

BIXOLON®

API 参考指南

BXL SDK for UPOS Compliant Android

Rev. 2.02

SPP-R210

SPP- R220 / R200II / R200III

SPP-R300 / R310

SPP-R400 / R410 / R418

SRP-275III

SRP-330II / 332II

SRP-340II / 342II

SRP-350III / 352III / 350plusIII / 352plusIII

SRP-380 / 382

SRP-F310II / F312II / F313II

SRP-S300

SRP-Q300 / Q302

SRP-QE300 / QE302

SRP-E300 / E302

STP-103III

<http://www.bixolon.com>

目录

1. 手册指南	4
2. 支持操作系统及界面	4
2-1 操作系统	4
2-2 支持设备及接口	4
3. 开发环境	5
3-1 配置环境	5
3-2 连接Android设备	5
3-2-1 Bluetooth	5
3-2-2 Network	6
3-2-3 Wi-Fi Direct	7
3-2-4 USB	8
3-2-5 设置Android设备开发者选项	12
4. 成套数码内容	13
4-1 手册	13
4-2 资源库	13
4-3 样本源代码	13
5. 常数值(Defines)	14
5-1 JposException	14
5-2 Event	15
5-2-1 StatusUpdate Event	15
5-2-2 Error Event	15
5-2-3 OutputComplete Event	15
5-2-4 Data Event	15
5-3 EscapeSequence	18
5-4 Transaction Print	19
5-5 Alignment	20
5-6 条形码类型	21
5-7 条形码文本位置	22
5-8 装置型号	22
5-9 页面模式打印方向	23
5-10 MSR 加密	23
5-11 SCR模式	23
5-12 Character Set	24
6. 各CLASS功能整理	25
6-1 BXLConfigLoader Class	25
6-1-1 openFile()	25
6-1-2 newFile()	25
6-1-3 getEntries()	26
6-1-4 addEntry()	27
6-1-5 removeEntry()	28
6-1-6 saveFile()	29
6-2 POSPrinter Class	30
6-2-1 open()	30
6-2-2 claim()	31
6-2-3 setDeviceEnabled()	32
6-2-4 release()	32

6-2-5 close()	33
6-2-6 checkHealth()	33
6-2-7 setAsyncMode()	34
6-2-8 setCharacterSet()	35
6-2-9 setCharacterEncoding()	36
6-2-10 cutPaper()	37
6-2-11 printBarCode()	38
6-2-12 printBitmap()	39
6-2-13 printBitmap()	40
6-2-14 printNormal()	41
6-2-15 printPDF()	42
6-2-16 printPDF()	43
6-2-17 setPageModePrintArea()	44
6-2-18 setPageModePrintDirection()	45
6-2-19 pageModePrint()	46
6-2-20 setPageModeHorizontalPosition()	47
6-2-21 setPageModeVerticalPosition()	48
6-2-22 transactionPrint()	49
6-3 MSR Class	55
6-3-1 open()	55
6-3-2 claim()	56
6-3-3 setDeviceEnabled()	57
6-3-4 release()	58
6-3-5 close()	58
6-3-6 setAutoDisable()	59
6-3-7 setDataEventEnabled()	60
6-3-8 setDataEncryptionAlgorithm()	61
6-3-9 getTrack1Data()	62
6-3-10 getTrack2Data()	63
6-3-11 getTrack1Data()	64
6-4 SmartCardRW Class	65
6-4-1 open()	65
6-4-2 claim()	66
6-4-3 setDeviceEnabled()	67
6-4-4 release()	67
6-4-5 close()	68
6-4-6 setSCSlot()	69
6-4-7 setIsoEmvMode()	70
6-4-8 readData()	71
6-5 CashDrawer Class	72
6-5-1 open()	72
6-5-2 claim()	73
6-5-3 setDeviceEnabled()	74
6-5-4 release()	74
6-5-5 close()	75
6-5-6 openDrawer()	75
6-5-7 getDrawerOpened()	76
7. 使用举例	77
7-1 打印文本	77
7-2 打印图像	78
7-3 打印PDF文件	79
7-4 Page mode打印	80

1. 手册指南

- 本SDK菜单内主要说明开发安卓应用程序所需程序库的有关内容。
- 说明SDK的使用方法、参数和限制。

2. 支持操作系统及界面

2-1 操作系统

- 本软件支持如下操作系统。
- Android 4.0 (Ice Cream Sandwich) 以上

2-2 支持设备及接口

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
SPP-R418	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
SRP-QE300	Ethernet / USB
SRP-QE302	Ethernet / USB
SRP-E300	Ethernet / USB
SRP-E302	Ethernet / USB

※ BLE: Bluetooth Low Energy

3. 开发环境

3-1 配置环境

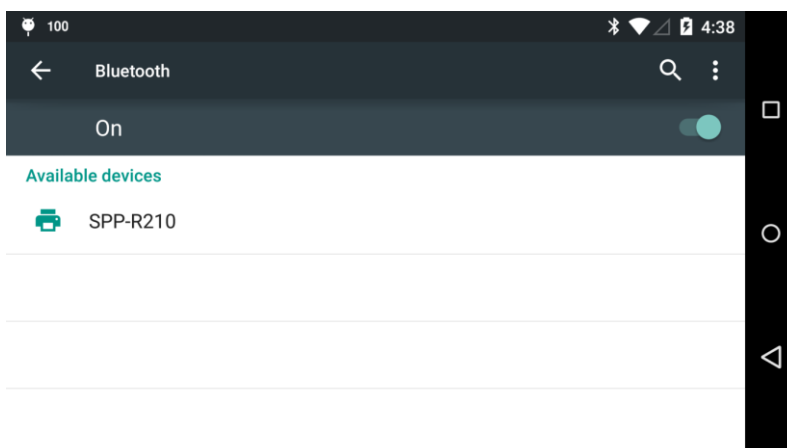
- Java Development Kit (JDK) 7
- Eclipse
- Android SDK Tools
- 参考: <http://developer.android.com/sdk/index.html>

3-2 连接Android设备

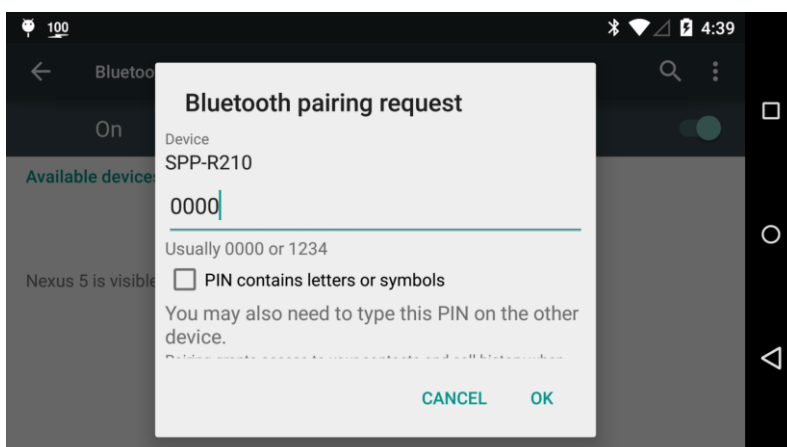
- 如下屏幕截图来自Nexus 5。
截图和项目名称可随Android操作系统或设备的不同而略有不同。

3-2-1 Bluetooth

1. 选择Settings。
2. 必须开启Bluetooth和打印机电源。
3. 选择Bluetooth后进行设置。

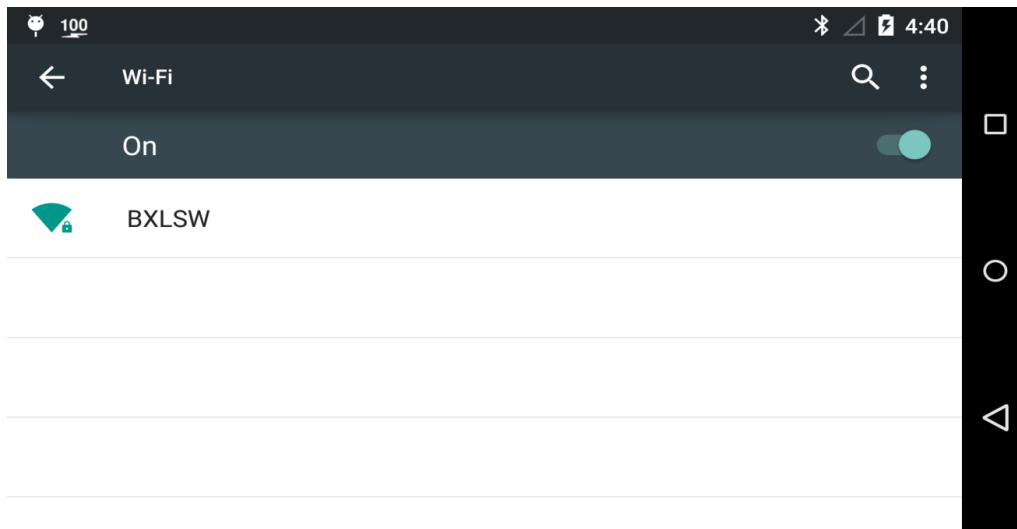


4. 选择Scan。搜索用于连接的打印机后配对。
5. 输入PIN code。(株)毕索龙的初始PIN code是“0000”。



3-2-2 Network

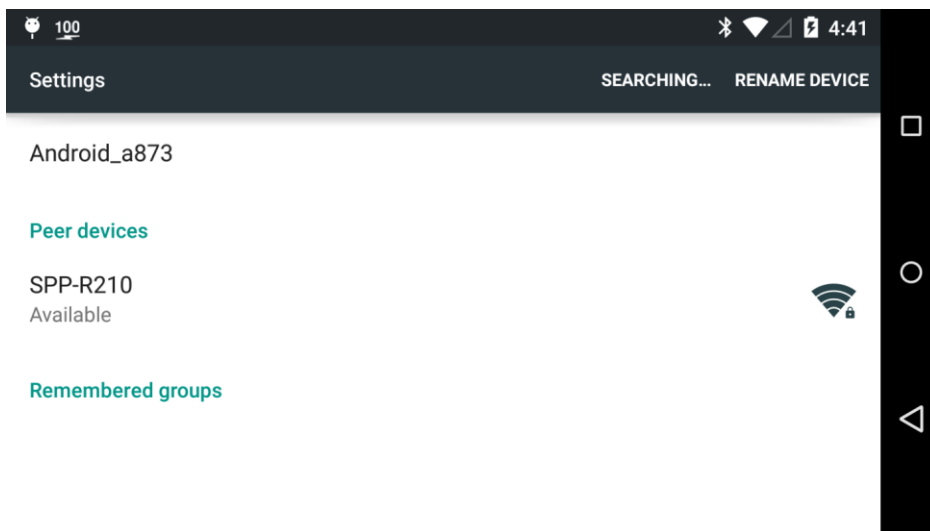
1. 将打印机连接至网络AP(Access Point)，分配IP地址或用DHCP设置。(株)毕索龙打印机初始设置为Ad-hoc，因此，应用本公司的Net Configuration Tool初始设置一次。Net Configuration Tool可在(株)毕索龙网站上下载。
(设置相关内容详见Net Configuration Tool菜单)
2. 选择Settings。
- 3.保持Wi-Fi开启。
- 4.连到(株)毕索龙打印机连接的网络。



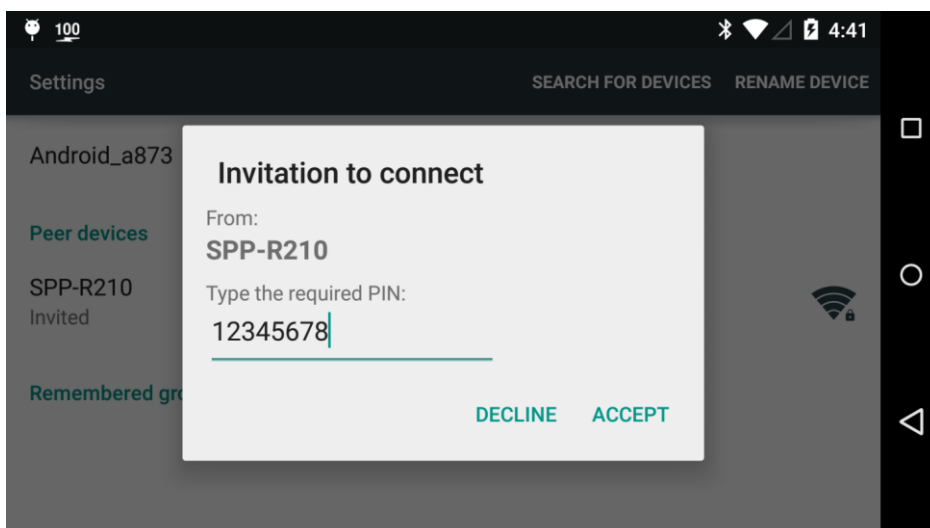
5. 在打印机的TCP/IP端口上连接安卓设备时无需追加设置。

3-2-3 Wi-Fi Direct

1. 安卓设备OS 4.0以上版本可用Wi-Fi Direct连接外围设备。
2. 安卓设备上无需安装(株)毕索龙的特定驱动程序和打印机软件。
3. 选择Settings。
4. 保持Wi-Fi开启。
5. 选择Wi-Fi Direct。

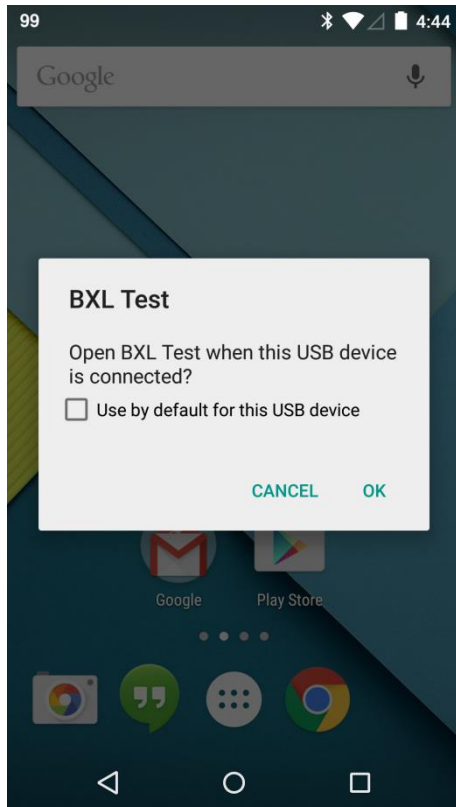


6. 选择并连接搜索到的打印机。
Wi-Fi Direct初始PIN号码是“12345678”。



3-2-4 USB

1. 安卓设备OS 3.1以上版本以连接USB外围设备。
2. 安卓设备无需安装 (株)毕索龙的特定驱动程序和打印机软件。
3. 随智能手机和平板电脑的不同，所需的USB线也略有不同。
请先确认要使用的安卓装置是否支持连接U盘后正确选用接线。
4. 第一次连接(株)毕索龙打印机时，随安卓设备的不同，可能会出现如下信息。



5. 将如下代码输入至以AndroidManifest.xml和Sample提供的BXLTest内res/xml/device_filter.xml，以连接U盘外围设备。

[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" />

  <!-- SPP-R418 -->
  <usb-device product-id="93" 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" />

<!-- SRP-QE300 -->
<usb-device product-id="153" vendor-id="5380" />

<!-- SRP-QE302 -->
<usb-device product-id="154" vendor-id="5380" />

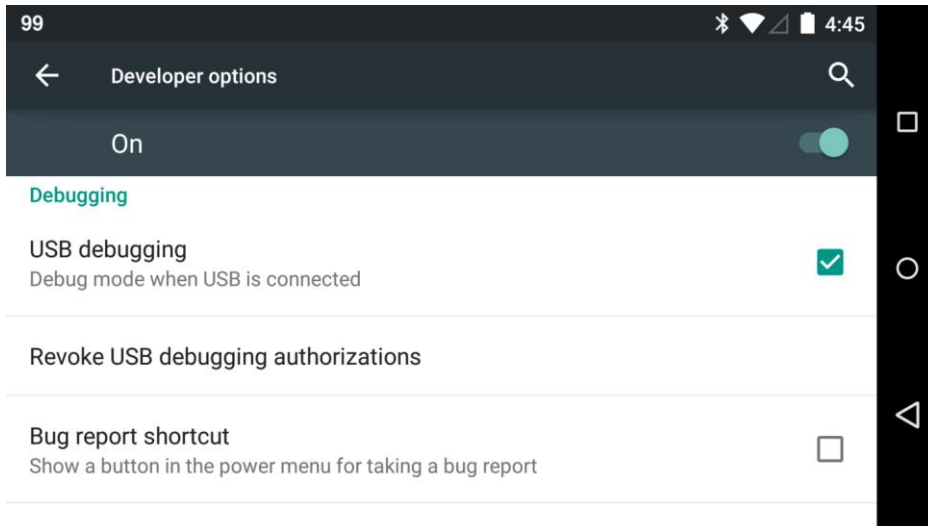
<!-- SRP-E300 -->
<usb-device product-id="152" vendor-id="5380" />

<!-- SRP-E302 -->
<usb-device product-id="156" vendor-id="5380" />

</resources>
```

3-2-5 设置Android设备开发者选项

1. 选择Settings。
2. 选择Developer options。
3. 激活USB debugging。



4. 成套数码内容

4-1 手册

菜单位置/名称	说明
docs/Manual_BXL SDK for Android_UPOS compliant API Reference Guide_english_Rev_x_xx	英文手册
docs/Manual_BXL SDK for Android_UPOS compliant API Reference Guide_japan_Rev_x_xx	日文手册
docs/Manual_BXL SDK for Android_UPOS compliant API Reference Guide_korean_Rev_x_xx	韩文手册
docs/Manual_BXL SDK for Android_UPOS compliant API Reference Guide_chinese_Rev_x_xx	中文手册

4-2 资源库

程序库位置/名称	说明
libs/bixolon_printer_Vxxx.jar	构建JavaPOS服务组件层级/打印机设置资源库
libs/jpos11x-controls.jar	JavaPOS设备控制及服务接口
libs/xerces.jar	构建apache.org XML服务(JavaPOS device control内需要)
libs/icu4j-58_1.jar	用于打印阿拉伯语/波斯语文字的程序库。
libs/PDF/bixolon_pdf.jar	打印PDF的资源库
libs/PDF/CPU type/libbxmlpdf.so	打印PDF的native资源库
libs/Image/OpenCV-2.4.11.jar	打印IMAGE的资源库
libs/Image/CPU type/libopencv_java.so	打印IMAGE的native资源库

4-3 样本源代码

范例位置/名称	说明
samples/BitmapTest	ビットマップ印刷サンプルアプリケーション
samples/BxlCustomerDisplay	BCD-3000文本、图像打印范例应用程序(仅限连接至SRP-Q300/Q302 Display port的装置)
samples/BXLTest	プリンターの設定、プリンターの接続、テキスト、バーコード、ビットマップ印刷、ページモード設定サンプルアプリケーション
samples/EventTest	Status/Error/OutputComplete Event范例应用程序
samples/FarsiTest	阿拉伯语和波斯语文字打印范例应用程序
samples/PDFTest	PDF打印范例应用程序
samples/TextDataTest	テキスト印刷オプション設定サンプルアプリケーション

5. 常数值(Defines)

5-1 JposException

- Method中执行特定功能的过程中出错时发生JposException例外。
发生例外时可以通过“printStackTrace”函数确认错误内容。

[Example]

```
import jpos.JposConst;

try
{
    .....
}
catch(JposException e)
{
    // Error
    e.printStackTrace();
}
```

5-2 Event

- 在JposConst, POSPrinterConst CLASS内定义各Event。

5-2-1 StatusUpdate Event

每次变更打印机状态时发生 StatusUpdate Event。

常数	值	说明
JPOS_SUE_POWER_ONLINE	2001	Printer Power on
JPOS_SUE_POWER_OFF_OFFLINE	2004	Printer Power off
PTR_SUE_COVER_OPEN	11	Cover Open
PTR_SUE_COVER_OK	12	Cover OK
PTR_SUE_REC_EMPTY	24	Receipt Paper Empty
PTR_SUE_REC_NEAREMPTY	25	Receipt Paper Near Empty
PTR_SUE_REC_PAPEROK	26	Receipt Paper OK
PTR_SUE_IDLE	1001	Printer Idle
PTR_SUE_OFF_LINE	53	Printer off-line
PTR_SUE_ON_LINE	54	Printer on-line
PTR_SUE_BAT_OK	55	Printer battery normal
PTR_SUE_BAT_LOW	56	Printer battery low

5-2-2 Error Event

常数	值	说明
JPOS_EPTR_COVER_OPEN	201	Cover Open
JPOS_EPTR_REC_EMPTY	203	Paper Empty
JPOS_EPTR_OFF_LINE	217	Printer off-line

5-2-3 OutputComplete Event

发生打印完毕Event。但应以Async模式使用。

5-2-4 Data Event

接收Direct I/O的应答及MSR Track信息等数据。

[Event Example]

```
try
{
    POSPrinter posPrinter = new POSPrinter(this);
    MSR msr = new MSR();
    posPrinter.addErrorListener(this);
    posPrinter.addStatusUpdateListener(this);
    posPrinter.addOutputCompleteListener(this);
    msr.addDataListener(this);
    posPrinter.setAsyncMode(true);
}
catch(JposException e)
{
    // Error
    e.printStackTrace();
}

@Override
public void outputCompleteOccurred(final OutputCompleteEvent e)
{
    runOnUiThread(new Runnable()
    {
        @Override
        public void run()
        {
            Toast.makeText(MainActivity.this, "complete print", Toast.LENGTH_SHORT).show();
        }
    });
}

@Override
public void dataOccurred(DataEvent arg0)
{
    // TODO Auto-generated method stub
    runOnUiThread(new Runnable()
    {
        @Override
        public void run()
        {
            try
            {
                String strData = new String(msr.getTrack1Data());
                strData += new String(msr.getTrack2Data());
                strData += new String(msr.getTrack3Data());
                Toast.makeText(MainActivity.this, strData, Toast.LENGTH_SHORT).show();
            }
            catch(JposException e)
            {
                Toast.makeText(MainActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    });
}
```



```
@Override
public void errorOccurred(final ErrorEvent arg0)
{
    // TODO Auto-generated method stub
    runOnUiThread(new Runnable()
    {
        @Override
        public void run()
        {
            switch (arg0.getErrorCodeExtended())
            {
                case POSPrinterConst.JPOS_EPTR_COVER_OPEN:
                    return "Cover open";
                case POSPrinterConst.JPOS_EPTR_REC_EMPTY:
                    return "Paper empty";
                case JposConst.JPOS_SUE_POWER_OFF_OFFLINE:
                    return "Power off";
                default:
                    return "Unknown";
            }
        }
    });
}
```

```
@Override
public void statusUpdateOccurred(final StatusUpdateEvent arg0)
{
    // TODO Auto-generated method stub
    runOnUiThread(new Runnable()
    {
        @Override
        public void run()
        {
            switch (arg0.getStatus())
            {
                case JposConst.JPOS_SUE_POWER_ONLINE:
                    return "Power on";
                case JposConst.JPOS_SUE_POWER_OFF_OFFLINE:
                    return "Power off";
                case POSPrinterConst.PTR_SUE_COVER_OPEN:
                    return "Cover Open";
                case POSPrinterConst.PTR_SUE_COVER_OK:
                    return "Cover OK";
                case POSPrinterConst.PTR_SUE_REC_EMPTY:
                    return "Receipt Paper Empty";
                case POSPrinterConst.PTR_SUE_REC_NEAREMPTY:
                    return "Receipt Paper Near Empty";
                case POSPrinterConst.PTR_SUE_REC_PAPEROK:
                    return "Receipt Paper OK";
                case POSPrinterConst.PTR_SUE_IDLE:
                    return "Printer Idle";
                case POSPrinterConst.PTR_SUE_BAT_LOW:
                    return "Battery-Low";
                case POSPrinterConst.PTR_SUE_BAT_OK:
                    return "Battery-OK";
                default:
                    return "Unknown";
            }
        }
    });
}
```

5-3 EscapeSequence

- 用于指定要打印文本的字体、粗细等选项的值。字符串形式，在打印数据前面追加使用。

定义	说明
String ESCAPE_CHARACTERS = new String(new byte[] {0x1b, 0x7c})	Escape Characters
ESCAPE_CHARACTERS + "N"	Normal
ESCAPE_CHARACTERS + "aM"	Font A (12x24)
ESCAPE_CHARACTERS + "bM"	Font B (9x17)
ESCAPE_CHARACTERS + "cM"	Font C (9x24)
ESCAPE_CHARACTERS + "lA"	Left justify
ESCAPE_CHARACTERS + "cA"	Center
ESCAPE_CHARACTERS + "rA"	Right justify
ESCAPE_CHARACTERS + "bC"	Bold
ESCAPE_CHARACTERS + "!bC"	Disabled bold
ESCAPE_CHARACTERS + "uC"	Underline
ESCAPE_CHARACTERS + "!uC"	Disabled underline
ESCAPE_CHARACTERS + "rvC"	Reverse video
ESCAPE_CHARACTERS + "!rvC"	Disabled reverse video
ESCAPE_CHARACTERS + "1C"	Single high and wide
ESCAPE_CHARACTERS + "2C"	Double wide
ESCAPE_CHARACTERS + "3C"	Double high
ESCAPE_CHARACTERS + "4C"	Double high and wide
ESCAPE_CHARACTERS + "1hC"	Scale 1 time horizontally
ESCAPE_CHARACTERS + "2hC"	Scale 2 times horizontally
ESCAPE_CHARACTERS + "3hC"	Scale 3 times horizontally
ESCAPE_CHARACTERS + "4hC"	Scale 4 times horizontally
ESCAPE_CHARACTERS + "5hC"	Scale 5 times horizontally
ESCAPE_CHARACTERS + "6hC"	Scale 6 times horizontally
ESCAPE_CHARACTERS + "7hC"	Scale 7 times horizontally
ESCAPE_CHARACTERS + "8hC"	Scale 8 times horizontally
ESCAPE_CHARACTERS + "1vC"	Scale 1 time vertically
ESCAPE_CHARACTERS + "2vC"	Scale 2 times vertically
ESCAPE_CHARACTERS + "3vC"	Scale 3 times vertically
ESCAPE_CHARACTERS + "4vC"	Scale 4 times vertically
ESCAPE_CHARACTERS + "5vC"	Scale 5 times vertically
ESCAPE_CHARACTERS + "6vC"	Scale 6 times vertically
ESCAPE_CHARACTERS + "7vC"	Scale 7 times vertically
ESCAPE_CHARACTERS + "8vC"	Scale 8 times vertically

[Example]

```

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open(logicalDeviceName);
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true)

    String ESCAPE_CHARACTERS = new String(new byte[] {0x1b, 0x7c})

    // 加粗
    String data = ESCAPE_SEQUENCE + "bC" + "Bixolon\n"
    posPrinter.printNormal(POSPrinterConst.PTR_S_RECEIPT, data);

    // 取消加粗
    String data = ESCAPE_SEQUENCE + "!bC" + "Bixolon\n"
    posPrinter.printNormal(POSPrinterConst.PTR_S_RECEIPT, data);
}
catch(JposException e)
{
    // Error
    e.printStackTrace();
}

```

5-4 Transaction Print

- 定义设置Transaction mode时可以使用数值。

常数	值	说明
PTR_TP_TRANSACTION	11	将Buffer初始化为空，开始Transaction Mode。
PTR_TP_NORMAL	12	结束Transaction mode，打印Buffer内积累的Data。

5-5 Alignment

- 定义指定排列时所需的数值。

[For Barcode]

常数	值	说明
PTR_BC_LEFT	-1	左对齐
PTR_BC_CENTER	-2	居中
PTR_BC_RIGHT	-3	右对齐

[For Image]

常数	值	说明
PTR_BM_LEFT	-1	左对齐
PTR_BM_CENTER	-2	居中
PTR_BM_RIGHT	-3	右对齐

[For PDF]

常数	值	说明
PTR_PDF_LEFT	-1	左对齐
PTR_PDF_CENTER	-2	居中
PTR_PDF_RIGHT	-3	右对齐

5-6 条形码类型

- 定义打印条形码时指定条形码所需的数值。

常数	值	说明								
PTR_BCS_UPCA	101	UPCA								
PTR_BCS_UPCE	102	UPCE								
PTR_BCS_JAN8	103	JAN8								
PTR_BCS_EAN8	103	EAN8								
PTR_BCS_JAN13	104	JAN13								
PTR_BCS_EAN13	104	EAN13								
PTR_BCS_TF	105	Standard(ordiscrete) 2 of 5								
PTR_BCS_ITF	106	Interleaved 2 of 5								
PTR_BCS_Codabar	107	Codabar								
PTR_BCS_Code39	108	Code39								
PTR_BCS_Code93	109	Code93								
PTR_BCS_Code128	110	Code 128 ※ Code128的Special Character								
		<table border="1"> <thead> <tr> <th>Special Charaters</th> <th>Ascii Represntation</th> </tr> </thead> <tbody> <tr> <td>Code A</td> <td>{A</td> </tr> <tr> <td>Code B</td> <td>{B</td> </tr> <tr> <td>Code C</td> <td>{C</td> </tr> </tbody> </table>	Special Charaters	Ascii Represntation	Code A	{A	Code B	{B	Code C	{C
		Special Charaters	Ascii Represntation							
		Code A	{A							
Code B	{B									
Code C	{C									
PTR_BCS_UPCA_S	111	UPC-A with supplemental barocde								
PTR_BCS_UPCE_S	112	UPC-E with supplemental barcode								
PTR_BCS_UPCD1	113	UPC-D1								
PTR_BCS_UPCD2	114	UPC-D2								
PTR_BCS_UPCD3	115	UPC-D3								
PTR_BCS_UPCD4	116	UPC-D4								
PTR_BCS_UPCD5	117	UPC-D5								
PTR_BCS_EAN8_S	118	EAN8 with supplemental barocde								
PTR_BCS_EAN13_S	119	EAN13 with supplemental barocde								
PTR_BCS_EAN128	120	EAN128								
PTR_BCS_OCRA	121	OCR "A"								
PTR_BCS_OCRB	122	OCR "B"								
PTR_BCS_Code128_Parsed	123	Code 128 with parsing								
PTR_BCS_GS1DATABAR	131	GS1 DataBar Omnidirectional								
PTR_BCS_GS1DATABAR_E	132	GS1 DataBar Stacked Omnidirectional								
PTR_BCS_GS1DATABAR_S	133	GS1 DataBar Expanded								
PTR_BCS_GS1DATABAR_E_S	134	GS1 DataBar Expanded Stacked								
PTR_BCS_PDF417	201	PDF 417								
PTR_BCS_MAXICODE	202	MAXI Code								
PTR_BCS_DATAMATRIX	203	Data Matrix								
PTR_BCS_QRCODE	204	QR Code								
PTR_BCS_UQRCODE	205	Micro QR Code								
PTR_BCS_AZTEC	206	Aztec								
PTR_BCS_UPDF417	207	Micro PDF 417								

5-7 条形码文本位置

- 属于支持打印文本的条形码时，指定是否打印条形码文本及位置。

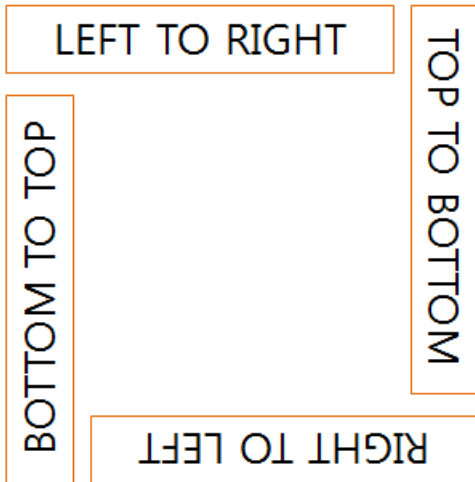
常数	值	说明
PTR_BC_TEXT_NONE	-11	不打印文本，只打印条件码。
PTR_BC_TEXT_ABOVE	-12	在条形码上方打印文本。
PTR_BC_TEXT_BELOW	-13	在条形码下方打印文本。

5-8 装置型号

常数	说明
PRODUCT_NAME_SPP_R210	SPP-R210
PRODUCT_NAME_SPP_R220	SPP-R220
PRODUCT_NAME_SPP_R200II	SPP-R200II
PRODUCT_NAME_SPP_R200III	SPP-R200III
PRODUCT_NAME_SPP_R300	SPP-R300
PRODUCT_NAME_SPP_R310	SPP-R310
PRODUCT_NAME_SPP_R400	SPP-R400
PRODUCT_NAME_SPP_R410	SPP-R410
PRODUCT_NAME_SPP_R418	SPP-R418
PRODUCT_NAME_SRP_350III	SRP-350III
PRODUCT_NAME_SRP_352III	SRP-352III
PRODUCT_NAME_SRP_330II	SRP-330II
PRODUCT_NAME_SRP_332II	SRP-332II
PRODUCT_NAME_SRP_340II	SRP-340II
PRODUCT_NAME_SRP_342II	SRP-342II
PRODUCT_NAME_SRP_350PLUSIII	SRP-350PLUSIII
PRODUCT_NAME_SRP_352PLUSIII	SRP-352PLUSIII
PRODUCT_NAME_SRP_380	SRP-380
PRODUCT_NAME_SRP_382	SRP-382
PRODUCT_NAME_SRP_S300	SRP-S300
PRODUCT_NAME_SRP_Q300	SRP-Q300
PRODUCT_NAME_SRP_Q302	SRP-Q302
PRODUCT_NAME_SRP_F310II	SRP-F310II
PRODUCT_NAME_SRP_F312II	SRP-F312II
PRODUCT_NAME_SRP_F313II	SRP-F313II
PRODUCT_NAME_STP_103III	STP-103III
PRODUCT_NAME_SRP_275III	SRP-275III
PRODUCT_NAME_SRP_QE300	SRP-QE300
PRODUCT_NAME_SRP_QE302	SRP-QE302
PRODUCT_NAME_SRP_E300	SRP-E300
PRODUCT_NAME_SRP_E302	SRP-E302

5-9 页面模式打印方向

常数	值	说明
PTR_PD_LEFT_TO_RIGHT	1	由左向右打印
PTR_PD_BOTTOM_TO_TOP	2	由下向上打印
PTR_PD_RIGHT_TO_LEFT	3	由右向左打印
PTR_PD_TOP_TO_BOTTOM	4	由上向下打印



5-10 MSR 加密

常数	值	说明
MSR_DE_NONE	1	不使用数据加密算法
MSR_DE_3DEA_DUKPT	2	使用数据加密算法

5-11 SCR模式

常数	值	说明
SC_CMODE_ISO	1	ISO模式
SC_CMODE_EMV	2	EMV模式

5-12 Character Set

常数	值	说明
CS_437_USA_STANDARD_EUROPE	437	PC437
CS_737_GREEK	737	PC737
CS_775_BALTIC	775	PC775
CS_850_MULTILINGUAL	850	PC850
CS_852_LATIN2	852	PC852
CS_855_CYRILLIC	855	PC855
CS_857_TURKISH	857	PC857
CS_858_EURO	858	PC858
CS_860_PORTUGUESE	860	PC860
CS_862_HEBREW_DOS_CODE	862	PC862
CS_863_CANADIAN_FRENCH	863	PC863
CS_864_ARABIC	864	PC864
CS_865_NORDIC	865	PC865
CS_866_CYRILLIC2	866	PC866
CS_928_GREEK	928	PC928
CS_1250_CZECH	1250	WPC1250
CS_1251_CYRILLIC	1251	WPC1251
CS_1252_LATIN1	1252	WPC1252
CS_1253_GREEK	1253	WPC1253
CS_1254_TURKISH	1254	WPC1254
CS_1255_HEBREW_NEW_CODE	1255	WPC1255
CS_1256_ARABIC	1256	WPC1256
CS_1257_BALTIC	1257	WPC1257
CS_1258_VIETNAM	1258	WPC1258
CS_FARSI	7065	FARSI
CS_KATAKANA	7565	KATAKANA
CS_KHMER_CAMBODIA	7572	KHMER
CS_THAI11	8411	THAI11
CS_THAI14	8414	THAI14
CS_THAI16	8416	THAI16
CS_THAI18	8418	THAI18
CS_THAI42	8442	THAI42
CS_KS5601	5601	KS5601
CS_BIG5	6605	BIG5
CS_GB2312	2312	GB2312
CS_SHIFT_JIS	8374	SHIFT-JIS
CS_TCVN_3_1	3031	TCVN-3(1)
CS_TCVN_3_2	3032	TCVN-3(2)

6. 各CLASS功能整理

6-1 BXLConfigLoader Class

- 用于保存要连接装置之设置信息Class。设置信息通过BXLConfigLoader Class管理装置信息。设置信息内包括装置的名称、产品名称、接口等，不正常保存相应信息将无法连接装置。应在调入Open函数之前调入该CLASS并保存设置信息。

Caption

无法连接未通过BXLConfigLoader保存的设备。

6-1-1 openFile()

打开已保存的设置文件。

[Syntax]

void openFile() throws Exception

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;

try
{
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
    bxlConfigLoader.openFile();
}
catch(Exception e)
{
    // Error
    e.printStackTrace();
}
```

6-1-2 newFile()

生成新设置保存文件。发生一般openFile例外时执行。

[Syntax]

void newFile() throws Exception

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;

try
{
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
    bxlConfigLoader.openFile();
}
catch(Exception e)
{
    e.printStackTrace();
    bxlConfigLoader.newFile();
}
```

6-1-3 getEntries()

调入已保存的设置信息。

[Syntax]

List<JposEntry> getEntries() throws Exception

[Return Values]

值	说明
JposEntry的List容器	对象的List排列

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;

try
{
    for (Object entry : bxlConfigLoader.getEntries())
    {
        JposEntry jposEntry = (JposEntry)entry;
        String strLogicalname = jposEntry.getLogicalName();
    }
}
catch(Exception e)
{
    e.printStackTrace();
}
```

6-1-4 addEntry()

追加装置连接信息。

[Syntax]

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

[Parameters]

- String logicalName : 输入装置的逻辑性名称(别名)。
- int deviceCategory : 选择装置的种类。

常数	值	说明
DEVICE_CATEGORY_CASH_DRAWER	0	Cash Drawer
DEVICE_CATEGORY_MSR	1	MSR
DEVICE_CATEGORY_POS_PRINTER	2	POS Printer
DEVICE_CATEGORY_SMART_CARD_RW	3	SCR

- String productName : 装置型号(“5-8 装置型号” 参考)
- int deviceBus : 选择界面类型。

常数	值	说明
DEVICE_BUS_BLUETOOTH	0	Bluetooth
DEVICE_BUS_ETHERNET	1	Ethernet
DEVICE_BUS_USB	2	USB
DEVICE_BUS_WIFI	3	WiFi
DEVICE_BUS_WIFI_DIRECT	4	WiFi-Direct
DEVICE_BUS_BLUETOOTH_LE	5	Bluetooth Low Energy

- String address : 输入装置的MAC或IP Address。
(BT : MAC Address, Network : IP Address)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;

try
{
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
    bxlConfigLoader.addEntry("SPP-R200III",
        BXLConfigLoader.DEVICE_CATEGORY_POS_PRINTER,
        BXLConfigLoader.PRODUCT_NAME_SPP_R200III,
        BXLConfigLoader.DEVICE_BUS_BLUETOOTH,
        "74:F0:7D:E4:11:AF");
}
catch(Exception e)
{
    e.printStackTrace();
}
```

6-1-5 removeEntry()

删除保存的设置信息。

[Syntax]

boolean removeEntry(String logicalName)

[Parameters]

- String logicalName : 输入装置的逻辑性名称(别称)。

[Return Values]

值	说明
true	成功时返回。
false	失败时返回。

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;

try
{
    for (Object entry : bxlConfigLoader.getEntries())
    {
        JposEntry jposEntry = (JposEntry)entry;
        String strLogicalname = jposEntry.getLogicalName();

        bxlConfigLoader.removeEntry(strLogicalname);
    }
}
catch(Exception e)
{
    e.printStackTrace();
}
```

6-1-6 saveFile()

通过addEntry、removeEntry保存变更后的信息。

[Syntax]

```
void saveFile()
```

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;

try
{
    for (Object entry : bxlConfigLoader.getEntries())
    {
        JposEntry jposEntry = (JposEntry)entry;
        String strLogicalname = jposEntry.getLogicalName();

        bxlConfigLoader.removeEntry(strLogicalname);
    }

    bxlConfigLoader.addEntry("SPP-R200III",
        BXLConfigLoader.DEVICE_CATEGORY_POS_PRINTER,
        BXLConfigLoader.PRODUCT_NAME_SPP_R200III,
        BXLConfigLoader.DEVICE_BUS_BLUETOOTH,
        "74:F0:7D:E4:11:AF");

    bxlConfigLoader.saveFile();
}
catch(Exception e)
{
    e.printStackTrace();
}
```

6-2 POSPrinter Class

- 用于控制POS Printer的Class。使用相应Class，执行连接及解除打印机和打印作业。
执行特定功能过程中出错时发生JposException。（“5-1 JposException” 参考）

6-2-1 open()

开始使用打印机CLASS，包含分配内存等初始化作业。
务必先执行，以便调用Claim以上的Method。
无法打开未通过BXLConfigLoader Class保存的装置。

[Syntax]

void open(String logicalDeviceName) throws JposException

[Parameters]

- String logicalDeviceName : 输入要Open装置的名称。

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
}
catch(JposException e)
{
    e.printStackTrace();
}
```

Caption

应以saveFile函数保存通过addEntry函数追加的设备。

6-2-2 claim()

尝试实际打开Device信息内包含的Port，包含初始化分配内存等部分初始化作业。务必先行执行，以激活Device使用。

[Syntax]

void claim(int timeout) throws JposException

[Parameters]

- int timeout : 尝试打开本参数中指定的时间动因Port。

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(3000);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-3 setDeviceEnabled()

包含是否使用装置。

DeviceEnabled值失效，可能无法使用功能。

[Syntax]

void setDeviceEnabled(boolean deviceEnabled) throws JposException

[Parameters]

- boolean deviceEnabled：输入装置激活状态。

值	说明
true	激活
false	失效

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-4 release()

用物理性方式结束使用Claim的Device的Port。

可以执行解除内存等作业。

[Syntax]

void release() throws JposException

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try
{
    posPrinter.release();
}
catch(JposException e)
{
    e.printStackTrace();
}
```


6-2-5 close()

结束使用Open的Device。
可以部分执行解除内存等作业。

[Syntax]

void close() throws JposException

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try
{
    posPrinter.close();
    posPrinter.setDeviceEnabled(false);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-6 checkHealth()

确认装置是否处于正常运行状态。
Open/Claim/DeviceEnabled为正常执行状态时可以使用。

[Syntax]

void checkHealth(int level) throws JposException

[Parameters]

- int level : 固定值 JposConst.JPOS_CH_INTERNAL

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.checkHealth(JposConst.JPOS_CH_INTERNAL);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-7 setAsyncMode()

包含是否使用非同步模式。

非同步模式为true时，相关方法以非同步模式运行。为False时，相关方法以同步模式运行。

[Syntax]

void setAsyncMode(boolean asyncMode) throws JposException

[Parameters]

- boolean asyncMode : 选择是否使用非同步模式。

值	说明
true	非同步模式
false	同步模式

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.setAsyncMode(true);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-8 setCharacterSet()

设置打印机的Character set。

[Syntax]

void setCharacterSet(int characterSet) throws JposException

[Parameters]

- int characterSet : 选择要在打印机中设置的Character set。
(“5-12 Character Set” 参考)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.setCharacterSet(BXLConst.CS_437_USA_STANDARD_EUROPE);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-9 setCharacterEncoding()

设置要向打印机传送数据的编码。

[Syntax]

void setCharacterEncoding(int characterEncoding) throws JposException

[Parameters]

- int characterEncoding : 选择数据编码类型。

常数	值	说明
CE_ASCII	0	ASCII(default)
CE_UTF8	1	UTF-8

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.setCharacterEncoding(BXLConst.CE_ASCII);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-10 cutPaper()

内置Auto Cutter的型号切纸。

[Syntax]

void cutPaper(int percentage) throws JposException

[Parameters]

- int percentage : 选择Full cut/Partial cut。
仅限有切纸刀的型号运行。

值	说明
100	Full cut
90	Partial cut

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.cutPaper(100);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-11 printBarcode()

打印条形码。

[Syntax]

void printBarcode(int station, String data, int symbology, int height, int width, int alignment, int textPosition) throws JposException

[Parameters]

- int station : 固定值PTR_S_RECEIPT
- String data : 条形码内要包含的Data、各条形码允许的Data可能有所不同。
- int symbology : 选择Barcode种类。(“5-6 条形码类型” 参考)
- int height : 指定Barcode的高度。
- int width : 指定Barcode的宽度。
- int alignment : 选择条形码排列。(“5-5 Alignment” 参考)
- int textPosition : 决定要与条形码一同打印的Text的Postion。
(“5-7 条形码文本位置” 参考)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.printBarcode(POSPrinterConst.PTR_S_RECEIPT,
                           "123456789",
                           POSPrinterConst.PTR_BCS_QRCODE,
                           8,
                           8,
                           POSPrinterConst.PTR_BC_CENTER,
                           POSPrinterConst.PTR_BC_TEXT_BELOW);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-12 printBitmap()

打印图像。(打印文件)

[Syntax]

void printBitmap(int station, String fileName, int width, int alignment) throws JposException

[Parameters]

- int station : 设置图像打印选项。(共由4字节构成)

Index	说明
第一个字节	固定值 : PTR_S_RECEIPT
第二个字节	亮度 : 0 ~ 100
第三个字节	是否使用压缩算法(RLE) (0x01 : RLE)
第四个字节	Reserved(0x00)

- String filename : 指定图像文件的路径。
- int width : 指定图像宽度。
- int alignment : 选择图像排列。(“5-5 Alignment” 参考)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    ByteBuffer buffer = ByteBuffer.allocate(4);
    buffer.put((byte) POSPrinterConst.PTR_S_RECEIPT);
    buffer.put((byte) 80); // brightness
    buffer.put((byte) 0x01); // compress
    buffer.put((byte) 0x00);

    posPrinter.printBitmap(buffer.getInt(0),
                           imagePath,
                           posPrinter.getRecLineWidth(),
                           POSPrinterConst.PTR_BM_LEFT);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-13 printBitmap()

打印图像。(Bitmap 文件)

[Syntax]

void printBitmap(int station, Bitmap bitmap, int width, int alignment) throws JposException

[Parameters]

- int station : 设置图像打印选项。(共由4字节构成)

Index	说明
第一个字节	固定值 : PTR_S_RECEIPT
第二个字节	亮度 : 0 ~ 100
第三个字节	是否使用压缩算法(RLE) (0x01 : RLE)
第四个字节	Reserved(0x00)

- Bitmap bitmap : 输入位数映射形式的图像数据。
- int width : 指定图像宽度。
- int alignment : 选择图像排列。(“5-5 Alignment” 参考)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    ByteBuffer buffer = ByteBuffer.allocate(4);
    buffer.put((byte) POSPrinterConst.PTR_S_RECEIPT);
    buffer.put((byte) 80); // brightness
    buffer.put((byte) 0x01); // compress
    buffer.put((byte) 0x00);

    posPrinter.printBitmap(buffer.getInt(0),
                           BitmapData,
                           posPrinter.getRecLineWidth(),
                           POSPrinterConst.PTR_BM_LEFT);
}
catch(JposException e)
{
    e.printStackTrace();
}
```


6-2-14 printNormal()

打印文本。

[Syntax]

void printNormal(int station, String data) throws JposException

[Parameters]

- int station : 固定值 PTR_S_RECEIPT
- String data : 指定要打印的Data。允许可打印的字符和escape sequences、carriage returns、line feeds Data。

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.printNormal(POSPrinterConst.PTR_S_RECEIPT, "Print Data\n");
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-15 printPDF()

打印PDF文件。

[Syntax]

void printPDF(int station, String fileName, int width, int alignment, int page, int brightness)
throws JposException

[Parameters]

- int station : 固定值 PTR_S_RECEIPT
- String filename : 指定PDF文件的路径。
- int width : 指定PDF宽度。
- int alignment : 选择图像排列。(“5-5 Alignment” 参考)
- int page : 指定要打印的PDF页号。
- int brightness : 指定亮度值。(0 ~ 100)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.printPDF(POSPrinterConst.PTR_S_RECEIPT,
                        "pdfFilePath",
                        posPrinter.getRecLineWidth(),
                        POSPrinterConst.PTR_PDF_CENTER,
                        1,
                        80);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-16 printPDF()

打印PDF文件。

[Syntax]

void printPDF(int station, String fileName, int width, int alignment, int startPage, int endPage, int brightness) throws JposException

[Parameters]

- int station : 固定值 PTR_S_RECEIPT
- String filename : 指定PDF文件的路径。
- int width : 指定PDF宽度。
- int alignment : 选择图像排列。(“5-5 Alignment” 参考)
- int startPage : 指定要开始打印的页号。
- int endPage : 指定要结束打印的页号。
- int brightness : 指定亮度值。(0 ~ 100)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.printPDF(POSPrinterConst.PTR_S_RECEIPT,
                        "pdfFilePath",
                        posPrinter.getRecLineWidth(),
                        POSPrinterConst.PTR_PDF_CENTER,
                        1,
                        3,
                        80);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-17 setPageModePrintArea()

指定页面模式区域。

[Syntax]

void setPageModePrintArea(String area) throws JposException

[Parameters]

- String area : 指定打印区域。
“区域的x坐标、区域的y坐标、区域的长、区域的宽”
ex) “0, 0, 576, 1600”

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.setPageModePrintArea("0, 0, 576, 1600");
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-18 setPageModePrintDirection()

指定页面模式打印方向。

[Syntax]

void setPageModePrintDirection(int direction) throws JposException

[Parameters]

- int direction : 指定打印方向。(“5-9 页面模式打印方向” 参考)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.setPageModePrintArea("0, 0, 576, 1600");
    posPrinter.setPageModePrintDirection(POSPrinterConst.PTR_PD_LEFT_TO_RIGHT);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-19 pageModePrint()

将打印机设置为页面模式。

[Syntax]

void pageModePrint(int control) throws JposException

[Parameters]

- int control : 指定页面模式。

常数	值	说明
PTR_PM_PAGE_MODE	1	激活页面模式
PTR_PM_NORMAL	2	变更为一般模式，打印保存在页面模式缓冲器内的数据。

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.setPageModePrintArea("0, 0, 576, 1600");
    posPrinter.setPageModePrintDirection(POSPrinterConst.PTR_PD_LEFT_TO_RIGHT);
    posPrinter.setPageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-20 setPageModeHorizontalPosition()

指定开始打印位置(横)。

[Syntax]

void setPageModeHorizontalPosition(int position) throws JposException

[Parameters]

- int position : 指定开始打印位置(横)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    ByteBuffer buffer = ByteBuffer.allocate(4);
    buffer.put((byte) POSPrinterConst.PTR_S_RECEIPT);
    buffer.put((byte) 80); // brightness
    buffer.put((byte) 0x01); // compress
    buffer.put((byte) 0x00);

    // 指定打印区域
    posPrinter.setPageModePrintArea("0, 0, 576, 1600");
    // 设置打印方向
    posPrinter.setPageModePrintDirection(POSPrinterConst.PTR_PD_LEFT_TO_RIGHT);
    // 转换为打印机页面模式
    posPrinter.setPageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);

    // 指定横/纵打印位置
    posPrinter.setPageModeHorizontalPosition(0);
    posPrinter.setPageModeVerticalPosition(0);
    // 传送要打印的数据(图像)
    posPrinter.printBitmap(buffer.getInt(0), imagePath, 384, POSPrinterConst.PTR_BM_LEFT);

    // 指定横/纵打印位置
    posPrinter.setPageModeHorizontalPosition(100);
    posPrinter.setPageModeVerticalPosition(100);
    // 传送要打印的数据(文本)
    posPrinter.printNormar(POSPrinterConst.PTR_S_RECEIPT, "Print Data\n");

    // 开始打印
    posPrinter.setPageModePrint(POSPrinterConst.PTR_PM_PAGE_NORMAL);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-21 setPageModeVerticalPosition()

指定开始打印位置(纵)。

[Syntax]

void setPageModeVerticalPosition(int position) throws JposException

[Parameters]

- int position : 指定开始打印位置(纵)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    ByteBuffer buffer = ByteBuffer.allocate(4);
    buffer.put((byte) POSPrinterConst.PTR_S_RECEIPT);
    buffer.put((byte) 80); // brightness
    buffer.put((byte) 0x01); // compress
    buffer.put((byte) 0x00);

    // 指定打印区域
    posPrinter.setPageModePrintArea("0, 0, 576, 1600");
    // 设置打印方向
    posPrinter.setPageModePrintDirection(POSPrinterConst.PTR_PD_LEFT_TO_RIGHT);
    // 转换为打印机页面模式
    posPrinter.setPageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);

    // 指定横/纵打印位置
    posPrinter.setPageModeHorizontalPosition(0);
    posPrinter.setPageModeVerticalPosition(0);
    // 传送要打印的数据(图像)
    posPrinter.printBitmap(buffer.getInt(0), imagePath, 384, POSPrinterConst.PTR_BM_LEFT);

    // 指定横/纵打印位置
    posPrinter.setPageModeHorizontalPosition(100);
    posPrinter.setPageModeVerticalPosition(100);
    // 传送要打印的数据(文本)
    posPrinter.printNormar(POSPrinterConst.PTR_S_RECEIPT, "Print Data\n");

    // 开始打印
    posPrinter.setPageModePrint(POSPrinterConst.PTR_PM_PAGE_NORMAL);
}
catch(JposException e)
{
    e.printStackTrace();
}
```


6-2-22 transactionPrint()

使用Transaction Mode打印。

[Syntax]

public void transactionPrint(int station, int control) throws JposException

[Parameters]

- int station : 固定值 PTR_S_RECEIPT
- int control : Transaction Mode(“5-4 Transaction Print” 参考)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    // 开始Transaction mode
    posPrinter.transactionPrint(POSPrinterConst.PTR_TP_TRANSACTION);

    // 传送要打印的数据(图像)
    posPrinter.printBitmap(buffer.getInt(0), imagePath, 384, POSPrinterConst.PTR_BM_LEFT);
    // 传送要打印的数据(文本)
    posPrinter.printNormar(POSPrinterConst.PTR_S_RECEIPT, "Print Data\n");

    // Transaction mode结束(开始打印)
    posPrinter.transactionPrint(POSPrinterConst.PTR_TP_NORMAL);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-23 displayString()

打印BCD-3000内的文本。
应将BCD-3000连接至SRP-Q300 DK port。

[Syntax]

public void displayString(String data) throws JposException

[Parameters]

- String data : BCD-3000内要打印的文本数据。

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SRP-Q300");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.displayString("BIXOLON Customer Display 3000");
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-24 cleanScreen()

将BCD-3000画面清屏。

应将BCD-3000连接至SRP-Q300 DK port。

[Syntax]

public void cleanScreen() throws JposException

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SRP-Q300");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.displayString("BIXOLON Customer Display 3000");
    posPrinter.cleanScreen();
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-25 storeImageFile()

在BCD-3000图像缓冲器内保存图像数据。
应将BCD-3000连接至SRP-Q300 DK port。

[Syntax]

public void storeImageFile(String filename, int width, int height, int imageNumber) throws JposException

[Parameters]

- String filename : 保存图像文本的路径。
- int width : 保存图像的宽度。(1 ~ 160)
- int height : 指定图像的高度。(1 ~ 32)
- int imageNumber : 指定要保存的图像数据号码。(1 ~ 5)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SRP-Q300");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.storeImageFile(imagePath, 160, 32, 1);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-26 displayImage()

保存BCD-3000图像缓冲器内保存的图像。
应将BCD-3000连接至SRP-Q300 DK port。

[Syntax]

public void displayImage(int imageNumber, int xPos, int yPos) throws JposException

[Parameters]

- int imageNumber : 指定要打印的图像数据号码。(1 ~ 5)
- int xPos : 输入要打印图像的X坐标。(0 ~ 159)
- int yPos : 输入要打印图像的Y坐标。(0 ~ 31)
 - ※ 图像超出要打印的范围时不显示图像。

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SRP-Q300");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.storeImageFile(imagePath, 160, 32, 1);
    posPrinter.displayImage(1, 0, 0);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-2-27 clearImage()

删除BCD-3000图像缓冲器内保存的图像。
 应将BCD-3000连接至SRP-Q300 DK port。

[Syntax]

public void clearImage(boolean isAll, int imageNumber) throws JposException

[Parameters]

- boolean isAll : 选择图像缓冲器清屏模式。

数值	说明
true	删除图像缓冲器的所有图像。
false	只删除指定号码的图像。imageNumber内应输入要删除的图像号码。

- int imageNumber : 指定要删除的图像数据号码。(1 ~ 5)

[Example]

```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;
import com.bxl.BXLConst;

try
{
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SRP-Q300");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    posPrinter.storeImageFile(imagePath, 160, 32, 1);
    posPrinter.displayImage(1, 0, 0);
    posPrinter.clearImage(false, 1);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-3 MSR Class

- 用于控制打印机MSR(Magnetic Stripe Reader)的Class。可以使用相应Class从MSR调入Card Track信息。执行特定功能过程中出错时发生JposException。
 (“5-1 JposException” 参考)

6-3-1 open()

开始使用MSR，包含分配内存等初始化作业。
务必先行执行，以调入Claim以上的Method。
无法打开未通过BXLConfigLoader Class保存的装置。

[Syntax]

void open(String logicalDeviceName) throws JposException

[Parameters]

- String logicalDeviceName : 要打开装置的名称

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataListener;

try
{
    MSR msr = new MSR();
    msr.open("SPP-R200III");
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-3-2 claim()

尝试实际打开Device信息内包含的Port，包含初始化分配内存等部分初始化作业。
务必先行执行，以激活Device使用。

[Syntax]

void claim(int timeout) throws JposException

[Parameters]

- int timeout : 尝试打开本参数内指定的时间动因Port。

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataListener;

try
{
    MSR msr = new MSR();
    msr.open("SPP-R200III");
    msr.claim(3000);
}
catch(JposException e)
{
    e.printStackTrace();
}
```


6-3-3 setDeviceEnabled()

包含是否使用装置。

DeviceEnabled值失效时，可能无法使用功能。

[Syntax]

void setDeviceEnabled(boolean deviceEnabled) throws JposException

[Parameters]

- boolean deviceEnabled：输入装置激活状态。

值	说明
true	激活
false	失效

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataListener;

try
{
    MSR msr = new MSR();
    msr.open("SPP-R200III");
    msr.claim(3000);
    msr.setDeviceEnabled(true);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-3-4 release()

采用物理性方法结束使用Claim的Device的Port。
可以执行解除内存等作业。

[Syntax]

void release() throws JposException

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataListener;

try
{
    msr.release();
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-3-5 close()

结束使用Open的Device。
可以部分执行解除内存等作业。

[Syntax]

void close() throws JposException

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataListener;

try
{
    msr.close();
    msr.setDeviceEnabled(false);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-3-6 setAutoDisable()

读取MSR后自动使MSR装置失效。

AutoDisable为true时，接收MSR数据后将DeviceEnabled变更为false。

[Syntax]

void setAutoDisable(boolean autoDisable) throws JposException

[Parameters]

- boolean autoDisable : 指定是否Auto Disable。

值	说明
true	激活
false	失效

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataListener;

try
{
    MSR msr = new MSR();
    msr.open("SPP-R200III");
    msr.claim(3000);
    msr.setDeviceEnabled(true);

    msr.setAutoDisable(true);
    // msr.setAutoDisable(false);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-3-7 setDataEventEnabled()

指定读取MSR后是否通过Data Event接收Track信息。

[Syntax]

void setDataEventEnabled(boolean dataEventEnabled) throws JposException

[Parameters]

- boolean dataEventEnabled : 指定是否使用Data Event。

值	说明
true	使用Event ("5-2-4 Data Event" 参考)
false	不使用Event

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataListener;

try
{
    MSR msr = new MSR();
    msr.open("SPP-R200III");
    msr.claim(3000);
    msr.setDeviceEnabled(true);

    msr.addDataListener(this);
    msr.setDataEventEnabled(true);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-3-8 setDataEncryptionAlgorithm()

指定MSR Track数据是否加密。

[Syntax]

void setDataEncryptionAlgorithm(int encryptAlgorithm) throws JposException

[Parameters]

- int encryptAlgorithm : 指定Track是否使用加密。(“5-10 MSR 加密 参考”)

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataListener;

try
{
    MSR msr = new MSR();
    msr.open("SPP-R200III");
    msr.claim(3000);
    msr.setDeviceEnabled(true);

    msr.setDataEncryptionAlgorithm(MSRConst.MSR_DE_NONE);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-3-9 getTrack1Data()

调入最近读取MSR Card的Track1 Data。

[Syntax]

byte[] getTrack1Data() throws JposException

[Return Values]

※ MSR Card的Track1 Data

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataListener;

try
{
    MSR msr = new MSR();
    msr.open("SPP-R200III");
    msr.claim(3000);
    msr.setDeviceEnabled(true);

    byte[] track1 = msr.getTrack1Data();
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-3-10 getTrack2Data()

调入最近读取MSR Card的Track2 Data。

[Syntax]

byte[] getTrack2Data() throws JposException

[Return Values]

※ MSR Card的Track2 Data

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataListener;

try
{
    MSR msr = new MSR();
    msr.open("SPP-R200III");
    msr.claim(3000);
    msr.setDeviceEnabled(true);

    byte[] track2 = msr.getTrack2Data();
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-3-11 getTrack1Data()

调入最近读取MSR Card的Track3 Data。

[Syntax]

byte[] getTrack3Data() throws JposException

[Return Values]

※ MSR Card的Track3 Data

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataListener;

try
{
    MSR msr = new MSR();
    msr.open("SPP-R200III");
    msr.claim(3000);
    msr.setDeviceEnabled(true);

    byte[] track3 = msr.getTrack3Data();
}
catch(JposException e)
{
    e.printStackTrace();
}
```


6-4 SmartCardRW Class

- 用于控制打印机SCR(Smart Card Reader)的Class。
执行特定功能过程中出错时发生 JposException。(“5-1 JposException” 参考)

6-4-1 open()

开始使用SmartCardRW，包含分配内存等初始化作业。
务必先行执行，以调入Claim以上的Method。
无法打开未通过BXLConfigLoader Class保存的装置。

[Syntax]

void open(String logicalDeviceName) throws JposException

[Parameters]

- String logicalDeviceName : 要Open的装置名称

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try
{
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R200III");
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-4-2 claim()

尝试实际打开Device信息内包含的Port，包含初始化分配内存等部分初始化作业。
务必先行执行，以激活使用Device。

[Syntax]

void claim(int timeout) throws JposException

[Parameters]

- int timeout : 尝试打开本参数内指定的时间动因Port。

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try
{
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R200III");
    smartCardRW.claim(3000);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-4-3 setDeviceEnabled()

包含是否使用装置。

DeviceEnabled值失效时，可能无法使用功能。

[Syntax]

void setDeviceEnabled(boolean deviceEnabled) throws JposException

[Parameters]

- boolean deviceEnabled：输入装置激活状态。

值	说明
true	激活
false	失效

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try
{
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R200III");
    smartCardRW.claim(3000);
    SmartCardRW.setDeviceEnabled(true);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-4-4 release()

采用物理性的方法结束使用Claim的Device的Port。

可以执行解除内存等作业。

[Syntax]

void release() throws JposException

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try
{
    smartCardRW.release();
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-4-5 close()

结束使用Open的Device。
可以部分执行解除内存等作业。

[Syntax]

void close() throws JposException

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try
{
    smartCardRW.close();
    smartCardRW.setDeviceEnabled(false);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-4-6 setSCSlot()

选择用于通信的卡插槽。

[Syntax]

void setSCSlot() throws JposException

[Parameters]

- int scSlot : 选择卡插槽。

值	说明
0x1000	Smart Card
0x0100	SAM1
0x0010	SAM2

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try
{
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R200III");
    smartCardRW.claim(3000);
    SmartCardRW.setDeviceEnabled(true);

    smartCardRW.setSCSlot(0x01 << (Integer.SIZE - 1)); // Smart Card
    smartCardRW.setSCSlot(0x01 << (Integer.SIZE - 2)); // SAM1
    smartCardRW.setSCSlot(0x01 << (Integer.SIZE - 3)); // SAM2
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-4-7 setIsoEmvMode()

选择ISO/EMV模式。

[Syntax]

void setIsoEmvMode(int isoEmvMode) throws JposException

[Parameters]

- int isoEmvMode : 选择模式。（“5-11 SCR模式”参考）

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try
{
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R200III");
    smartCardRW.claim(3000);
    SmartCardRW.setDeviceEnabled(true);

    smartCardRW.setIsoEmvMode(SmartCardRWConst.SC_CMODE_EMV); // EMV Mode
    smartCardRW.setIsoEmvMode(SmartCardRWConst.SC_CMODE_ISO); // ISO Mode
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-4-8 readData()

读/写Smart Card chip内的数据。

[Syntax]

void readData(int action, int[] count, String[] data) throws JposException

[Parameters]

- int action : 固定值 SmartCardRWConst.SC_READ_DATA
- int[] count : 响应数据尺寸
- String[] data : RW缓冲

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try
{
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R200III");
    smartCardRW.claim(3000);
    SmartCardRW.setDeviceEnabled(true);

    String[] data = new String[] {
        new String(new byte[] {
            0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38,
            0x39, 0x3a, 0x3b, 0x3c, 0x3d, 0x3e, 0x3f, 0x40})
    };
    int[] count = new int[1];

    smartCardRW.readData(SmartCardRWConst.SC_READ_DATA, count, data);
    int rspSize = count[0];
    byte[] rspData = data[0].getBytes();
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-5 CashDrawer Class

- 用于控制CashDrawer的Class。执行特定功能中出错时发生JposException。
（“5-1 JposException” 参考）

6-5-1 open()

开始使用CashDrawer，包含分配内存等初始化作业。
务必先行执行，以调入Claim以上的Method。
无法开启未通过BXLConfigLoader Class保存的装置。

[Syntax]

void open(String logicalDeviceName) throws JposException

[Parameters]

- String logicalDeviceName : 要Open的装置名称

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try
{
    CashDrawer cashDrawer = new CashDrawer();
    cashDrawer.open("SPP-R200III");
}
catch(JposException e)
{
    e.printStackTrace();
}
```


6-5-2 claim()

尝试实际打开Device信息内包含的Port，包含初始化分配内存等部分初始化作业。
务必先行执行，以激活Device使用。

[Syntax]

void claim(int timeout) throws JposException

[Parameters]

- int timeout : 尝试打开本参数内指定的时间动因Port。

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try
{
    CashDrawer cashDrawer = new CashDrawer();
    cashDrawer.open("SPP-R200III");
    cashDrawer.claim(3000);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-5-3 setDeviceEnabled()

包含是否使用装置。

DeviceEnabled值失效时，可能无法使用功能。

[Syntax]

void setDeviceEnabled(boolean deviceEnabled) throws JposException

[Parameters]

- boolean deviceEnabled：输入装置激活状态。

值	说明
true	激活
false	失效

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try
{
    CashDrawer cashDrawer = new CashDrawer();
    cashDrawer.open("SPP-R200III");
    cashDrawer.claim(3000);
    cashDrawer.setDeviceEnabled(true);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-5-4 release()

以物理性方法结束使用Claim的Device的Port。

可以执行解除内存等作业。

[Syntax]

void release() throws JposException

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try
{
    cashDrawer.release();
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-5-5 close()

结束使用Open的Device。
可以部分执行解除内存等作业。

[Syntax]

void close() throws JposException

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try
{
    cashDrawer.close();
    cashDrawer.setDeviceEnabled(false);
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-5-6 openDrawer()

打开Cash Drawer。

[Syntax]

void openDrawer() throws JposException

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try
{
    CashDrawer cashDrawer = new CashDrawer();
    cashDrawer.open("SPP-R200III");
    cashDrawer.claim(3000);
    cashDrawer.setDeviceEnabled(true);

    cashDrawer.openDrawer();
}
catch(JposException e)
{
    e.printStackTrace();
}
```

6-5-7 getDrawerOpened()

返回Cash Drawer Open状态。

[Syntax]

boolean getDrawerOpened() throws JposException

[Return Values]

值	说明
true	Cash drawer is opened.
false	Cash drawer is closed.

[Example]

```
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try
{
    CashDrawer cashDrawer = new CashDrawer();
    cashDrawer.open("SPP-R200III");
    cashDrawer.claim(3000);
    cashDrawer.setDeviceEnabled(true);

    if(cashDrawer.getDrawerOpened())
    {
        // opened
    }
    else
    {
        // closed
    }
}
catch(JposException e)
{
    e.printStackTrace();
}
```

7. 使用举例

7-1 打印文本

```

import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try
{
    // BXLConfigLoader生成/设置文件open
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
    bxlConfigLoader.openFile();
}
catch(JposException e)
{
    e.printStackTrace();
    bxlConfigLoader.newFile();
}

try
{
    // 追加装置信息
    bxlConfigLoader.addEntry("SPP-R200III",
        BXLConfigLoader.DEVICE_CATEGORY_POS_PRINTER,
        BXLConfigLoader.PRODUCT_NAME_SPP_R200III,
        BXLConfigLoader.DEVICE_BUS_BLUETOOTH,
        "74:F0:7D:E4:11:AF");

    // 保存设置文件
    bxlConfigLoader.saveFile();

    // 打印机Open/Claim/DeviceEnabled
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    // 打印文本
    posPrinter.printNormal(POSPrinterConst.PTR_S_RECEIPT, "Print Data\n");
    posPrinter.printNormal(POSPrinterConst.PTR_S_RECEIPT,
        EscapeSequence.getString(7) + "www.bixelon.com\n");

    posPrinter.close();
}
catch(JposException e)
{
    e.printStackTrace();
}

```

7-2 打印图像

```

import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try
{
    // BXLConfigLoader生成/设置文件open
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
    bxlConfigLoader.openFile();
}
catch(JposException e)
{
    e.printStackTrace();
    bxlConfigLoader.newFile();
}

try
{
    // 追加装置信息
    bxlConfigLoader.addEntry("SPP-R200III",
                            BXLConfigLoader.DEVICE_CATEGORY_POS_PRINTER,
                            BXLConfigLoader.PRODUCT_NAME_SPP_R200III,
                            BXLConfigLoader.DEVICE_BUS_BLUETOOTH,
                            "74:F0:7D:E4:11:AF");

    // 保存设置文件
    bxlConfigLoader.saveFile();

    // 打印机Open/Claim/DeviceEnabled
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    // 打印图像
    ByteBuffer buffer = ByteBuffer.allocate(4);
    buffer.put((byte) POSPrinterConst.PTR_S_RECEIPT);
    buffer.put((byte) 80); // brightness
    buffer.put((byte) 0x01); // compress
    buffer.put((byte) 0x00);

    posPrinter.printBitmap(buffer.getInt(0),
                           "/storage/emulated/0/kicc/sample_image.png",
                           384,
                           POSPrinterConst.PTR_BM_LEFT);

    posPrinter.close();
}
catch(JposException e)
{
    e.printStackTrace();
}

```

7-3 打印PDF文件

```

import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try
{
    // BXLConfigLoader生成/设置文件open
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
    bxlConfigLoader.openFile();
}
catch(JposException e)
{
    e.printStackTrace();
    bxlConfigLoader.newFile();
}

try
{
    // 追加装置信息
    bxlConfigLoader.addEntry("SPP-R200III",
                            BXLConfigLoader.DEVICE_CATEGORY_POS_PRINTER,
                            BXLConfigLoader.PRODUCT_NAME_SPP_R200III,
                            BXLConfigLoader.DEVICE_BUS_BLUETOOTH,
                            "74:F0:7D:E4:11:AF");

    // 保存设置文件
    bxlConfigLoader.saveFile();

    // 打印机Open/Claim/DeviceEnabled
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    // PDF打印
    posPrinter.printPDF(POSPrinterConst.PTR_S_RECEIPT,
                        "/storage/emulated/0/sample.pdf",
                        384,
                        POSPrinterConst.PTR_PDF_CENTER,
                        0,
                        2,
                        80);

    posPrinter.close();
}
catch(JposException e)
{
    e.printStackTrace();
}

```

7-4 Page mode打印

```

import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try
{
    // BXLConfigLoader生成/设置文件open
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
    bxlConfigLoader.openFile();
}
catch(JposException e)
{
    e.printStackTrace();
    bxlConfigLoader.newFile();
}

try
{
    // 追加装置信息
    bxlConfigLoader.addEntry("SPP-R200III",
                            BXLConfigLoader.DEVICE_CATEGORY_POS_PRINTER,
                            BXLConfigLoader.PRODUCT_NAME_SPP_R200III,
                            BXLConfigLoader.DEVICE_BUS_BLUETOOTH,
                            "74:F0:7D:E4:11:AF");

    // 保存设置文件
    bxlConfigLoader.saveFile();

    // 打印机Open/Claim/DeviceEnabled
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(0);
    posPrinter.setDeviceEnabled(true);

    ByteBuffer buffer = ByteBuffer.allocate(4);
    buffer.put((byte) POSPrinterConst.PTR_S_RECEIPT);
    buffer.put((byte) 80); // brightness
    buffer.put((byte) 0x01); // compress
    buffer.put((byte) 0x00);

    // Page mode设置打印区域
    posPrinter.setPageModePrintArea("0,0,384,1200");
    // Page mode设置打印方向
    posPrinter.setPageModePrintDirection(POSPrinterConst.PTR_PD_LEFT_TO_RIGHT);
    // Page mode转换
    posPrinter.pageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);

    // 设置打印位置：横/纵
    posPrinter.setPageModeHorizontalPosition(0);
    posPrinter.setPageModeVerticalPosition(0);
    posPrinter.printBitmap(buffer.getInt(0),
                            "/storage/emulated/0/kicc/sample_image.png",
                            384,
                            POSPrinterConst.PTR_BM_LEFT);
}

```



```
posPrinter.setPageModeHorizontalPosition(130);  
posPrinter.setPageModeVerticalPosition(200);  
posPrinter.printNormal(POSPrinterConst.PTR_S_RECEIPT,  
    EscapeSequence.getString(0) + // Normal  
    EscapeSequence.getString(7) + // Bold  
    EscapeSequence.getString(18) + //Scale_2_time_horizontally  
    EscapeSequence.getString(26) + // Scale_2_time_vertically  
    "www.bixolon.com\n");
```

```
posPrinter.setPageModeHorizontalPosition(340);  
posPrinter.setPageModeVerticalPosition(550);  
String data = "www.bixolon.com\n";  
posPrinter.printBarCode(POSPrinterConst.PTR_S_RECEIPT,  
    data,  
    POSPrinterConst.PTR_BCS_QRCODE,  
    0, 6/*1to8*/,  
    POSPrinterConst.PTR_BC_LEFT,  
    POSPrinterConst.PTR_BC_TEXT_NONE);
```

```
// 开始页面模式打印
```

```
posPrinter.pageModePrint(POSPrinterConst.PTR_PM_NORMAL);
```

```
posPrinter.close();
```

```
}  
catch(JposException e)  
{  
    e.printStackTrace();  
}
```

Copyright Owned by BIXOLON

This User Manual and product are protected under copyright law.

It is strictly prohibited to copy, duplicate, translate or convert into electronic form the whole or any part of the manual and product without the prior written approval of BIXOLON.

BIXOLON maintains ongoing efforts to enhance and upgrade the functions and quality of all our products. In the following, product specifications and/or user manual content may be changed without prior notice.

The BIXOLON logo is the registered trademark of BIXOLON.

Warning - U.S.A

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Notice - Canada

This Apparatus complies with class "A" limits for radio interference as specified in the Canadian department of communications radio interference regulations. Get appareil est conforme aux normes class "A" d'interference radio tel que specifier par ministre canadien des communications dans les reglements d'interference radio.

Caution

Some semiconductor devices are easily damaged by static electricity. You should turn the printer "OFF", before you connect or remove the cables on the rear side, in order to guard the printer against the static electricity. If the printer is damaged by the static electricity, you should turn the printer "OFF".

Copyright © BIXOLON Co., Ltd. 2017. All rights reserved.

