

# iOS 程式設計

## 影像輸出入

Brian Wang 王昱景

[brian.wang.frontline@gmail.com](mailto:brian.wang.frontline@gmail.com)

# iOS SDK影像處理API

- iOS影像處理的Framework為Core Graphics Framework。
- iOS 4抽出為ImageIO Framework。
- 可取得影像的大小、DPI、EXIF等資訊。

# Image I/O Programming Guide

## Introduction

- The Image I/O programming interface allows applications to **read and write most image file formats**. Originally part of the **Core Graphics framework**, Image I/O resides in its own framework to allow developers to use it independently of Core Graphics (Quartz 2D). Image I/O provides the definitive way to access image data because it is highly efficient, allows easy access to metadata, and provides color management.
- The Image I/O interface is available in Mac OS X v10.4 and later and **in iOS 4 and later**.

# ImageIO Framework API

- 進行處理影像輸入和取得資訊：  
**CGImageSource**。
- 提供影像輸出入的呈現方式：  
**CGImageDestination**。
- 提供影像屬性資訊說明：  
**CGImageProperties**。



# ImageIO Framework使用三步驟

1. 建立CGImageSourceRef物件（可用url路徑、data物件或data provider取得影像）。
2. 使用CGImageSourceCopyPropertiesAtIndex取得影像物件的屬性dictionary。
3. 取得屬性dictionary後，再使用CFDictionaryGetValue查詢影像屬性dictionary內的細部資訊。

# ImageIOSample01 (1/13)

1. 建置一個 **View-based** 的專案，名稱『**ImageIOSample01**』。
2. 從 Targets > Build Phases > Link Binary With Libraries 加入 **ImageIO Framework**。

# ImageIOSample01 (2/13)

3. 加入 `ImageIO/ImageIO.h` 的 `import`。

4. 在 `ImageIOSample01ViewController.h` 中  
新增 `getImageFromWeb` 方法。

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

@interface ImageIOSample01ViewController :
    UIViewController

- (void)getImageFromWeb;

@end
```

# ImageOSample01 (3/13)

## 5. 在 ImageOSample01ViewController.m 中實作 getImageFromWeb 方法。

```
- (void)getImageFromWeb
{
    CFURLRef url = (__bridge CFURLRef) [NSURL URLWithString:
@"http://pic.pimg.tw/rita26/ff6b8b83bf6868641013c85a56df783d.jpg"];

    CGImageSourceRef imageSource = CGImageSourceCreateWithURL(url, NULL);

    CFDictionaryRef imageProperties =
CGImageSourceCopyPropertiesAtIndex(imageSource, 0, NULL);

    NSString *imageWidth = (NSString *)
CFDictionaryGetValue(imageProperties, kCGImagePropertyPixelWidth);
    NSString *imageHeight = (NSString *)
CFDictionaryGetValue(imageProperties, kCGImagePropertyPixelHeight);

    NSLog(@"Image width :%@ ,Image Height : %@", imageWidth, imageHeight);

    CFRelease(imageProperties);
    CFRelease(imageSource);
}
```



# Image Source建立影像時可設定的值

設定值	說明
kCGImageSourceTypeIdentifierHint	影像格式的UTI(uniform type identifier)
kCGImageSourceShouldAllowFloat	允許浮動值，這樣可能會超出邊界需要再作另外處理
kCGImageSourceShouldCache	解碼時進行快取
kCGImageSourceCreateThumbnailFromImageIfAbsent	當沒縮圖時，是否自動建立縮圖
kCGImageSourceCreateThumbnailFromImageAlways	總是建立縮圖，沒指定縮圖最大尺寸的話，會是原圖大小
kCGImageSourceThumbnailMaxPixelSize	縮圖最大像素
kCGImageSourceCreateThumbnailWithTransform	縮圖是否需要旋轉或是縮放例如根據Exif標記自動旋轉

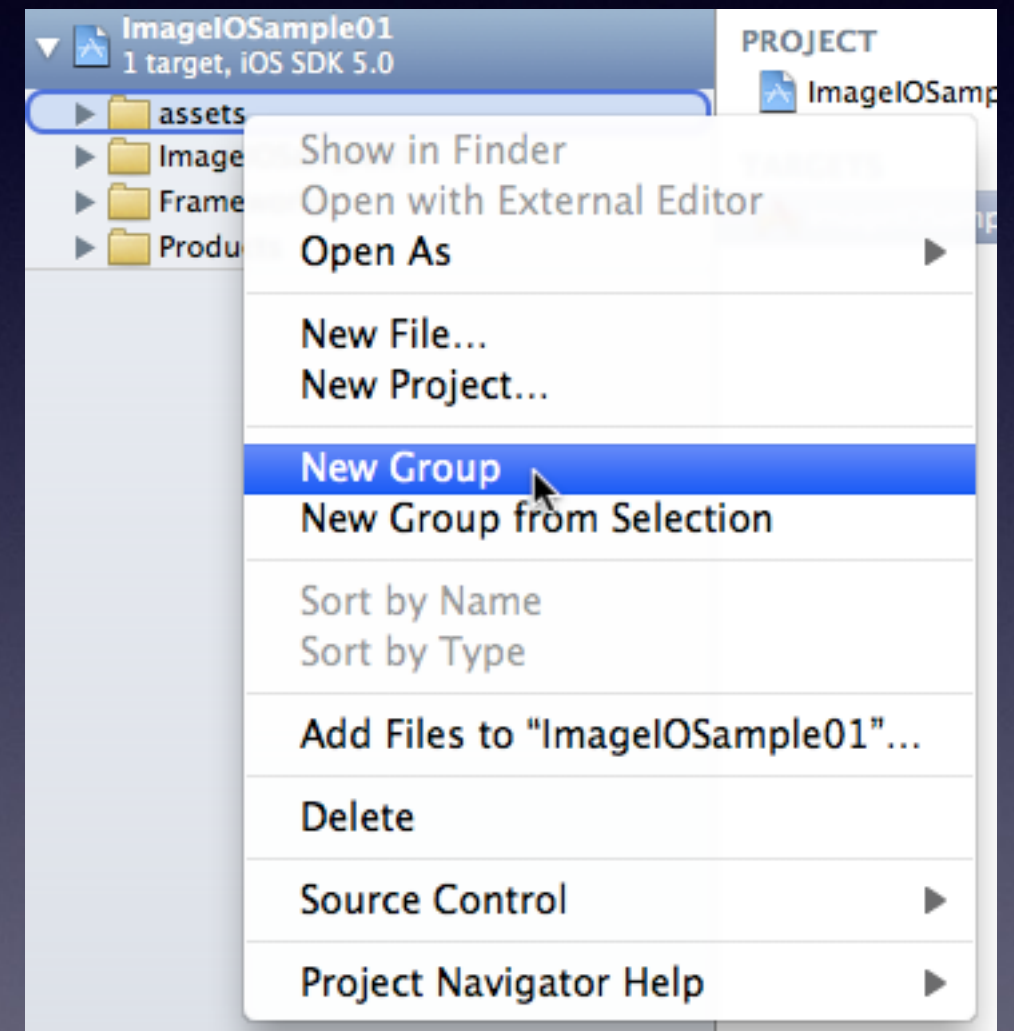
# ImageIOSample01 (4/13)

6. 在viewDidLoad方法中加入以下程式碼後執行。

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    [self getImageFromWeb];
}
```

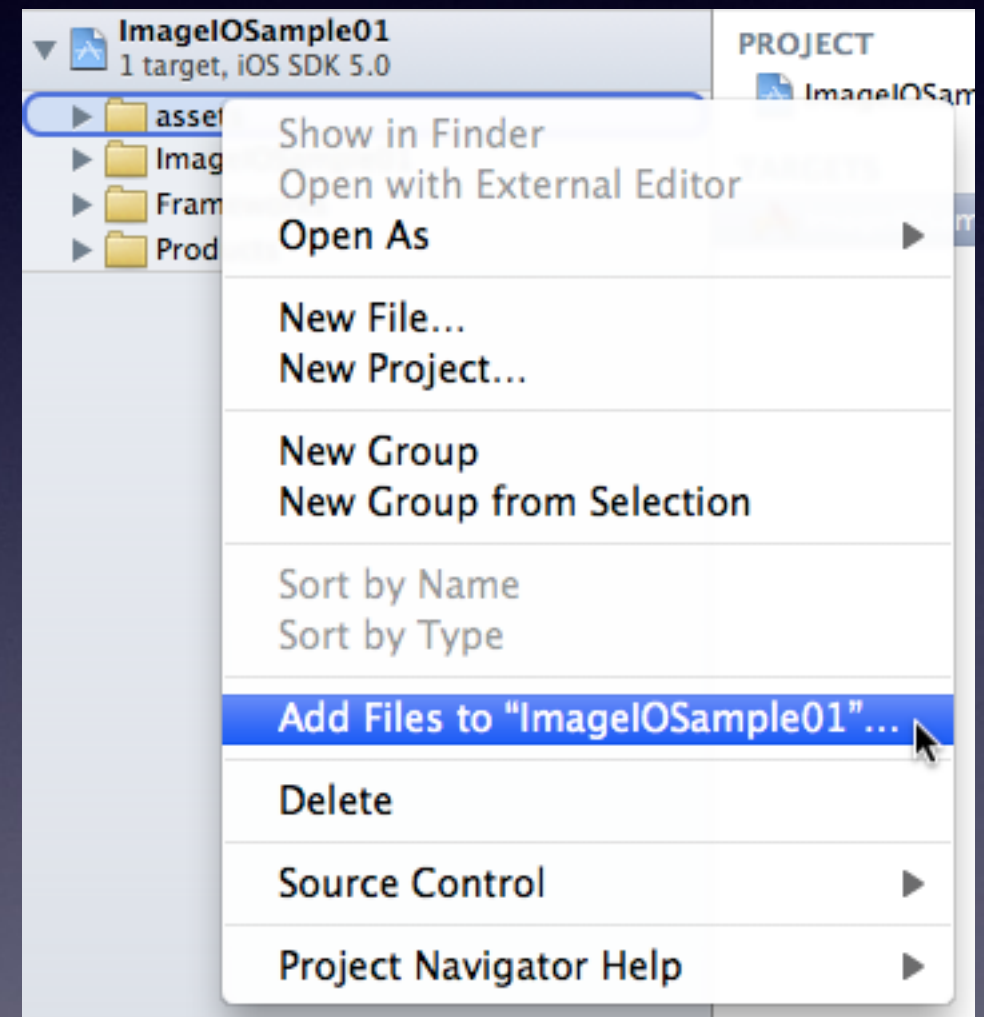
# ImageIOSample01 (5/13)

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



# ImageIOSample01 (6/13)

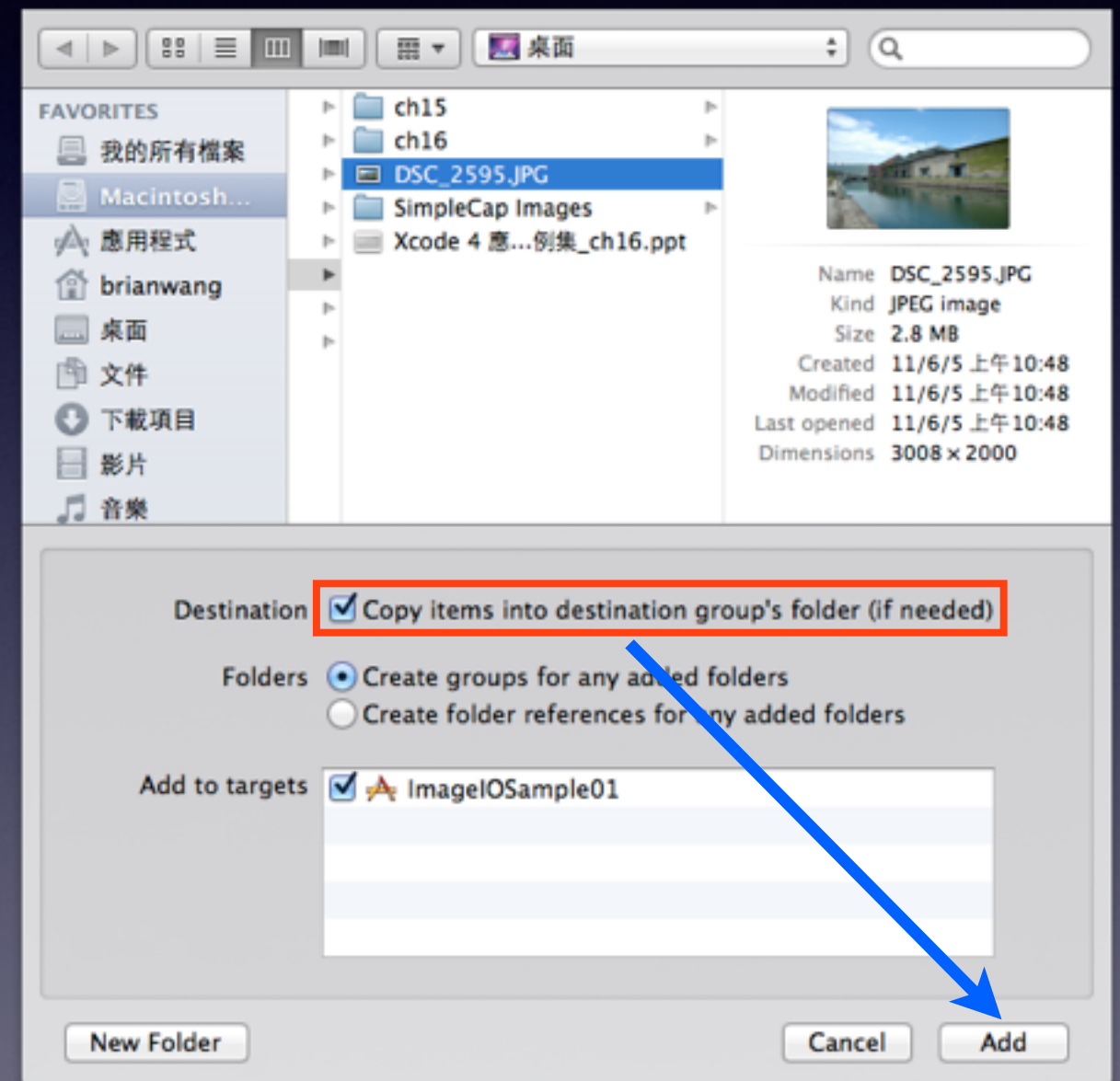
8. 點選滑鼠右鍵，選擇 **Add File to**  
“ImageIOSample01”  
，加入影像檔案。





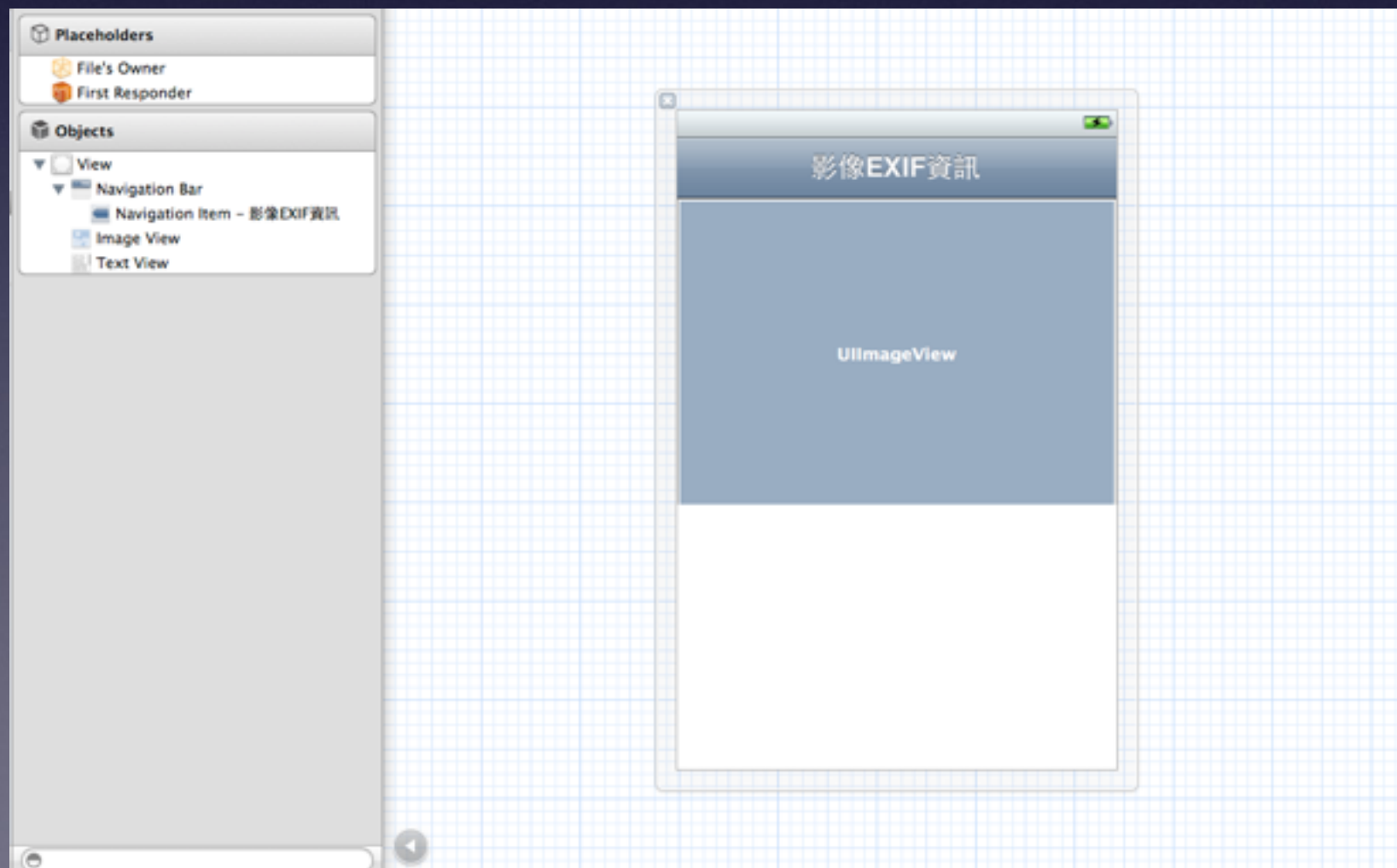
# ImageIOSample01 (7/13)

9. 選擇影像檔案  
後，並勾選  
Copy items into  
destination  
group's folder  
後  
點選 Add 。



# ImageOSample01 (8/13)

10. 在 View 上加入一個 **Navigation Bar**、**Image View** 和 **Text View**，並將 View 的配置完成如下圖所示。



# ImageIOSample01 (9/13)

- 1. 在 **ImageIOSample01ViewController.h** 中加入 **UIImageView** 及 **UITextView** 的 Outlet，名稱如下所示並與元件建立參考。

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

@interface ImageIOSample01ViewController : UIViewController {
    IBOutlet UIImageView *myImageView;
    IBOutlet UITextView *myImageInfo;
}

@property (nonatomic, retain) IBOutlet UIImageView *myImageView;
@property (nonatomic, retain) IBOutlet UITextView *myImageInfo;

- (void)getImageFromWeb;

@end
```

# ImageIOSample01 (10/13)

## 12. 新增 **showImage** 方法。

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

@interface ImageIOSample01ViewController : UIViewController {
    IBOutlet UIImageView *myImageView;
    IBOutlet UITextView *myImageInfo;
}

@property (nonatomic, retain) IBOutlet UIImageView *myImageView;
@property (nonatomic, retain) IBOutlet UITextView *myImageInfo;

- (void)getImageFromWeb;
- (void)showImage;

@end
```



# EXIF (Exchangeable image file format)

- EXIF (Exchangeable image file format)是**可交換圖像文件**的縮寫，是專門為**數位相機**的照片設定的，可以記錄數位照片的屬性信息和拍攝數據。
- EXIF最初由日本電子工業發展協會在1996年制定，版本為1.0。1998年，升級到2.1，增加了對音頻文件的支持。2002年3月，發表了**2.2**版。
- EXIF可以附加於**JPEG**、**TIFF**、**RIFF**等文件之中，為其增加有關數位相機拍攝信息的內容和索引圖或圖像處理軟體的版本信息。

# EXIF (Exchangeable image file format)

項目	資訊
製造廠商	Canon
相機型號	Canon EOS-1Ds Mark III
影像方向	正常 (upper-left)
影像解析度 X	300
影像解析度 Y	300
解析度單位	dpi
Software	Adobe Photoshop CS Macintosh
最後異動時間	2005:10:06 12:53:19
YCbCrPositioning	2
曝光時間	0.00800 (1/125) sec
光圈值	F22
拍攝模式	光圈優先

ISO 感光值	100
EXIF 資訊版本	30,32,32,31
影像拍攝時間	2005:09:25 15:00:18
影像存入時間	2005:09:25 15:00:18
曝光補償 (EV+-)	0
測光模式	點測光 (Spot)
閃光燈	關閉
鏡頭實體焦長	12 mm
Flashpix 版本	30,31,30,30
影像色域空間	sRGB
影像尺寸 X	5616 pixel
影像尺寸 Y	3744 pixel

# ImageOSample01 (11/13)

## 13. 在 ImageOSample01ViewController.m 中實作 showImage 方法。

```
- (void)showImage{
    NSURL *modelURL = [[NSBundle mainBundle] URLForResource:
@"test.JPG" withExtension:nil];

    CGImageSourceRef myImageSource = CGImageSourceCreateWithURL(
(CFURLRef) modelURL, NULL);

    CFDictionaryRef imageProperties =
CGImageSourceCopyPropertiesAtIndex(myImageSource, 0, NULL);

    CGImageRef image = CGImageSourceCreateImageAtIndex(myImageSource, 0, NULL);

    [myImageView setImage:[UIImage imageWithCGImage:image]];

    CFDictionaryRef tiff = CFDictionaryGetValue(
imageProperties, kCGImagePropertyTIFFDictionary);
    CFDictionaryRef exif = CFDictionaryGetValue(
imageProperties, kCGImagePropertyExifDictionary);
```

# ImageIOSample01 (12/13)

```
NSString *model = (NSString *)
    CFDictionaryGetValue(tiff, kCGImagePropertyTIFFModel);
NSString *make = (NSString *)
    CFDictionaryGetValue(tiff, kCGImagePropertyTIFFMake);
NSString *focallength = (NSString *)
    CFDictionaryGetValue(exif, kCGImagePropertyExifFocalLength);
NSString *datetime = (NSString *)
    CFDictionaryGetValue(exif, kCGImagePropertyExifDateTimeOriginal);
NSString *aperture = (NSString *)
    CFDictionaryGetValue(exif, kCGImagePropertyExifApertureValue);
NSString *shutterspeed = (NSString *)
    CFDictionaryGetValue(exif, kCGImagePropertyExifShutterSpeedValue);
NSString *ExposureTime = (NSString *)
    CFDictionaryGetValue(exif, kCGImagePropertyExifExposureTime);

myImageInfo.text = [NSString stringWithFormat:
    @"製造商： %@\n設備型號： %@\n焦距： %@\n拍攝時間： %@\n光圈： %@\n快門速度： %@\n曝光時間： %@\n",
    make, model, focallength, datetime, aperture, shutterspeed, ExposureTime];

myImageInfo.backgroundColor = [UIColor redColor];

CFRelease(myImageSource);
CFRelease(imageProperties);
}
```



# ImageIOSample01 (13/13)

9. 在viewDidLoad方法中加入以下程式碼後執行。

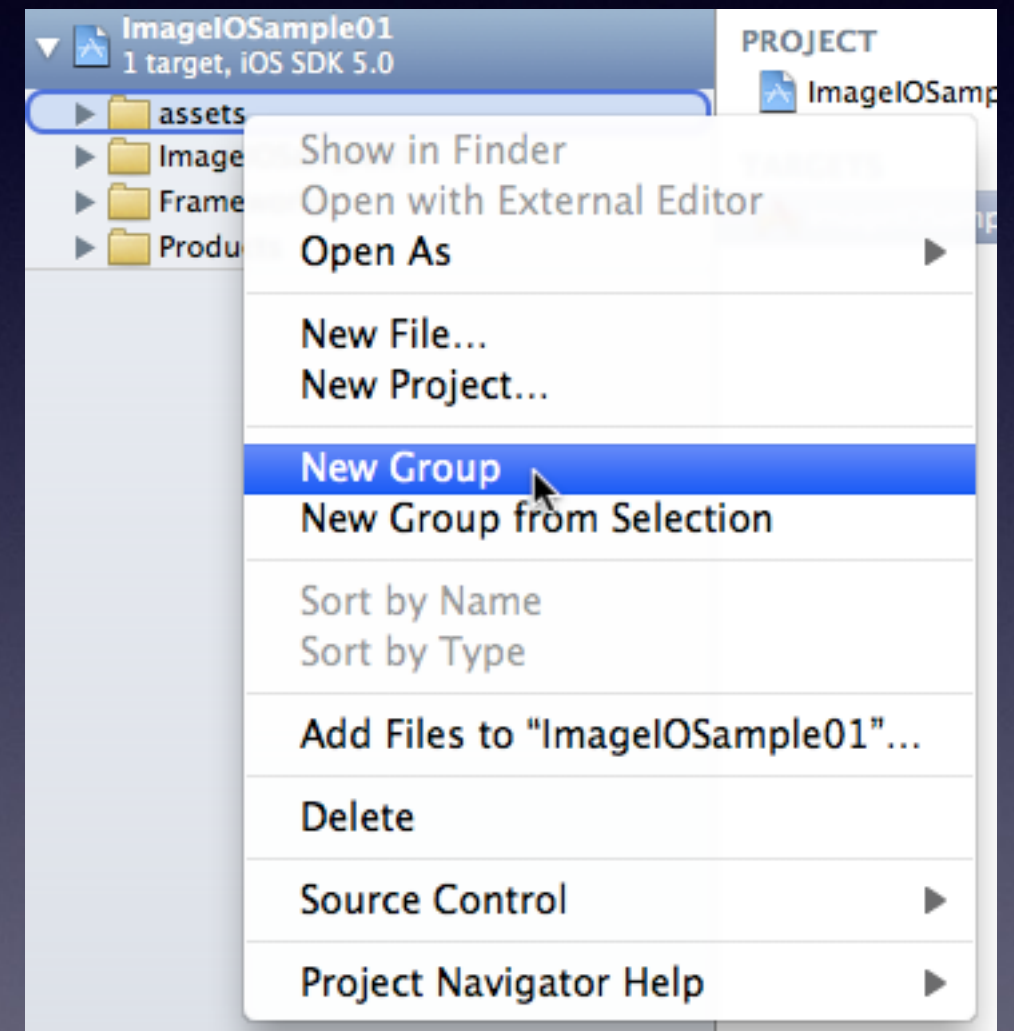
```
- (void)viewDidLoad
{
    [super viewDidLoad];
    [self getImageFromWeb];
    [self showImage];
}
```

# ImageIO Sample02(1/10)

1. 建置一個 **View-based** 的專案，名稱『**ImageIO Sample02**』。
2. 從 Targets > Build Phases > Link Binary With Libraries 加入 **ImageIO Framework**。

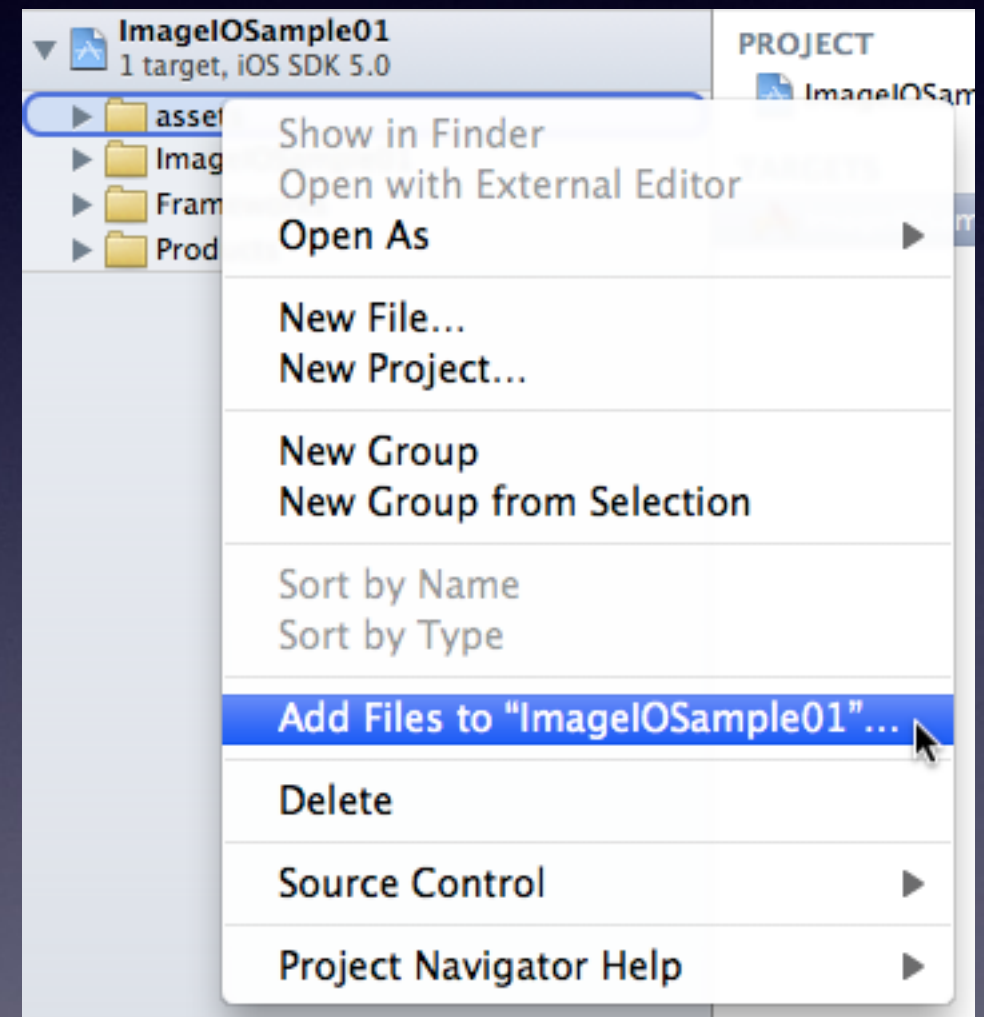
# ImageIOSample02(2/10)

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



# ImageIOSample02(3/10)

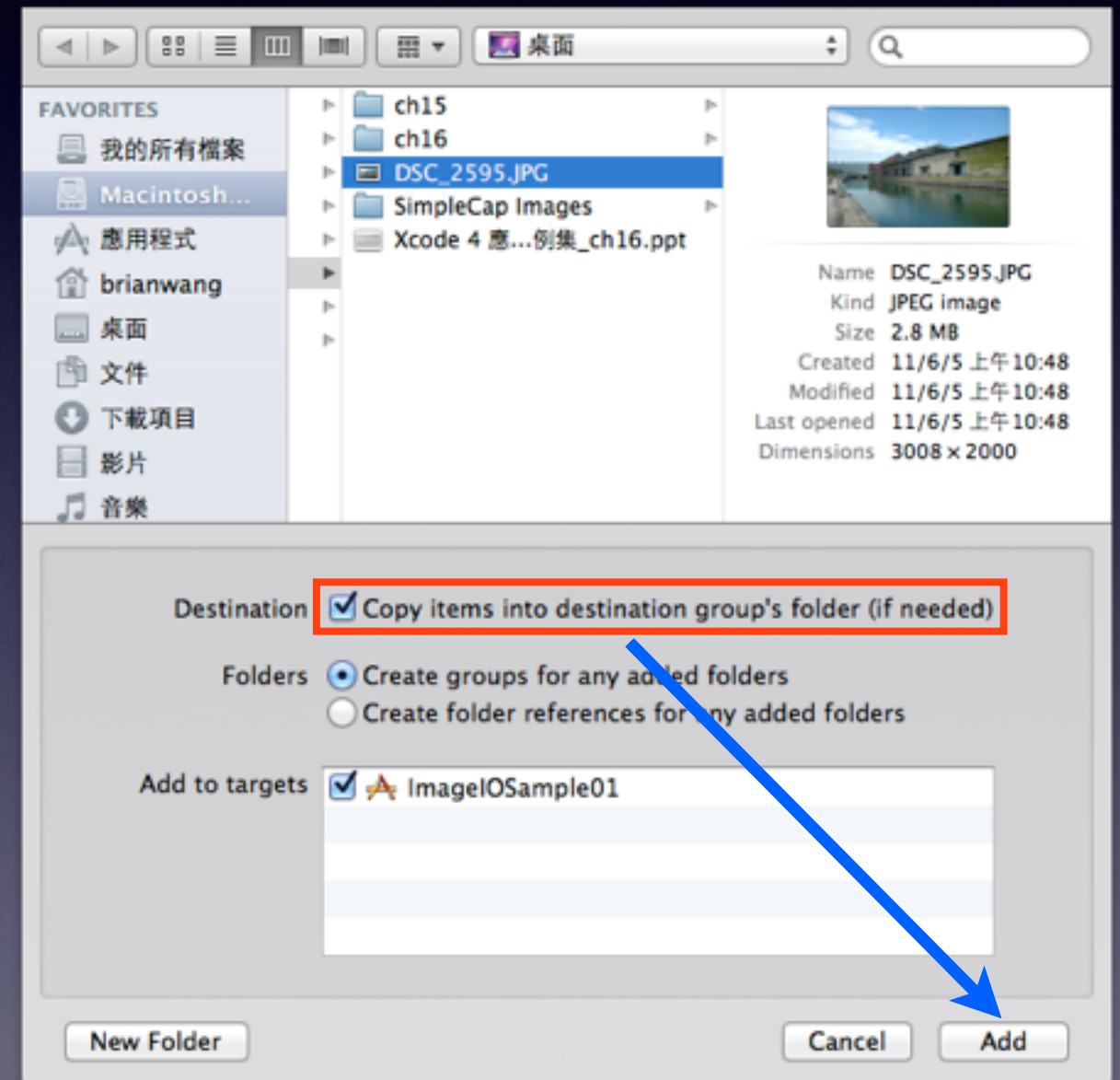
4.點選滑鼠右鍵，選擇 **Add File to**  
“ImageIOSample02”  
，加入影像檔案。





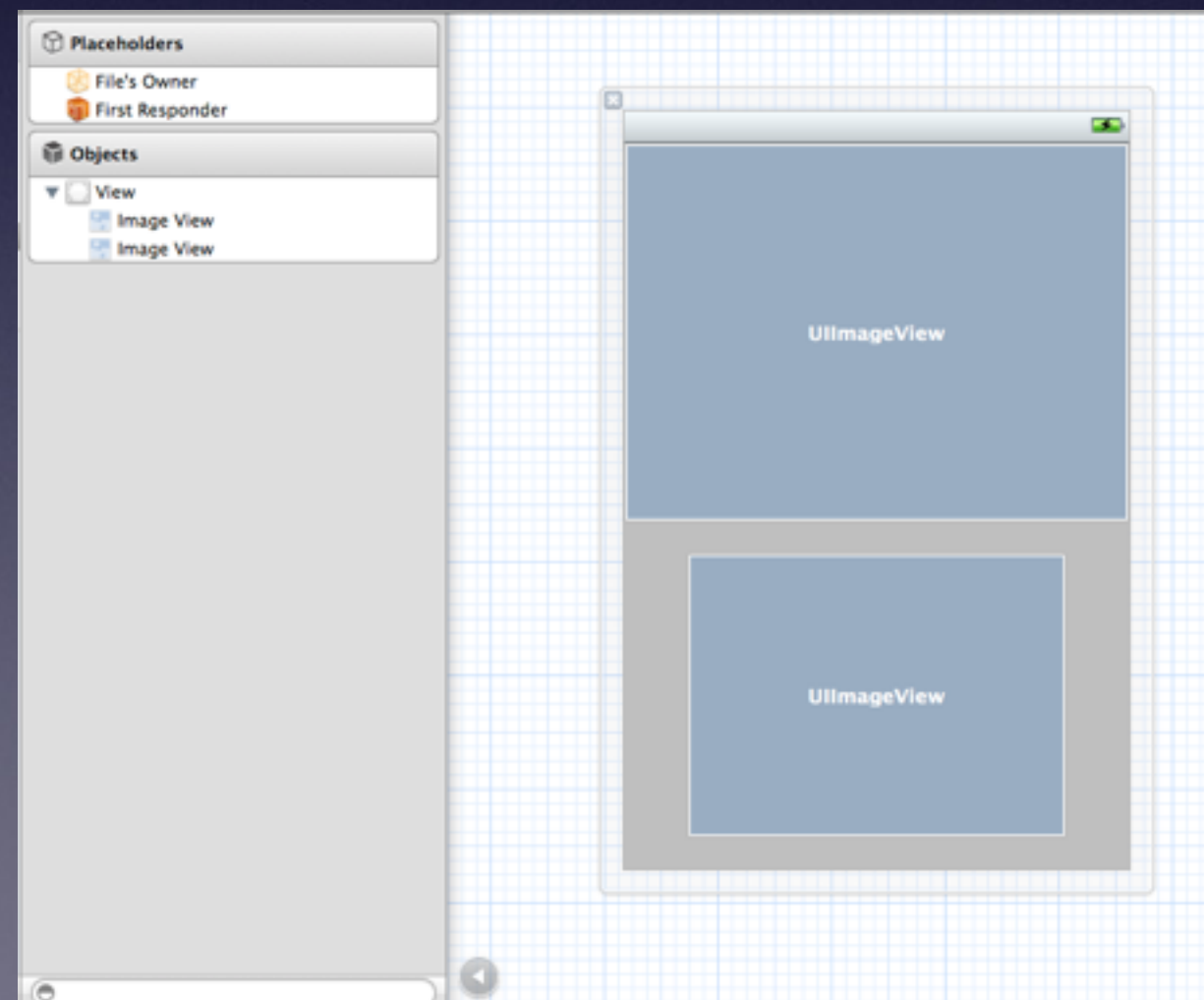
# ImageIOSample02(4/10)

5. 選擇影像檔案  
後，並勾選  
Copy items into  
destination  
group's folder  
後  
點選 Add 。



# ImageOSample02(5/10)

6. 在 View 上加入兩個 **Image View**，並將 View 的配置完成如下圖所示。



# ImageIOSample02(6/10)

7. 在 **ImageIOSample02ViewController.h** 中加入兩個 **UIImageView** 的 Outlet，名稱如下所示並與元件建立參考。

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

@interface ImageIOSample02ViewController : UIViewController {
    IBOutlet UIImageView *myImageView;
    IBOutlet UIImageView *myThumbnailView;
}

@property (nonatomic, retain) IBOutlet UIImageView *myImageView;
@property (nonatomic, retain) IBOutlet UIImageView *myThumbnailView;

@end
```

# ImageIOSample02(7/10)

8. 加入 **ImageIO/ImageIO.h** 的 import。

9. 在 **ImageIOSample02ViewController.h** 中新增 **showImages** 方法。

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

@interface ImageIOSample02ViewController : UIViewController {
    IBOutlet UIImageView *myImageView;
    IBOutlet UIImageView *myThumbnailView;
}

@property (nonatomic, retain) IBOutlet UIImageView *myImageView;
@property (nonatomic, retain) IBOutlet UIImageView *myThumbnailView;

- (void)showImages;

@end
```



# ImageIOSample02(8/10)

10. 在 ImageIOSample02ViewController.m 中實作 showImage 方法。

```
- (void)showImages
{
    NSURL *modelURL = [[NSBundle mainBundle]
    URLForResource:@"test.JPG" withExtension:nil];
    CGImageSourceRef myImageSource =
    CGImageSourceCreateWithURL((CFURLRef) modelURL, NULL);
    CFDictionaryRef imageProperties =
    CGImageSourceCopyPropertiesAtIndex(myImageSource, 0, NULL);

    int imageSize = 240;

    CFNumberRef thumbnailSize =
    CFNumberCreate(NULL, kCFNumberIntType, &imageSize);
    CFTypeRef isCreateThumbnail = kCFBooleanTrue;
```

# ImageIOSample02(9/10)

```
NSString *createThumbnailFromImage = (NSString *)
    kCGImageSourceCreateThumbnailFromImageIfAbsent;
    NSString *ThumbnailMaxSize = (NSString *)
    kCGImageSourceThumbnailMaxPixelSize;

    NSDictionary *options = [[NSDictionary alloc]
initWithObjectsAndKeys:(void *) isCreateThumbnail,(void *)
createThumbnailFromImage,(void *) thumbnailSize, (void *)
ThumbnailMaxSize, nil];

    CGImageRef image =
CGImageSourceCreateImageAtIndex(myImageSource, 0, NULL);
    CGImageRef thumbnailimage =
CGImageSourceCreateThumbnailAtIndex(myImageSource, 0,
(CFDictionaryRef) options);

    [myImageView setImage:[UIImage imageWithCGImage:image]];
    [myThumbnailView setImage:
[UIImage imageWithCGImage:thumbnailimage]];

    CFRelease(myImageSource);
    CFRelease(imageProperties);
}
```

# ImageIOSample02(10/10)

||.在viewDidLoad方法中加入以下程式碼後執行。

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    [self showImages];
}
```