Create Hybrid iOS App Scanner plugin

Download the cordova 3.3.0
Install the Cordova cli tools from command line in cordova directory cordova-3.3.0
command

```
$ npm install -g cordova
```

This will install Cordova globally on this machine, will require administrator password.

Now we can create hybrid App with cordova

```
$ cordova create haloscanner com.example.haloscanner “HaloScanner”
```

This command sets up new directory haloscanner and places cordova JS files

```
$ cd haloscanner
```

No we need to add target platform to our new project to be able to build for iOS, Android etc. We will add iOS only for now.

```
$ cordova platform add ios
```

This command will add iOS related SDK libraries and tools to our project and we can start working with XCode.

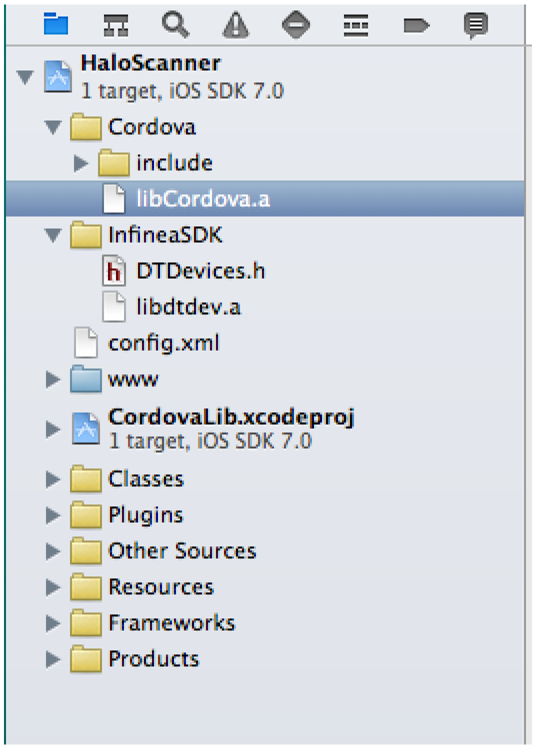
Now add iOS platform library to your project

```
$ cordova build ios
```

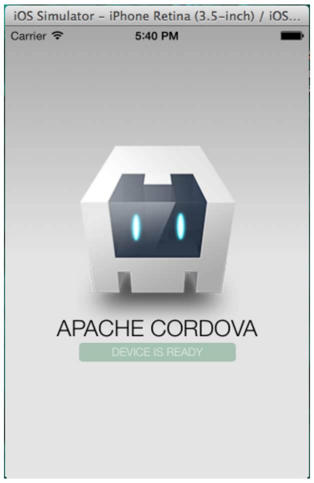
Add cordova iOS .h and .a files to te XCode project, create a directory Cordova and copy new generated *include* folder with all contents.

If we examine our directory *platforms/ios* we should see HaloScanner.xcodepoj file this is new XCOde project created for Hybrid application.
Open the project with XCode and you can start working with it.

Lets add the new files to the project from XCode tool right-click on the project and choose **Add Files to HaloScanner** select Cordova Folder and add to the project



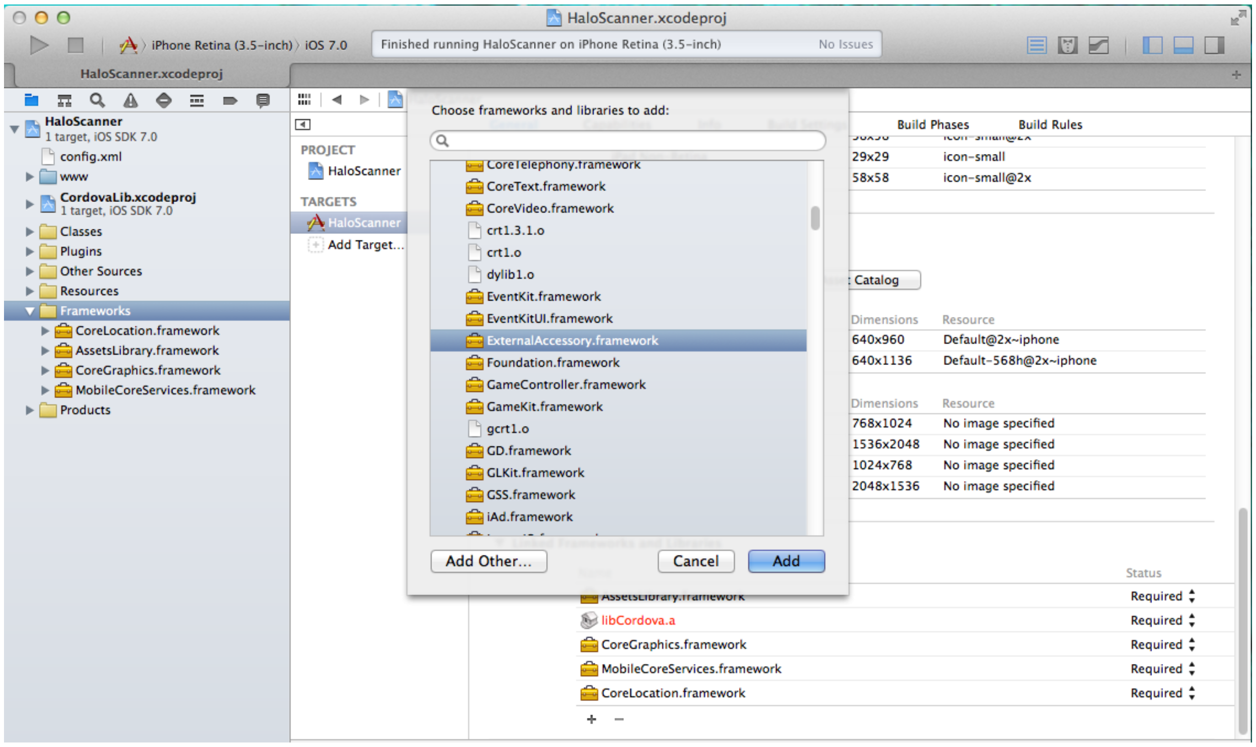
At this point we can build and run the app in simulator to validate everything is working



Now the project is running lets start working with our scanner device plugin.

1. Download Infinate Perifirals Unversal SDK v1.82, this will require registration for Developer Portal and sign the basic NDA to get to download page.
Download SDK file - DTDevices-iOS_2013-10-29_v1.82_Universal_XCode5

2. Add required Framework to project - ExternalAccessory.framework

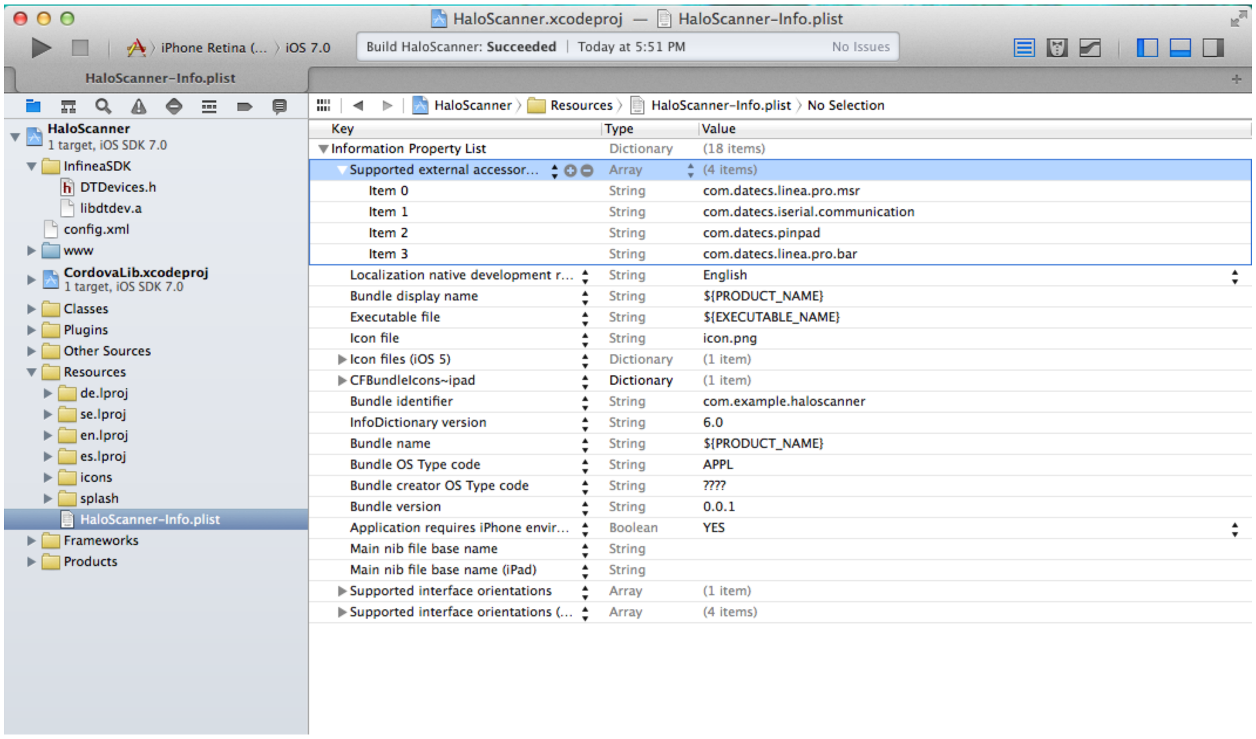


3. Add scanner SDK DTDevices related files to project
   a. Create InfineaSDK Group in project
   b. Copy DTDevice.h and libdtdev.a files into new SDK group

If you find that project has errors then it is effect of a bug. XCode 5 has a bug for adding new files to project described here how to fix that: https://devforums.apple.com/message/898557#898557 remove extra garbage / \ that xcode places in Library search paths.

4. Add Supported External Accessory Products items to project plist property file
Add 4 items

com.datecs.linea.pro.bar
com.datecs.linea.pro.msr
com.datecs.iserial.communication
com.datecs.pinpad


HaloScanner.xcodeproj — HaloScanner-Info.plist

iPhone Retina (... 〉 iOS 7.0 | Build HaloScanner: **Succeeded** | Today at 5:51 PM | No Issues

HaloScanner-Info.plist

```
▼ HaloScanner
    1 target, iOS SDK 7.0
  ▼ InfineaSDK
      DTDevices.h
      libdtdev.a
    config.xml
  ▶ www
  ▶ CordovaLib.xcodeproj
      1 target, iOS SDK 7.0
  ▶ Classes
  ▶ Plugins
  ▶ Other Sources
  ▼ Resources
    ▶ de.lproj
    ▶ se.lproj
    ▶ en.lproj
    ▶ es.lproj
    ▶ icons
    ▶ splash
      HaloScanner-Info.plist
  ▶ Frameworks
  ▶ Products
```

HaloScanner 〉 Resources 〉 HaloScanner-Info.plist 〉 No Selection

Key	Type	Value
▼ Information Property List	Dictionary	(18 items)
▼ Supported external accessor...	Array	(4 items)
Item 0	String	com.datecs.linea.pro.msr
Item 1	String	com.datecs.iserial.communication
Item 2	String	com.datecs.pinpad
Item 3	String	com.datecs.linea.pro.bar
Localization native development r...	String	English
Bundle display name	String	${PRODUCT_NAME}
Executable file	String	${EXECUTABLE_NAME}
Icon file	String	icon.png
▶ Icon files (iOS 5)	Dictionary	(1 item)
▶ CFBundleIcons~ipad	Dictionary	(1 item)
Bundle identifier	String	com.example.haloscanner
InfoDictionary version	String	6.0
Bundle name	String	${PRODUCT_NAME}
Bundle OS Type code	String	APPL
Bundle creator OS Type code	String	????
Bundle version	String	0.0.1
Application requires iPhone envir...	Boolean	YES
Main nib file base name	String	
Main nib file base name (iPad)	String	
▶ Supported interface orientations	Array	(1 item)
▶ Supported interface orientations (...	Array	(4 items)

5. Create basic Plugin *IPCardScanner* class derive from CDVPlugin to manage delegate and device interface. When .m .h files are generated they have incorrect import *<Cordova/Cordova.h>* fix it by replacing that with *import "CDV.h"* for plugin development.

6. Include DTDevice.h file.
Your header file will look like this now.

```
#import "CDV.h"
#import "DTDevices.h"

@interface IPCardScanner : CDVPlugin {
    DTDevices *dtdev;
}

@end
```

7. Create native Plugin interface methods for JScript to handle device notifications to HTML side

```
-(void) initScanner:(CDVInvokedUrlCommand*)command;
-(void) scanBarcode:(NSString*)num;
-(void) scanPaymentCard:(NSString*)num;
-(void) scannerConect:(NSString*)num;
-(void) scannerBattery:(NSString*)num;
```

Now the plugin interface will look like this:

```
#import "CDV.h"
#import "DTDevices.h"

@interface IPCardScanner : CDVPlugin {
   DTDevices *dtdev;
}

// Cordova command method initialize Scanner device
-(void) initScanner:(CDVInvokedUrlCommand*)command;

-(void) scanBarcode:(NSString*)num;
-(void) scanPaymentCard:(NSString*)num;
-(void) scannerConect:(NSString*)num;
-(void) scannerBattery:(NSString*)num;

@end
```

8. Implement DTDevices interfaces for card scan and magnetic swipe in plugin *PICardScanner*. Now we write some Objective-c code to implement interface methods we defined. IN this plugin we used 2 Cordova methods to pass data events back and force for native to web layer.

Command method - Java Script calling native method.
initScanner:(CDVInvokedUrlCommand)command* - this is more of a reply to method call. this method is invoked from JavaScript via Cordova interface and reply is sent back as a command.

```
cordova.exec(onSuccessScan, onErrorScan, "PICardScanner", "initScanner", []);
```

In this case *onSuccessScan* and *onErrorScan* is javascript asynchronous response handler methods. These methods will be called when *initScanner* native methods returns. This initialization method is called by our HTML page to start the scanner device.

Native Method Initiating a call to JavaScript function using WebView object

```
NSString *jsStatement = [NSString stringWithFormat:@"onSuccessScanBarcode('%@');", num];
[self.webView stringByEvaluatingJavaScriptFromString:jsStatement];
[self.viewController dismissViewControllerAnimated:YES completion:nil];
```

This is a way to call JavaScript from Native code.
in our implementation for each native plugin method we will call a corresponding JavaScript function and perform some action on the UI

Scanner Device Notification Methods

Our mobile scanner device Universal SDK handles various attachments from Infinate Peripherals for this example we are using iPad scanner device attachment Infinea Tab 4 that has

dual function to scan barcodes and magnetic strip on credit cards.
We need to implement following methods in our plugin

*-(void)barcodeData:(NSString *)barcode type:(int)type*
*- (void)magneticCardData:(NSString *)track1 track2:(NSString *)track2 track3:(NSString *)track3*
-(void)connectionState:(int)state