WirelessPrinter SDK Reference

一、引言

1.1、编写目的

- 通过调用本开发包的接口函数，可以轻松实现文字、图片、条码等打印元素的排版和打印，无需了解复杂的打印机指令及原理。
- 客户可以使用 SDK 实现打印指令的生成以及打印机通讯。SDK 采用.framework 的形式。使用方法和系统动态框架一样。

  > SDK文件：`PrinterCommand.framework`、`PrinterDispatch.framwrok`

- WirelessPinter 的主要作用是测试SDK 功能、测试打印机的功能、向客户演示功能以及提供针对 MPT-8 机型的文档打印功能。

1.2、预期读者

本文档的预期读者为开发本公司打印机关联的应用程序的用户、开发人员、项目经理、测试人员和文档编写人员，用来对项目涉及的业务知识和系统开发所需解决的问题、满足的需求达成共识。

1.2.1、开发人员要求

使用 WirelessPrinter iOS SDK 进行二次开发，您需要了解：

- 如何使用 framework 动态库（Dynamic Library）或静态库（Static Library）框架
- 稍微了解 CoreBluetooth 框架
- 稍微了解打印机的指令集
- 了解您的打印机支持的指令集
- 了解您的打印机支持的通讯模块（蓝牙或者 WiFi）
- 了解真机调试
- 若需要引入 `PrinterCommandSwift.framework` ,需要具备 `Objective-C` 和 `Swift` 联合编程经验

1.3、文中特殊的定义和缩写

1.3.1、定义

`characteristic`：BLE 蓝牙的特征值

`services`：BLE 蓝牙的服务

`RSSI`：BLE 蓝牙的信号强度

`Dispatcher`：通讯调度员，管理和打印机的 WiFi 和 Bluetooth 4.0 通讯

1.3.2、缩写

`BLE`：蓝牙4.0、低功耗蓝牙，区别于4.0以前的传统蓝牙

`SDK`：指 PrinterCommand.framework，PrinterDispatch.framework

1.4、参考资料

- 《 esc-pos product v10.5.pdf 》
- 《 TSPL_TSPL2_Programming.pdf 》

二、SDK 更新及变化

> 变更类型：（A 增加），（ M 修订），（ D 删除）

2.1、PrinterCommand.framework

> Objective-C 语言框架

版本	时间	变更摘要	变更
v1.1.0	2016.3.8	支持 TSC-TSPL 指令集	A
v1.1.2	2016.3.21	支持 ESC-POS 指令集	M
v1.1.3	2016.3.25	新增支持 PDF 转图片打印。	A
v1.2.0	2016.4.1	新增支持两种图片压缩算法，Beta 版的 Office 文档打印	A

2.2、PrinterDispatch.framework

版本	时间	变更摘要	变更
v1.0.0	2016.2	支持蓝牙	A
v1.0.3	2016.2.29	支持 WiFi	A
v1.1.1	2016.3.10	PTDispatcher 统一管理蓝牙和 WiFi	A

2.2、PrinterCommandSwift.framework

> Swift 语言框架

版本	时间	变更摘要	变更
V1.0.0	2016.4.21	新增 CPCL 指令集	A

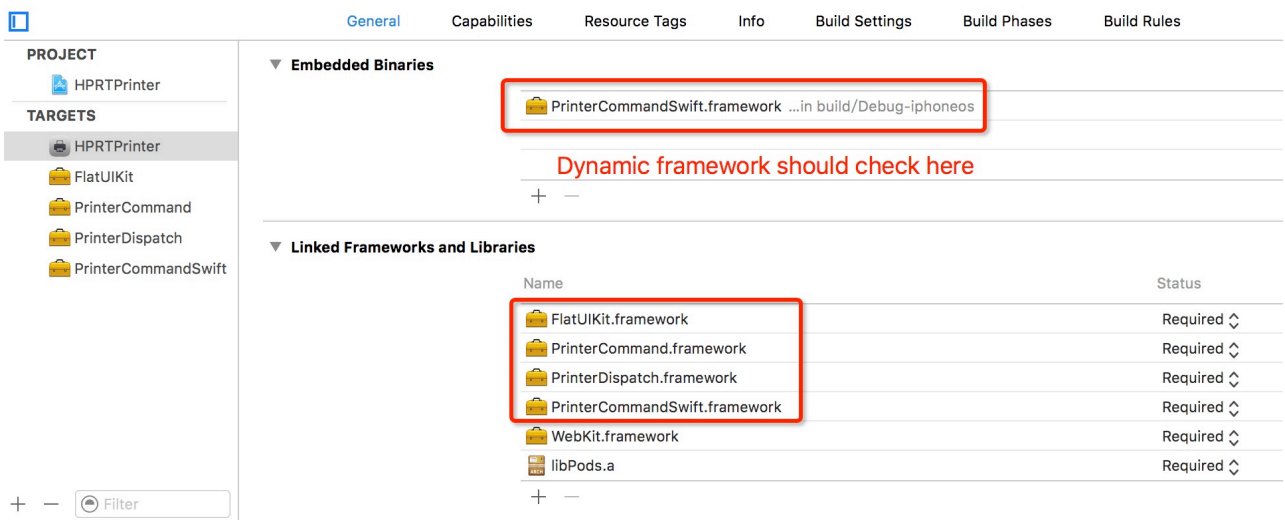
三、开发环境配置

3.1、SDK 版本要求

框架	iOS 系统要求
PrinterCommand.framework	支持 iOS 6.0及以上
PrinterCommandSwift.framework	支持 iOS 8.0 及以上
PrinterDispatch.framework	支持 iOS 6.0及以上
PTPrinter Demo	支持 iOS 8.0 及以上

3.2 、SDK 引入工程

3.2.1 导入 `PrinterCommand.framework`、`PrinterDispatch.framework`



- 解压提供的 WirelessPrinter Demo 工程
- 再解压后的工程目录找到 `PrinterCommand.framework`、`PrinterDispatch.framework`，拖动到您的工程中
- 检查工程是否连接 framework

  - 点击工程
  - 点击 `Targets` 中的应用名字
  - 点击在 `General` 选项卡
  - 找到`Embedded Binaries`一项，点击 '+' 号添加所需要的 framework
  - 检查`Link Binary With Libraries`一项，是否包含所需要的 framework，没有则添加
  - 一般添加`Embedded Binaries`中时，会自动添加到`Link Binary With Libraries`。当编译出现问题时，尝试删除这两项中的 framework 连接，重新添加。

- 在要使用 SDK 的类文件中引入 SDK 框架：

```
#import <PrinterCommand/PrinterCommand.h>
#import <PrinterDispatch/PrinterDispatch.h>
```

- 愉快的使用 SDK

3.2.2 导入 `PrinterCommandSwift.framework`

> PrinterCommandSwift.framework 使用了 Swift 语言编写，当前支持 CPCL 指令集，需要配合 PrinterCommand.framework 进行连接打印机和处理图片等相关操作。

```objc
#import "PTTestCPCLSlim.h"
@import PrinterCommandSwift;
#import <PrinterDispatch/PrinterDispatch.h>

@implementation PTTestCPCLSlim

- (void)testLabelStart {

    PTCommandCPCL *cmd = [[PTCommandCPCL alloc] init];
    [cmd cpclLabelSessionConfigWithOffset:0 horizontalRes:200
        verticalRes:200 height:210 quantity:1];
    [cmd cpclCenter];
    [cmd cpclTextWithRotate:0 fontNameOrNumber:4 fontSize:0 x:10 y:0
        text:@"This is a ! command."];
    [cmd cpclPrint];

    NSData *cmdData = [cmd cpclGetCMDData];
    [[PTDispatcher share] sendData:cmdData];
}
```

```swift
import Foundation
import PrinterCommandSwift
import PrinterDispatch

public class PTTestCPCL: NSObject {

    public func testLabelStart() {

        let cmd = PTCommandCPCL()
        cmd.cpclLabelSessionConfigWithOffset(0, horizontalRes: 200,
            verticalRes: 200, height: 210, quantity: 1)
        cmd.cpclCenter()
        cmd.cpclTextWithRotate(0, fontNameOrNumber: 4, fontSize: 0, x: 10,
            y: 0, text: "This is a ! command.")
        cmd.cpclPrint()

        let cmdData = cmd.cpclGetCMDData()
        PTDispatcher.share().sendData(cmdData)
    }
```

> 左边使用 Swift 调用，右边使用 Objective-C 调用

- 引用框架文件流程同 3.2.1
- 在 Swift 类中引用该框架

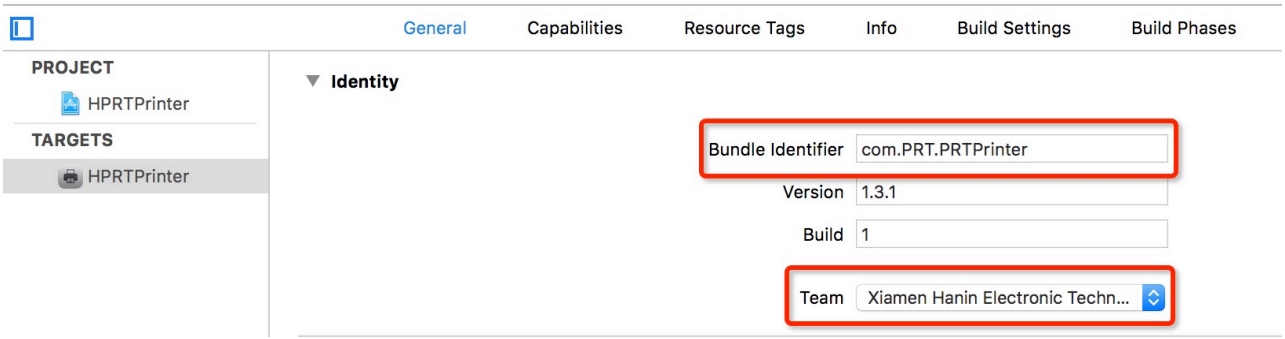
```swift
/** 需要同时引用两个框架 */
import PrinterCommand
import PrinterCommandSwift
```

- 在 Objective-C 类中引用该框架

```objc
/** 需要同时引用两个框架 */
@import PrinterCommandSwift;
#import <PrinterCommand/PrinterCommand.h>
```

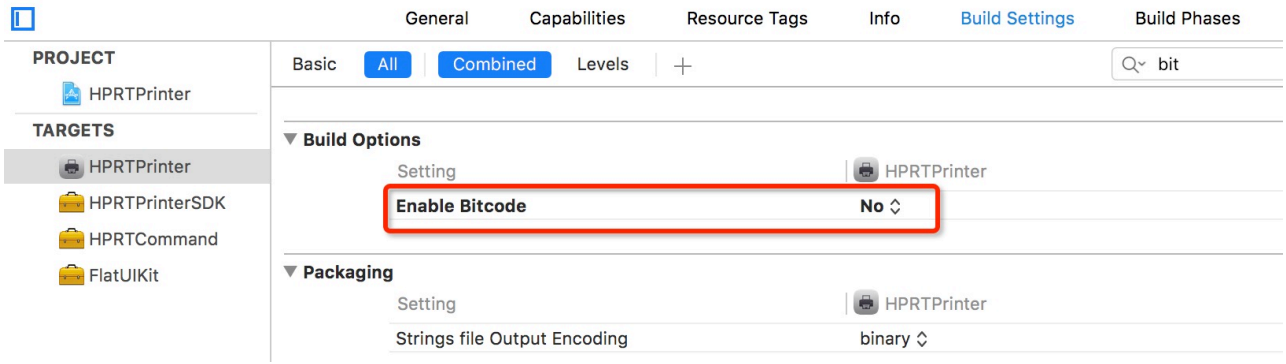
- 愉快的使用 SDK

3.3 真机运行和调试



- 真机运行时，您需要修改 `Bundle Identifier` 和 `Team` 项
- `Bundle Identifier` 是程序包的唯一标识符，可以修改为任意字符串，但不能和 Apple 服务器已有记录的标志符冲突
- `Team` 选择您的 iOS 开发者帐号

3.4 Archive 上架



- 当通过 Archive 打包准备上线 App Store 时，请将您的应用中的`Enable Bitcode`设置为`NO`.

四、WirelessPrinter Demo 介绍

根据不同的指令集不同，大致上提供了一下功能：

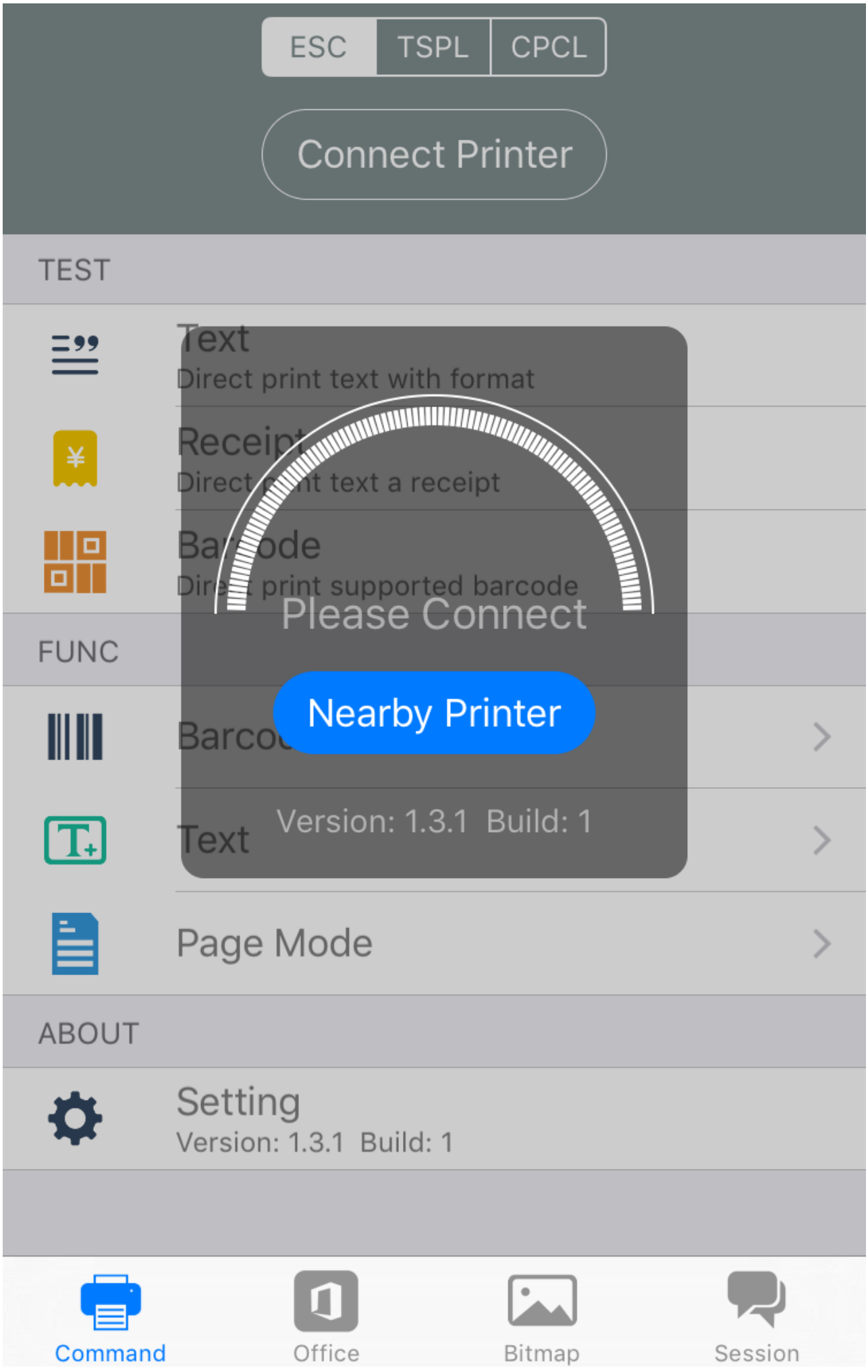
- 打印条形码
- 打印二维码
- 打印文本或者带格式的文本
- 打印图片（黑白图片、半色调图片）
- 打印小票（预定义格式的小票）

通过本 Demo，您可以了解到：

- 如何导入、链接和使用`PrinterCommand.framework`和`PrinterDispatch.framework`两个框架
- 如何通过 Bluetooth 4.0和 便携式蓝牙打印机进行通讯
- 如何通过 WiFi 和便携式 WiFi 打印机进行通讯
- 如何使用打印机指令集中的基础指令，并根据自己的需求分装成为功能

  > - Demo 中在`PTFunctionESC.m`、`PTFunctionTSPL.m`等类中演示了如何将基础指令封装形成功能，以便能够执行打印条形码、二维码、图片、文本等功能。
  > - Demo 中在`PTTesESC.m`、`PTTestTSPL.m`、`PTTestCPCL.swift`等类中展示了测试打印机功能和 SDK 功能的函数。




ESC
TSPL
CPCL
Connect Printer
TEST
Text
Direct print text with format
Receipt
Direct print text a receipt
Barcode
Direct print supported barcode
Please Connect
Nearby Printer
FUNC
Barcode
Text
Version: 1.3.1 Build: 1
Page Mode
ABOUT
Setting
Version: 1.3.1 Build: 1
Command
Office
Bitmap
Session

五、PrinterDispatch.framework

5.1、支持的通讯方式

- `BLE`：Bluetooth 4.0及以上，低功耗蓝牙，不支持传统蓝牙（Bluetooth 4.0 以下）
- `WiFi`：需要在 iOS 系统设置中连接 WiFi 打印机的热点

> 在一些 iOS 机型上，同时打开蓝牙和 WiFi 会导致蓝牙频繁断线，当频繁断线是，请尝试关闭 iOS 系统的WiFi 以保证蓝牙连接的稳定性。

- 了解如何通过 WiFi 或 Bluetooth 4.0 连接打印机

六、PrinterCommand.framework

6.1、支持的打印机指令集

> 您需要了解所使用的打印机是否支持以下任意一种或多种打印指令集

打印指令集	所在框架	介绍
ESC-POS	PTPrinterSDK.framework	主要应用于票据打印机
TSC-TSPL	PTPrinterSDK.framework	主要应用于标签打印机
CPCL	PTCommand.framework	Zebra 兼容打印机

- 了解如何使用 ESC-POS 指令集
- 了解如何使用 TSC-TSPL 指令集

七、支持的打印机

- 打印机至少支持 WiFi 或者BLE 其中一种通讯方式。
- 打印机必须至少支持 ESC 、TSC、CPCL 其中一种指令集。

八、 PrinterCommandSwift.framework

- Command 部分： （Using Swift）

类名	功能
PTCommandCPCL	CPCL 指令集

> 本框架当前只有指令部分，没有通讯部分和图片处理部分
>
> 当您不需要 CPCL 指令集时，不需要引入此框架
>
> 当您需要 CPCL 指令集时，除了需要引入此框架，还需要引入 PTPrinterSDK.Framework

七、API 导航

- 大致分为 `Command`、`Dispatch`、`Function` 三个部分：
- Command：打印机指令集部分（Using Objective-C）

类名	功能
PTCommandESC	ESC-POS 指令集
PTCommandTSPL	TSC-TSPL指令集
PTBitmap	图像处理功能，可以生成二值图像和经过抖动的视觉灰度图像。
PTCommandCPCL	CPCL指令集

- Dispatch：打印机通讯（调度）部分（Using Objective-C、C）

类名	功能
PTDispatcher	封装 PTNetwork和 PTBluetooth，蓝牙和 WiFi 两种通讯 API。提供统一的调用接口，是唯一可以外部调用的通讯接口。