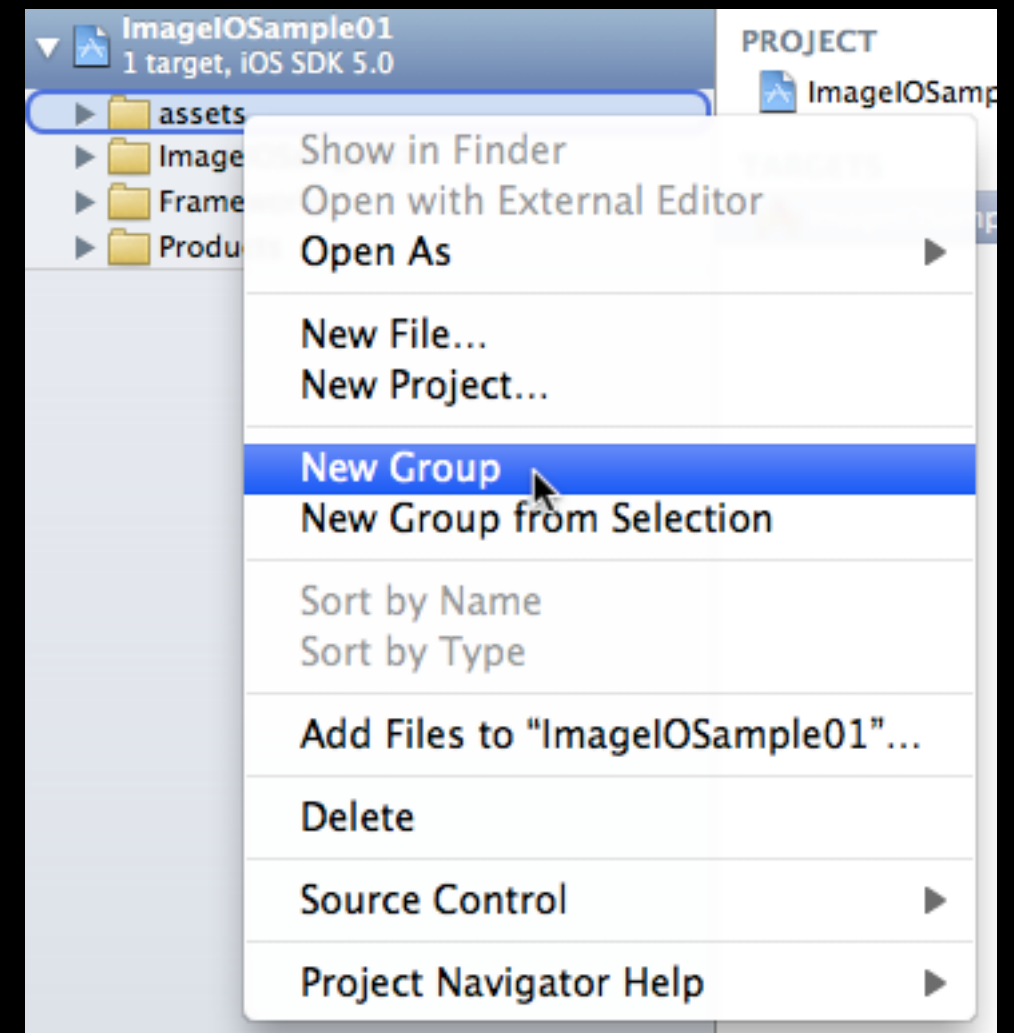
- AV Foundation is an advanced Cocoa framework. To use it effectively, you must have:
 - A solid understanding of fundamental Cocoa development tools and techniques
 - A basic grasp of **blocks**
 - A basic understanding of **key-value coding** and key-value observing
 - For playback, a basic understanding of Core Animation (see **Core Animation Programming Guide**)

MediaSample(1/13)

1. 建置一個 **Single View Application** 的專案，名稱 『**MediaSample**』。
2. 從 Targets > Build Phases > Link Binary With Libraries 加入 **AVFoundation Framework**。

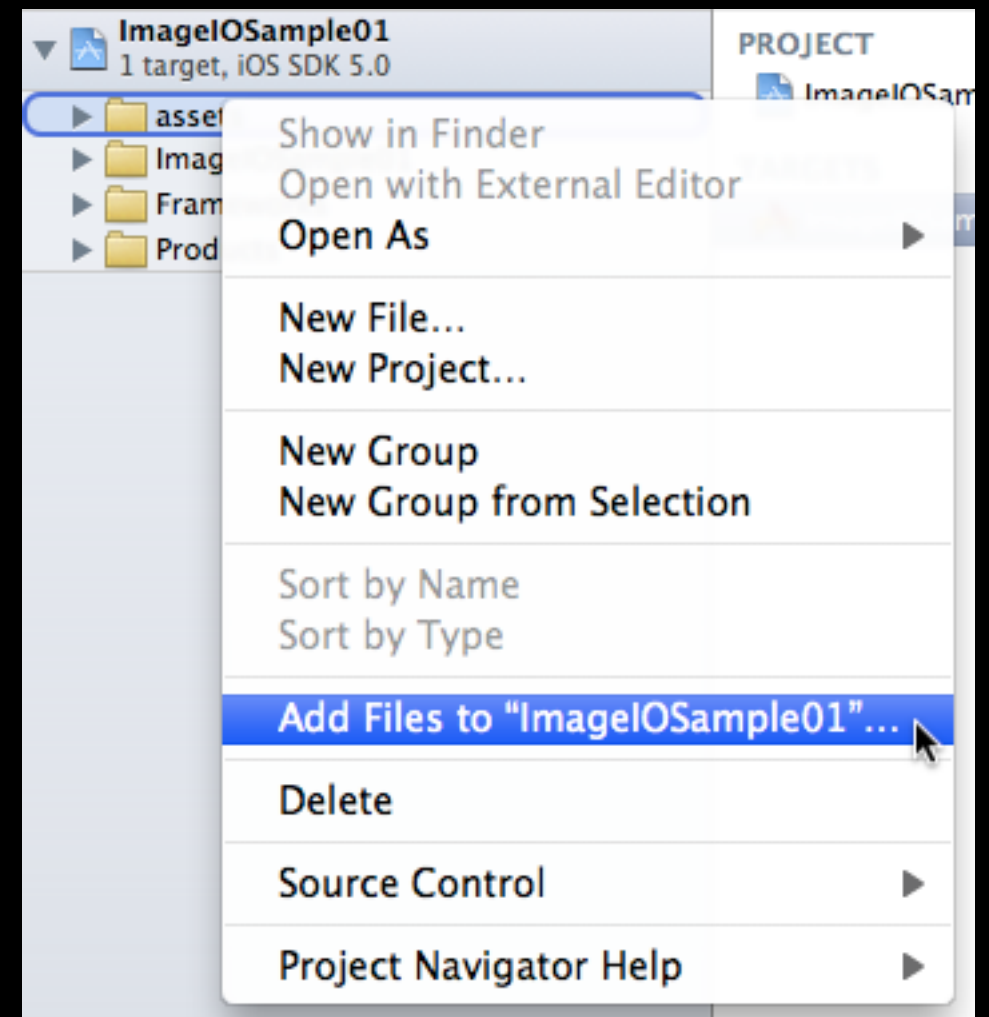
MediaSample(2/13)

3. 在專案檔案列表點選滑鼠右鍵，選擇 **New Group** 建立 Group 名為 **assets**。



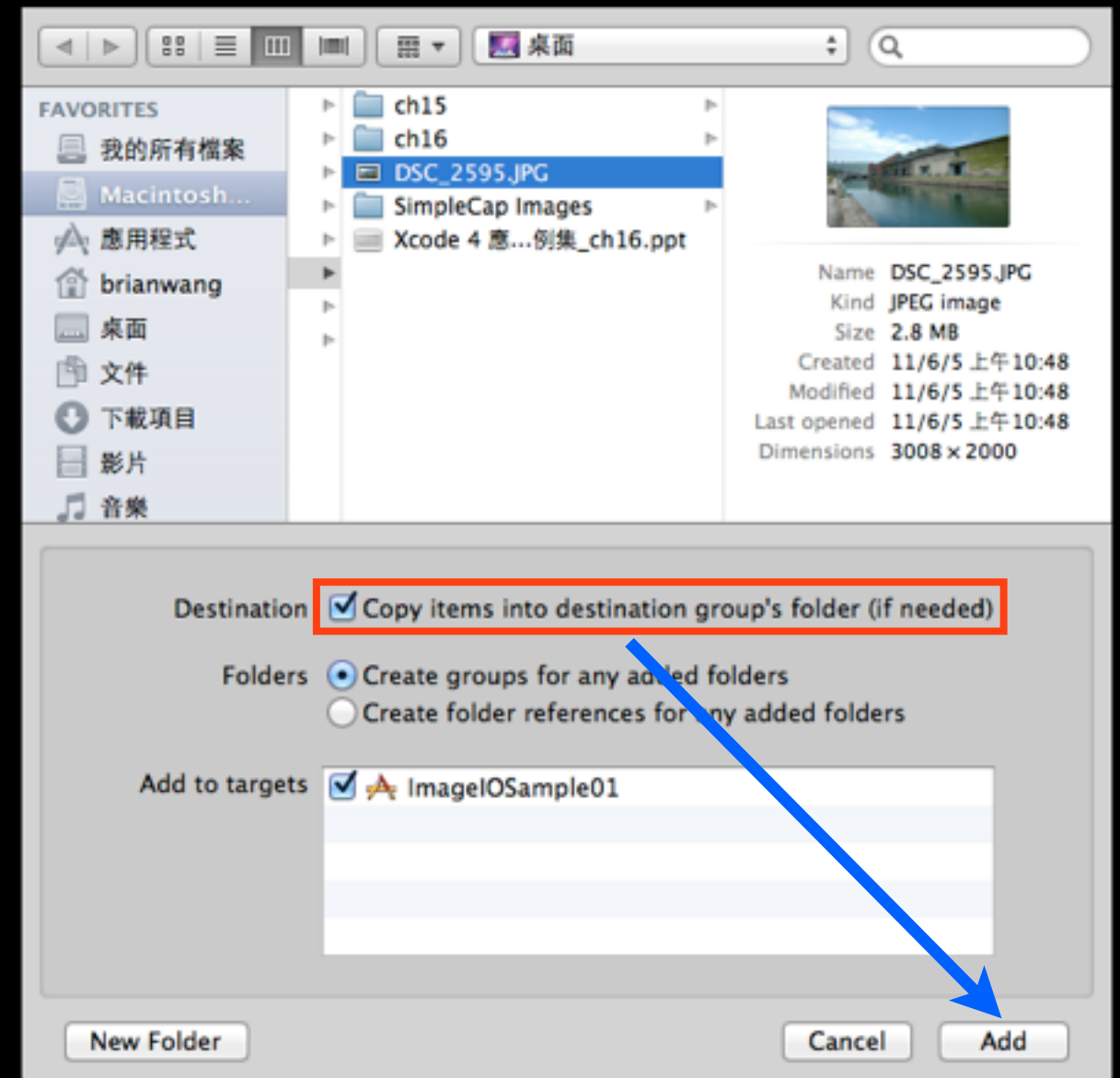
MediaSample(3/13)

4.點選滑鼠右鍵，選擇 **Add File to “MediaSample”**，加入音訊檔案。



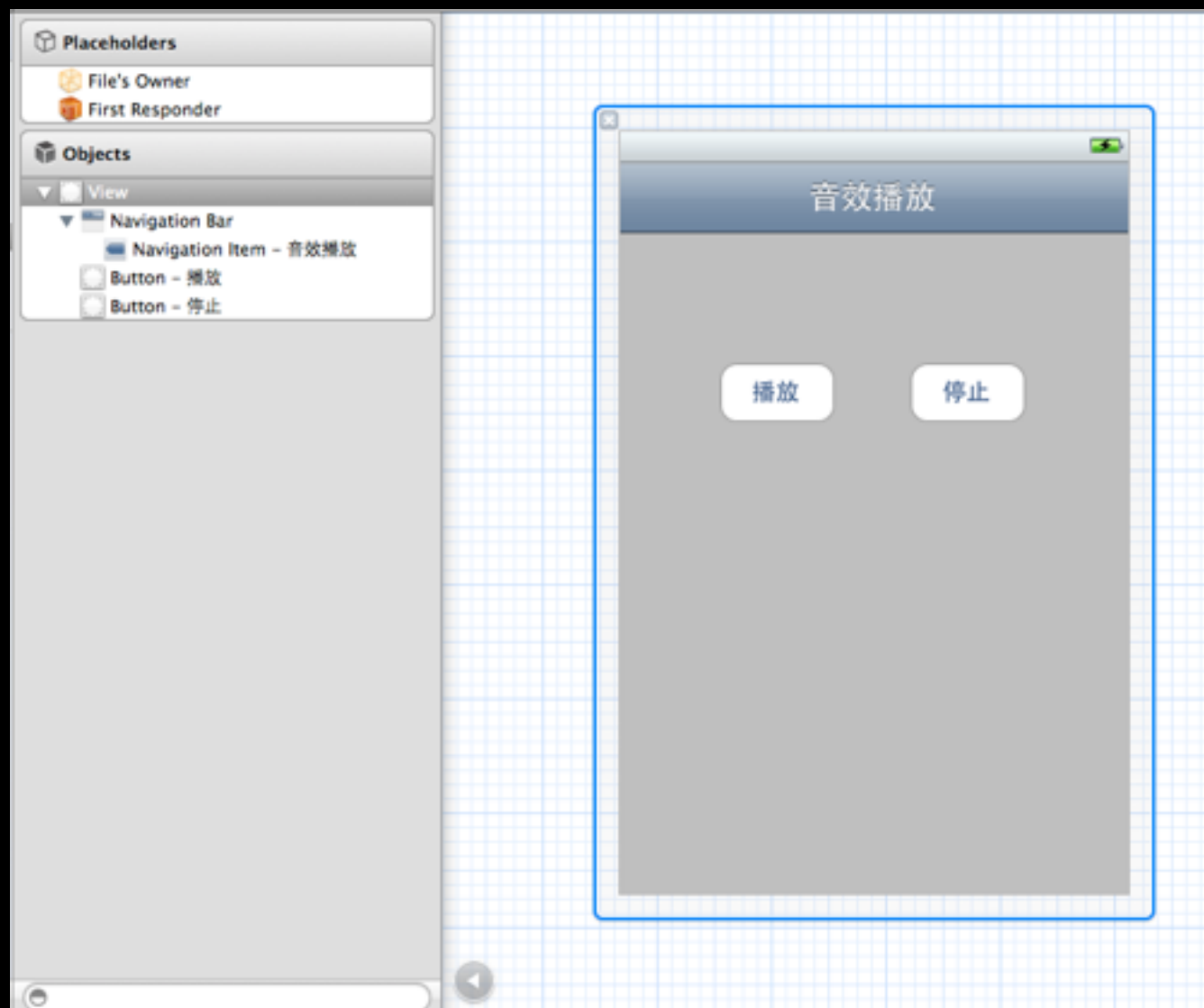
MediaSample(4/13)

5. 選擇音訊檔案
後，並勾選
Copy items into
destination
group's folder
後
點選 Add。



MediaSample(5/13)

1. 在View上加入一個 **Navigation Bar** 和兩個 **Button**，並將View的配置完成如下圖所示。



MediaSample(6/13)

7. 在 **MediaSample01ViewController.h** 中加入兩個 **UIButton** 的 **Outlet**，名稱如下所示並與元件建立參考。

```
#import <UIKit/UIKit.h>

@interface MediaSample01ViewController : UIViewController
{
    IBOutlet UIButton *playBtn;
    IBOutlet UIButton *stopBtn;
}

@property (nonatomic, retain) IBOutlet UIButton *playBtn;
@property (nonatomic, retain) IBOutlet UIButton *stopBtn;

@end
```

MediaSample(7/13)

8. 加入 **AVFoundation/AVFoundation.h** 的 import。

9. 新增 **AVAudioPlayer** 屬性，並加上合成方法。

```
#import <UIKit/UIKit.h>
#import <AVFoundation/AVFoundation.h>

@interface MediaSample01ViewController : UIViewController {
    IBOutlet UIButton *playBtn;
    IBOutlet UIButton *stopBtn;
    AVAudioPlayer *audioPlayer;
}

@property (nonatomic, retain) IBOutlet UIButton *playBtn;
@property (nonatomic, retain) IBOutlet UIButton *stopBtn;
@property (nonatomic, retain) AVAudioPlayer *audioPlayer;

@end
```


MediaSample(8/13)

10. 新增兩個 IBAction，名稱為『**play**』和『**stop**』與 View 上的 Button 元件的『**Touch Up Inside**』事件建立關聯。

```
#import <UIKit/UIKit.h>
#import <AVFoundation/AVFoundation.h>

@interface MediaSample01ViewController : UIViewController {
    IBOutlet UIButton *playBtn;
    IBOutlet UIButton *stopBtn;
    AVAudioPlayer *audioPlayer;
}

@property (nonatomic, retain) IBOutlet UIButton *playBtn;
@property (nonatomic, retain) IBOutlet UIButton *stopBtn;
@property (nonatomic, retain) AVAudioPlayer *audioPlayer;

- (IBAction)play:(id)sender;
- (IBAction)stop:(id)sender;

@end
```

MediaSample(9/13)

- 1. 在 **MediaSample01ViewController.m** 中加上 **audioPlayer** 的 **synthesize** 和實作 **dealloc** 方法。

```
#import "MediaSample01ViewController.h"

@implementation MediaSample01ViewController

@synthesize playBtn, stopBtn, audioPlayer;

- (void)dealloc
{
    [super dealloc];
    [playBtn release];
    [stopBtn release];
    [audioPlayer release];
}
```

MediaSample(10/13)

12. 實作 **play** 方法。

```
- (IBAction)play:(id)sender
{
    // Make sure the audio is at the start of the stream.
    self.audioPlayer.currentTime = 0;

    [self.audioPlayer play];
}
```

MediaSample(11/13)

13. 實作 **stop** 方法。

```
- (IBAction)stop:(id)sender
{
    [self.audioPlayer stop];
}
```

MediaSample(12/13)

14. 在viewDidLoad方法中加入以下程式碼後執行。

```
- (void)viewDidLoad
{
    [super viewDidLoad];

    // Get the file path to the song to play.
    NSString *filePath = [[NSBundle mainBundle]
        pathForResource:@"rainbow" ofType:@"mp3"];

    // Convert the file path to a URL.
    NSURL *fileURL =
        [[NSURL alloc] initWithPath:filePath];
```

MediaSample(13/13)

```
//Initialize the AVAudioPlayer.  
self.audioPlayer = [[AVAudioPlayer alloc]  
    initWithContentsOfURL:fileURL error:nil];  
  
// Preloads the buffer and prepares the audio for playing.  
[self.audioPlayer prepareToPlay];  
  
[filePath release];  
[fileURL release];  
}
```