AsReader

SDK 1.4 User Guide

Revision History

Version	Date	Changes
Rev. 00	2014-10-08	Initially released
Rev. 01	2014-10-09	Add description about Optional.
Rev. 02	2014-10-22	Modify manual title to indicate the supported AsReader version to 1.4, add 2.2 and modify 4.1 add 4.2
Rev. 03	2014-11-03	Add information about optional delegate methods 4.3
Rev. 04	2017-12-15	AsReaderSDK Function

Requirqements

To use AsReader you have to add ExternalAccessory.framework to your project.

1. Add AsReader SDK framework to your project.

1.1 Click target -> Build phases –> Link Binary With Libraries -> +

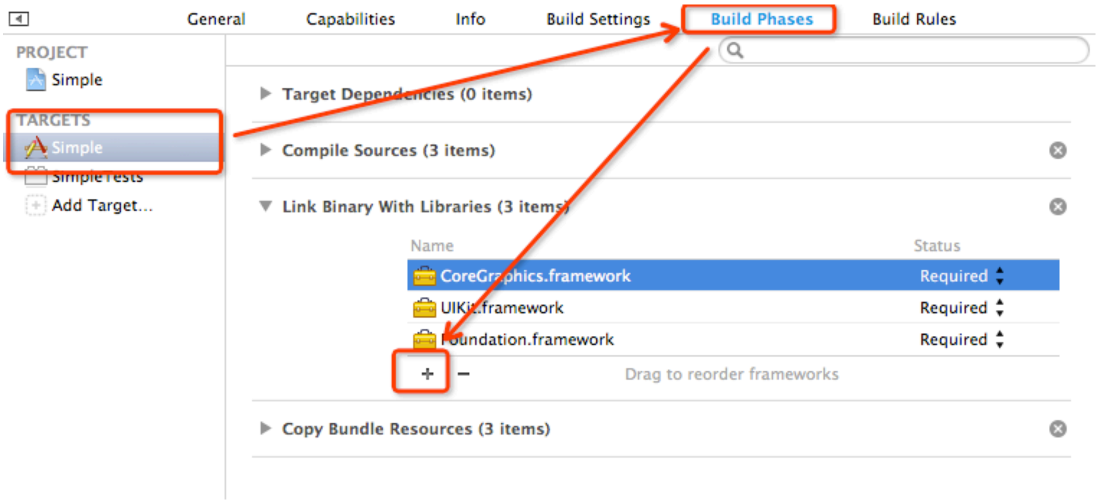
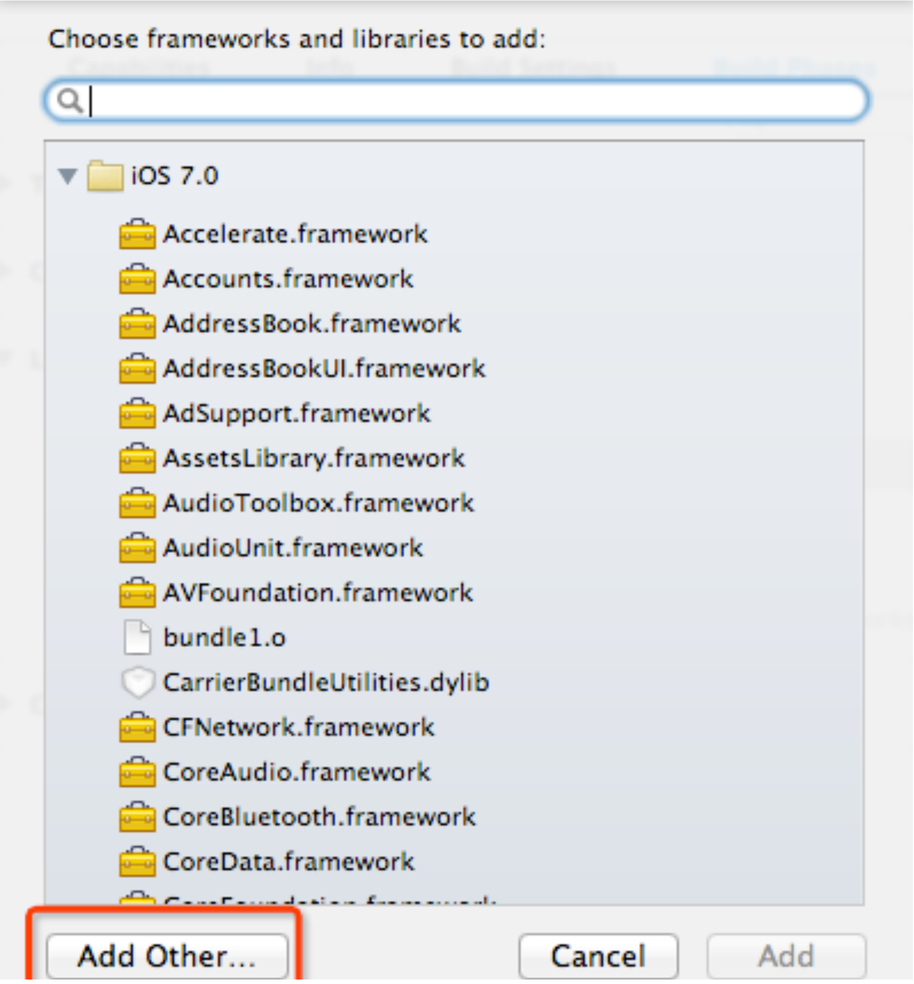
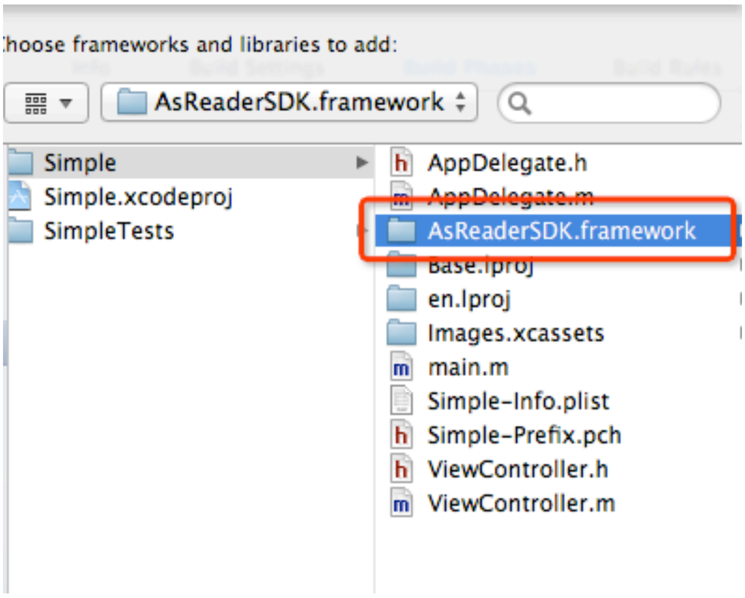


Figure 1

1.2 Click “Add Other …” at the bottom of the popup.



1.3 Find “AsReaderSDK.framework” and click “Open”



:hoose frameworks and libraries to add:

AsReaderSDK.framework

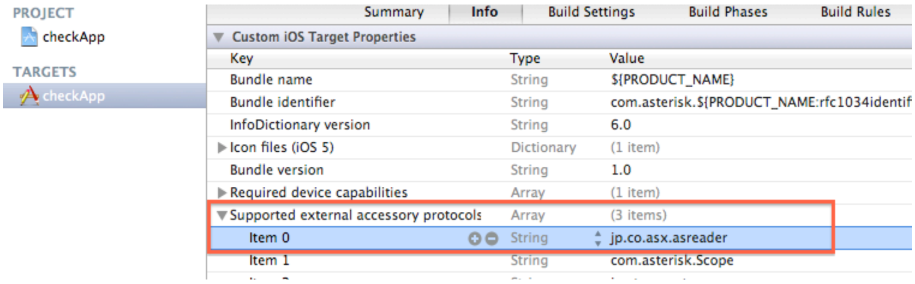
- Simple
- Simple.xcodeproj
- SimpleTests

- AppDelegate.h
- AppDelegate.m
- AsReaderSDK.framework
- Base.lproj
- en.lproj
- Images.xcassets
- main.m
- Simple-Info.plist
- Simple-Prefix.pch
- ViewController.h
- ViewController.m

Figure 3

2. Set your project to support AsReader Protocol

2.1 Add protocol “jp.co.asx.asreader” to Supported external accessory protocols.



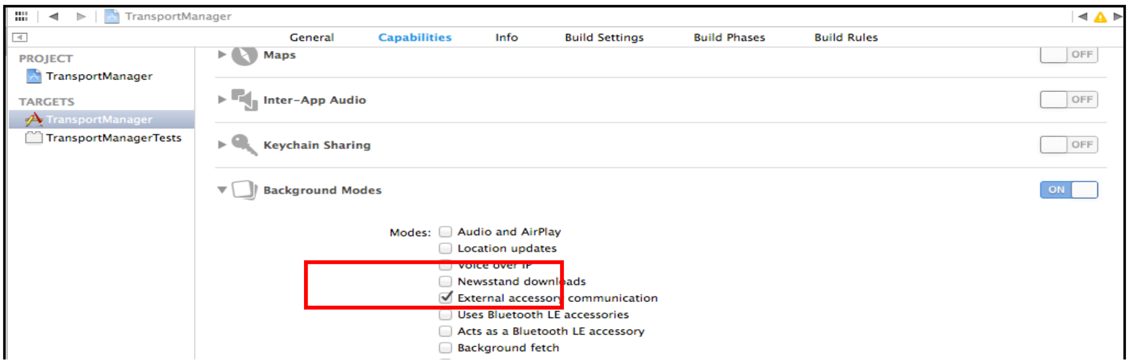
Custom iOS Target Properties

Key	Type	Value
Bundle name	String	${PRODUCT_NAME}
Bundle identifier	String	com.asterisk.${PRODUCT_NAME:rfc1034identif
InfoDictionary version	String	6.0
▶Icon files (iOS 5)	Dictionary	(1 item)
Bundle version	String	1.0
▶Required device capabilities	Array	(1 item)
▼Supported external accessory protocols	Array	(3 items)
Item 0	String	jp.co.asx.asreader
Item 1	String	com.asterisk.Scope

Figure 4

2.2 Because of iOS’s specification, External Accessory connection may be disconnected while the device enter into sleep mode, We recommend to do the following setting to keep the program simple and efficient.

Find Capabilities > Background Modes and CHECK the External accessory communication checkbox.



- Maps OFF
- Inter-App Audio OFF
- Keychain Sharing OFF
- Background Modes ON

Modes:
□ Audio and AirPlay
□ Location updates
□ Voice over IP
□ Newsstand downloads
☑ External accessory communication
□ Uses Bluetooth LE accessories
□ Acts as a Bluetooth LE accessory
□ Background fetch

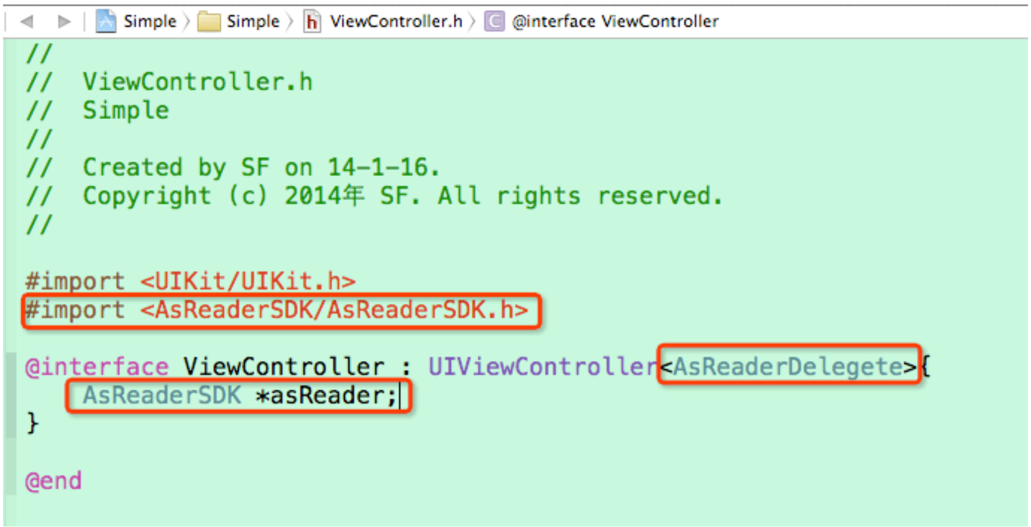
3. Play with AsReader

Now we can make use of AsReaderSDK to implement barcode scanning function.

3.1 First, import the header file like this:#import <AsReaderSDK/AsReaderSDK.h>

3.2 Next, implement the AsReaderDelegate protocol**,** Declare a global AsReader object like this:

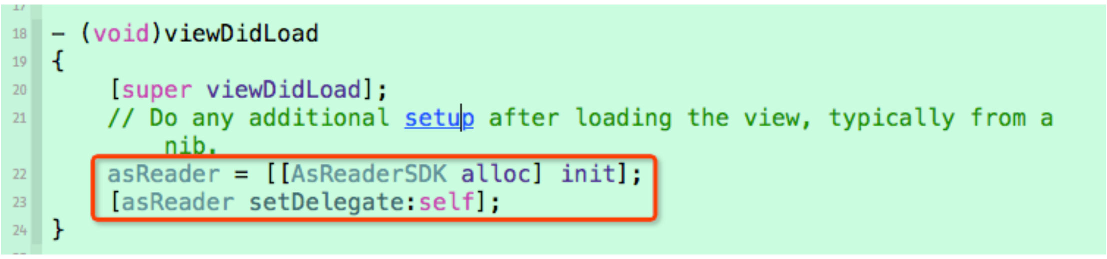
AsReaderSDK *asReader;



```
//
//  ViewController.h
//  Simple
//
//  Created by SF on 14-1-16.
//  Copyright (c) 2014年 SF. All rights reserved.
//

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

@interface ViewController : UIViewController<AsReaderDelegete>{
    AsReaderSDK *asReader;
}

@end
```

Figure 6

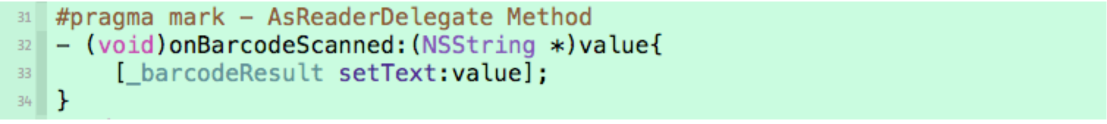
3.3 In the viewdidload method, initialize the asReader object. And set self as delegate.



```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a
       nib.
    asReader = [[AsReaderSDK alloc] init];
    [asReader setDelegate:self];
}
```

Figure 7

3.4 Finally, implement the AsReaderDelegate's onBarcodeScanned method, this is where the scanned barcode will come in as a NSString , so we can consume the barcode here.



```
#pragma mark - AsReaderDelegate Method
- (void)onBarcodeScanned:(NSString *)value{
     [_barcodeResult setText:value];
}
```

Figure 8

4. Optional

4.1 If you don’t want to keep the connection while the app enter into background mode, you have to call the following method to start or stop AsReader session whenever you have to.

- (void)stopAsReaderSession;
- (BOOL)startAsReaderSession;

4.2 It is strongly recommended to make sure you have only ONE instance in the app, because AsReader will not work with more than one instance. What we usually do is to initiate the AsReader SDK instance and keep it as singleton in the App Delegate or wherever can be accessed from all over your app.

4.3 Connection status event.

There are 2 optional delegate method can be used to receive notice when AsReader is connected to or disconnected from iPod/iPhone. Just as easy as implementing the following to delegate method in your AsReader delegate.
- (void)whenAsReaderConnect;
This method will be called when our SDK detected a AsReader connected to the iOS device.
- (void) whenAsReaderDisconnect;
This method will be called when our SDK detected a AsReader disconnected from iOS device.

5.AsReaderSD Function

When use the device with AsReaderSDK to scan barcode, the corresponding Class can be called when the class implements AsReadeSDK Delegate Function.
*It is recommended declare AsReaderSDK delegate in -(void)viewWillAppear(BOOL)animated. then it avoids possibility that fails to return to scan barcode after page redirection.

- -(void)onBarcodeScanned:(NSString *)value;(must be implemented in Delegate)
  The value is the scanned barcode value.
- -(void)whenAsReaderConnect;(can be skipped in Delegate)
  It is called when device is connected.
- -(void)whenAsReaderDisconnect;(can be skipped in Delegate)
  it is called when device is disconnected.
- -(void)stopAsReaderSession;(This is Class function, not Delegate funciton)
  cut off device connection and stop scanning.
- -(BOOL)startAsReaderSession; (This is Class function, not Delegate funciton)
  connet to device to start scanning.

Notes：To avoid the situation that the device doesn’t work when iOS device is woken up from background, do the following：

In Class :AppDelegate

- -(void)applicationWillEnterForeground☹UIApplication *)application
- -(void)applicationDidBecomeActive:(UIApplication *)application

the two functions should implement“stopAsReaderSession”.

Pay attention to the above.