API 레퍼런스 가이드

BXL SDK for UPOS Compliant Android

Rev. 2.05
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 / 383
SRP-F310II / F312II / F313II
SRP-S300
SRP-Q300 / Q302
SRP-QE300 / QE302
SRP-E300 / E302
STP-103III
BK3-3

http://www.bixolon.co.kr
목차

1. 매뉴얼 안내 ............................................................................................................... 5

2. 지원 OS 및 인터페이스 ....................................................................................... 5
   2-1 운영체제 ........................................................................................................... 5
   2-2 지원 장치 및 인터페이스 ............................................................................... 5

3. 개발 환경 .............................................................................................................. 6
   3-1 환경 설정 ......................................................................................................... 6
   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 .............................................................. 11

4. 패키지 콘텐츠 ....................................................................................................... 12
   4-1 매뉴얼 ............................................................................................................ 12
   4-2 라이브러리 ...................................................................................................... 12
   4-3 샘플 소스 코드 ............................................................................................... 12

5. 상수값(Defines) .................................................................................................. 13
   5-1 JposException .............................................................................................. 13
   5-2 Event .............................................................................................................. 14
      5-2-1 StatusUpdate Event ................................................................................. 14
      5-2-2 Error Event ............................................................................................. 14
      5-2-3 OutputComplete Event .......................................................................... 14
      5-2-4 Data Event ............................................................................................... 14
      5-2-5 DirectIO Event ......................................................................................... 14
   5-3 EscapeSequence ............................................................................................ 18
   5-4 Transaction Print ........................................................................................... 19
   5-5 Alignment ....................................................................................................... 20
   5-6 바코드 타입 .................................................................................................... 21
   5-7 바코드 텍스트 위치 ....................................................................................... 22
   5-8 장치 모델명 .................................................................................................. 22
   5-9 페이지 모드 인쇄 방향 ................................................................................ 23
   5-10 MSR 암호화 ................................................................................................ 23
   5-11 SCR 모드 ...................................................................................................... 23
   5-12 Character Set ............................................................................................... 24

6. 각 클래스별 기능 정리 ......................................................................................... 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

6-2-1 open() ................................................................. 30
6-2-2 claim() ................................................................. 30
6-2-3 setDeviceEnabled() .................................................. 31
6-2-4 release() .............................................................. 32
6-2-5 close() ................................................................. 32
6-2-6 checkHealth() ......................................................... 33
6-2-7 setAsyncMode() ....................................................... 34
6-2-8 setCharacterSet() ..................................................... 35
6-2-9 setCharacterSetEncoding() ........................................ 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
6-2-23 displayString() ..................................................... 50
6-2-24 cleanScreen() ....................................................... 51
6-2-25 storeImageFile() ................................................... 52
6-2-26 displayImage() ...................................................... 53
6-2-27 clearImage() ......................................................... 54
6-2-28 directIO() ........................................................... 55
6-2-29 markFeed() .......................................................... 56

6-3 MSR Class ................................................................. 57
6-3-1 open() ................................................................. 57
6-3-2 claim() ................................................................. 58
6-3-3 setDeviceEnabled() .................................................. 59
6-3-4 release() .............................................................. 60
6-3-5 close() ................................................................. 60
6-3-6 setAutoDisable() ..................................................... 61
6-3-7 setDataEventEnabled() ............................................. 62
6-3-8 setDataEncryptionAlgorithm() .................................. 63
6-3-9 getTrack1Data() ..................................................... 64
6-3-10 getTrack2Data() ................................................... 65
6-3-11 getTrack1Data() ................................................... 66

6-4 SmartCardRW Class .................................................. 67
6-4-1 open() ................................................................. 67
6-4-2 claim() ................................................................. 68
6-4-3 setDeviceEnabled() .................................................. 69
6-4-4 release() .............................................................. 69
6-4-5 close() ................................................................. 70
6-4-6 setSCSlot() .......................................................... 71
6-4-7 setIsoEmvMode() .................................................... 72
6-4-8 beginInsertion() .................................................... 73
6-4-9 endInsertion() ....................................................... 74
6-4-10 beginRemoval() ............................................................... 75
6-4-11 endRemoval() ................................................................. 76
6-4-12 readData() ..................................................................... 77
6-5 CashDrawer Class ................................................................. 78
  6-5-1 open() ........................................................................ 78
  6-5-2 claim() ......................................................................... 79
  6-5-3 setDeviceEnabled() ......................................................... 80
  6-5-4 release() ...................................................................... 80
  6-5-5 close() ......................................................................... 81
  6-5-6 openDrawer() ................................................................. 81
  6-5-7 getDrawerOpened() ......................................................... 82

7. 사용 예제 ................................................................................ 83
  7-1 텍스트 인쇄 ...................................................................... 83
  7-2 이미지 인쇄 ..................................................................... 84
  7-3 PDF 파일 인쇄 .................................................................. 85
  7-4 Page mode 인쇄 ............................................................... 86
1. 매뉴얼 안내

- 이 SDK 매뉴얼에는 Android용 응용 프로그램 개발에 필요한 Library에 대한 내용이 기술되어 있습니다.
- SDK의 사용방법, 사양, 제약에 대해 설명합니다.

2. 지원 OS 및 인터페이스

2-1 운영체제
- 이 소프트웨어는 아래 운영체제를 지원합니다.
  - Android 4.0 (Ice Cream Sandwich) 이상

2-2 지원 장치 및 인터페이스

<table>
<thead>
<tr>
<th>Models</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPP-R200II</td>
<td>Bluetooth / WLAN / USB</td>
</tr>
<tr>
<td>SPP-R200III</td>
<td>Bluetooth / WLAN / USB</td>
</tr>
<tr>
<td>SPP-R210</td>
<td>Bluetooth / WLAN / USB</td>
</tr>
<tr>
<td>SPP-R220</td>
<td>Bluetooth / BLE / WLAN / USB</td>
</tr>
<tr>
<td>SPP-R300</td>
<td>Bluetooth / WLAN / USB</td>
</tr>
<tr>
<td>SPP-R310</td>
<td>Bluetooth / WLAN / USB</td>
</tr>
<tr>
<td>SPP-R400</td>
<td>Bluetooth / BLE / WLAN / USB</td>
</tr>
<tr>
<td>SPP-R410</td>
<td>Bluetooth / BLE / WLAN / USB</td>
</tr>
<tr>
<td>SPP-R418</td>
<td>Bluetooth / BLE / WLAN / USB</td>
</tr>
<tr>
<td>SRP-350plusIII</td>
<td>Bluetooth / WLAN / Ethernet / USB</td>
</tr>
<tr>
<td>SRP-352plusIII</td>
<td>Bluetooth / WLAN / Ethernet / USB</td>
</tr>
<tr>
<td>SRP-350III</td>
<td>Ethernet / USB</td>
</tr>
<tr>
<td>SRP-352III</td>
<td>Ethernet / USB</td>
</tr>
<tr>
<td>SRP-F310II</td>
<td>Bluetooth / WLAN / Ethernet / USB</td>
</tr>
<tr>
<td>SRP-F312II</td>
<td>Bluetooth / WLAN / Ethernet / USB</td>
</tr>
<tr>
<td>SRP-F313II</td>
<td>Bluetooth / WLAN / Ethernet / USB</td>
</tr>
<tr>
<td>SRP-380</td>
<td>Bluetooth / WLAN / Ethernet / USB</td>
</tr>
<tr>
<td>SRP-382</td>
<td>Bluetooth / WLAN / Ethernet / USB</td>
</tr>
<tr>
<td>SRP-383</td>
<td>Bluetooth / WLAN / Ethernet / USB</td>
</tr>
<tr>
<td>SRP-330II</td>
<td>Ethernet / USB</td>
</tr>
<tr>
<td>SRP-332II</td>
<td>Ethernet / USB</td>
</tr>
<tr>
<td>SRP-S300</td>
<td>Bluetooth / WLAN / Ethernet / USB</td>
</tr>
<tr>
<td>SRP-340II</td>
<td>Ethernet / USB</td>
</tr>
<tr>
<td>SRP-342II</td>
<td>Ethernet / USB</td>
</tr>
<tr>
<td>STP-103III</td>
<td>USB</td>
</tr>
<tr>
<td>SRP-275III</td>
<td>Ethernet / USB</td>
</tr>
<tr>
<td>SRP-Q300</td>
<td>Bluetooth / WLAN / Ethernet / USB</td>
</tr>
<tr>
<td>SRP-Q302</td>
<td>Bluetooth / WLAN / Ethernet / USB</td>
</tr>
<tr>
<td>SRP-QE300</td>
<td>Ethernet / USB</td>
</tr>
<tr>
<td>SRP-QE302</td>
<td>Ethernet / USB</td>
</tr>
<tr>
<td>SRP-E300</td>
<td>Ethernet / USB</td>
</tr>
<tr>
<td>SRP-E302</td>
<td>Ethernet / USB</td>
</tr>
<tr>
<td>BK3-3</td>
<td>USB</td>
</tr>
</tbody>
</table>

※ BLE: Bluetooth Low Energy
3. 개발 환경

3-1 환경 설정
- Java Development Kit (JDK) 7
- Eclipse
- Android SDK Tools

3-2 Android 장치 연결
- 아래 스クリーン샷은 Nexus 5에서 캡쳐한것입니다. 스クリーン샷과 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. ㈜빅솔론의 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, 당사 Net Configuration Tool로 최초 한번은 설정을 해야 됩니다. Net Configuration Tool은 (주)빅솔론의 웹사이트에서 다운로드 가능합니다. (설정과 관련된 자세한 사항은 Net Configuration Tool 매뉴얼 참고)

2. Settings를 선택합니다.

3. Wi-Fi가 켜져 있어야 합니다.

4. (주)빅솔론의 프린터가 연결되어있는 네트워크와 동일한 네트워크에 연결합니다.

5. 안드로이드 장치를 프린터의 TCP/IP 포트에 연결하기 위해 추가 설정은 필요하지 않습니다.
3-2-3 Wi-Fi Direct

1. 안드로이드 장치는 OS 버전 4.0이상부터 Wi-Fi Direct로 주변장치 연결이 가능합니다.

2. 안드로이드 장치에 뷰익솔론의 특정 드라이버나 프린터 소프트웨어가 설치 되어 있을 필요는 없습니다.


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. 안드로이드 장치는 OS 버전 3.1 이상부터 USB 주변장치 연결이 가능합니다.

2. 안드로이드 장치에 주bcrypt의 특정 드라이버나 프린터 소프트웨어가 설치 되어있을 필요는 없습니다.

3. 필요한 USB 케이블은 안드로이드 장치에 따라 달라질 수 있습니다. 사용하려는 안드로이드 장치가 USB 연결을 지원하는지 확인 후 알맞은 케이블을 사용하시기 바랍니다.

4. 주bcrypt의 프린터를 처음 연결하는 경우, 안드로이드 장치에 따라서 아래와 같은 메시지가 나타날 수 있습니다.
5. USB 주변장치를 연결하기 위해서는 아래 코드를 AndroidManifest.xml과 Sample로 제공되는 BXLTest내 res/xml/device_filter.xml에 입력해야 합니다.

[AndroidManifest.xml]

```xml
...<uses-feature android:name="android.hardware.usb.host"/>
...
<intent-filter>
  <action android:name="android.hardware.usb.action.USBDEVICE_ATTACHED"/>
</intent-filter>
<meta-data
  android:name="android.hardware.usb.action.USBDEVICE_ATTACHED"
  android:resource="@xml/device_filter"/>
```

[device_filter.xml]

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <usb-device
    class="7"
    protocol="2"
    subclass="1"/>
</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

<table>
<thead>
<tr>
<th>매뉴얼 위치/이름</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>docs/Manual_BXL SDK for Android_UPOS compliant API Reference Guide_english_Rev_x_xx</td>
<td>영문 매뉴얼</td>
</tr>
<tr>
<td>docs/Manual_BXL SDK for Android_UPOS compliant API Reference Guide_japan_Rev_x_xx</td>
<td>일본어 매뉴얼</td>
</tr>
<tr>
<td>docs/Manual_BXL SDK for Android_UPOS compliant API Reference Guide_korean_Rev_x_xx</td>
<td>한글 매뉴얼</td>
</tr>
<tr>
<td>docs/Manual_BXL SDK for Android_UPOS compliant API Reference Guide_chinese_Rev_x_xx</td>
<td>중국어 매뉴얼</td>
</tr>
</tbody>
</table>

4-2 라이브러리

<table>
<thead>
<tr>
<th>라이브러리 위치/이름</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>libs/bixolon_printer_Vxxx.jar</td>
<td>JavaPOS 서비스 컴포넌트 계층의 구현. 프린터 설정 라이브러리</td>
</tr>
<tr>
<td>libs/jpos11x-controls.jar</td>
<td>JavaPOS 장치 컨트롤과 서비스 인터페이스</td>
</tr>
<tr>
<td>libs/xerces.jar</td>
<td>apache.org XML 서비스구현 (JavaPOS device control에서 필요)</td>
</tr>
<tr>
<td>libs/icu4j-58_1.jar</td>
<td>아랍어/페르시아어 문자를 인쇄하기 위해 사용되는 라이브러리</td>
</tr>
<tr>
<td>libs/PDF/bixolon_pdf.jar</td>
<td>PDF 인쇄를 위한 라이브러리</td>
</tr>
<tr>
<td>libs/PDF/CPU type/libbxmlpdf.so</td>
<td>PDF 인쇄를 위한 native 라이브러리</td>
</tr>
<tr>
<td>libs/Image/opencv-2.4.11.jar</td>
<td>Image 인쇄를 위한 라이브러리</td>
</tr>
<tr>
<td>libs/Image/CPU type/libopencv_java.so</td>
<td>Image 인쇄를 위한 native 라이브러리</td>
</tr>
</tbody>
</table>

4-3 샘플 소스 코드

<table>
<thead>
<tr>
<th>샘플 위치/이름</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>samples/BixolonSample</td>
<td>프린터/MSR/SRC/CashDrawer 사용 샘플 애플리케이션</td>
</tr>
</tbody>
</table>

★ SDK 활용 이해를 높이기 위한 샘플 레퍼런스 클래스 제공
- BixolonPrinter.java(src\main\java\com\bixolon\sample\PrinterControl)
5 상수값(Defines)

5-1 JposException
**Method**에서 특정 기능 수행 중 에러 발생시 JposException 예외를 발생 시킵니다. 에외 발생 시 "printStackTrace" 함수를 통해 에러 내용을 확인할 수 있습니다.

[Example]
```java
import jpos.JposConst;

try {
    ...
} catch(JposException e) {
    // Error
    e.printStackTrace();
}
```
5-2 Event

각 Event는 JposConst, POSPrinterConst 클래스에 정의되어 있습니다.

5-2-1 StatusUpdate Event

프린터 상태가 변경될 때마다 StatusUpdate 이벤트가 발생합니다.

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPOS_SUE_POWER_ONLINE</td>
<td>2001</td>
<td>Printer Power on</td>
</tr>
<tr>
<td>JPOS_SUE_POWER_OFF_OFFLINE</td>
<td>2004</td>
<td>Printer Power off</td>
</tr>
<tr>
<td>PTR_SUE_COVER_OPEN</td>
<td>11</td>
<td>Cover Open</td>
</tr>
<tr>
<td>PTR_SUE_COVER_OK</td>
<td>12</td>
<td>Cover OK</td>
</tr>
<tr>
<td>PTR_SUE_REC_EMPTY</td>
<td>24</td>
<td>Receipt Paper Empty</td>
</tr>
<tr>
<td>PTR_SUE_REC_NEAREMPTY</td>
<td>25</td>
<td>Receipt Paper Near Empty</td>
</tr>
<tr>
<td>PTR_SUE_REC_PAPEROK</td>
<td>26</td>
<td>Receipt Paper OK</td>
</tr>
<tr>
<td>PTR_SUE_IDLE</td>
<td>1001</td>
<td>Printer Idle</td>
</tr>
<tr>
<td>PTR_SUE_OFF_LINE</td>
<td>53</td>
<td>Printer off-line</td>
</tr>
<tr>
<td>PTR_SUE_ON_LINE</td>
<td>54</td>
<td>Printer on-line</td>
</tr>
<tr>
<td>PTR_SUE_BAT_OK</td>
<td>55</td>
<td>Printer battery normal</td>
</tr>
<tr>
<td>PTR_SUE_BAT_LOW</td>
<td>56</td>
<td>Printer battery low</td>
</tr>
</tbody>
</table>

5-2-2 Error Event

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPOS_EPTR_COVER_OPEN</td>
<td>201</td>
<td>Cover Open</td>
</tr>
<tr>
<td>JPOS_EPTR_REC_EMPTY</td>
<td>203</td>
<td>Paper Empty</td>
</tr>
<tr>
<td>JPOS_EPTR_OFF_LINE</td>
<td>217</td>
<td>Printer off-line</td>
</tr>
</tbody>
</table>

5-2-3 OutputComplete Event

인쇄 완료 이벤트를 발생시킵니다. 단, Async 모드로 사용해야 합니다.

5-2-4 Data Event

MSR Track 정보 데이터를 수신 합니다.

5-2-5 DirectIO Event

Direct I/O의 응답 데이터를 수신 합니다.
try {
    POSPrinter posPrinter = new POSPrinter(this);
    MSR msr = new MSR();
    posPrinter.addErrorListener(this);
    posPrinter.addStatusUpdateListener(this);
    posPrinter.addOutputCompleteListener(this);
    posPrinter.addDirectIOLe

@Override
public void directIOOccurred(DirectIOEvent directIOEvent) {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(MainActivity.this, new String((byte[]) directIOEvent.getObject(), 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

- 인쇄할 텍스트에 폰트, 굵기 등의 옵션을 지정하기 위한 값입니다.
  - 문자열 형식이며 인쇄 데이터 앞에 추가하여 사용합니다.

<table>
<thead>
<tr>
<th>정의</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>String ESCAPE_CHARACTERS = new String(new byte[] {0x1b, 0x7c})</td>
<td>Escape Characters</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;N&quot;</td>
<td>Normal</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;aM&quot;</td>
<td>Font A (12x24)</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;bM&quot;</td>
<td>Font B (9x17)</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;cM&quot;</td>
<td>Font C (9x24)</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;IA&quot;</td>
<td>Left justify</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;cA&quot;</td>
<td>Center</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;rA&quot;</td>
<td>Right justify</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;bC&quot;</td>
<td>Bold</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;IbC&quot;</td>
<td>Disabled bold</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;uC&quot;</td>
<td>Underline</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;luC&quot;</td>
<td>Disabled underline</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;rvC&quot;</td>
<td>Reverse video</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;lrvC&quot;</td>
<td>Disabled reverse video</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;1C&quot;</td>
<td>Single high and wide</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;2C&quot;</td>
<td>Double wide</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;3C&quot;</td>
<td>Double high</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;4C&quot;</td>
<td>Double high and wide</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;1hC&quot;</td>
<td>Scale 1 time horizontally</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;2hC&quot;</td>
<td>Scale 2 times horizontally</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;3hC&quot;</td>
<td>Scale 3 times horizontally</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;4hC&quot;</td>
<td>Scale 4 times horizontally</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;5hC&quot;</td>
<td>Scale 5 times horizontally</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;6hC&quot;</td>
<td>Scale 6 times horizontally</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;7hC&quot;</td>
<td>Scale 7 times horizontally</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;8hC&quot;</td>
<td>Scale 8 times horizontally</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;1vC&quot;</td>
<td>Scale 1 time vertically</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;2vC&quot;</td>
<td>Scale 2 times vertically</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;3vC&quot;</td>
<td>Scale 3 times vertically</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;4vC&quot;</td>
<td>Scale 4 times vertically</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;5vC&quot;</td>
<td>Scale 5 times vertically</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;6vC&quot;</td>
<td>Scale 6 times vertically</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;7vC&quot;</td>
<td>Scale 7 times vertically</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;8vC&quot;</td>
<td>Scale 8 times vertically</td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;[code]B&quot;</td>
<td>Print NV Image</td>
</tr>
<tr>
<td>[code] : 0 ~ 255(image code)</td>
<td></td>
</tr>
<tr>
<td>ESCAPE_CHARACTERS + &quot;[percentage]fP&quot;</td>
<td>Feed Cut</td>
</tr>
<tr>
<td>[percentage] : 100 전체절단</td>
<td></td>
</tr>
<tr>
<td>[percentage] : 0~90 부분절단</td>
<td></td>
</tr>
</tbody>
</table>
try {
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open(logicalDeviceName);
    posPrinter.claim(5000);
    posPrinter.setDeviceEnabled(true)

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

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

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

    // NV Image
    String data = ESCAPE_SEQUENCE + "0B"
    posPrinter.printNormal(POSPrinterConst.PTR_S_RECEIPT, data);

    // Feed cut
    String data = ESCAPE_SEQUENCE + "90fP"
    posPrinter.printNormal(POSPrinterConst.PTR_S_RECEIPT, data);
}

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

### 5-4 Transaction Print

- **Transaction mode** 설정 시 사용할 수 있는 값에 대한 정의입니다.

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR_TP_TRANSACTION</td>
<td>11</td>
<td>Buffer를 비어있는 상태로 초기화 하고 Transaction Mode를 시작합니다</td>
</tr>
<tr>
<td>PTR_TP_NORMAL</td>
<td>12</td>
<td>Transaction mode를 종료하고, Buffer에 쌓여 있는 Data를 출력합니다</td>
</tr>
</tbody>
</table>
5-5 Alignment

• 정렬을 지정할 때 필요한 값에 대한 정의입니다.

[For Barcode]

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR_BC_LEFT</td>
<td>-1</td>
<td>좌측 정렬</td>
</tr>
<tr>
<td>PTR_BC_CENTER</td>
<td>-2</td>
<td>중앙 정렬</td>
</tr>
<tr>
<td>PTR_BC_RIGHT</td>
<td>-3</td>
<td>우측 정렬</td>
</tr>
</tbody>
</table>

[For Image]

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR_BM_LEFT</td>
<td>-1</td>
<td>좌측 정렬</td>
</tr>
<tr>
<td>PTR_BM_CENTER</td>
<td>-2</td>
<td>중앙 정렬</td>
</tr>
<tr>
<td>PTR_BM_RIGHT</td>
<td>-3</td>
<td>우측 정렬</td>
</tr>
</tbody>
</table>

[For PDF]

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR_PDF_LEFT</td>
<td>-1</td>
<td>좌측 정렬</td>
</tr>
<tr>
<td>PTR_PDF_CENTER</td>
<td>-2</td>
<td>중앙 정렬</td>
</tr>
<tr>
<td>PTR_PDF_RIGHT</td>
<td>-3</td>
<td>우측 정렬</td>
</tr>
</tbody>
</table>
# 5-6 바코드 타입
- 바코드 출력 시 바코드 지정에 필요한 값에 대한 정의 입니다.

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR_BCS_UPCA</td>
<td>101</td>
<td>UPCA</td>
</tr>
<tr>
<td>PTR_BCS_UPCE</td>
<td>102</td>
<td>UPCE</td>
</tr>
<tr>
<td>PTR_BCS_JAN8</td>
<td>103</td>
<td>JAN8</td>
</tr>
<tr>
<td>PTR_BCS_EAN8</td>
<td>103</td>
<td>EAN8</td>
</tr>
<tr>
<td>PTR_BCS_JAN13</td>
<td>104</td>
<td>JAN13</td>
</tr>
<tr>
<td>PTR_BCS_EAN13</td>
<td>104</td>
<td>EAN13</td>
</tr>
<tr>
<td>PTR_BCS_TF</td>
<td>105</td>
<td>Standard(ordiscrete) 2 of 5</td>
</tr>
<tr>
<td>PTR_BCS_ITF</td>
<td>106</td>
<td>Interleaved 2 of 5</td>
</tr>
<tr>
<td>PTR_BCS_Codabar</td>
<td>107</td>
<td>Codabar</td>
</tr>
<tr>
<td>PTR_BCS_Code39</td>
<td>108</td>
<td>Code39</td>
</tr>
<tr>
<td>PTR_BCS_Code93</td>
<td>109</td>
<td>Code93</td>
</tr>
<tr>
<td>PTR_BCS_Code128</td>
<td>110</td>
<td>Code 128</td>
</tr>
<tr>
<td></td>
<td></td>
<td>※ Code128 의 Special Character</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Special Charaters Ascii Representation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code A {A}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code B {B}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code C {C}</td>
</tr>
<tr>
<td>PTR_BCS_UPCA_S</td>
<td>111</td>
<td>UPC-A with supplemental barcode</td>
</tr>
<tr>
<td>PTR_BCS_UPCE_S</td>
<td>112</td>
<td>UPC-E with supplemental barcode</td>
</tr>
<tr>
<td>PTR_BCS_UPCD1</td>
<td>113</td>
<td>UPC-D1</td>
</tr>
<tr>
<td>PTR_BCS_UPCD2</td>
<td>114</td>
<td>UPC-D2</td>
</tr>
<tr>
<td>PTR_BCS_UPCD3</td>
<td>115</td>
<td>UPC-D3</td>
</tr>
<tr>
<td>PTR_BCS_UPCD4</td>
<td>116</td>
<td>UPC-D4</td>
</tr>
<tr>
<td>PTR_BCS_UPCD5</td>
<td>117</td>
<td>UPC-D5</td>
</tr>
<tr>
<td>PTR_BCS_EAN8_S</td>
<td>118</td>
<td>EAN8 with supplemental barcode</td>
</tr>
<tr>
<td>PTR_BCS_EAN13_S</td>
<td>119</td>
<td>EAN13 with supplemental barcode</td>
</tr>
<tr>
<td>PTR_BCS_EAN128</td>
<td>120</td>
<td>EAN128</td>
</tr>
<tr>
<td>PTR_BCS_OCRA</td>
<td>121</td>
<td>OCR “A”</td>
</tr>
<tr>
<td>PTR_BCS_OCRA</td>
<td>122</td>
<td>OCR “B”</td>
</tr>
<tr>
<td>PTR_BCS_Code128_Parsed</td>
<td>123</td>
<td>Code 128 with parsing</td>
</tr>
<tr>
<td>PTR_BCS_GS1DATABAR</td>
<td>131</td>
<td>GS1 DataBar Omnidirectional</td>
</tr>
<tr>
<td>PTR_BCS_GS1DATABAR_E</td>
<td>132</td>
<td>GS1 DataBar Stacked Omnidirectional</td>
</tr>
<tr>
<td>PTR_BCS_GS1DATABAR_S</td>
<td>133</td>
<td>GS1 DataBar Expanded</td>
</tr>
<tr>
<td>PTR_BCS_GS1DATABAR_E_S</td>
<td>134</td>
<td>GS1 DataBar Expanded Stacked</td>
</tr>
<tr>
<td>PTR_BCS_PDF417</td>
<td>201</td>
<td>PDF 417</td>
</tr>
<tr>
<td>PTR_BCS_MAXICODE</td>
<td>202</td>
<td>MAXI Code</td>
</tr>
<tr>
<td>PTR_BCS_DATAMATRIX</td>
<td>203</td>
<td>Data Matrix</td>
</tr>
<tr>
<td>PTR_BCS_QRCODE</td>
<td>204</td>
<td>QR Code</td>
</tr>
<tr>
<td>PTR_BCS_UQRCODE</td>
<td>205</td>
<td>Micro QR Code</td>
</tr>
<tr>
<td>PTR_BCS_AZTEC</td>
<td>206</td>
<td>Aztec</td>
</tr>
<tr>
<td>PTR_BCS_UPDF417</td>
<td>207</td>
<td>Micro PDF 417</td>
</tr>
</tbody>
</table>
5-7 바코드 텍스트 위치
- 텍스트 출력을 지원하는 바코드 일 경우, 바코드 텍스트의 출력 여부 또는 위치를 지정 합니다.

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR_BC_TEXT_NONE</td>
<td>-11</td>
<td>Text 를 출력하지 않습니다. 바코드만 출력 합니다.</td>
</tr>
<tr>
<td>PTR_BC_TEXT_ABOVE</td>
<td>-12</td>
<td>Text 를 바코드의 상단에 출력 합니다.</td>
</tr>
<tr>
<td>PTR_BC_TEXT_BELOW</td>
<td>-13</td>
<td>Text 를 바코드의 하단에 출력합니다.</td>
</tr>
</tbody>
</table>

5-8 장치 모델명

<table>
<thead>
<tr>
<th>상수명</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODUCT_NAME_SPP_R210</td>
<td>SPP-R210</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_R220</td>
<td>SPP-R220</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_R200II</td>
<td>SPP-R200II</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_R200III</td>
<td>SPP-R200III</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_R300</td>
<td>SPP-R300</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_R310</td>
<td>SPP-R310</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_R400</td>
<td>SPP-R400</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_R410</td>
<td>SPP-R410</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_R418</td>
<td>SPP-R418</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_RPC300II</td>
<td>SPP-300II</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_RPC302II</td>
<td>SPP-302II</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_RPC310II</td>
<td>SPP-310II</td>
</tr>
<tr>
<td>PRODUCT_NAME_SPP_RPC312III</td>
<td>SPP-312III</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_350III</td>
<td>SRP-350III</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_352III</td>
<td>SRP-352III</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_330II</td>
<td>SRP-330II</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_332II</td>
<td>SRP-332II</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_340II</td>
<td>SRP-340II</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_342II</td>
<td>SRP-342II</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_350PLUSIII</td>
<td>SRP-350PLUSIII</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_352PLUSIII</td>
<td>SRP-352PLUSIII</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_380</td>
<td>SRP-380</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_382</td>
<td>SRP-382</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_383</td>
<td>SRP-383</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_S300</td>
<td>SRP-S300</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_Q300</td>
<td>SRP-Q300</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_Q302</td>
<td>SRP-Q302</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_F310II</td>
<td>SRP-F310II</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_F312II</td>
<td>SRP-F312II</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_F313II</td>
<td>SRP-F313II</td>
</tr>
<tr>
<td>PRODUCT_NAME_STP_103III</td>
<td>STP-103III</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_275III</td>
<td>SRP-275III</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_QE300</td>
<td>SRP-QE300</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_QE302</td>
<td>SRP-QE302</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_E300</td>
<td>SRP-E300</td>
</tr>
<tr>
<td>PRODUCT_NAME_SRP_E302</td>
<td>SRP-E302</td>
</tr>
<tr>
<td>PRODUCT_NAME_BK3_3</td>
<td>BK3-3</td>
</tr>
</tbody>
</table>
### 5-9 페이지 모드 인쇄 방향

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR_PD_LEFT_TO_RIGHT</td>
<td>1</td>
<td>왼쪽에서 오른쪽으로 인쇄</td>
</tr>
<tr>
<td>PTR_PD_BOTTOM_TO_TOP</td>
<td>2</td>
<td>아래에서 위로 인쇄</td>
</tr>
<tr>
<td>PTR_PD_RIGHT_TO_LEFT</td>
<td>3</td>
<td>오른쪽에서 원쪽으로 인쇄</td>
</tr>
<tr>
<td>PTR_PD_TOP_TO_BOTTOM</td>
<td>4</td>
<td>위에서 아래로 인쇄</td>
</tr>
</tbody>
</table>

### 5-10 MSR 암호화

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSR_DE_NONE</td>
<td>1</td>
<td>데이터 암호화 알고리즘 사용 안함</td>
</tr>
<tr>
<td>MSR_DE_3DEA_DUKPT</td>
<td>2</td>
<td>데이터 암호화 알고리즘 사용</td>
</tr>
</tbody>
</table>

### 5-11 SCR 모드

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC_CMODE_ISO</td>
<td>1</td>
<td>ISO 모드</td>
</tr>
<tr>
<td>SC_CMODE_EMV</td>
<td>2</td>
<td>EMV 모드</td>
</tr>
</tbody>
</table>
### 5-12 Character Set

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS_437_USA_STANDARD_EURO</td>
<td>437</td>
<td>PC437</td>
</tr>
<tr>
<td>CS_737_GREEK</td>
<td>737</td>
<td>PC737</td>
</tr>
<tr>
<td>CS_775_BALTIC</td>
<td>775</td>
<td>PC775</td>
</tr>
<tr>
<td>CS_850_MULTILINGUAL</td>
<td>850</td>
<td>PC850</td>
</tr>
<tr>
<td>CS_852_LATIN2</td>
<td>852</td>
<td>PC852</td>
</tr>
<tr>
<td>CS_855_CYRILLIC</td>
<td>855</td>
<td>PC855</td>
</tr>
<tr>
<td>CS_857_TURKISH</td>
<td>857</td>
<td>PC857</td>
</tr>
<tr>
<td>CS_858_EURO</td>
<td>858</td>
<td>PC858</td>
</tr>
<tr>
<td>CS_860_PORTUGUESE</td>
<td>860</td>
<td>PC860</td>
</tr>
<tr>
<td>CS_862_HEBREW_DOS_CODE</td>
<td>862</td>
<td>PC862</td>
</tr>
<tr>
<td>CS_863_CANADIAN_FRENCH</td>
<td>863</td>
<td>PC863</td>
</tr>
<tr>
<td>CS_864_ARABIC</td>
<td>864</td>
<td>PC864</td>
</tr>
<tr>
<td>CS_865_NORDIC</td>
<td>865</td>
<td>PC865</td>
</tr>
<tr>
<td>CS_866_CYRILLIC2</td>
<td>866</td>
<td>PC866</td>
</tr>
<tr>
<td>CS_928_GREEK</td>
<td>928</td>
<td>PC928</td>
</tr>
<tr>
<td>CS_1250_CZECH</td>
<td>1250</td>
<td>WPC1250</td>
</tr>
<tr>
<td>CS_1251_CYRILLIC</td>
<td>1251</td>
<td>WPC1251</td>
</tr>
<tr>
<td>CS_1252_LATIN1</td>
<td>1252</td>
<td>WPC1252</td>
</tr>
<tr>
<td>CS_1253_GREEK</td>
<td>1253</td>
<td>WPC1253</td>
</tr>
<tr>
<td>CS_1254_TURKISH</td>
<td>1254</td>
<td>WPC1254</td>
</tr>
<tr>
<td>CS_1255_HEBREW_NEW_CODE</td>
<td>1255</td>
<td>WPC1255</td>
</tr>
<tr>
<td>CS_1256_ARABIC</td>
<td>1256</td>
<td>WPC1256</td>
</tr>
<tr>
<td>CS_1257_BALTIC</td>
<td>1257</td>
<td>WPC1257</td>
</tr>
<tr>
<td>CS_1258_VIETNAM</td>
<td>1258</td>
<td>WPC1258</td>
</tr>
<tr>
<td>CS_FARSI</td>
<td>7065</td>
<td>Farsi</td>
</tr>
<tr>
<td>CS_KATAKANA</td>
<td>7565</td>
<td>Katakana</td>
</tr>
<tr>
<td>CS_KHMER_CAMBODIA</td>
<td>7572</td>
<td>Khmer</td>
</tr>
<tr>
<td>CS_THAI11</td>
<td>8411</td>
<td>THAI11</td>
</tr>
<tr>
<td>CS_THAI14</td>
<td>8414</td>
<td>THAI14</td>
</tr>
<tr>
<td>CS_THAI16</td>
<td>8416</td>
<td>THAI16</td>
</tr>
<tr>
<td>CS_THAI18</td>
<td>8418</td>
<td>THAI18</td>
</tr>
<tr>
<td>CS_THAI42</td>
<td>8442</td>
<td>THAI42</td>
</tr>
<tr>
<td>CS_KS5601</td>
<td>5601</td>
<td>KS5601</td>
</tr>
<tr>
<td>CS_BIG5</td>
<td>6605</td>
<td>BIG5</td>
</tr>
<tr>
<td>CS_GB2312</td>
<td>2312</td>
<td>GB2312</td>
</tr>
<tr>
<td>CS_SHIFT_JIS</td>
<td>8374</td>
<td>SHIFT-JIS</td>
</tr>
<tr>
<td>CS_TCVN_3_1</td>
<td>3031</td>
<td>TCVN-3(1)</td>
</tr>
<tr>
<td>CS_TCVN_3_2</td>
<td>3032</td>
<td>TCVN-3(2)</td>
</tr>
</tbody>
</table>
6. 각 클래스별 기능 정리

6-1 BXLConfigLoader Class

- 연결하고자 하는 장치 설정 정보를 저장하기 위한 Class 입니다. 설정 정보는 BXLConfigLoader Class를 통해 장치 정보를 관리 합니다. 설정 정보에는 장치 이름, 제품명, 인터페이스 등이 있으며 해당 정보가 정상적으로 저장되어 있지 않으면 장치에 연결할 수 없습니다. Open 함수를 호출하기 전에 이 클래스를 호출하여 설정 정보를 저장해야 합니다.

주의: BXLConfigLoader를 통해 저장되지 않은 장치는 연결 할 수 없습니다.

6-1-1 openFile()

기존 저장된 설정 파일을 Open 합니다.

[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()

새로운 설정 저장 파일을 생성 합니다. 보통 openFile 예외 발생 시 수행 합니다.

[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()

저장된 설정 정보를 얻어옵니다.

[Syntax]
List<JposEntry> getEntries() throws Exception

[Return Values]

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>JposEntry의 List 컨테이너</td>
<td>객체의 리스트 배열</td>
</tr>
</tbody>
</table>

[Example]

```java
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()

장치 연결 정보를 추가 합니다.

**Syntax**

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

**Parameters**

- **String logicalName** : 장치의 논리적 이름(별칭)을 입력합니다.
- **int deviceCategory** : 장치의 종류를 선택 합니다.

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVICE_CATEGORY_CASH_DRAWER</td>
<td>0</td>
<td>Cash Drawer</td>
</tr>
<tr>
<td>DEVICE_CATEGORY_MSR</td>
<td>1</td>
<td>MSR</td>
</tr>
<tr>
<td>DEVICE_CATEGORY.Pos_printer</td>
<td>2</td>
<td>POS Printer</td>
</tr>
<tr>
<td>DEVICE_CATEGORY.SMART_CARD_RW</td>
<td>3</td>
<td>SCR</td>
</tr>
</tbody>
</table>

- **String productName** : 장치 모델명("5-8 장치 모델명" 참조)
- **int deviceBus** : 인터페이스 타입을 선택 합니다.

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVICE_BUS_BLUETOOTH</td>
<td>0</td>
<td>Bluetooth</td>
</tr>
<tr>
<td>DEVICE_BUS.ETHERNET</td>
<td>1</td>
<td>Ethernet</td>
</tr>
<tr>
<td>DEVICE_BUS_USB</td>
<td>2</td>
<td>USB</td>
</tr>
<tr>
<td>DEVICE_BUS.WIFI</td>
<td>3</td>
<td>WiFi</td>
</tr>
<tr>
<td>DEVICE_BUS.WIFI_DIRECT</td>
<td>4</td>
<td>WiFi-Direct</td>
</tr>
<tr>
<td>DEVICE_BUS.BLUETOOTH_LE</td>
<td>5</td>
<td>Bluetooth Low Energy</td>
</tr>
</tbody>
</table>

- **String address** : 장치의 MAC 또는 IP Address를 입력합니다.

( BT : MAC Address, Network : IP Address)

**Example**

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

try {
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
}
catch(Exception e) {
    e.printStackTrace();
}
```
6-1-5 removeEntry()

저장된 설정 정보를 삭제 합니다.

[Syntax]
boolean removeEntry(String logicalName)

[Parameters]
• String logicalName : 장치의 논리적 이름(별칭)을 입력합니다.

[Return Values]

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>성공 시 리턴 됩니다.</td>
</tr>
<tr>
<td>false</td>
<td>실패 시 리턴 됩니다.</td>
</tr>
</tbody>
</table>

[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()

addEntry, removeEntry를 통해 변경된 정보를 저장 합니다.

[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();
}

주의 addEntry 함수를 통해 추가된 장치는 반드시 saveFile 함수를 통해 저장해야 합니다.
6-2 POSPrinter Class
• POS Printer 제어를 위한 Class 입니다. 해당 Class를 사용하여 프린터 연결 및 해제와 인쇄 작업을 수행 합니다. 특정 기능 수행 중 에러 발생 시 JposException을 발생시킵니다. (“5-1 JposException” 참조)

6-2-1 open()

프린터 클래스 사용을 시작하여 메모리 할당 등의 초기화 작업이 포함되어있습니다.
Claim 이상의 Method 호출을 위해서 반드시 선행 되어야 합니다.
BXLConfigLoader Class를 통해 저장되지 않은 장치는 open되지 않습니다.

[Syntax]
void open(String logicalDeviceName) throws JposException

[Parameters]
• String logicalDeviceName : Open 하려는 장치의 이름을 입력 합니다.

[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();
}
6-2-2 claim()

Device 정보에 포함되어있는 Port 를 실제로 Open 하기 위해 시도하며, 메모리 할당 초기화 등의 초기화 작업이 일부 포함되어있습니다.
Device 사용을 활성화하기 위해서 반드시 선행되어야 합니다.

[Syntax]
void claim(int timeout) throws JposException

[Parameters]
• int timeout : 이 파라미터에서 지정한 시간동안 Port open를 시도합니다.

[Example]
```java
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(5000);
}
catch(JposException e) {
    e.printStackTrace();
}
```
6-2-3 setDeviceEnabled()
장치 사용 여부를 포함하고 있습니다.
DeviceEnabled의 값이 비활성화 되어 있으면 기능 이용이 불가능 할 수 있습니다.

**[Syntax]**
void setDeviceEnabled(boolean deviceEnabled) throws JposException

**[Parameters]**
- boolean deviceEnabled: 장치 활성화 상태를 입력 합니다.

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>활성화</td>
</tr>
<tr>
<td>false</td>
<td>비활성화</td>
</tr>
</tbody>
</table>

**[Example]**
```java
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(5000);
    posPrinter.setDeviceEnabled(true);
}
catch(JposException e) {
    e.printStackTrace();
}
```

6-2-4 release()
Claim 된 Device의 Port 사용을 물리적으로 종료합니다.
메모리 해제 등의 작업이 수행될 수 있습니다.

**[Syntax]**
void release() throws JposException

**[Example]**
```java
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()

Open 된 Device의 사용을 종료합니다. 메모리 해제 등의 작업이 일부 수행될 수 있습니다.

[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()

장치가 정상적으로 동작하는 상태인지 확인 합니다.
Open / Claim / DeviceEnabled 가 정상적으로 수행된 상태일 때 사용 가능 합니다.

[Syntax]
void checkHealth(int level) throws JposException

[Parameters]
• int level : 고정값 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(5000);
    posPrinter.setDeviceEnabled(true);

    posPrinter.checkHealth(JposConst.JPOS_CH_INTERNAL);
} catch(JposException e) {
    e.printStackTrace();
}
6-2-7 setAsyncMode()

비동기 모드 사용 여부를 포함하고 있습니다. 비동기 모드가 true일 경우 관련 메소드가 비동기 모드로 동작합니다. False일 경우 관련 메소드는 동기 모드로 동작 합니다.

**[Syntax]**

```java
void setAsyncMode(boolean asyncMode) throws JposException
```

**[Parameters]**

- boolean asyncMode : 비동기 모드 사용 여부를 선택 합니다.

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>비동기 모드</td>
</tr>
<tr>
<td>false</td>
<td>동기 모드</td>
</tr>
</tbody>
</table>

**[Example]**

```java
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(5000);
    posPrinter.setDeviceEnabled(true);

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

6-2-8 setCharacterSet()

프린터의 Character set을 설정 합니다.

[Syntax]
void setCharacterSet(int characterSet) throws JposException

[Parameters]
• int characterSet : 프린터에 설정할 Character set을 선택 합니다.
  (“5-12 Character Set” 참조)

[Example]
```java
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(5000);
    posPrinter.setDeviceEnabled(true);

    posPrinter.setCharacterSet(BXLConst.CS_437_USA_STANDARD_EUROPE);
}
catch(JposException e) {
    e.printStackTrace();
}
```
6-2-9 setCharacterEncoding()

프린터로 전송할 데이터의 인코딩을 설정 합니다.

[Syntax]
void setCharacterEncoding(int characterEncoding) throws JposException

[Parameters]
• int characterEncoding : 데이터 인코딩 타입을 선택 합니다.

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE_ASCII</td>
<td>0</td>
<td>ASCII(default)</td>
</tr>
<tr>
<td>CE_UTF8</td>
<td>1</td>
<td>UTF-8</td>
</tr>
</tbody>
</table>

[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(5000);
    posPrinter.deviceEnabled(true);

    posPrinter.setCharacterEncoding(BXLConst.CE_ASCII);
}
catch(JposException e) {
    e.printStackTrace();
}
```
6-2-10 cutPaper()

Auto Cutter 가 내장된 모델의 경우 종이를 절단합니다.

[Syntax]
void cutPaper(int percentage) throws JposException

[Parameters]
• int percentage : Full cut/Partial cut 을 선택합니다.
  커터가 달린 모델에 한해 동작 합니다.

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Full cut</td>
</tr>
<tr>
<td>90</td>
<td>Partial cut</td>
</tr>
</tbody>
</table>

[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(5000);
    posPrinter.setDeviceEnabled(true);

    posPrinter.cutPaper(100);
}
catch(JposException e)
{
    e.printStackTrace();
}
6-2-11 printBarCode()

바코드를 인쇄 합니다.

**[Syntax]**

```java
void printBarCode(int station, String data, int symbology, int height, int width, int alignment, int textPosition) throws JposException
```

**[Parameters]**

- **int station**: 고정값 PTR_S_RECEIPT
- **String data**: 바코드에 포함할 Data. 바코드 타입 별 허용되는 Data가 다를 수 있습니다.
- **int symbology**: Barcode 종류를 선택 합니다. ("5-6 바코드 타입" 참조)
- **int height**: Barcode 의 높이를 지정합니다.
- **int width**: Barcode 의 너비를 지정합니다.
- **int alignment**: 바코드 정렬을 선택합니다. ("5-5 Alignment" 참조)
- **int textPosition**: 바코드와 함께 출력 될 Text의 Position 을 결정합니다. ("5-7 바코드 텍스트 위치" 참조)

**[Example]**

```java
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(5000);
    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()

이미지를 인쇄 합니다.(파일 인쇄)

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

[Parameters]
• int station : 이미지 인쇄 옵션을 설정 합니다.(총 4바이트로 구성)
  
  Index | 설명 |
  |---|---|
  첫번째 바이트 | 고정값 : PTR_S_RECEIPT |
  두번째 바이트 | 밝기 : 0 ~ 100 |
  세번째 바이트 | 압축 알고리즘(RLE) 사용여부(0x01 : RLE) |
  네번째 바이트 | Reserved(0x00) |
• String fileName : 이미지 파일의 경로를 지정합니다.
• int width : 이미지 너비를 지정합니다.
• int alignment : 이미지 정렬을 선택합니다. ("5-5 Alignment" 참조)

[Example]
```java
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(5000);
    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()

이미지를 인쇄 합니다.(Bitmap 데이터)

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

[Parameters]
- int station : 이미지 인쇄 옵션을 설정 합니다.(총 4바이트로 구성)

<table>
<thead>
<tr>
<th>Index</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>첫번째 바이트</td>
<td>고정값 : PTR_S_RECEIPT</td>
</tr>
<tr>
<td>두번째 바이트</td>
<td>밝기 : 0 ~ 100</td>
</tr>
<tr>
<td>세번째 바이트</td>
<td>압축 알고리즘(RLE) 사용여부(0x01 : RLE)</td>
</tr>
<tr>
<td>네번째 바이트</td>
<td>Reserved(0x00)</td>
</tr>
</tbody>
</table>
- Bitmap bitmap : 비트맵 형식의 이미지 데이터를 입력합니다.
- int width : 이미지 너비를 지정합니다.
- int alignment : 이미지 정렬을 선택합니다. (“5-5 Alignment” 참조)

[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(5000);
    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()

텍스트를 인쇄합니다.

[Syntax]
void printNormal(int station, String data) throws JposException

[Parameters]
• int station : 고정값 PTR_S_RECEIPT
• String data : 출력할 Data를 지정합니다. 출력 가능한 문자와 escape sequences, carriage returns, line feeds Data가 허용됩니다.

[Example]
```java
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(5000);
    posPrinter.setDeviceEnabled(true);

    posPrinter.printNormal(POSPrinterConst.PTR_S_RECEIPT, "Print Data\n");
} catch(JposException e) {
    e.printStackTrace();
}
```
6-2-15 printPDF()

PDF 파일을 인쇄 합니다.

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

[Parameters]
- int station : 고정값 PTR_S_RECEIPT
- String fileName : PDF 파일의 경로를 지정 합니다.
- int width : PDF 너비를 지정합니다.
- int alignment : 이미지 정렬을 선택합니다. (“5-5 Alignment” 참조)
- int page : 인쇄할 PDF 페이지 번호를 지정 합니다.
- int brightness : 밝기값을 지정 합니다. (0 ~ 100)

[Example]
```java
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(5000);  // 게시권 할당
    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()

PDF 파일을 인쇄 합니다.

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

[Parameters]
• int station : 고정값 PTR_S_RECEIPT
• String filename : PDF 파일의 경로를 지정 합니다.
• int width : PDF 너비를 지정합니다.
• int alignment : 이미지 정렬을 선택합니다. (“5-5 Alignment” 참조)
• int startPage : 인쇄를 시작할 페이지 번호를 지정 합니다.
• int endPage : 인쇄를 끝낼 페이지 번호를 지정 합니다.
• int brightness : 밝기값을 지정 합니다. (0 ~ 100)

[Example]
```java
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(5000);
    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()

페이지 모드 영역을 지정 합니다.

[Syntax]
void setPageModePrintArea(String area) throws JposException

[Parameters]
• String area : 인쇄 영역을 지정 합니다.
  “영역의 x 좌표, 영역의 y 좌표, 영역의 가로, 영역의 세로”
  ex) “0, 0, 576, 1600”

[Example]
```java
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(5000);
    posPrinter.setDeviceEnabled(true);
    posPrinter.setPageModePrintArea("0, 0, 576, 1600");
}
catch(JposException e) {
    e.printStackTrace();
}
```
6-2-18 setPageModePrintDirection()

페이지 모드 인쇄 방향을 지정 합니다.

[Syntax]
void setPageModePrintDirection(int direction) throws JposException

[Parameters]
• int direction : 인쇄 방향을 지정 합니다.(“5-9 페이지 모드 인쇄 방향” 참조)

[Example]
```java
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(5000);
    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()

프린터를 페이지 모드로 설정 합니다.

[Syntax]
void pageModePrint(int control) throws JposException

[Parameters]
• int control : 페이지 모드를 지정 합니다.

<table>
<thead>
<tr>
<th>상수명</th>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR_PM_PAGE_MODE</td>
<td>1</td>
<td>페이지 모드 활성화</td>
</tr>
<tr>
<td>PTR_PM_NORMAL</td>
<td>2</td>
<td>일반 모드로 변경되며 페이지 모드 버퍼에 저장된 데이터가 인쇄 됩니다.</td>
</tr>
</tbody>
</table>

[Example]
```java
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(5000);
    posPrinter.setDeviceEnabled(true);

    posPrinter setPageModePrintArea("0, 0, 576, 1600");
    posPrinter setPageModePrintDirection(POSPrinterConst.PTR_PD_LEFT_TO_RIGHT);
    posPrinter setPageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);

    posPrinter setPageModePrintArea("0, 0, 576, 1600");
    posPrinter setPageModePrintDirection(POSPrinterConst.PTR_PD_LEFT_TO_RIGHT);
    posPrinter setPageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);

    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()

인쇄 시작 위치(가로)를 지정 합니다.

[Syntax]
void setPageModeHorizontalPosition(int position) throws JposException

[Parameters]
• int position : 인쇄 시작 위치(가로)

[Example]
```java
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(5000);
    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);

    // 인쇄 영역 지정
    posPrinter setPageModePrintArea("0, 0, 576, 1600");
    // 인쇄 방향 설정
    posPrinter setPageModePrintDirection(POSPrinterConst.PTR_PD_LEFT_TO_RIGHT);
    // 프린터 페이지 모드로 변환
    posPrinter setPageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);

    // 가로/세로 인쇄 위치 지정
    posPrinter setPageModeHorizontalPosition(0);
    posPrinter setPageModeVerticalPosition(0);
    // 인쇄할 데이터 전송(이미지)
    posPrinter.printBitmap(buffer.getInt(0), imagePath, 384, POSPrinterConst.PTR_BM_LEFT);

    // 가로/세로 인쇄 위치 지정
    posPrinter setPageModeHorizontalPosition(100);
    posPrinter setPageModeVerticalPosition(100);
    // 인쇄할 데이터 전송(텍스트)
    posPrinter.printNormar(POSPrinterConst.PTR_S_RECEIPT, "Print Data\n");

    // 인쇄 시작
    posPrinter setPageModePrint(POSPrinterConst.PTR_PM_PAGE_NORMAL);
} catch (JposException e) {
    e.printStackTrace();
}
```
6-2-21 setPageModeVerticalPosition()

인쇄 시작 위치(세로)를 지정 합니다.

**[Syntax]**

```java
void setPageModeVerticalPosition(int position) throws JposException
```

**[Parameters]**

- `int position` : 인쇄 시작 위치(세로)

**[Example]**

```java
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(5000);
    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);

    // 인쇄 영역 지정
    posPrinter.setPageModePrintArea("0, 0, 576, 1600");
    // 인쇄 방향 설정
    posPrinter.setPageModePrintDirection(POSPrinterConst.PTR_PD_LEFT_TO_RIGHT);
    // 프린터 페이지 모드로 변환
    posPrinter.setPageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);

    // 가로/세로 인쇄 위치 지정
    posPrinter.setPageModeHorizontalPosition(0);
    posPrinter.setPageModeVerticalPosition(0);
    // 인쇄할 데이터 전송(이미지)
    posPrinter.printBitmap(buffer.getInt(0), imagePath, 384, POSPrinterConst.PTR_BM_LEFT);

    // 가로/세로 인쇄 위치 지정
    posPrinter.setPageModeHorizontalPosition(100);
    posPrinter.setPageModeVerticalPosition(100);
    // 인쇄할 데이터 전송(텍스트)
    posPrinter.printNormar(POSPrinterConst.PTR_S_RECEIPT, "Print Data\n");

    // 인쇄 시작
    posPrinter.setPageModePrint(POSPrinterConst.PTR_PM_PAGE_NORMAL);
} catch(JposException e) {
    e.printStackTrace();
}
```
6-2-22 transactionPrint()

Transaction Mode를 사용하여 인쇄 합니다.

[Syntax]
public void transactionPrint(int station, int control) throws JposException

[Parameters]
• int station : 고정값 PTR_S_RECEIPT
• int control : Transaction Mode(“5-4 Transaction Print” 참조)

[Example]
```java
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(5000);
    posPrinter.setDeviceEnabled(true);

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

    // 인쇄할 데이터 전송(이미지)
    posPrinter.printBitmap(buffer.getInt(0), imagePath, 384, POSPrinterConst.PTR_BM_LEFT);
    // 인쇄할 데이터 전송(텍스트)
    posPrinter.printNormar(POSPrinterConst.PTR_S_RECEIPT, "Print Data\n");

    // Transaction mode 종료(인쇄 시작)
    posPrinter.transactionPrint(POSPrinterConst.PTR_TP_NORMAL);
}
catch(JposException e)
{
    e.printStackTrace();
}
6-2-23 displayString()

BCD-3000에 텍스트를 출력 합니다.
단, BCD-3000은 SRP-Q300 DK port에 연결되어 있어야 합니다.

**Syntax**
public void displayString(String data) throws JposException

**Parameters**
• String data : BCD-3000에 출력할 텍스트 데이터

**Example**
```java
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(5000);
    posPrinter.setDeviceEnabled(true);

    posPrinter.displayString("BIXOLON Customer Display 3000");
} catch(JposException e) {
    e.printStackTrace();
}
```
6-2-24 cleanScreen()

BCD-3000 화면을 클리어 합니다.
단, BCD-3000은 SRP-Q300 DK port에 연결되어 있어야 합니다.

[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(5000);
    posPrinter.setDeviceEnabled(true);

    posPrinter.displayString("BIXOLON Customer Display 3000");
    posPrinter.cleanScreen();
}
catch(JposException e)
{
    e.printStackTrace();
}
6-2-25 storeImageFile()

BCD-3000 이미지 버퍼에 이미지 데이터를 저장합니다.
단, BCD-3000은 SRP-Q300 DK port에 연결되어 있어야 합니다.

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

[Parameters]
• String filename : 이미지 파일의 경로를 지정합니다.
• int width : 이미지 너비를 지정합니다.(1 ~ 160)
• int height : 이미지 높이를 지정합니다.(1 ~ 32)
• int imageNumber : 저장할 이미지 데이터 번호를 지정합니다.(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(5000);
    posPrinter.setDeviceEnabled(true);
    posPrinter.storeImageFile(imagePath, 160, 32, 1);
}
catch(JposException e)
{
    e.printStackTrace();
}
6-2-26 displayImage()

BCD-3000 이미지 버퍼에 저장된 이미지를 출력 합니다.
단, BCD-3000은 SRP-Q300 DK port에 연결되어 있어야 합니다.

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

[Parameters]
• int imageNumber : 출력할 이미지 데이터 번호를 지정합니다.(1 ~ 5)
• int xPos : 이미지를 출력할 X 좌표를 입력 합니다.(0 ~ 159)
• int yPos : 이미지를 출력할 Y 좌표를 입력 합니다.(0 ~ 31)
※ 이미지가 출력할 영역을 넘어갈 경우 이미지는 Display 되지 않습니다.

[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(5000);
    posPrinter.setDeviceEnabled(true);

    posPrinter.storeImageFile(imagePath, 160, 32, 1);
    posPrinter.displayImage(1, 0, 0);
} catch(JposException e) {
    e.printStackTrace();
}
6-2-27 clearImage()

BCD-3000 이미지 버퍼에 저장된 이미지를 삭제합니다。
단, BCD-3000은 SRP-Q300 DK port에 연결되어 있어야 합니다.

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

[Parameters]
• boolean isAll : 이미지 버퍼 클리어 모드를 선택 합니다.

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>이미지 버퍼의 모든 이미지를 삭제합니다.</td>
</tr>
<tr>
<td>false</td>
<td>저장된 번호의 이미지만 삭제 합니다. imageNumber에 삭제할 이미지 번호를 입력해야 합니다.</td>
</tr>
</tbody>
</table>

• int imageNumber : 삭제할 이미지 데이터 번호를 지정합니다. (1 ~ 5)

[Example]
```java
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(5000);
    posPrinter.setDeviceEnabled(true);

    posPrinter.storeImageFile(imagePath, 160, 32, 1);
    posPrinter.displayImage(1, 0, 0);
    posPrinter.clearImage(false, 1);
}
catch(JposException e) {
    e.printStackTrace();
}
```
6-2-28 directIO()

사용자 정의 데이터를 프린터에 전송합니다.
데이터에 따라 응답값은 directIOOccurred Event로 전달 됩니다.

[Syntax]
public void directIO(int command, int[] data, Object object) throws JposException

[Parameters]
• int command : direct I/O command 종류를 선택 합니다.

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>사용자 정의 데이터 전송.</td>
</tr>
<tr>
<td>2</td>
<td>배터리 상태 체크.</td>
</tr>
<tr>
<td>3</td>
<td>Power off time(모바일 프린터만 사용 가능) : 0 ~ 90(초)</td>
</tr>
</tbody>
</table>

• int[] data : Power off time을 지정 합니다.(command = 3)
• Object object : 사용자 정의 명령어를 입력 합니다.

[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.addDirectIOListener(this);
    posPrinter.open("SRP-Q300");
    posPrinter.claim(5000);
    posPrinter.setDeviceEnabled(true);
    byte[] data = new byte{0x1d, 0x49, 0x43};
    posPrinter.directIO(1, null, data);
}
catch(JposException e) {
    e.printStackTrace();
}

@Override
public void directIOOccurred(DirectIOEvent directIOEvent) {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(MainActivity.this, new String((byte[]) directIOEvent.getObject(), Toast.LENGTH_SHORT).show();
        }
    });
}
6-2-29 markFeed()

다음 인쇄 위치까지 용지를 Feeding 합니다.

[Syntax]
public void markFeed(int type) throws JposException

[Parameters]
• int type : Mark type을 지정 합니다.(고정값 : 0)

[Example]
```java
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-R310");
    posPrinter.claim(5000);
    posPrinter.setDeviceEnabled(true);

    posPrinter.markFeed(0);
}
catch(JposException e) {
    e.printStackTrace();
}
```
6-3 MSR Class

- 프린터의 MSR(Magnetic Stripe Reader) 제어를 위한 Class입니다. 해당 Class를 사용하여 MSR로부터 Card Track 정보를 얻어낼 수 있습니다. 특정 기능 수행 중 에러 발생 시 JposException을 발생시킵니다. (“5-1 JposException” 참조)

6-3-1 open()


[Syntax]
void open(String logicalDeviceName) throws JposException

[Parameters]
- String logicalDeviceName : Open 하려는 장치 이름

[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-R200II");
}
catch(JposException e)
{
    e.printStackTrace();
}
6-3-2 claim()

Device 정보에 포함되어있는 Port 를 실제로 Open 하기 위해 시도 하며, 메모리 활당 초기화 등의 초기화 작업이 일부 포함되어있습니다. 
Device 사용을 활성화 하기 위해서 반드시 선행 되어야 합니다.

[Syntax]
void claim(int timeout) throws JposException

[Parameters]
• int timeout : 이 파라미터에서 지정한 시간동안 Port open을 시도합니다.

[Example]
```java
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()

장치 사용 여부를 포함하고 있습니다. 
DeviceEnabled 의 값이 비활성화 되어 있으면 기능 이용이 불가능 할 수 있습니다.

[Syntax]
void setDeviceEnabled(boolean deviceEnabled) throws JposException

[Parameters]
• boolean deviceEnabled : 장치 활성화 상태를 입력 합니다.

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>활성화</td>
</tr>
<tr>
<td>false</td>
<td>비활성화</td>
</tr>
</tbody>
</table>

[Example]
```java
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()

Claim 된 Device의 Port 사용을 물리적으로 종료 합니다. 메모리 해제 등의 작업이 수행 될 수 있습니다.

[Syntax]
void release() throws JposException

[Example]
```java
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()

Open 된 Device의 사용을 종료 합니다. 메모리 해제 등의 작업이 일부 수행 될 수 있습니다.

[Syntax]
void close() throws JposException

[Example]
```java
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()

MSR 리딩 후 자동으로 MSR 장치를 비활성화 합니다. 
AutoDisable가 true일 경우 MSR 데이터 수신 후 DeviceEnabled를 false로 변경 합니다.

[Syntax]
void setAutoDisable(boolean autoDisable) throws JposException

[Parameters]
• boolean autoDisable : Auto Disable 여부를 지정 합니다.

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>활성화</td>
</tr>
<tr>
<td>false</td>
<td>비활성화</td>
</tr>
</tbody>
</table>

[Example]
```java
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()

MSR 리딩 후 Track 정보를 Data Event를 통해 수신할지 여부를 지정합니다.

[Syntax]
void setDataEventEnabled(boolean dataEventEnabled) throws JposException

[Parameters]
• boolean dataEventEnabled : Data Event 사용 여부를 지정 합니다.

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>Event 사용(&quot;5-2.4 Data Event&quot; 참조)</td>
</tr>
<tr>
<td>false</td>
<td>Event 미사용</td>
</tr>
</tbody>
</table>

[Example]
```java
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()

MSR Track 데이터의 암호화 여부를 지정합니다.

**Syntax**

```java
void setDataEncryptionAlgorithm(int encryptAlgorithm) throws JposException
```

**Parameters**

- **int encryptAlgorithm** : Track 암호화 사용 여부를 지정 합니다. (5-10 MSR 암호화" 참조)

**Example**

```java
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()

가장 최근에 읽은 MSR Card의 Track1 Data 를 가져옵니다.

[Syntax]
babye[] getTrack1Data() throws JposException

[Return Values]
※ MSR Card의 Track1 Data

[Example]
```java
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()

가장 최근에 읽은 MSR Card의 Track2 Data 를 가져옵니다.

[Syntax]
byte[] getTrack2Data() throws JposException

[Return Values]
※ MSR Card의 Track2 Data

[Example]
```java
import jpos.JposConst;
import jpos.JposException;
import jpos.MSR;
import jpos.MSRConst;
import jpos.events.DataEvent;
import jpos.events.DataAccessListener;

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()

가장 최근에 읽은 MSR Card의 Track3 Data를 가져옵니다.

[Syntax]
byte[] getTrack3Data() throws JposException

[Return Values]
※ MSR Card의 Track3 Data

[Example]
```java
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

- 프린터의 SCR(Smart Card Reader) 제어를 위한 Class 입니다. 특정 기능 수행 중 에러 발생 시 JposException을 발생시킵니다. (“5-1 JposException” 참조)

6-4-1 open()

SmartCardRW 클래스 사용을 시작하며 메모리 할당 등의 초기화 작업이 포함되어있습니다.
Claim 이상의 Method 호출을 위해서 반드시 선행 되어야 합니다.
BXLConfigLoader Class를 통해 저장되지 않은 장치는 open되지 않습니다.

[Syntax]
void open(String logicalDeviceName) throws JposException

[Parameters]
- String logicalDeviceName : Open 하려는 장치 이름

[Example]
```
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try
{
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R210");
}
catch(JposException e)
{
    e.printStackTrace();
}
```
6-4-2 claim()

Device 정보에 포함되어있는 Port 를 실제로 Open 하기 위해 시도 하며, 메모리 할당 초기화 등의 초기화 작업이 일부 포함되어있습니다.
Device 사용을 활성화 하기 위해서 반드시 선행 되어야 합니다.

[Syntax]
void claim(int timeout) throws JposException

[Parameters]
• int timeout : 이 파라미터에서 지정한 시간동안 Port open을 시도합니다.

[Example]
```java
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try {
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R210");
    smartCardRW.claim(3000);
} catch(JposException e) {
    e.printStackTrace();
}
```
6-4-3 setDeviceEnabled()

장치 사용 여부를 포함하고 있습니다. DeviceEnabled의 값이 비활성화 되어 있으면 기능 이용이 불가능 할 수 있습니다.

[Syntax]
void setDeviceEnabled(boolean deviceEnabled) throws JposException

[Parameters]
• boolean deviceEnabled : 장치 활성화 상태를 입력합니다.

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>활성화</td>
</tr>
<tr>
<td>false</td>
<td>비활성화</td>
</tr>
</tbody>
</table>

[Example]
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try
{
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R210");
    smartCardRW.claim(3000);
    SmartCardRW.setDeviceEnabled(true);
}
catch(JposException e)
{
    e.printStackTrace();
}

6-4-4 release()

Claim 된 Device의 Port 사용을 물리적으로 종료 합니다. 메모리 해제 등의 작업이 수행 될 수 있습니다.

[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()

Open 된 Device의 사용을 종료 합니다. 메모리 해제 등의 작업이 일부 수행 될 수 있습니다.

[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()

통신을 위한 Card slot을 선택 합니다.

**[Syntax]**
void setSCSlot() throws JposException

**[Parameters]**
- int scSlot : 카드 솔롯을 선택 합니다.

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1000</td>
<td>Smart Card</td>
</tr>
<tr>
<td>0x0100</td>
<td>SAM1</td>
</tr>
<tr>
<td>0x0010</td>
<td>SAM2</td>
</tr>
</tbody>
</table>

**[Example]**
```java
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try {
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R210");
    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()

ISO/EMV 모드를 선택 합니다.

[Syntax]
void setIsoEmvMode(int isoEmvMode) throws JposException

[Parameters]
• int isoEmvMode : 모드를 선택 합니다.(“5-11 SCR 모드” 참조)

[Example]
```java
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try {
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R210");
    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 beginInsertion()

지정한 시간 동안 Smart Card가 삽입되었는지 체크 합니다.
해당 함수 호출 후 반드시 endInsertion() 함수를 호출해야 합니다.

[Syntax]
void beginInsertion(int timeout) throws JposException

[Parameters]
• int timeout : 카드 삽입 체크 시간 설정

[Example]
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try
{
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R210");
    smartCardRW.claim(5000);
    SmartCardRW.setDeviceEnabled(true);

    smartCardRW.beginInsertion(5000);
    smartCardRW.endInsertion();
}
catch(JposException e)
{
    e.printStackTrace();
}
6-4-9 endInsertion()

삽입된 Smart Card chip에 파워를 인가합니다.
해당 함수 호출 전 반드시 beginInsertion 함수를 호출해야 합니다.

[Syntax]
void endInsertion() throws JposException

[Example]
```java
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try {
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R210");
    smartCardRW.claim(5000);
    SmartCardRW.setDeviceEnabled(true);

    smartCardRW.beginInsertion(5000);
    smartCardRW.endInsertion();
} catch(JposException e) {
    e.printStackTrace();
}
```
6-4-10 beginRemoval()

Smart Card chip에 인가된 파워를 해지합니다.
해당 함수 호출 후 반드시 endRemoval 함수를 호출해야 합니다.

**[Syntax]**
void beginRemoval(int timeout) throws JposException

**[Parameters]**
- int timeout : 파워 해지 시간 설정

**[Example]**
```java
import jpos.JposConst;
import jpos.JposException;
import jpos.SmartCardRW;
import jpos.SmartCardRWConst;

try {
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R210");
    smartCardRW.claim(5000);
    smartCardRW.setDeviceEnabled(true);

    smartCardRW.beginRemoval(5000);
    smartCardRW.endRemoval();
}
catch(JposException e) {
    e.printStackTrace();
}
```
6-4-11 endRemoval()

Smart Card chip에 파워가 정상적으로 해지 되었는지 체크 합니다.
해당 함수 호출 전 반드시 beginRemoval 함수를 호출해야 합니다.

**[Syntax]**

void endRemoval() throws JposException

**[Example]**

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

try{
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R210");
    smartCardRW.claim(5000);
    SmartCardRW.setDeviceEnabled(true);

    smartCardRW.beginRemoval(5000);
    smartCardRW.endRemoval();
}
catch(JposException e)
{
    e.printStackTrace();
}
```
6-4-12 readData()

Smart Card chip에 데이터를 R/W 합니다.

**Syntax**

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

**Parameters**

- **int action**: 고정값 SmartCardRWConst.SC_READ_DATA
- **int[] count**: 응답 데이터 사이즈
- **String[] data**: R/W 버퍼(“cp437” 인코딩)

**Example**

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

try {
    SmartCardRW smartCardRW = new SmartCardRW();
    smartCardRW.open("SPP-R210");
    smartCardRW.claim(3000);
    smartCardRW.setDeviceEnabled(true);

    // Power up
    smartCardRW.beingInsertion(5000);
    smartCardRW.endInsertion();

    String[] data = new String[]{
        new String(new byte[]{0x00, (byte) 0xA4, 0x04, 0x00, 0x07, (byte) 0xD4, 0x10, 0x65, 0x09, (byte) 0x90, 0x00, 0x10}, "CP437")
    }

    int[] count = new int[1];

    smartCardRW.readData(SmartCardRWConst.SC_READ_DATA, count, data);
    int rspSize = count[0];
    byte[] rspData = data[0].getBytes("CP437");

    // Power down
    smartCardRW.beginRemoval(5000);
    smartCardRW.endRemoval();
}
```

```
6-5 CashDrawer Class

CashDrawer 제어를 위한 Class 입니다. 특정 기능 수행 중 에러 발생 시 JposException을 발생시킵니다. (“5-1 JposException” 참조)

6-5-1 open()

CashDrawer 클래스 사용을 시작하며 메모리 할당 등의 초기화 작업이 포함되어있습니다. Claim 이상의 Method 호출을 위해서 반드시 선행 되어야 합니다. BXLConfigLoader Class를 통해 저장되지 않은 장치는 open되지 않습니다.

[Syntax]
void open(String logicalDeviceName) throws JposException

[Parameters]
- String logicalDeviceName : Open 하려는 장치 이름

[Example]
```java
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try
{
    CashDrawer cashDrawer = new CashDrawer();
    cashDrawer.open("SRP-Q300");
}
catch(JposException e)
{
    e.printStackTrace();
}
```
6-5-2 claim()

Device 정보에 포함되어있는 Port 를 실제로 Open 하기 위해 시도 하며, 메모리 할당 초기화 등의 초기화 작업이 일부 포함되어있습니다. Device 사용을 활성화 하기 위해서 반드시 선행되어야 합니다.

[Syntax]
void claim(int timeout) throws JposException

[Parameters]
• int timeout : 이 파라미터에서 지정한 시간동안 Port open을 시도합니다.

[Example]
```java
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try
{
    CashDrawer cashDrawer = new CashDrawer();
    cashDrawer.open("SRP-Q300");
    cashDrawer.claim(3000);
}
catch(JposException e)
{
    e.printStackTrace();
}
```
6-5-3 setDeviceEnabled()

장치 사용 여부를 포함하고 있습니다. DeviceEnabled의 값이 비활성화 되어 있으면 기능 이용이 불가능 할 수 있습니다.

[Syntax]
void setDeviceEnabled(boolean deviceEnabled) throws JposException

[Parameters]
• boolean deviceEnabled: 장치 활성화 상태를 입력 합니다.

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>활성화</td>
</tr>
<tr>
<td>false</td>
<td>비활성화</td>
</tr>
</tbody>
</table>

[Example]
```java
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try {
    CashDrawer cashDrawer = new CashDrawer();
    cashDrawer.open("SRP-Q300");
    cashDrawer.claim(3000);
    cashDrawer.setDeviceEnabled(true);
} catch(JposException e) {
    e.printStackTrace();
}
```

6-5-4 release()

Claim된 Device의 Port 사용을 물리적으로 종료 합니다. 메모리 해제 등의 작업이 수행 될 수 있습니다.

[Syntax]
void release() throws JposException

[Example]
```java
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try {
    cashDrawer.release();
} catch(JposException e) {
    e.printStackTrace();
}
```
6-5-5 close()

Open 된 Device의 사용을 종료 합니다. 메모리 해제 등의 작업이 일부 수행 될 수 있습니다.

[Syntax]
void close() throws JposException

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

6-5-6 openDrawer()

Cash Drawer를 Open 합니다.

[Syntax]
void openDrawer() throws JposException

[Example]
```java
try {
    CashDrawer cashDrawer = new CashDrawer();
    cashDrawer.open("SRP-Q300");
    cashDrawer.claim(3000);
    cashDrawer.setDeviceEnabled(true);

    cashDrawer.openDrawer();
} catch (JposException e) {
    e.printStackTrace();
}
```
6-5-7 getDrawerOpened()

Cash Drawer Open 상태를 리턴 합니다.

[Syntax]
boolean getDrawerOpened() throws JposException

[Return Values]

<table>
<thead>
<tr>
<th>값</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>Cash drawer is opened.</td>
</tr>
<tr>
<td>false</td>
<td>Cash drawer is closed.</td>
</tr>
</tbody>
</table>

[Example]
```java
import jpos.JposConst;
import jpos.JposException;
import jpos.CashDrawer;
import jpos.CashDrawerConst;

try {
    CashDrawer cashDrawer = new CashDrawer();
    cashDrawer.open(“SRP-Q300”);
    cashDrawer.claim(3000);
    cashDrawer.setDeviceEnabled(true);

    if(cashDrawer.getDrawerOpened())
    {
        // opened
    }
    else
    {
        // closed
    }
}
catch(JposException e)
{
    e.printStackTrace();
}
```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try {
    // BXLConfigLoader 생성 / 설정 파일 open
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
    bxlConfigLoader.openFile();
} catch (JposException e) {
    e.printStackTrace();
    bxlConfigLoader.newFile();
}

try {
    // 장치 정보 추가
    bxlConfigLoader.addEntry("SPP-R200III",
            BXLConfigLoader.DEVICE_CATEGORY_POS_PRINTER,
            BXLConfigLoader.PRODUCT_NAME_SPP_R200III,
            BXLConfigLoader.DEVICE_BUS_BLUETOOTH,
            "74:F0:7D:E4:11:AF");

    // 설정 파일 저장
    bxlConfigLoader.saveFile();

    // 프린터 Open/Claim/DeviceEnabled
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(5000);
    posPrinter.setDeviceEnabled(true);

    // 텍스트 인쇄
    posPrinter.printNormal(POSPrinterConst.PTR_S_RECEIPT, "Print Data\n");
    posPrinter.printNormal(POSPrinterConst.PTR_S_RECEIPT,
            EscapeSequence.getString(7) + "www.bixolon.com\n");

    posPrinter.close();
} catch (JposException e) {
    e.printStackTrace();
}
7-2 이미지 인쇄

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

try {
    // BXLConfigLoader 생성 / 설정 파일 open
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
    bxlConfigLoader.openFile();
}
catch(JposException e) {
    e.printStackTrace();
    bxlConfigLoader.newFile();
}

try {
    // 장치 정보 추가
    bxlConfigLoader.addEntry("SPP-R200III",
                            BXLConfigLoader.DEVICE_CATEGORY_POS_PRINTER,
                            BXLConfigLoader.PRODUCT_NAME_SPP_R200III,
                            BXLConfigLoader.DEVICE_BUS_BLUETOOTH,
                            "74:F0:7D:E4:11:AF");

    // 설정 파일 저장
    bxlConfigLoader.saveFile();

    // 프린터 Open/Claim/DeviceEnabled
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(5000);
    posPrinter.setDeviceEnabled(true);

    // 이미지 인쇄
    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();
}
```
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

ty{ // BXLConfigLoader 생성 / 설정 파일 open
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
    bxlConfigLoader.openFile();
}
catch(JposException e)
{
    e.printStackTrace();
    bxlConfigLoader.newFile();
}

ty{
    // 장치 정보 추가
    bxlConfigLoader.addEntry("SPP-R200III",
        BXLConfigLoader.DEVICE_CATEGORY_POS_PRINTER,
        BXLConfigLoader.PRODUCT_NAME_SPP_R200III,
        BXLConfigLoader.DEVICE_BUS_BLUETOOTH,
        "74:F0:7D:E4:11:AF");

    // 설정 파일 저장
    bxlConfigLoader.saveFile();

    // 프린터 Open/Claim/DeviceEnabled
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(5000);
    posPrinter.setDeviceEnabled(true);

    // PDF 인쇄
    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();
}
import com.bxl.config.editor.BXLConfigLoader;
import jpos.config.JposEntry;
import jpos.POSPrinter;
import jpos.POSPrinterConst;

try {
    // BXLConfigLoader 생성 / 설정 파일 open
    BXLConfigLoader bxlConfigLoader = new BXLConfigLoader(this);
    bxlConfigLoader.openFile();
} catch (JposException e) {
    e.printStackTrace();
    bxlConfigLoader.newFile();
}

try {
    // 장치 정보 추가
    bxlConfigLoader.addEntry("SPP-R200III",
        BXLConfigLoader.DEVICE_CATEGORY_POS_PRINTER,
        BXLConfigLoader.PRODUCT_NAME_SPP_R200III,
        BXLConfigLoader.DEVICE_BUS_BLUETOOTH,
        "74:F0:7D:E4:11:AF");

    // 설정 파일 저장
    bxlConfigLoader.saveFile();

    // 프린터 Open/Claim/DeviceEnabled
    POSPrinter posPrinter = new POSPrinter(this);
    posPrinter.open("SPP-R200III");
    posPrinter.claim(5000);
    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 인쇄 영역 설정
    posPrinter setPageModePrintArea("0,0,384,1200");

    // Page mode 인쇄 방향 설정
    posPrinter setPageModePrintDirection(POSPrinterConst.PTR_PD_LEFT_TO_RIGHT);

    // Page mode 전환
    posPrinter.pageModePrint(POSPrinterConst.PTR_PM_PAGE_MODE);

    // 인쇄 위치 설정 : 가로/세로
    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);

        // 페이지 모드 인쇄 시작
        posPrinter.pageModePrint(POSPrinterConst.PTR_PM_NORMAL);

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

이 사용설명서와 제품은 저작권법에 의해 보호되어 있습니다.
(주)빅솔론의 사전 서면 동의 없이 사용 설명서 및 제품의 일부 또는 전체를
복사, 복제, 번역 또는 전자매체나 기계가 읽을 수 있는 형태로 바꿀 수 없습니다.

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

빅솔론 로고는 (주)빅솔론의 등록상표입니다.

사용시 주의사항

프린터와 같은 전자 제품은 정전기에 의해 쉽게 훼손될 수 있습니다.
정전기로부터 프린터를 보호하기 위해서는 프린터 후면 부에 케이블을 연결하거나
제거하기 전에 반드시 프린터 전원을 고심시오. 만약 프린터가 정전기로부터 손상을
입었을 경우에는 가까운 구입처에 문의하십시오.

Copyright © BIXOLON Co., Ltd. 2017. All rights reserved.
<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00</td>
<td>17.08.21</td>
<td>-</td>
<td>신규제작</td>
</tr>
<tr>
<td>2.01</td>
<td>17.11.22</td>
<td>4,11,13,23,24,40,41,43,44</td>
<td>SRP-QE300/SRP-QE302 모델추가, 누락내용추가</td>
</tr>
<tr>
<td>2.02</td>
<td>18.01.09</td>
<td>4,11,14,16,18,23,26,30,51,52,53,54,55</td>
<td>SRP-E300/SRP-E302 모델추가. BCD-3000 제어 API 설명 추가. 배터리 상태 이벤트 추가 외</td>
</tr>
<tr>
<td>2.03</td>
<td>18.05.28</td>
<td>15,17,19,58,59,76~80</td>
<td>샘플 애플리케이션 변경 및 레퍼런스 클래스 추가. SmartCard 제어 API 설명 추가. 누락내용추가.</td>
</tr>
<tr>
<td>2.04</td>
<td>18.08.30</td>
<td>1,5,10,13,19,20,23,68,69,70,72,73,79,81,83</td>
<td>SRP-383 모델추가. 누락내용추가</td>
</tr>
<tr>
<td>2.05</td>
<td>18.10.17</td>
<td>1,5,18,19,22</td>
<td>BK3-3 모델추가. FeedCut 설명 추가.</td>
</tr>
</tbody>
</table>