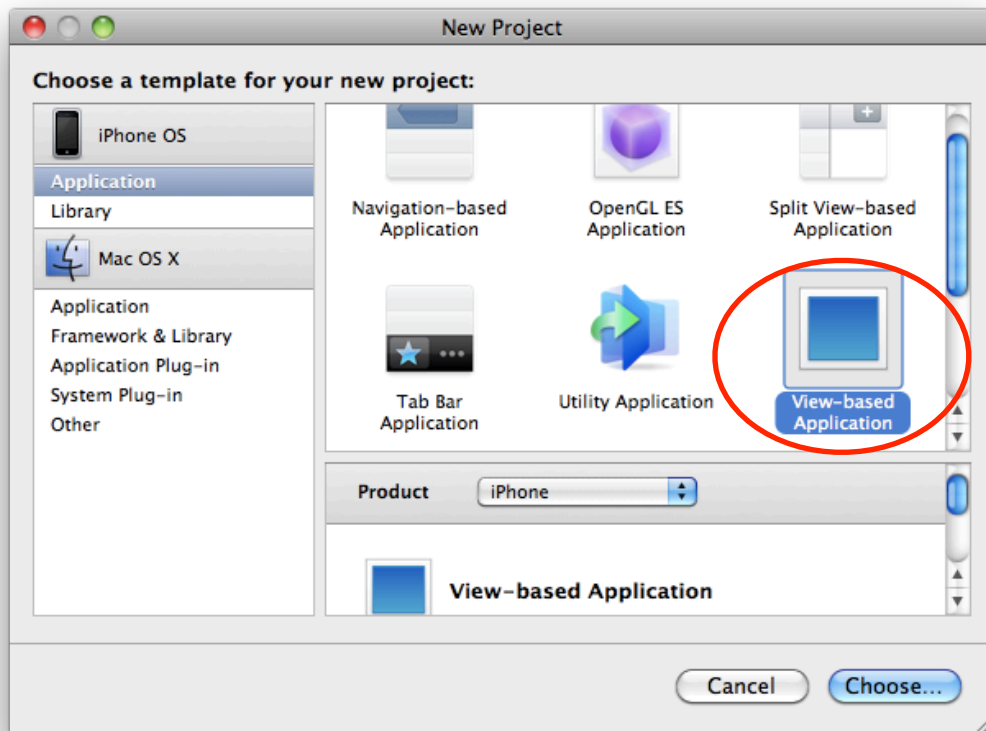
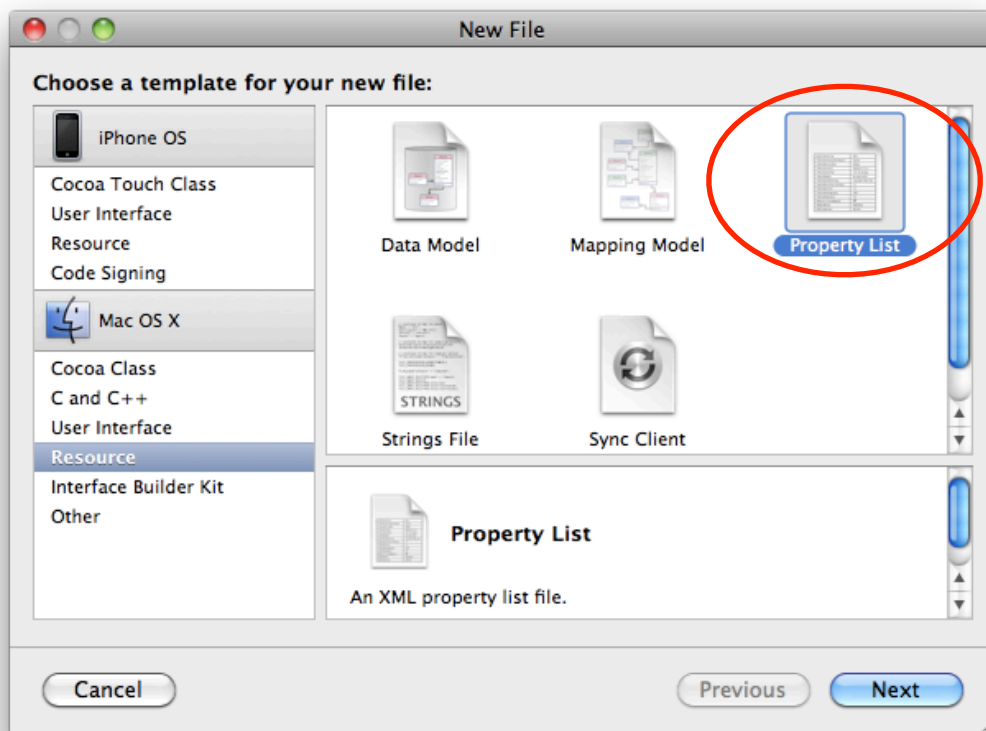


Lab FileReadWrite

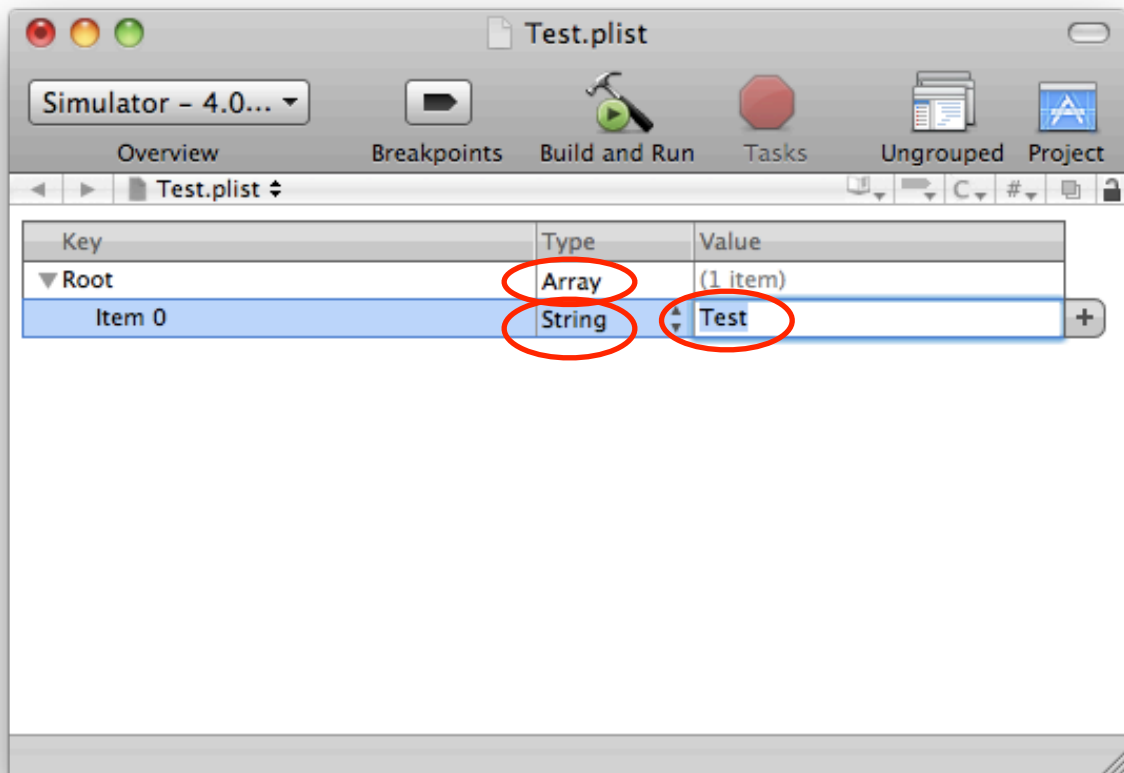
Step1. 在File開啓一個新的project, 在Application目錄裡面選擇 view based application, 將project命名為 lab_FileReadWrite



Step2. 在File開啓一個新的File, 在Mac OS X裡Resource裡面選擇Property List, 命名為 Test



Step3. 開啓剛新增 Test.plist 將Type 設定為Array, 並按右鍵add Row加入第一個Item: Test , 注意Type為String

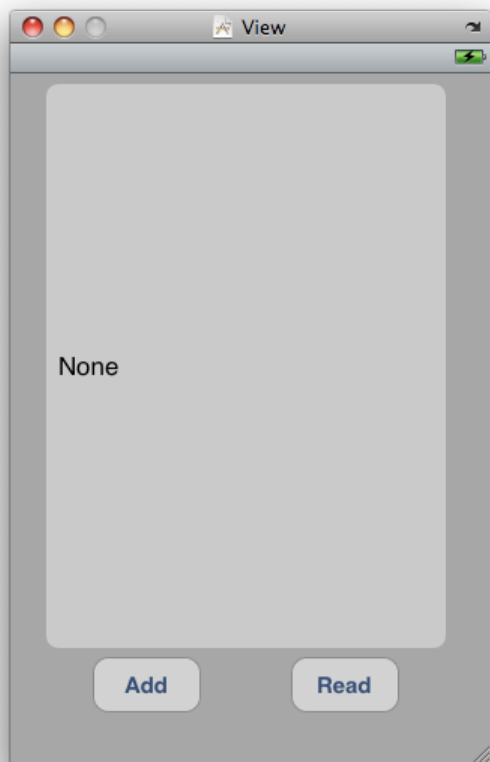


Step4. 在lab_FileReadWriteViewController.h裡加入View裡對應的物件, 物件對應的Action, 以及我們會用到的變數和method

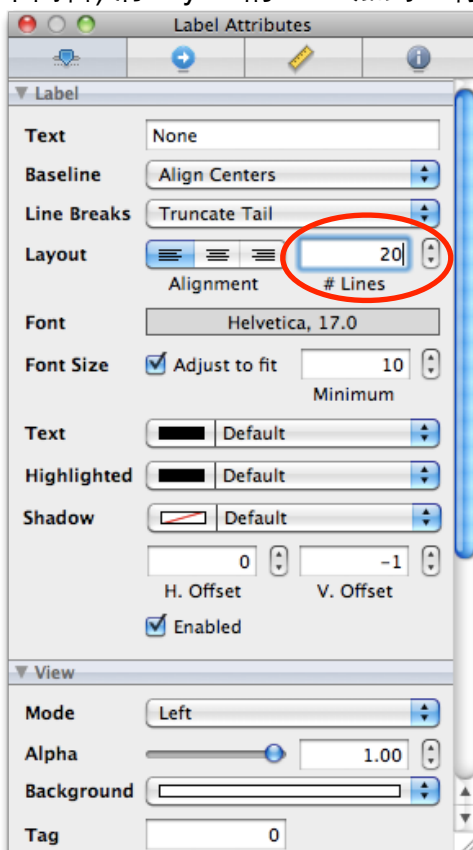
```
#import <UIKit/UIKit.h>
```

```
@interface lab_FileReadWriteViewController : UIViewController {  
    IBOutlet UILabel *myLabel;  
    IBOutlet UIButton *myAddButton;  
    IBOutlet UIButton *myReadButton;  
    NSMutableString *information;  
    NSMutableArray *setArray;  
    NSMutableArray *newArray;  
}  
- (IBAction) add;  
- (IBAction) read;  
- (NSString *) fileLocation;  
@end
```

Step5. 在Resources裡開啓lab_FileReadWriteViewController.xib, 加入我們需要的物件: 一個Label和兩個Button, 並將Label 預設的 text 設為None, 左邊的Button 的 Title 設為Add, 右邊的Button 的 Title 設為 Read. 並將加入的Label的Frame盡量拉大, 以便顯示更多資訊,



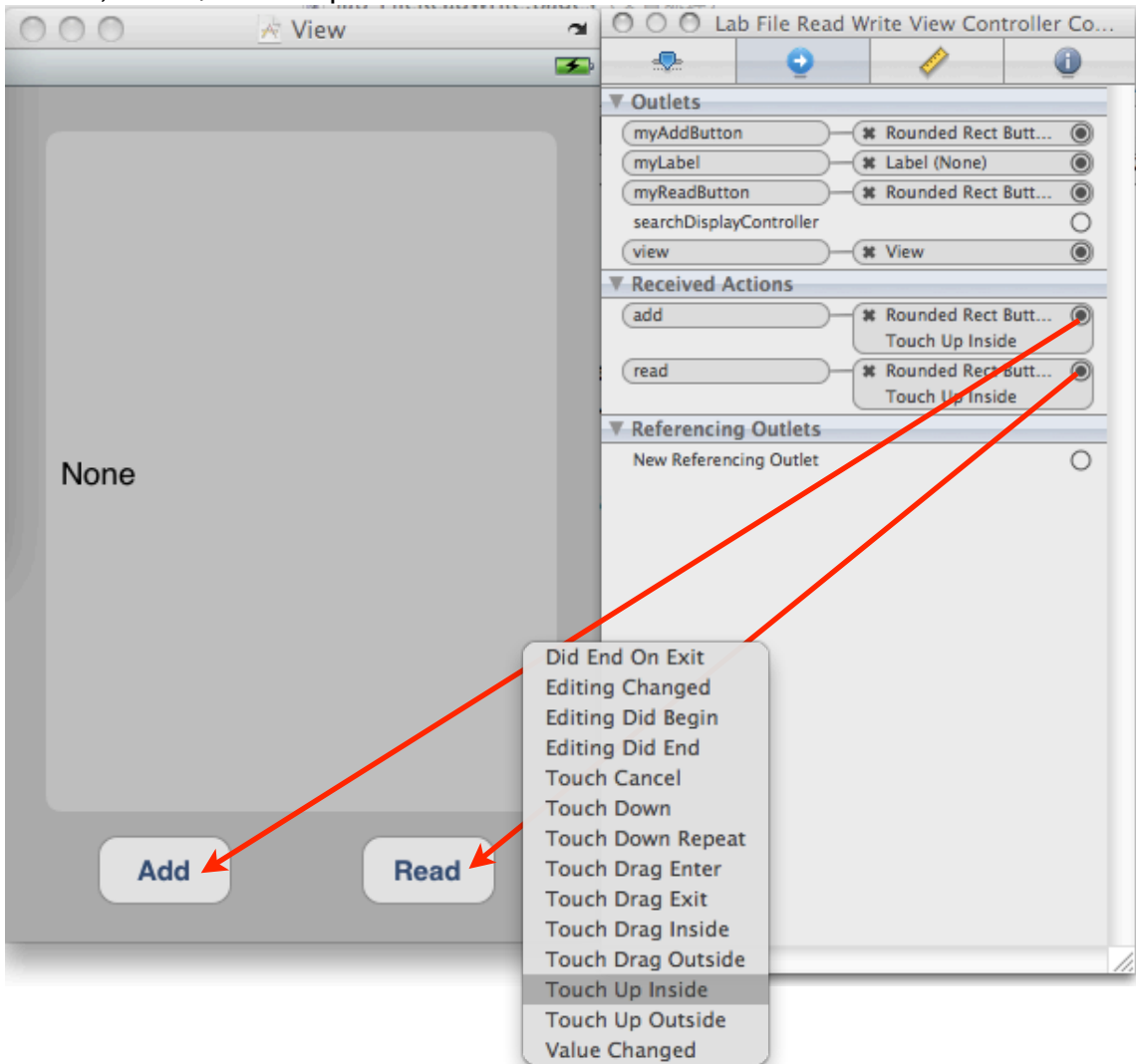
Step6. 點選新增的Label, 在他的Inspector裡(若無開啓在點選Label之後 command + shift + I 來開啓) 將Layout的Lines加到20行, 以便顯示更多我們新加入的資訊



Step7. 點選File's Owner來開啓他的Inspector, 在Connections連結所有的宣告的物件以及對應的IBAction

myLabel連到上放的Label; myAddButton連到左邊的button; myReadButton連到右邊的Button

add這個IBAction連到左邊的Button, 並選擇Touch Up Inside; read這個IBAction連到右邊的Button, 並選擇Touch Up Inside



Step8. 在lab_FileReadWriteViewController.m裡, 將viewDidLoad這個Method的Mark去掉, 加入初始化顯示用的NSMutableString “information” 以及從剛剛新增的plist檔案的Array來作初始化的 NSMutableArray “setArray”

// Implement viewDidLoad to do additional setup after loading the view, typically from a nib.

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    information = [[NSMutableString alloc] init];  
    setArray = [[NSMutableArray alloc] initWithContentsOfFile:[self  
fileLocation]];
```

```
}
```

Step9. 同樣在lab_FileReadWriteViewController.m裡，我們實作對應兩個UIButton的IBAction，
add這個method在剛剛初始化的setArray新增一個由現在時間產生的NSString物件
[[NSDate date] description]來顯示新增物件的時間，並再寫入Test.plist這個檔案裡
read這個method則負責將Test.plist裡Array裡的每個String取出來並且顯示在myLabel裡
並實作fileLocation這個method去每次動態取得Test.plist位置的Path來給每次寫入和讀出使用

```
- (IBAction)add{
    [setArray addObject:@"Hello at " stringByAppendingString:[NSDate
date] description]];
    [setArray writeToFile:[self fileLocation] atomically:YES];
}

- (IBAction)read{
    newArray = [[NSMutableArray alloc] initWithContentsOfFile:[self
fileLocation]];
    for(NSString *ele in newArray){
        [information appendString:[ele
stringByAppendingString:@"\n"]];
    }
    [myLabel setText:information];
    [information setString:@""];
}

- (NSString * ) fileLocation {
    return [[NSBundle mainBundle] pathForResource:@"Test"
ofType:@"plist"];
}
```

Step10. Simulation (command + enter)

一開啓Label顯示預設text “None”,按一下Read, 顯示我們一開始寫入plist的第一個Item “Test”



按一下Add這個Button, 可加入現在時間的NSString物件並可再按一次Read這個Button來顯示出來,多按幾次Add Button, 再按Read Button可以顯示我們剛在不同時間加入的顯示時間的NSString Object

