



# **BXL SDK for Android\_UPOS**

## **Compliant API 레퍼런스 가이드**

# **POS 프린터**

---

<b>Rev. 1.11</b>	<b>SPP-R200II</b>
	<b>SPP-R200III</b>
	<b>SPP-R210</b>
	<b>SPP-R220</b>
	<b>SPP-R300</b>
	<b>SPP-R310</b>
	<b>SPP-R400</b>
	<b>SPP-R410</b>
	<b>SRP-350plusIII/352plusIII</b>
	<b>SRP-350III/352III</b>
	<b>SRP-F310II/F312II/F313II</b>
	<b>SRP-380/382</b>
	<b>SRP-330II/332II</b>
	<b>SRP-S300</b>
	<b>SRP-340II/342II</b>
	<b>STP-103III</b>
	<b>SRP-275III</b>
	<b>SRP-Q300/Q302</b>

## ■ 목차

<b>1. 매뉴얼 안내.....</b>	<b>4</b>
<b>2. 사용 환경 .....</b>	<b>5</b>
2-1 운영체제 .....	5
2-2 지원 장치 및 인터페이스.....	5
<b>3. 개발 환경 .....</b>	<b>5</b>
3-1 개발 환경 설정 .....	5
3-2 Android 장치 연결.....	6
3-2-1 Bluetooth .....	6
3-2-2 Network .....	7
3-2-3 Wi-Fi Direct.....	8
3-2-4 USB .....	9
3-2-5 Android 장치 개발자 옵션 설정 .....	12
<b>4. 패키지 콘텐츠 .....</b>	<b>13</b>
4-1 매뉴얼 .....	13
4-2 라이브러리 .....	13
4-3 샘플 소스 코드 .....	13
<b>5. BXL SDK for Android_UPOS compliant 설정 .....</b>	<b>14</b>
5-1 BXLConfigLoader .....	14
5-1-1 메소드 실행 순서 .....	14
5-1-2 생성자 .....	14
5-1-3 addEntry .....	15
5-1-4 getAddress .....	16
5-1-5 getDeviceBus .....	16
5-1-6 getDeviceCategory .....	16
5-1-7 getEntries .....	17
5-1-8 getProductName.....	17
5-1-9 modifyEntry.....	17
5-1-10 newFile .....	17
5-1-11 openFile.....	18
5-1-12 removeEntry .....	18
5-1-13 removeAllEntries .....	18
5-1-14 saveFile .....	18
5-2 jpos.xml .....	19
5-2-1 JposEntries.....	19
5-2-2 JposEntry.....	19
5-2-3 creation.....	20
5-2-4 vendor.....	20
5-2-5 jpos.....	20
5-2-6 product.....	20
5-2-7 prop .....	20

<b>6. 샘플프로그램</b>	<b>21</b>
6-1 Bitmap Test	21
6-2 BXL Test	21
6-3 PDF Test	21
6-4 PhoneGapSample	21
6-4-1 환경설정	21
6-4-2 PhongGap을 사용하는 애플리케이션에서 BXL SDK for Android_UPOS compliant를 사용하기 위한 설정	21
6-5 Text Data Test	21
<b>7. User 애플리케이션</b>	<b>22</b>
7-1 라이브러리 포함	22
7-2 애플리케이션에서 control component 사용	22
7-2-1 Constructor	22
7-2-2 연결 순서	23
7-2-3 Claim	23
7-3 Bitmap 인쇄	23
7-4 PDF 인쇄	24
7-4-1 라이브러리 포함	24
7-4-2 APIs	24
<b>8. UnifiedPOS 지원 명세</b>	<b>25</b>
8-1 Properties	25
8-1-1 POSPrinter 지원 properties	25
8-1-2 CashDrawer 지원 properties	26
8-1-3 MSR 지원 properties	27
8-1-4 SmartCardRW 지원 properties	27
8-2 Methods	28
8-3 Events	28
8-4 Errors	28

## **1. 매뉴얼 안내**

BXL SDK for Android\_UPOS compliant는 UnifiedPOS 버전을 준수합니다.

BXL SDK for Android\_UPOS compliant는(주)빅솔론의 POS 프린터에 대해 JavaPOS에서 제공하는 POSPrinter service를 구현한 것입니다. 이 소프트웨어는 애플리케이션이 (주)빅솔론의 프린터에 쉽게 접근 할 수 있도록 Java class framework을 제공합니다.

이 매뉴얼은 BXL SDK for Android\_UPOS compliant의 사용방법, 사양, 제약에 대해 설명합니다. 또한 이 매뉴얼은 JavaPOS 장치를 사용해서 애플리케이션 시스템을 설계하는 개발자를 위해 제공됩니다.

(주)빅솔론의 프린터를 사용하기 전에 BXL SDK for Android\_UPOS compliant에 포함된 BXLConfigLoader class 또는 jpos.xml 파일을 이용하여 장치를 설정해야 합니다.

[참고사이트]

<http://monroeecs.com/unifiedpos.htm>: UnifiedPOS 위원회

<http://www.bixolon.com>: SDK 업데이트

<http://www.javapos.com>: JavaPOS 공식 사이트

<http://developer.android.com/index.html>: Android 개발자 사이트

저희 (주)빅솔론은 제품의 기능과 품질 향상을 위하여 지속적인 개선을 하고 있습니다. 이로 인하여 제품의 사양과 매뉴얼의 내용은 사전 통보 없이 변경될 수 있습니다.

## 2. 사용 환경

### 2-1 운영체제

이 소프트웨어는 아래 운영체제를 지원합니다.

- Android 4.0 (Ice Cream Sandwich) 이상

### 2-2 지원 장치 및 인터페이스

Models	Interface
SPP-R200II	Bluetooth / WLAN / USB
SPP-R200III	Bluetooth / WLAN / USB
SPP-R210	Bluetooth / WLAN / USB
SPP-R220	Bluetooth / BLE / WLAN / USB
SPP-R300	Bluetooth / WLAN / USB
SPP-R310	Bluetooth / WLAN / USB
SPP-R400	Bluetooth / WLAN / USB
SPP-R410	Bluetooth / BLE / WLAN / USB
SRP-350plusIII	Bluetooth / WLAN / Ethernet / USB
SRP-352plusIII	Bluetooth / WLAN / Ethernet / USB
SRP-350III	Ethernet / USB
SRP-352III	Ethernet / USB
SRP-F310II	Bluetooth / WLAN / Ethernet / USB
SRP-F312II	Bluetooth / WLAN / Ethernet / USB
SRP-F313II	Bluetooth / WLAN / Ethernet / USB
SRP-380	Bluetooth / WLAN / Ethernet / USB
SRP-382	Bluetooth / WLAN / Ethernet / USB
SRP-330II	Ethernet / USB
SRP-332II	Ethernet / USB
SRP-S300	Bluetooth / WLAN / Ethernet / USB
SRP-340II	Ethernet / USB
SRP-342II	Ethernet / USB
STP-103III	USB
SRP-275III	Ethernet / USB
SRP-Q300	Bluetooth / WLAN / Ethernet / USB
SRP-Q302	Bluetooth / WLAN / Ethernet / USB

※BLE : Bluetooth Low Energy

## 3. 개발 환경

### 3-1 개발 환경 설정

- Java Development Kit (JDK) 7
- Eclipse
- Android SDK Tools
- 참고: <http://developer.android.com/sdk/index.html>

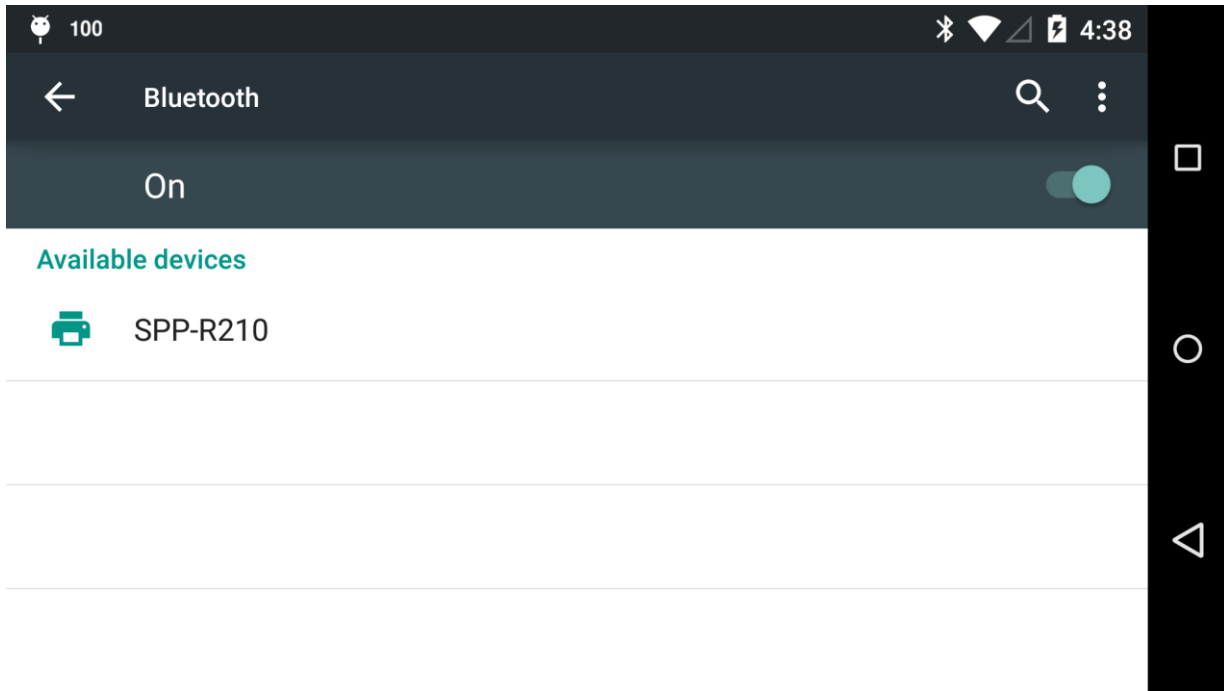
## 3-2 Android 장치 연결

아래 스크린샷은 Nexus 5에서 캡처한것입니다.

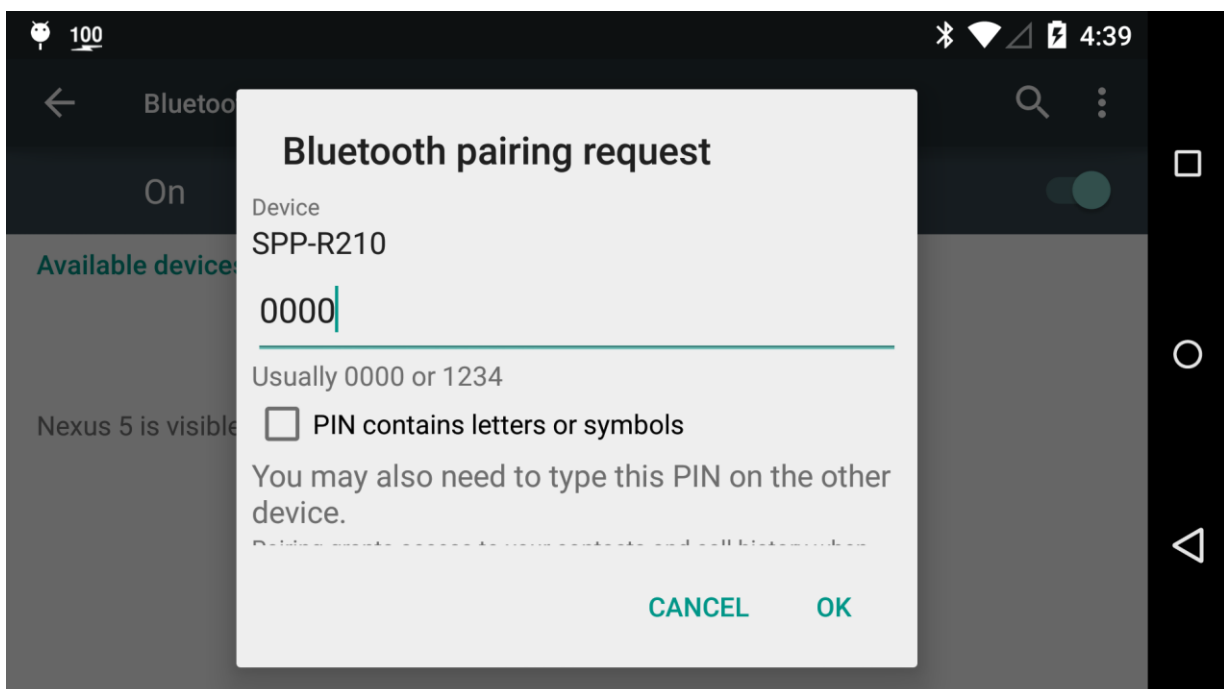
스크린샷과 항목이름은 Android 운영체제 또는 장치에 따라 다를 수 있습니다.

### 3-2-1 Bluetooth

1. Settings를 선택합니다.
2. 반드시 Bluetooth가 켜져 있고, 프린터의 전원이 켜져 있어야 합니다.
3. 설정을 위해 Bluetooth를 선택합니다.

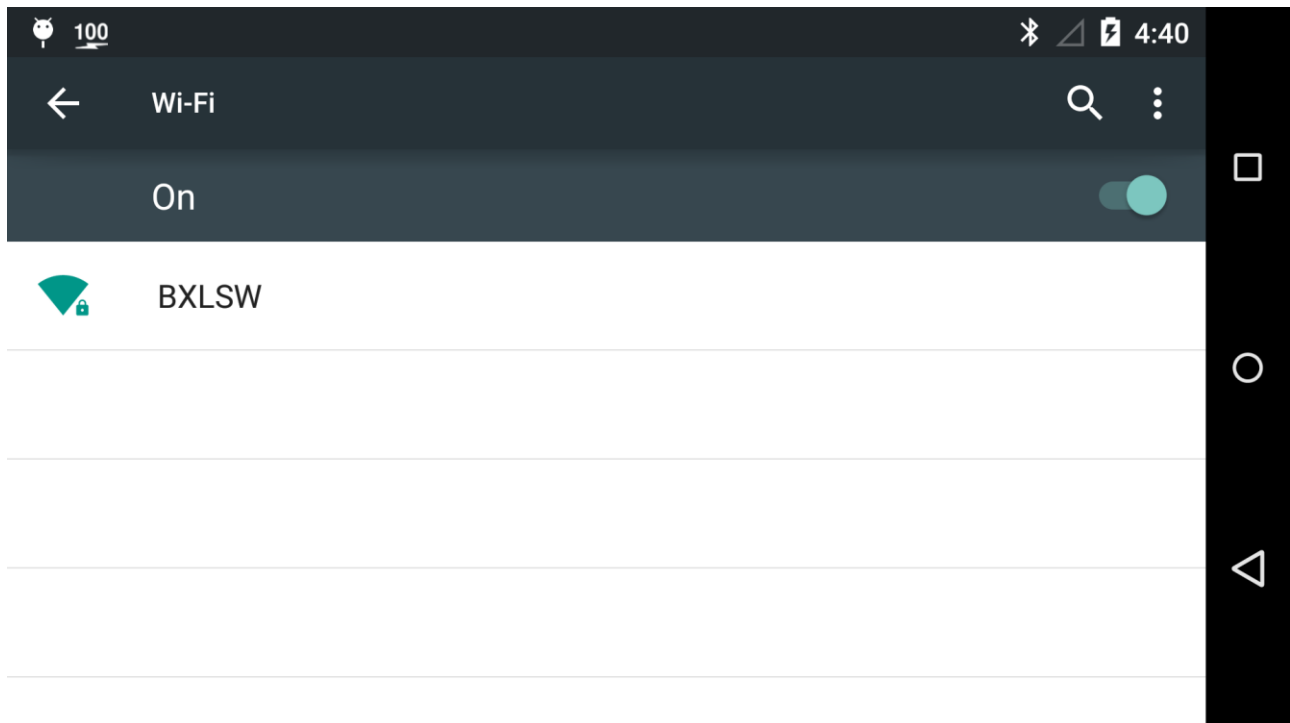


4. Scan을 선택합니다. 연결하기 위한 프린터를 검색하고 페어링합니다.
5. PIN code를 입력합니다. (주)박솔론의 초기 PIN code는 “0000” 입니다.



### 3-2-2 Network

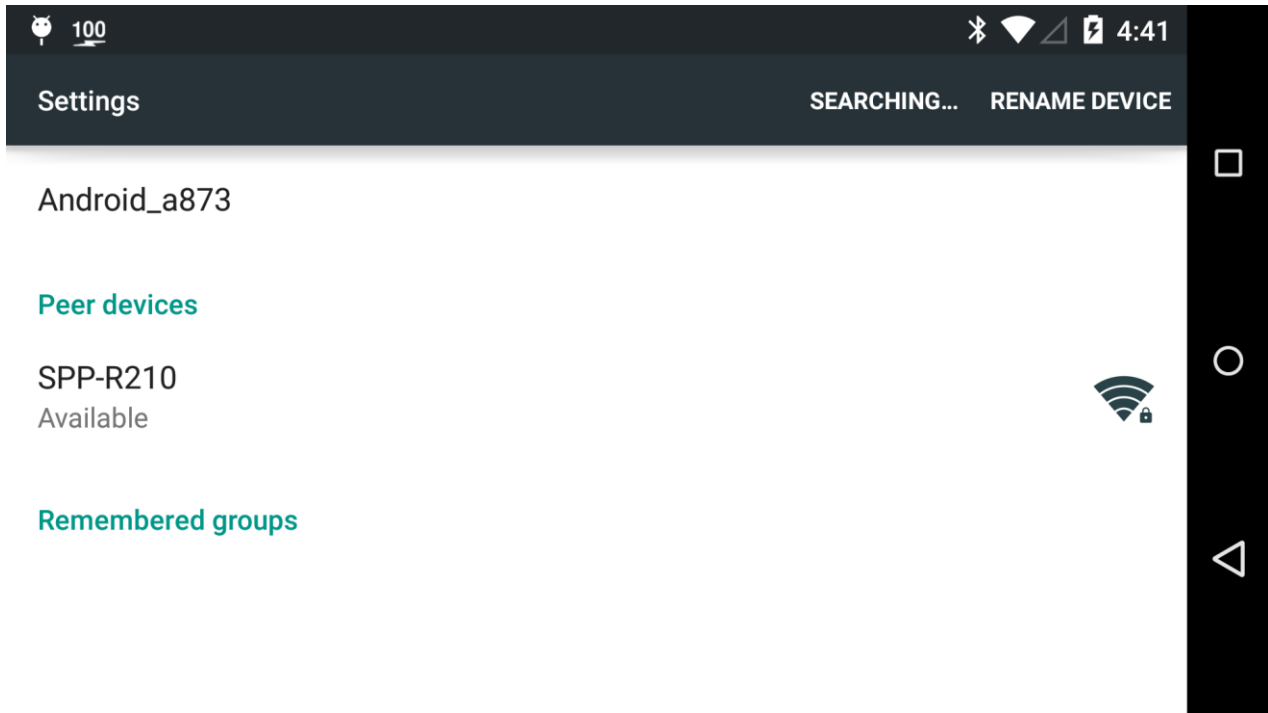
1. 프린터를 네트워크 AP(Access Point)에 연결하고 IP 주소를 할당하거나 DHCP로 설정합니다. (주)빅솔론의 프린터는 초기에 Ad-hoc으로 설정되어 있기 때문에, 마스터 CD에 포함되어있는 Net Configuration Tool로 최초 한번은 설정을 해야 됩니다. Net Configuration Tool은 [\(주\)빅솔론의 웹사이트](#)에서도 다운로드 받으실 수 있습니다.
2. Settings를 선택합니다.
3. Wi-Fi가 켜져 있어야 합니다.
4. (주)빅솔론의 프린터가 연결되어있는 네트워크와 동일한 네트워크에 연결합니다.



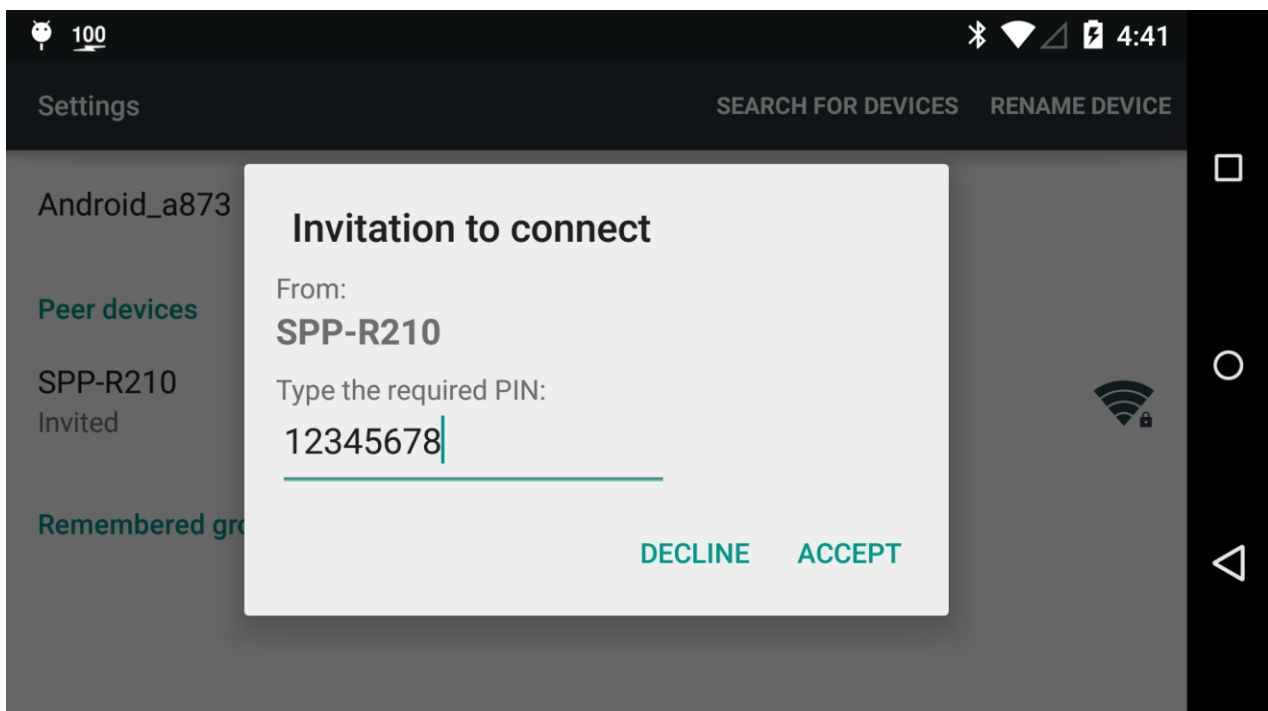
5. 안드로이드장치를 프린터의 TCP/IP 포트에 연결하기 위해 추가설정은 필요하지 않습니다.

### 3-2-3 Wi-Fi Direct

1. 안드로이드장치는 OS 버전 4.0이상부터 Wi-Fi Direct로 주변장치 연결이 가능합니다.
2. 안드로이드장치에 (주)빅솔론의 특정드라이버나 프린터 소프트웨어가 설치되어있을 필요는 없습니다.
3. Settings를 선택합니다.
4. Wi-Fi가 켜져 있어야 합니다.
5. Wi-Fi Direct를 선택합니다.



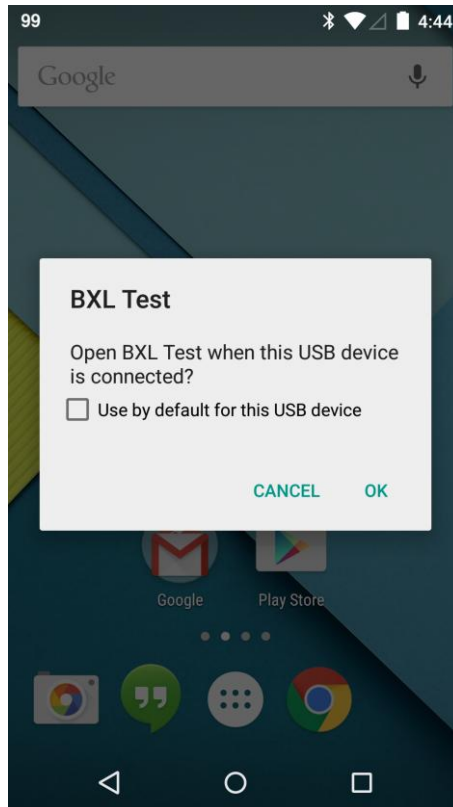
6. 검색된 프린터를 선택해서 연결합니다.  
Wi-Fi Direct 초기 PIN 번호는 “12345678” 입니다.





### 3-2-4 USB

1. 안드로이드장치는 OS 버전 3.1 이상부터 USB 주변장치 연결이 가능합니다.
2. 안드로이드장치에 (주)빅솔론의 특정드라이버나 프린터소프트웨어가 설치되어있을 필요는 없습니다.
3. 필요한 USB 케이블은 스마트폰이나 태블릿에 따라 달라질 수 있습니다.  
대부분의 안드로이드장치는 A to B USB 케이블을 제공하지 않습니다. Mini/micro USB 케이블이나 아답터/dock이 필요할 수 있습니다.  
사용하려는 안드로이드장치에 알맞은 케이블을 사용하는지 확인해야 합니다.
4. (주)빅솔론의 프린터를 처음 연결하는 경우, 안드로이드장치에 따라서 아래와 같은 메시지가 나타날 수 있습니다.



5. USB 주변장치를 연결하기 위해서는 아래 코드를 AndroidManifest.xml과 res/xml/device\_filter.xml에 입력해야 합니다.

[AndroidManifest.xml]

```
...
<uses-feature android:name="android.hardware.usb.host" />
...

<intent-filter>
    <action android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" />
</intent-filter>

<meta-data
    android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED"
    android:resource="@xml/device_filter" />
```

[device\_filter.xml]

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <!-- SPP-R200II -->
    <usb-device product-id="40" vendor-id="5380" />

    <!-- SPP-R200III -->
    <usb-device product-id="91" vendor-id="5380" />

    <!-- SPP-R210 -->
    <usb-device product-id="81" vendor-id="5380" />

    <!-- SPP-R220 -->
    <usb-device product-id="106" vendor-id="5380" />

    <!-- SPP-R300 -->
    <usb-device product-id="33" vendor-id="5380" />

    <!-- SPP-R310 -->
    <usb-device product-id="92" vendor-id="5380" />

    <!-- SPP-R400 -->
    <usb-device product-id="41" vendor-id="5380" />

    <!-- SPP-R410 -->
    <usb-device product-id="75" vendor-id="5380" />

    <!-- SRP-350plusIII -->
    <usb-device product-id="61" vendor-id="5380" />

    <!-- SRP-352plusIII -->
    <usb-device product-id="62" vendor-id="5380" />

    <!-- SRP-350III -->
    <usb-device product-id="43" vendor-id="5380" />

    <!-- SRP-352III -->
    <usb-device product-id="60" vendor-id="5380" />

    <!-- SRP-F310II -->
    <usb-device product-id="87" vendor-id="5380" />

    <!-- SRP-F312II -->
    <usb-device product-id="88" vendor-id="5380" />

    <!-- SRP-F313II -->
    <usb-device product-id="90" vendor-id="5380" />

    <!-- SRP-380 -->
    <usb-device product-id="96" vendor-id="5380" />
```

```
<!-- SRP-382 -->
<usb-device product-id="97" vendor-id="5380" />

<!-- SRP-330II -->
<usb-device product-id="110" vendor-id="5380" />
<!-- SRP-332II-->
<usb-device product-id="111" vendor-id="5380" />

<!-- SRP-S300-->
<usb-device product-id="82" vendor-id="5380" />

<!-- SRP-340II -->
<usb-device product-id="114" vendor-id="5380" />

<!-- SRP-342II -->
<usb-device product-id="115" vendor-id="5380" />

<!-- STP-103III -->
<usb-device product-id="83" vendor-id="5380" />

<!-- SRP-275III -->
<usb-device product-id="89" vendor-id="5380" />

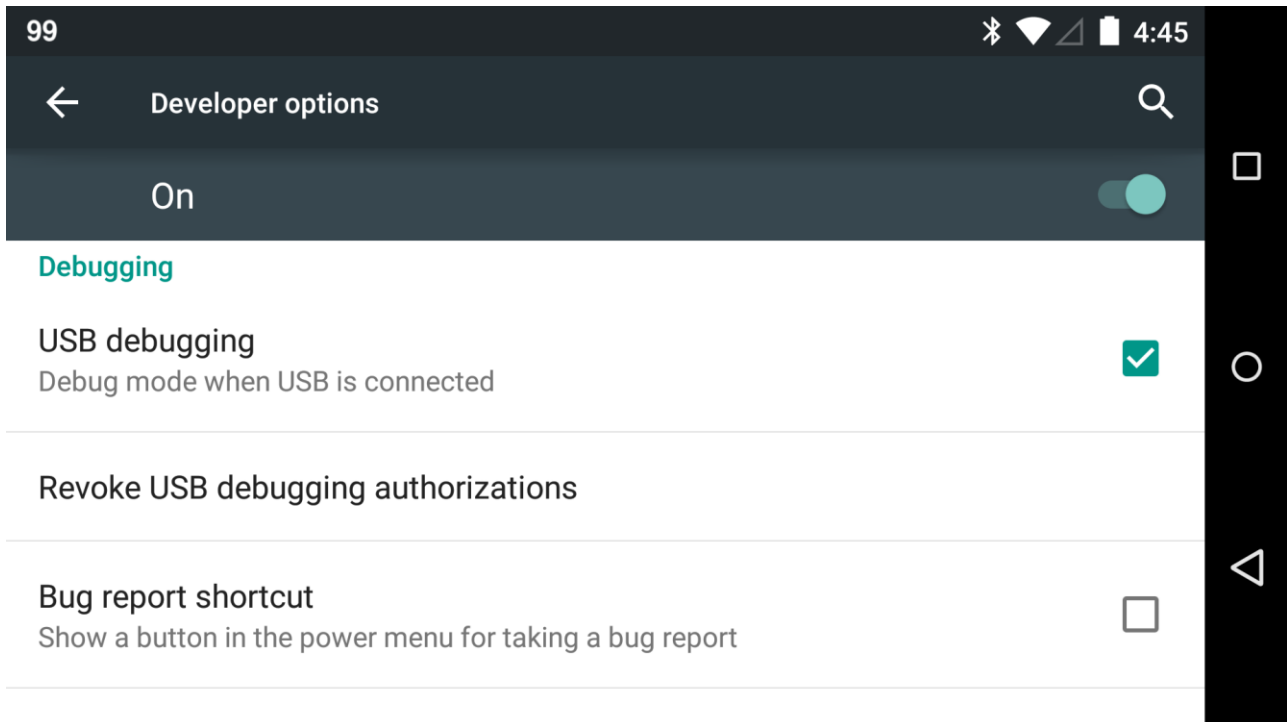
<!-- SRP-Q300 -->
<usb-device product-id="86" vendor-id="5380" />

<!-- SRP-Q302 -->
<usb-device product-id="89" vendor-id="5380" />
```

```
</resources>
```

### 3-2-5 Android 장치 개발자 옵션 설정

1. Settings를 선택합니다.
2. Developer options를 선택합니다.
3. USB debugging을 활성화합니다.



## 4. 패키지 콘텐츠

### 4-1 매뉴얼

- docs/Manual\_BXL SDK for Android\_UPOS compliant API Reference Guide\_english\_Rev\_x\_xx: 영문 매뉴얼
- docs/Manual\_BXL SDK for Android\_UPOS compliant API Reference Guide\_japan\_Rev\_x\_xx: 일본어 매뉴얼
- docs/Manual\_BXL SDK for Android\_UPOS compliant API Reference Guide\_korean\_Rev\_x\_xx: 한글 매뉴얼

### 4-2 라이브러리

- libs/bixolon\_printer.jar: JavaPOS 서비스 컴포넌트 계층의 구현 / 프린터 설정 라이브러리
- libs/jpos114-controls.jar: JavaPOS 장치 컨트롤과 서비스 인터페이스
- libs/xerces.jar: apache.org XML 서비스구현 (JavaPOS device control에서 필요)
- libs/PDF/bixolon\_pdf.jar: PDF 인쇄를 위한 라이브러리
- libs/PDF/armeabi-v7a/libbxmlpdf.so: PDF 인쇄를 위한 native 라이브러리

### 4-3 샘플 소스 코드

- samples/android-support-v7-appcompat: v7 Support library 프로젝트 (BXLTest에서 참조)
- samples/BitmapTest: 비트맵 인쇄 샘플 애플리케이션
- samples/BXLTest: 프린터 설정, 프린터 연결, 텍스트, 바코드, 비트맵 인쇄, 페이지 모드 설정 샘플 애플리케이션
- samples/PDFTTest: PDF 인쇄 샘플 애플리케이션
- samples/PhoneGap Plugin: PhoneGap용 plugin
- samples/PhoneGapSample: PhoneGap을 사용한 Web App 샘플
- samples/TextDataTest: 텍스트 인쇄 옵션 설정 샘플 애플리케이션

## **5. BXL SDK for Android\_UPOS compliant 설정**

BIXOLON UPOS Printer SDK for Android는 장치 설정을 제공하기 위해 BXLConfigLoader class 또는 jpos.xml 파일을 사용합니다.

### **5-1 BXLConfigLoader**

jpos.xml 파일을 직접 수정하지 않고 라이브러리를 통해서 장치 설정을 저장합니다.  
라이브러리 내부에서 jpos.xml 파일을 관리합니다.  
BXLTest 샘플 소스 코드를 참고하시기 바랍니다.

#### **5-1-1 메소드 실행 순서**

1. openFile (저장된 jpos.xml이 없어서 exception이 발생하는 경우 newFile)
2. addEntry, modifyEntry 또는 removeEntry
3. saveFile

#### **5-1-2 생성자**

```
public BXLConfigLoader(Context context)
```

- Parameters

*Context:* POSPrinterService에 전달하기 위한 context 객체.

## 5-1-3 addEntry

public void addEntry(String logicalName, int deviceCategory, String productName,  
int deviceBus, String address) throws IllegalArgumentException

장치를 추가합니다.

## - Parameters

*logicalName*: 저장할 장치의 logical name

*deviceCategory*: 저장할 장치의 device category.

상수	값
DEVICE_CATEGORY_CASH_DRAWER	0
DEVICE_CATEGORY_MSR	1
DEVICE_CATEGORY_POS_PRINTER	2
DEVICE_CATEGORY_SMART_CARD_RW	3

*productName*: 저장할 장치의 product name. deviceCategory가  
DEVICE\_CATEGORY\_POS\_PRINTER인 경우 실제 모델명과  
동일한 값으로 설정해야 됨.

*deviceBus*: 저장할 장치의 device bus

상수	값
DEVICE_BUS_BLUETOOTH	0
DEVICE_BUS_ETHERNET	1
DEVICE_BUS_USB	2
DEVICE_BUS_WIFI	3
DEVICE_BUS_WIFI_DIRECT	4

*address*: 저장할 프린터의 주소

Device bus	address
DEVICE_BUS_BLUETOOTH	블루투스 MAC 주소
DEVICE_BUS_ETHERNET	IP 주소
DEVICE_BUS_USB	없음
DEVICE_BUS_WIFI	IP 주소
DEVICE_BUS_WIFI_DIRECT	WLAN MAC 주소

## - Throws

*IllegalArgumentException*: deviceCategory, deviceBus의 값이 지정된 상수가  
아니거나 productName이 null인 경우

#### 5-1-4 getAddress

```
public String getProductName(String logicalName)
```

저장된 장치들에서 `logicalName`을 검색해 해당 장치의 `address`를 반환합니다.

- Parameters  
*logicalName*: Address를 반환할 장치의 logical name
- Returns  
`logicalName`과 동일한 이름의 장치가 저장되어 있는 경우 해당 장치의 `address`를 반환, 그렇지 않으면 `null`을 반환

#### 5-1-5 getDeviceBus

```
public int getDeviceBus(String logicalName)
```

저장된 장치들에서 `logicalName`을 검색해 해당 장치의 `device bus`를 반환합니다.

- Parameters  
*logicalName*: Device bus를 반환할 장치의 logical name
- Returns  
`logicalName`과 동일한 이름의 장치가 저장되어 있는 경우 해당 장치의 `device bus`를 반환, 그렇지 않으면 `-1`을 반환

상수	값
DEVICE_BUS_BLUETOOTH	0
DEVICE_BUS_ETHERNET	1
DEVICE_BUS_USB	2
DEVICE_BUS_WIFI	3
DEVICE_BUS_WIFI_DIRECT	4

#### 5-1-6 getDeviceCategory

```
public int getDeviceCategory(String logicalName)
```

저장된 장치들에서 `logicalName`을 검색해 해당 장치의 `device category`를 반환합니다.

- Parameters  
*logicalName*: Device category를 반환할 장치의 logical name
- Returns  
`logicalName`과 동일한 이름의 장치가 저장되어 있는 경우 해당 장치의 `device category`를 반환, 그렇지 않으면 `-1`을 반환

상수	값
DEVICE_CATEGORY_CASH_DRAWER	0
DEVICE_CATEGORY_MSIR	1
DEVICE_CATEGORY_POS_PRINTER	2
DEVICE_CATEGORY_SMART_CARD_RW	3



### 5-1-7 getEntries

public List<?> getEntries() throws Exception

jpos.xml 파일에 저장된 장치 정보를 가져옵니다.

- Returns  
저장된 장치의 목록. JposEntry를 element로 갖는 List.

### 5-1-8 getProductName

public String getProductName(String logicalName)

저장된 장치들에서 logicalName을 검색해 해당 장치의 product name을 반환합니다.

- Parameters  
*logicalName*: Product name를 반환할 장치의 logical name
- Returns  
logicalName과 동일한 이름의 장치가 저장되어 있는 경우 해당 장치의 product name을 반환, 그렇지 않으면 null을 반환

### 5-1-9 modifyEntry

public boolean modifyEntry(String logicalName, int deviceBus, String address)

저장된 장치의 device bus 또는 address를 수정합니다.

- Parameters  
*logicalName*: 수정할 장치의 logical name

*deviceBus*: 수정할 deviceBus

상수	값
DEVICE_BUS_BLUETOOTH	0
DEVICE_BUS_ETHERNET	1
DEVICE_BUS_USB	2
DEVICE_BUS_WIFI	3
DEVICE_BUS_WIFI_DIRECT	4

*address*: 수정할 address

- Returns  
logicalName과 동일한 이름의 장치가 저장되어 있는 경우 해당 장치의 device bus와 address를 설정한 값으로 수정하고 true를 반환, 그렇지 않으면 false를 반환

### 5-1-10 newFile

public void newFile()

저장된 jpos.xml 파일이 없는 경우 새로 생성합니다.

### **5-1-11 openFile**

public void openFile() throws Exception

저장된 장치의 정보를 읽거나 수정하기 위해 jpos.xml을 open합니다.

- Throws  
*Exception:* 저장된 jpos.xml 파일이 없는 경우

### **5-1-12 removeEntry**

public boolean removeEntry(String logicalName)

저장된 장치를 삭제합니다.

- Parameters  
*logicalName:* 삭제할 장치의 logical name
- Returns  
logicalName과 동일한 이름의 장치가 저장되어 있는 경우 해당 장치를 삭제하고 true를 반환, 그렇지 않으면 false를 반환

### **5-1-13 removeAllEntries**

public boolean removeAllEntries()

저장된 장치를 모두 삭제합니다.

### **5-1-14 saveFile**

public void saveFile() throws Exception

설정된 장치의 정보를 jpos.xml 파일에 저장합니다.

- Throws  
*Exception:* 파일을 저장하는 동안 에러가 발생하는 경우

## 5-2 jpos.xml

jpos.xml 파일을 직접 수정해서 프린터 설정을 저장합니다. jpos.xml 파일은 아래와 같은 형태로 저장됩니다.

PDFTest 또는 PhoneGapSample 샘플 소스 코드를 참고하시기 바랍니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE JposEntries PUBLIC "-//JavaPOS//DTD//EN"
                        "jpos/res/jcl.dtd">
<JposEntries>

    <JposEntry logicalName="SPP-R210" >
        <creation
            factoryClass="com.bxl.loader.ServiceInstanceFactory"
            serviceClass="com.bxl.services.posprinter.POSPrinterService" />

        <vendor
            name="BIXOLON"
            url="http://www.bixolon.com" />

        <jpos
            category="POSPrinter"
            version="1.14" />

        <product
            name="SPP-R210"
            description="Slim yet Rugged 2 inch Thermal Mobile Portable Printer"
            url="http://www.bixolon.com" />

        <prop
            name="deviceBus"
            type="String"
            value="Bluetooth" />
        <prop
            name="address"
            type="String"
            value="74:F0:7D:E2:50:36" />
    </JposEntry>
</JposEntries>
```

### 5-2-1 JposEntries

최상위 element 입니다.

### 5-2-2 JposEntry

하나의 장치에 해당하는 element 입니다.

장치가 추가되는 경우 JposEntry element를 추가하면 됩니다.

**5-2-3 creation**

(주)빅솔론의 `service`를 설정하기 위한 `element`입니다. 아래 `attribute`를 고정 값으로 사용해야 됩니다.

```
factoryClass="com.bxl.loader.ServiceInstanceFactory"
serviceClass="com.bxl.services.posprinter.POSPrinterService"
```

**5-2-4 vendor**

(주)빅솔론의 `vendor` 정보에 해당하는 `element`입니다. 아래 `attribute`를 고정 값으로 사용해야 됩니다.

```
name="BIXOLON"
url="http://www.bixelon.com"
```

**5-2-5 jpos**

장치의 `jpos` 정보를 저장하기 위한 `element` 입니다. `category attribute`는 아래 값 중 한 가지로 설정하고 `version attribute`는 고정 값으로 사용해야 됩니다.

```
category="CashDrawer" | "MSR" | "POSPrinter" | "SmartCardRW"
version="1.14"
```

**5-2-6 product**

장치의 상세 정보를 저장하기 위한 `element` 입니다. `name attribute`는 실제 장치의 모델명과 동일하게 설정하고 다른 `attribute`는 임의로 설정이 가능합니다.

```
name="SPP-R210"
description="Slim yet Rugged 2 inch Thermal Mobile Portable Printer"
url="http://www.bixelon.com" />
```

**5-2-7 prop**

장치의 연결 정보를 저장하기 위한 `element` 입니다. `Name attribute`를 `deviceBus`로 갖고 있는 `element`와 `address`로 갖고 있는 `element`가 반드시 설정되어야 합니다.

```
name="deviceBus"
type="String"
value="Bluetooth" | "Ethernet" | "USB" | "Wi-Fi" | "Wi-Fi Direct"
```

```
name="address"
type="String"
value="..."
```

## 6. 샘플프로그램

### 6-1 Bitmap Test

BXL SDK for Android\_UPOS compliant를 사용해서 안드로이드 애플리케이션을 개발하기 위한 레퍼런스입니다. 안드로이드 장치 저장공간에 저장된 비트맵과 애플리케이션 리소스에 포함된 비트맵을 인쇄할 수 있습니다.

### 6-2 BXL Test

BXL SDK for Android\_UPOS compliant를 사용해서 안드로이드 애플리케이션을 개발하기 위한 레퍼런스입니다. 장치 연결, 텍스트, 바코드, 비트맵 인쇄, Cash drawer, MSR, SCR을 포함합니다.

### 6-3 PDF Test

BXL SDK for Android\_UPOS compliant를 사용해서 PDF 파일을 인쇄하는 애플리케이션을 개발하기 위한 레퍼런스입니다.

### 6-4 PhoneGapSample

BXL SDK for Android\_UPOS compliant를 사용해서 안드로이드 웹 애플리케이션을 개발하기 위한 레퍼런스입니다.

PhoneGap 버전 3.1.0-0.15.0를 사용해서 빌드되었습니다.

#### 6-4-1 환경설정

- Node.js 설치 (<http://nodejs.org> 참고)
- PhoneGap 설치 (<http://docs.phonegap.com/en/edge/index.html> 참고)
- 필요한 플러그인 추가 및 빌드  
([http://docs.phonegap.com/en/edge/guide\\_cli\\_index.md.html#The%20Command-Line%20Interface](http://docs.phonegap.com/en/edge/guide_cli_index.md.html#The%20Command-Line%20Interface) 참고)

#### 6-4-2 PhoneGap을 사용하는 애플리케이션에서 BXL SDK for Android\_UPOS compliant를 사용하기 위한 설정

- 애플리케이션에 PhoneGap Plugin을 추가합니다.
- PhoneGap 애플리케이션의 res/xml/config.xml에 아래코드를 추가합니다.

```
<feature name="BXLService">
    <param name="android-package"
        value="com.bxl.service.phonegap.BXLService " />
</feature>
```

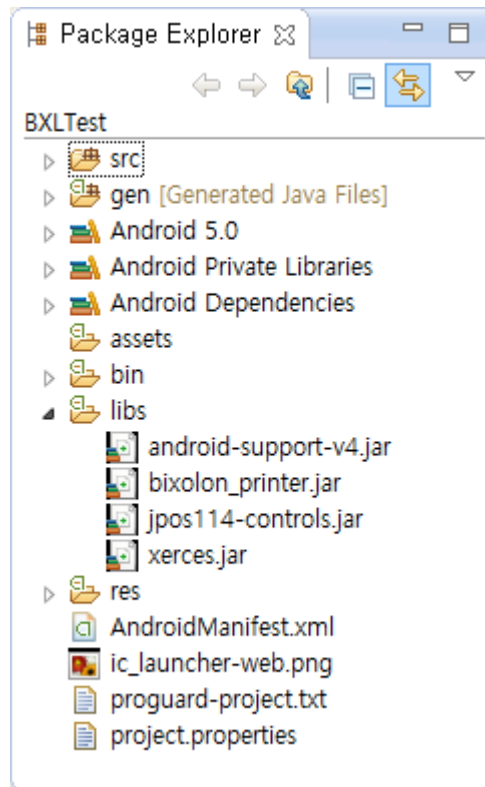
### 6-5 Text Data Test

BXL SDK for Android\_UPOS compliant를 사용해서 안드로이드 애플리케이션을 개발하기 위한 레퍼런스입니다. Bold, underline 등 텍스트 인쇄 시 사용되는 옵션을 설정할 수 있습니다.

## 7. User 애플리케이션

### 7-1 라이브러리 포함

애플리케이션 프로젝트에 bixon\_printer.jar, jpos114-controls.jar, xerces.jar가 모두 포함되어야 합니다.



### 7-2 애플리케이션에서 control component 사용

#### 7-2-1 Constructor

POSPrinter 객체 생성 시 android.content.Context 객체를 인자로 전달합니다.  
CashDrawer, MSR, SmartCardRW는 default constructor를 사용합니다.

```
public class MainActivity extends Activity {

    ...

    POSPrinter posPrinter = new POSPrinter(this);
    CashDrawer cashDrawer = new CashDrawer();
    MSR msr = new MSR();
    SmartCardRW smartCardRW = new SmartCardRW();
    ...

}
```

**7-2-2 연결 순서**

CashDrawer, MSR, POSPrinter, SmartCardRW 객체를 사용해서 각 장치를 연결하기 위해서는 아래 순서대로 메소드를 실행해야 됩니다.

```
posPrinter.open(logicalName);
posPrinter.claim(timeout);
posPrinter.setDeviceEnabled(true);
```

**7-2-3 Claim**

CashDrawer, MSR, SmartCardRW는 물리적으로 프린터에 내장되어 있기 때문에 POSPrinter와 연결(connection)을 공유합니다. POSPrinter의 `getClaimed()`가 `true`일 경우에만 CashDrawer, MSR, SmartCardRW의 `claim()`이 가능합니다.

```
If (posPrinter.getClaimed()) {
    cashDrawer.claim();
    msr.claim();
    smartCardRW.claim();
}
```

**7-3 Bitmap 인쇄**

`void jpos.POSPrinter.printBitmap(int station, String fileName, int width, int alignment)`  
throws JposException

지원 가능 비트맵 이미지 포맷: JPEG, GIF, PNG, BMP

Bitmap을 인쇄하는 경우 아래와 같은 방법으로 `brightness`를 설정할 수 있습니다.

`Brightness`의 값은 0부터 100까지 가능합니다. 유효한 값의 범위는 13부터 88 입니다.

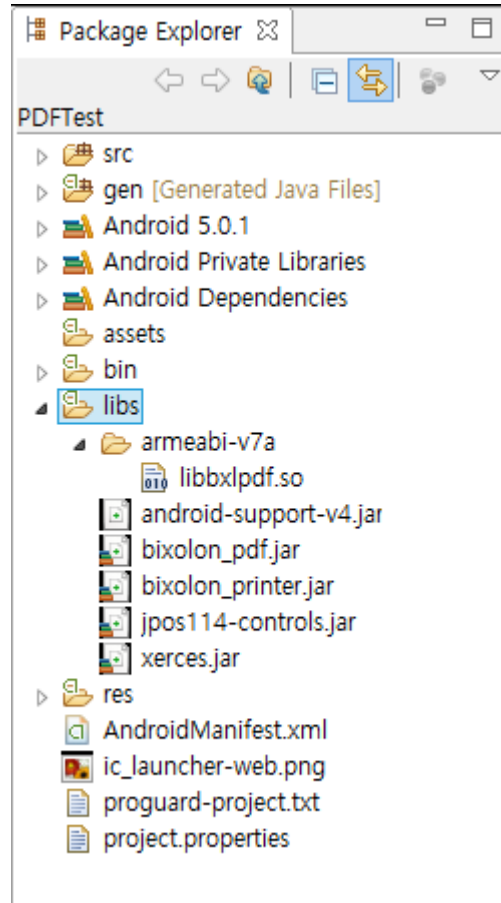
```
ByteBuffer buffer = ByteBuffer.allocate(4);
buffer.put((byte) POSPrinterConst.PTR_S_RECEIPT);
buffer.put((byte) brightness); // brightness
buffer.put((byte) 0x00);       // dummy
buffer.put((byte) 0x00);       // dummy

posPrinter.printBitmap(buffer.getInt(0), fileName, width, alignment);
```

## 7-4 PDF 인쇄

### 7-4-1 라이브러리 포함

Software\_BXL SDK for Android\_UPOS compliant\_Vx.x.x/libs/PDF 경로에 있는 armeabi-v7a/libbxmlpdf.so와 bixolon\_pdf.jar 라이브러리가 추가로 포함되어야 합니다.



### 7-4-2 APIs

PDF 파일을 인쇄하는 경우 아래 API를 사용해야 됩니다.

- Syntax  
void jpos.POSPrinter.printPDF(int station, String fileName, int width, int alignment, int page, int brightness) throws JposException

Parameter	Description
Station	PTR_S_RECEIPT
fileName	인쇄할 PDF 파일의 절대 경로
Width	인쇄할 페이지의 너비
Alignment	인쇄할 페이지의 배치
Page	인쇄할 페이지
Brightness	인쇄할 페이지의 밝기 (13 ~ 88)



## 8. UnifiedPOS 지원 명세

### 8-1 Properties

#### 8-1-1 POSPrinter 지원 properties

\* 공통 property value 목록

Properties	Initial value
CheckHealthText	“”
Claimed	False
DeviceEnabled	False
OutputID	0
PowerState	UNKNOWN
State	CLOSED
CapRecNearEndSensor	프린터 모델에 의존
CapRecPapercut	프린터 모델에 의존
CapRecMarkFeed	프린터 모델에 의존
AsyncMode	False
CharacterSet	437
CharacterSetList	"437, 737, 775, 850, 852, 855, 857, 858, 860, 862, 863, 864, 865, 866, 928, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 7065(Farsi), 7565(Katakana), 7572(Khmer), 8411(Thai11), 8414(Thai14), 8416(Thai16), 8418(Thai18), 8442(Thai42)"
CoverOpen	False
PageModeArea	프린터 모델에 의존
PageModeHorizontalPosition	0
PageModePrintArea	“0,0,0,0”
PageModeVerticalPosition	0
RecLineChars	프린터 모델에 의존
RecLineCharsList	프린터 모델에 의존
RecLineWidth	프린터 모델에 의존
RecEmpty	False

\* 프린터 모델별 property value 목록

Model	PageModeArea	RecLine Chars	RecLine CharsList	RecLine Width
SPP-R200II	"384,2400"	32	"32,42"	384
SPP-R200III	"384,2400"	32	"32,42"	384
SPP-R210	"384,2400"	32	"32,42"	384
SPP-R220	"384,2400"	32	"32,42"	384
SPP-R300	"576,2400"	48	"48,64"	576
SPP-R310	"576,2400"	48	"48,64"	576
SPP-R400	"832,2400"	69	"69,92"	832
SPP-R410	"832,2400"	69	"69,92"	832
SRP-350plusIII	"512,1662"	42	"42,56"	512
SRP-352plusIII	"576,1662"	48	"48,64"	576
SRP-350III	"512,1662"	42	"42,56"	512
SRP-352III	"576,1662"	48	"48,64"	576
SRP-F310II	"512,1662"	42	"42,56"	512
SRP-F312II	"576,1662"	48	"48,64"	576
SRP-F313II	"640,1662"	50	"50,67"	640
SRP-380	"512,1662"	42	"42,56"	512
SRP-382	"576,1662"	48	"48,64"	576
SRP-330II	"512,1662"	42	"42,56"	512
SRP-332II	"576,1662"	48	"48,64"	576
SRP-S300	"576,1662"	48	"48,64"	576
SRP-340II	"512,1662"	42	"42,56"	512
SRP-342II	"576,1662"	48	"48,64"	576
STP-103III	"384,1662"	32	"32,42"	384
SRP-275III	-	33	"33,44"	400
SRP-Q300	"512,1662"	42	"42,56"	512
SRP-Q302	"576,1662"	48	"48,64"	576

## 8-1-2 CashDrawer 지원 properties

Properties	Initial value
CheckHealthText	"
Claimed	False
DeviceEnabled	False
PowerState	UNKNOWN
State	CLOSED
DrawerOpened	False

## 8-1-3 MSR 지원 properties

Properties	Initial value
AutoDisable	False
CheckHealthText	“”
Claimed	False
DeviceEnabled	False
PowerState	UNKNOWN
State	CLOSED
CapDataEncryption	False
DataEncryptionAlgorithm	NONE
Track1Data	길이 0의 byte 배열
Track1EncryptedData	길이 0의 byte 배열
Track1EncryptedDataLength	0
Track2Data	길이 0의 byte 배열
Track2EncryptedData	길이 0의 byte 배열
Track2EncryptedDataLength	0
Track3Data	길이 0의 byte 배열
Track3EncryptedData	길이 0의 byte 배열
Track3EncryptedDataLength	0
TracksToRead	NONE

## 8-1-4 SmartCardRW 지원 properties

Properties	Initial value
CheckHealthText	“”
Claimed	False
DeviceEnabled	False
PowerState	UNKNOWN
State	CLOSED
CapInterfaceMode	APDU
CapIsoEmvMode	EMV
CapSCPpresentSensor	SLOT1   SLOT2   SLOT3
CapSCSlots	SLOT1   SLOT2   SLOT3
CapTransmissionProtocol	PROTOCOL_T0   PROTOCOL_T1
InterfaceMode	APDU
IsoEmvMode	EMV
SCPpresentSensor	0
SCSlot	0

## 8-2 Methods

POSPrinter	CashDrawer	MSR	SmartCardRW
open	open	open	open
close	close	close	close
claim	claim	claim	claim
release	release	release	release
checkHealth	checkHealth	checkHealth	checkHealth
directIO	openDrawer	clearInput clearInputProperties	clearInput clearInputProperties readData
clearOutput			
clearPrintArea			
cutPaper			
markFeed			
pageModePrint			
printBarCode			
printBitmap			
printNormal			
transactionPrint			
updateFirmware			

## 8-3 Events

Events	POSPrinter	CashDrawer	MSR	SmartCardRW
DataEvent	X	X	O	X
OutputCompleteEvent	O	X	X	X
StatusUpdateEvent	O	O	O	O

## 8-4 Errors

Method	Errors
open	NOSERVICE
claim	ILLEGAL TIMEOUT
release checkHealth clearOutput clearPrintArea cutPaper markFeed pageModePrint printBarCode printBitmap printNormal transactionPrint	ILLEGAL