

Sanei ESC/POS Android SDK User Manual

February 13, 2024 Rev1.4.0

This manual provides information on the design guidelines concerning Sanei ESC/POS Android SDK that customers need to build Android applications.



History of revision of this manual

Revision	Date	SDK Ver	Description of revision
Rev1.0.0	December 25, 2019	1.0.0	Release of 1st edition
Rev1.1.0	April 13, 2020	1.1.0	(1) SM4-21 is added to the supportable printer. (2) BLE/WLAN is expanded in the interface specification. (3) BuildConfig class is added. (4) In getStatus method, STATUS_2_BOOKED is removed and STATUS_2_LOWBATTERY is newly added
Rev1.2.0	August 02, 2021	1.2.0	(1) SM4-31, SK1-41, SP1-21 is added to the supportable printer.
Rev1.2.1	January 14, 2022	1.2.1	(1) Android 12 is supported to build
Rev1.3.0	January 5, 2023	1.3.0	(1) SK5-31 is added to the supportable printer.
Rev1.4.0	February 13, 2024	1.4.0	(1) The sample program is now compatible with Android 12 or later.

Caution

- Sanei ESC/POS Android SDK is a copyrighted work of Sanei Electric Co., Ltd. (hereinafter referred to as Sanei Electric). Copyright and other rights concerning this product belong to Sanei Electric.
- Sanei Electric grants the user the right to use Sanei ESC/POS Android SDK (free to copy and distribute) for the purpose of using Sanei Electric products that are compatible with Sanei ESC/POS Android SDK.
- Sanei Electric does not guarantee the absence of defects in the Sanei ESC/POS Android SDK and is not liable for any damages resulting from the use of the information contained in this manual.
- Sanei Electric shall never be liable under any circumstances for any direct or indirect loss or damage in connection with the use of Sanei ESC/POS Android SDK.
- Users cannot directly or indirectly export all or part of the Sanei ESC/POS Android SDK without obtaining necessary permission from the Japanese government or the government of the relevant country.

Sanei Electric Co., Ltd. 2020

Unauthorized reproduction prohibited.

The contents of this document are subject to change without notice.

Android is a trademark of Google Inc.

Windows is a trademark of the US Microsoft Corporation in the United States, Japan, and other countries.

The Bluetooth wordmark and logo are registered trademarks of Bluetooth SIG, Inc.

They are trademarks or registered trademarks, and they shall be used under license.

Other product names and company names are trademarks or registered trademarks of the respective companies.

1. Introduction	6
1.1 Building the Sanei ESC/POS Android	6
1.2 Supported Printers	8
1.3 Related Software	8
1.4 Installing Libraries	8
2. Printer Device Class	11
3. Printer Device Class Methods	12
3.1 discoverUsbPrinter method	13
3.1.1 discoverBlePrinter method	13
3.2 connectPrinter method (For USB)	14
3.2.1 connectPrinter method (For BLE)	14
3.2.2 connectPrinter method (For WLAN)	14
3.3 disconnectPrinter method	15
3.4 isPrinterConnected method	15
3.5 isSupportedPrinterDevice method (For BLE)	15
3.6 printString method	16
3.7 setFontStyle method	17
3.8 setFontType method	17
3.9 setFontMagnification method	18
3.10 setFontColor method	18
3.11 setFontSmoothing method	19
3.12 setAlignment method	19
3.13 setCodePage method	20
3.14 setInternationalChar method	21
3.15 printBarcode method	22
3.16 printQR method	23
3.17 printPDF417 method	24
3.18 setBarcodeModification method	25
3.19 printBitmap method	26
3.20 printBitmap method	27
3.21 printBitmapRaster method	28
3.22 lineFeed method	29
3.23 paperFeed method	30
3.24 printStringAndroidFont method	31
3.25 startPageMode method	32
3.26 endPageMode method	32
3.27 setPageAttribute method	33
3.28 getStatus method	34
3.29 initPrinter method	35
3.30 outputRawData method	36
3.31 outputRawData method	36
3.32 inputRawData method	37
3.33 inputRawData method	37

Appendix 1. Printer Status..... 38

Appendix 2. Page Mode..... 42

Appendix 3. Confirmation of SDK version..... 44

1. Introduction

1.1 Building the Sanei ESC/POS Android

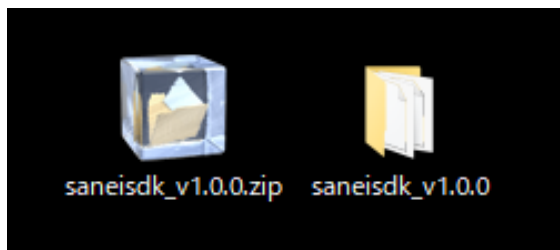
To build the Android SDK, obtain the package from the following site.

<https://www.sanei-elec.co.jp/en/support/downloads/android-sdk/>

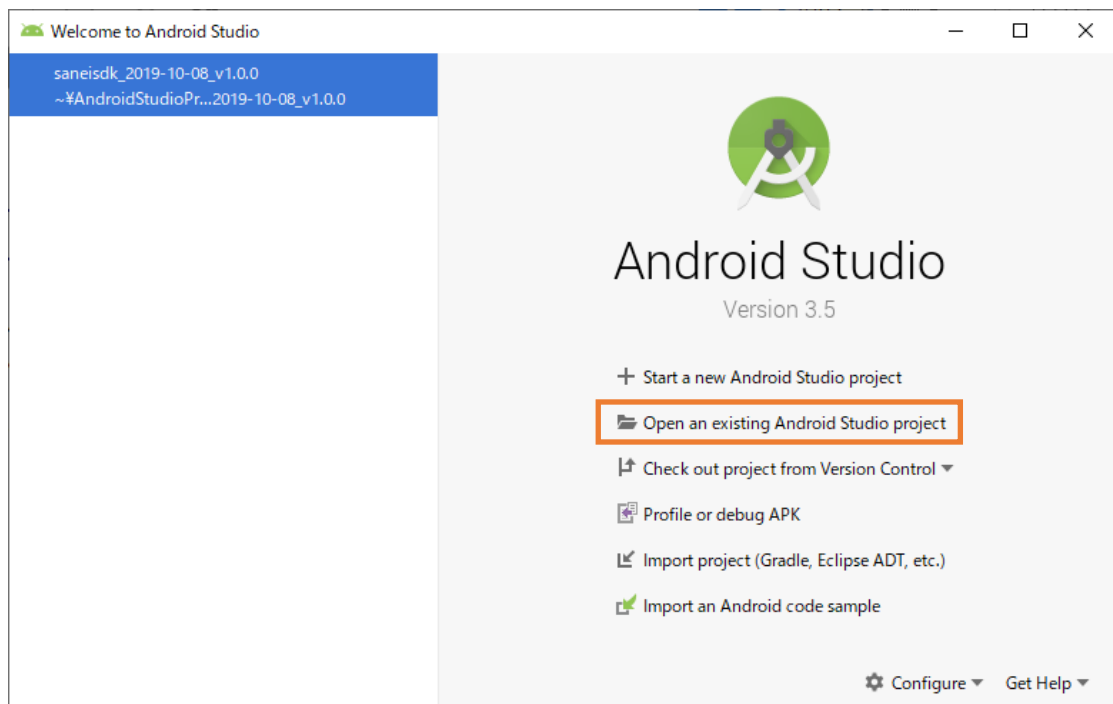
Following this procedure, open the Android SDK project in the Android Studio project where the customer plans to create the Android application.

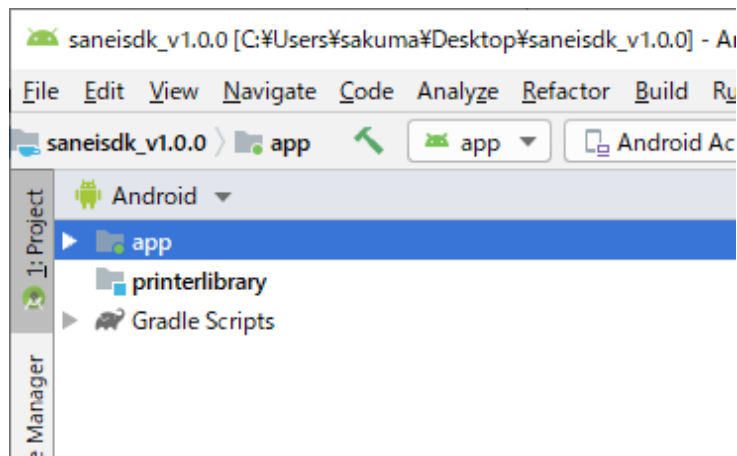
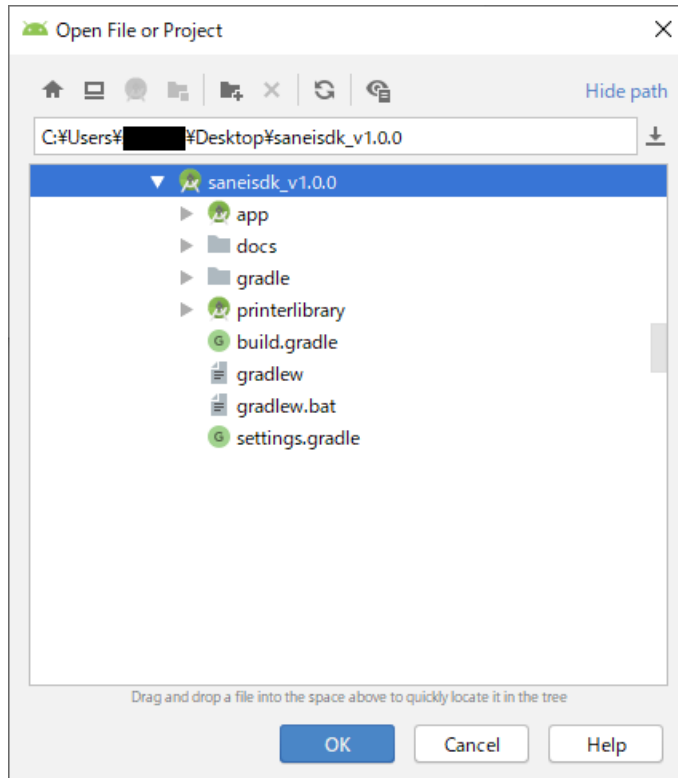
It is assumed that Android Studio Giraffe or higher is already installed on the PC on which the application is developed.

1. Unzip the Sanei ESC/POS SDK package.

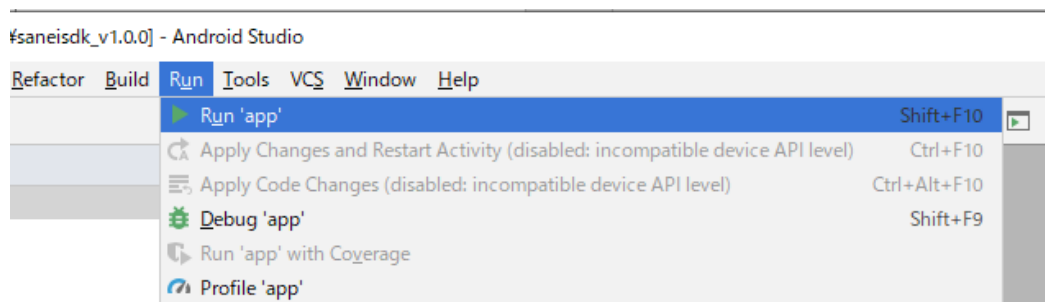


2. Run Android Studio and open the Sanei ESC/POS SDK package.





3. Click “Run” in the menu bar at the top and select “Run 'app'”.



1.2 Supported Printers

Supported printer models		F/W version	Compatible interface	Supported command
KIOSK printers	SK4-21	V1.02.00 or later	USB	MODE-A
	SK4-31	V1.02.00 or later	USB	MODE-A
	SK1-2x1	V2.60.00 or later	USB	MODE-A
	SK1-3x1	V2.60.00 or later	USB	MODE-A
	SK1-21H	V2.60.00 or later	USB	MODE-A
	SK1-31H	V2.60.00 or later	USB	MODE-A
	SK1-41	V3.01.00 or later	USB	MODE-A
	SK5-31	V1.00.00 or later	USB,LAN	MODE-A
Desktop printers	SD3-22	V1.05.00 or later	USB	MODE-A
	SD3-21	V1.05.00 or later	USB	MODE-A
Mobile printer	SM4-21	V1.01.00 or later	USB, BLE, WLAN	MODE-A
	SM4-31	V1.01.00 or later	USB, BLE, WLAN	MODE-A
Panel printer	SP1-21	V1.54.00 or later	USB	MODE-A

★ Memory switch settings

When using the Sanei ESC/POS Android SDK, set the memory switch settings for all printer models as follows.

OFFLINE BUSY = OFF

Act. For Driver = INVALID

1.3 Related Software

None in the current version

1.4 Installing Libraries

1. In the Sanei ESC/POS Android SDK, set the API level on the application side to 15 or higher.

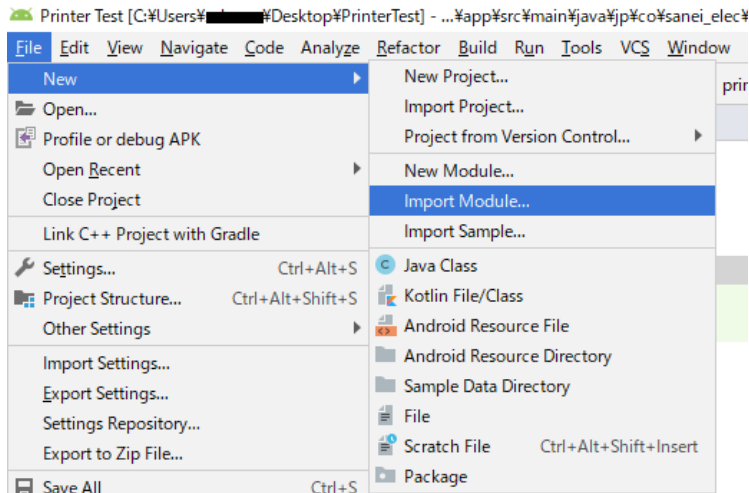
You can check the value of “minSdkVersion” by opening the “build.gradle” of the application.

```

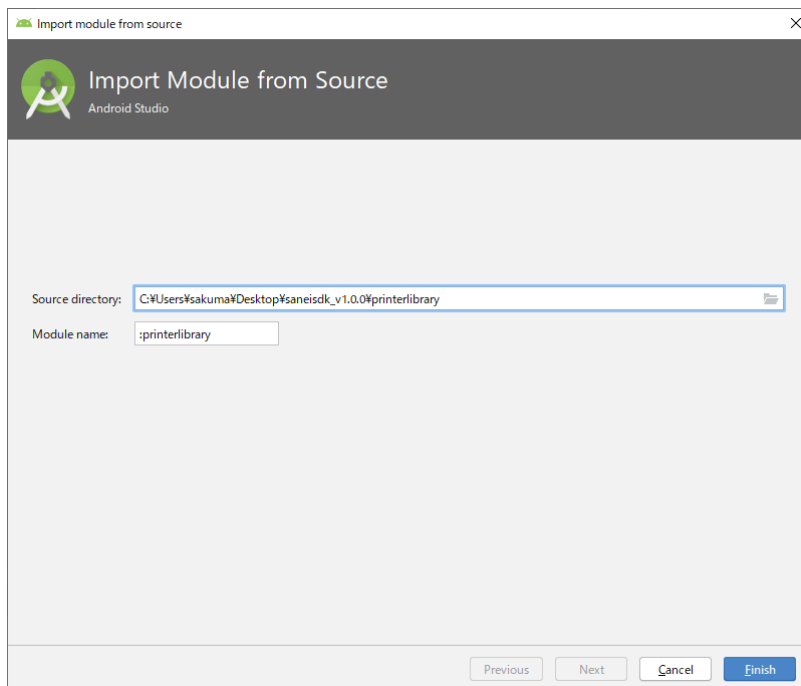
1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 29
5      defaultConfig {
6          applicationId "jp.co.sanei_elec.printertest"
7          minSdkVersion 15
8          targetSdkVersion 29
9          versionCode 1
10         versionName "1.0"
11         testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
12     }
13     buildTypes {
14         release {
15             minifyEnabled false
16             proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
17         }
18     }
19 }
20

```


2. Select Android Studio File menu → New → Import Module.



3. Specify the “printerlibrary” folder in the unzipped folder in the Source Directory.




4. Add the following line to “dependencies” in the “build.gradle” of the application.

[implementation project(':printerlibrary')]

```
21 dependencies {
22     implementation project(':printerlibrary')
23     implementation fileTree(dir: 'libs', include: ['*.jar'])
24     implementation 'androidx.appcompat:appcompat:1.1.0'
25     implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
26     testImplementation 'junit:junit:4.12'
27     androidTestImplementation 'androidx.test:runner:1.2.0'
28     androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
29 }
```

5. Add the following import specification to the source you want to use.

```
import jp.co.sanei_elec.printerlibrary.api.BarcodeSystem;
import jp.co.sanei_elec.printerlibrary.api.HRI;
import jp.co.sanei_elec.printerlibrary.api.HorizontalAlignment;
import jp.co.sanei_elec.printerlibrary.api.InternationalCharset;
import jp.co.sanei_elec.printerlibrary.api.PageDirection;
import jp.co.sanei_elec.printerlibrary.api.PrinterDevice;
import jp.co.sanei_elec.printerlibrary.api.PrinterListener;
import jp.co.sanei_elec.printerlibrary.api.PrinterStatus;
```



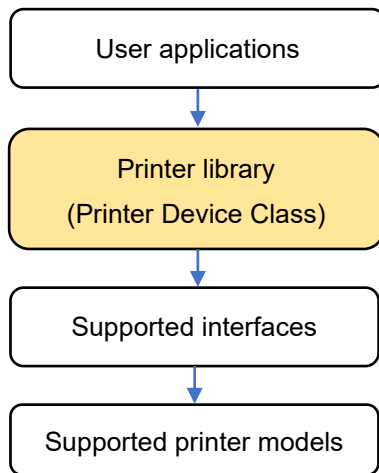
```
1 package jp.co.sanei_elec.printertest;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import jp.co.sanei_elec.printerlibrary.api.BarcodeSystem;
6 import jp.co.sanei_elec.printerlibrary.api.HRI;
7 import jp.co.sanei_elec.printerlibrary.api.HorizontalAlignment;
8 import jp.co.sanei_elec.printerlibrary.api.InternationalCharset;
9 import jp.co.sanei_elec.printerlibrary.api.PageDirection;
10 import jp.co.sanei_elec.printerlibrary.api.PrinterDevice;
11 import jp.co.sanei_elec.printerlibrary.api.PrinterListener;
12 import jp.co.sanei_elec.printerlibrary.api.PrinterStatus;
13
14 import android.os.Bundle;
15
16 public class MainActivity extends AppCompatActivity {
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_main);
22     }
23 }
24
```

6. You will go to use the SDK.

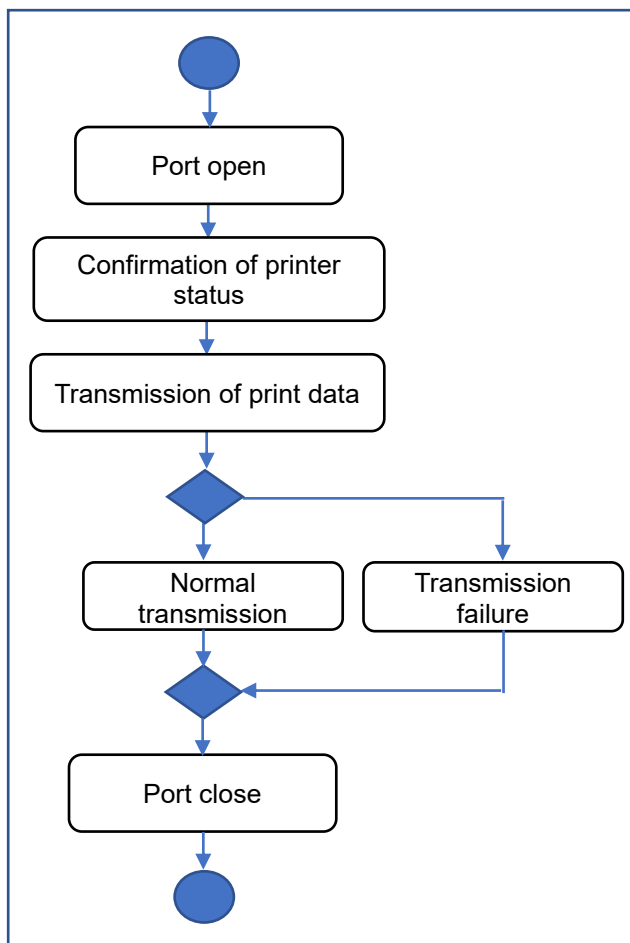
2. Printer Device Class

The overall configuration using the Printer Device Class and the process up to printing are shown below.

- Overall configuration



- Process up to printing



Images of sample codes:

```
// port connection
If (ConnectPrinter ("USB:SK1-31:00") == true)
{
    // confirmation of printer status
    int[] sts = GetStatus();

    If (sts[0] == Ready)
    {
        // transmission of print data
        If (PrintString ("Hello World!¥n") == false)
        {
            System.out.println("Print Error");
        }
    }
}

// port close
DisconnectPrinter();
}
```

3. Printer Device Class Methods

The list of printer device class methods is as follows.

Category	Method	Function
Connect Disconnect	discoverUsbPrinter	Discovers the names of USB-connected printer ports.
	discoverBlePrinter	Discovers the names of Bluetooth (BLE) connected printer port.
	connectPrinter (For USB)	Connects to the specified printer port to be USB connected.
	connectPrinter (For BLE)	Connects to the specified printer port to be Bluetooth (BLE) connected.
	connectPrinter (For WLAN)	Connects to the printer port at specified IP address and port number.
	disconnectPrinter	Disconnects the specified printer port.
	isPrinterConnected	Checks whether the device is connected to all device printer port.
	isSupportedPrinterDevice	Checks whether the Bluetooth (BLE) connected device is supportable printer.
Standard printing	printString	Specifies the character string with Unicode (kanji characters are also supported) and prints out the printer font.
	setFontStyle	Sets character modifiers for the printer fonts.
	setFontType	Sets the printer font type (font A or font B).
	setFontMagnification	Sets the printer font size (vertical multiple and horizontal multiple).
	setFontColor	Specifies or cancels white/black inversion for printer fonts.
	setFontSmoothing	Specifies or cancels smoothing of printer fonts for which vertical or horizontal multiples have been specified.
	setAlignment	Specifies the print position for print data (left alignment, centering, and right alignment).
	setCodePage	Specifies the code page for the printer font.
	setInternationalChar	Specifies the international characters for the printer font.
	printBarcode	Prints out the 1D barcode.
	printQR	Prints out the QR code.
	printPDF417	Prints out PDF417.
	setBarcodeModification	Sets the 1D bar code modification information (height, HRI character).
	printBitmap	Converts the Android class bitmap object into bit image format and prints.
	printBitmap	Converts the data in bit map array format into bit image format and prints.
	printBitmapRaster	Converts the data in bit map array format into a raster image format and prints.
	lineFeed	Inserts the specified number of carriage returns (line feeds).
	paperFeed	Executes paper feeding in the forward or reverse direction for the specified number of dot lines.
	printStringAndroidFont	Converts the specified character string into Android font and prints in bit image format.
	initPrinter	Initializes the modification information set in the printer.
Page printing	startPageMode	Shifts from standard mode into page mode.
	endPageMode	Prints the specified area of the page memory and shifts from page mode to standard mode.
	setPageAttribute	Sets the page mode attribution information (printing direction and printing area).
Status	getStatus	Gets the printer status into the printer status object.
Binary	outputRawdata	Sends binary data in byte array format to the printer port.
	outputRawdata	Sends 1-byte binary data to the printer port.
	inputRawdata	Gets binary data read into byte array format from the printer port.
	inputRawdata	Gets 1-byte binary data from the printer port.

3.1 discoverUsbPrinter method

Discovers the USB-connected printer port names.

Declaration:

```
public android.hardware.usb.UsbDevice[] discoverUsbPrinter(int timeoutMillis)
```

Argument:

timeoutMillis	Internal control and timeout time in this method
---------------	--

Return value:

UsbDevice[]	Name of the USB-connected printer port
-------------	--

Remarks:

Get all USB-connected printer ports that are allowed to be used.

3.1.1 discoverBlePrinter method

Discovers the Bluetooth (BLE) connected printer port names.

Declaration :

```
public android.bluetooth.BluetoothDevice[] discoverBlePrinter(int timeoutMillis)
```

Argument:

timeoutMillis	Internal control and timeout time in this method
---------------	--

Return value:

UsbDevice[]	Name of the Bluetooth (BLE) connected printer port
-------------	--

Remarks:

Get all Bluetooth (BLE) connected printer ports that are allowed to be used.

3.2 connectPrinter method (For USB)

Connects to the specified printer port to be USB connected.

Declaration:

```
public void connectPrinter (android.hardware.usb.UsbDevice device)
```

Argument:

device One USB-connected printer port

Remarks:

java.io.IOException When an error occurs in the connected printer port, an exception is notified.

3.2.1 connectPrinter method (For BLE)

Connects to the specified printer port to be Bluetooth (BLE) connected.

Declaration:

```
public void connectPrinter (android.bluetooth.BluetoothDevice device)
```

Argument:

device One Bluetooth (BLE) connected printer port

Remarks:

java.io.IOException When an error occurs in the connected printer port, an exception is notified.

3.2.2 connectPrinter method (For WLAN)

Connects to the printer port at specified IP address and port number.

Declaration:

```
public void connectPrinter (String host, int port);
```

Argument:

host Specify IP address (Input example 192.168.127.10, specify "192.168.127.10")

port Specify port number (Input example 9100, specify 9100)

Remarks:

java.io.IOException When an error occurs in the connected printer port, an exception is notified.

3.3 disconnectPrinter method

Disconnects the specified printer port.

Declaration:

```
public void disconnectPrinter()
```

Argument: None

Return Value: None

3.4 isPrinterConnected method

Checks whether the printer port is connected.

Declaration:

```
public boolean isPrinterConnected()
```

Argument: None

Return Value: true Connected to the printer port.
 false Not connected to the printer port.

3.5 isSupportedPrinterDevice method (For BLE)

Checks whether the Bluetooth (BLE) connected device is supportable printer.

Declaration:

```
public boolean isSupportedPrinterDevice(android.bluetooth.BluetoothDevice device)
```

Argument:

device One Bluetooth (BLE) connected printer port

Return Value: true Supportable printer
 false Not supportable printer.

Remarks:

Referring to the internal model list of the Bluetooth device name and checks if it is the supportable printer or not.

3.6 printString method

Specifies a character string with Unicode and prints with a printer font.

Declaration:

```
public void printString(java.lang.String data)
public void printString(java.lang.String data, java.lang.String charsetName)
```

Argument:

data Unicode character string

charsetName Specifies the encoding character set with a character string.

The encoding character set is properly used depending on the device, you can check the "IANA Charset Registry" for the canonical name.

Also, the encoding character set should be supported by an implementation of the Java platform.

Examples of character strings that can be specified
"US-ASCII"
"ISO-8859-1"
"UTF-8"
"UTF-16"
"Extended_UNIX_Code_Packed_Format_Japanese" (*1)

(*1) Specifies it instead EUC-JP.

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.7 setFontStyle method

Sets the character modifier for the printer font.

Declaration:

```
public void setFontStyle(boolean bold, boolean italic, boolean underline)
```

Argument:

bold	Specification (true) or cancellation (false) of bold typeface
italic	Specification (true) or cancellation (false) of italic typeface
underline	Specification (true) or cancellation (false) of underlining

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.
It is also effective for kanji characters in the printer fonts.

3.8 setFontType method

Sets the printer font type (font A or font B).

Declaration:

```
public void setFontType(boolean compact)
```

Argument:

compact	Font A (false), Font B (true)
---------	-------------------------------

Return Value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.
It is also effective for kanji characters in printer fonts.

3.9 setFontMagnification method

Sets the printer font size (vertical multiple and horizontal multiple).

Declaration:

```
public void setFontMagnification(int horizontalRatio, int verticalRatio)
```

Argument:

horizontalRatio Specification of the horizontal multiple between 1 and 8

verticalRatio Specification of the vertical multiple between 1 and 8

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

It is also effective for kanji characters in printer fonts.

3.10 setFontColor method

Specifies or cancels white/black inversion for printer fonts.

Declaration:

```
public void setFontColor(boolean reverse)
```

Argument:

reverse Specification (true) or cancellation (false) of white/black inversion

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

It is also effective for kanji characters in printer fonts.

3.11 setFontSmoothing method

Specifies or cancels the smoothing of printer fonts for which vertical or horizontal multiples have been specified.

Declaration:

```
public void setFontSmoothing(boolean on)
```

Argument:

on Specification (true) or cancellation (false) of smoothing

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.
It is also effective for kanji characters in printer fonts.

3.12 setAlignment method

Specifies the print position (left alignment, centering, and right alignment) for print data.

Declaration:

```
public void setAlignment(HorizontalAlignment alignment)
```

Argument:

alignment Print position (select from HorizontalAlignment class)

HorizontalAlignment
Left: Left alignment
Center: Centering
Right: Right alignment

Return value: None

Remarks:

java.io.IOException If error occurs in the connected printer port, an exception is notified.

3.13 setCodePage method

Specifies the code page for the printer font.

Declaration:

```
public void setCodePage(CodePage codePage)
```

Argument:

codePage Code page (select from CodePage class)

CodePage
katakana
PC1253
PC437
PC737
PC850
PC852
PC857
PC858
PC860
PC862
PC863
PC864
PC865
PC866
WPC1250
WPC1251
WPC1252
WPC1252_2
WPC1254

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.14 setInternationalChar method

Specifies the international character for the printer font.

Declaration:

```
public void setInternationalChar(InternationalCharset internationalCharset)
```

Argument:

internationalCharset International character (select from InternationalCharset class)

InternationalCharset
Denmark
England
France
Germany
Italy
Japan
Spain
Sweden
USA

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.15 printBarcode method

Prints the 1D barcode.

Declaration:

```
public void printBarcode(BarcodeSystem barcodeSystem, byte[] barcode)
public void printBarcode(BarcodeSystem barcodeSystem, java.lang.String barcode)
```

Argument:

barcodeSystem Barcode type (select from BarcodeSystem class)

BarcodeSystem
CODABAR
CODE128
CODE39
ITF
JAN13
JAN8
UPCA
UPCE

barcode Barcode data

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

The barcode variable type is properly used depending on the presence or absence of a control code in the barcode data.

3.16 printQR method

Prints the QR code.

Declaration:

```
public void printQR(int size, int eccLevel, byte[] barcode)
public void printQR(int size, int eccLevel, java.lang.String barcode)
```

Argument:

size Specify the symbol size (1-40).

eccLevel Specify the error control level (1 to 4).

1: L (7%)

2: M (15%)

3: Q (25%)

4: H (30%)

barcode Barcode data

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

The barcode variable type is properly used depending on the presence or absence of a control code in the barcode data.

3.17 printPDF417 method

Prints PDF417.

Declaration:

```
public void printPDF417(boolean truncate,boolean binaryEncode, int eccLevel, int size, byte[] barcode)
public void printPDF417(boolean truncate,boolean binaryEncode, int eccLevel, int size,
                        java.lang.String barcode)
```

Argument:

truncate Specify (true) of not specify (false) as (compact) PDF417

binaryEncode Byte encoding mode (true) or automatic encoding mode (false)

eccLevel The error correction level (0 to 7) is specified.

size The barcode size is specified as shown in the combination table.

size	Details (X: Row / Y = Step)	size	Details (X: Row / Y = Step)
0	X 2: Y 4	8	X 12: Y 4
1	X 2: Y 9	9	X 12: Y 9
2	X 2: Y 15	10	X 12: Y 15
3	X 2: Y 20	11	X 12: Y 20
4	X 7: Y 4	12	X 20: Y 4
5	X 7: Y 9	13	X 20: Y 9
6	X 7: Y 15	14	X 20: Y 15
7	X 7: Y 20	15	X 20: Y 20

barcode Barcode data

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

The barcode variable type is properly used depending on the presence or absence of a control code in the barcode data.

3.18 setBarcodeModification method

Sets the 1D barcode modification information (height, HRI characters).

Declaration:

```
public void setBarcodeModification(HRI hri, int width, int height)
```

Argument:

hri HRI character modification (select from HRI class)

HRI
Above: HRI characters are printed above the barcode.
Both: HRI characters are printed above and below the barcode.
None: HRI characters are not printed.
Under: HRI characters are printed below the barcode.

width The barcode module width (1 to 4) is specified.

height The barcode height (1-255 dot pitch) is specified.

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.19 printBitmap method

An android bitmap class object is converted into a bit image command and printed.

Declaration:

```
public void printBitmap(android.graphics.Bitmap bitmap)
```

```
public void printBitmap(android.graphics.Bitmap bitmap,HorizontalAlignment align,boolean dither)
```

Argument:

bitmap Bitmap data

alignment Print position (select from HorizontalAlignment class)

HorizontalAlignment
Left: Left alignment
Center: Centering
Right: Right alignment

dither true Execution of Floyd Steinberg image dithering is specified.
 false Execution of image dithering is no specified.

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

Specifying only the bitmap specification works with alignment = Left and dither = true.

3.20 printBitmap method

An int array image data is converted into a bit image command and printed.

Declaration:

```
public void printBitmap(int[] argb, int width, int height, HorizontalAlignment align, boolean dither)
```

Argument:

argb Int array image data (1 dot = 32-bit color)

width Image data width

height image data height

alignment Print position (select from HorizontalAlignment class)

HorizontalAlignment
Left: Left alignment
Center: Centering
Right: Right alignment

dither true Execution of Floyd Steinberg image dithering is specified.

 false Execution of image dithering is no specified.

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.21 printBitmapRaster method

An int array image data is converted into a raster bit image command and printed.

Declaration:

```
public void printBitmapRaster(int[] argb, int width, int height, HorizontalAlignment align,  
                             boolean dither, boolean crop)
```

Argument:

argb Int array image data (1 dot = 32-bit color)

width Image data width

height image data height

alignment Print position (select from HorizontalAlignment class)

HorizontalAlignment
Left: Left alignment
Center: Centering
Right: Right alignment

dither true Execution of Floyd Steinberg image dithering is specified.
false Execution of image dithering is no specified.

crop true Excess margin is deleted when converting to a bit image.
false Excess margin is not deleted

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.22 lineFeed method

Specified number of carriage returns (line feeds) is executed.

Declaration:

```
public void lineFeed(int lines)
```

Argument:

lines Number of carriage returns (line feeds) is specified (1 to 255).

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

In the standard mode, the data in the line buffer is printed and the paper is fed forward by the specified number of carriage returns (line feeds).

In page mode, the data is shifted in the y-axis direction by the specified number of carriage returns (line feeds).

3.23 paperFeed method

Paper feed is executed in the forward or reverse direction by the specified number of dot lines.

Declaration:

```
public void paperFeed(int lines)
```

Argument:

lines Paper feed is specified by a dot pitch value of -255 to 255.

If the parameter is negative, the command executes printing and reverse paper feeding.

If the parameter is an integer, printing and forward paper feeding are executed.

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

In the standard mode, the data in the line buffer is printed, and the paper is fed forward by the specified dot pitch. In page mode, the data is shifted in the y-axis direction by the specified dot pitch.

3.24 printStringAndroidFont method

A specified character string is converted into an Android font and printed as a bit image.

Declaration:

```
public void printStringAndroidFont(java.lang.String text, int x, int y, android.graphics.Paint paint)
```

Argument:

text A character string is specified.

x Horizontal print position

y Vertical print position

paint Android Paint class.
 Android font attributes (size, style, color, etc.) are specified.

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.25 startPageMode method

Shift from standard mode to page mode.

Declaration:

```
public void startPageMode()
```

Argument: None

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.26 endPageMode method

The specified area of page memory is printed and the page mode is shifted to standard mode.

Declaration:

```
public void endPageMode()
```

Argument: None

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.27 setPageAttribute method

The specified area (print direction, print area) in page mode is set.

Declaration:

```
public void setPageAttribute(int x, int y, int width, int height)
public void setPageAttribute(int x, int y, int width, int height, PageDirection direction)
```

Argument:

x Starting point of the print area (x axis)

y Starting point of the print area (y axis)

width Horizontal width of the starting point of the print area (x axis)

height Height of the starting point of the print area (y axis)

direction Direction of expansion within the print area (select from PageDirection class)

PageDirection
Normal: Forward direction
Clockwise270: 90-degree rotation to the left
Clockwise180: Reverse direction
Clockwise90: 90-degree rotation to the right

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

The print area can be specified any number of times until the page mode is printed.

When printing in page mode, the printer automatically prints within the maximum print area at that time.

The specifiable range of the print area depends on the printer model and MSW settings.

3.28 getStatus method

The printer status is acquired as a printerStatus class object.

Declaration:

```
public PrinterStatus getStatus()  
public PrinterStatus getStatus(int timeoutMillis)
```

Argument:

timeoutMillis Internal control and timeout time in this method

Return value:

PrinterStatus PrinterStatus class

PrinterStatus	
status1	STATUS_1_ERROR STATUS_1_MOVING STATUS_1_WAITING
status2	STATUS_2_NORMAL STATUS_2_DEVICE_ERROR STATUS_2_HEAD_OPEN STATUS_2_NEAR_END STATUS_2_PAPER_EMPTY STATUS_2_PAPER_IN_BEZEL STATUS_2_PULL_OUT_THE_PAPER STATUS_2_LOWBATTERY
status3	STATUS_3_NO_ERROR STATUS_3_PAPER_JAM_ERROR STATUS_3_TEMPERATURE_ERROR STATUS_3_VOLTAGE_ERROR
status4	0 (Reserved)

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

For details on the printer status, refer to “Appendix 1. Printer Status.”

3.29 initPrinter method

Modification information set in the printer is initialized.

Declaration:

```
public void initPrinter()
```

Argument: None

Return value: None

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.30 outputRawData method

Binary data in byte array format is sent to the printer port.

Declaration:

```
public void outputRawData(byte[] output)
```

Argument:

Output	Binary data in byte array format
--------	----------------------------------

Return value:	None
---------------	------

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.31 outputRawData method

1-byte binary data is sent to the printer port.

Declaration:

```
public void outputRawData(int b)
```

Argument:

b	1-byte binary data
---	--------------------

Return value:	None
---------------	------

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.32 inputRawData method

Binary data in byte array format is acquired from the printer port.

Declaration:

```
public int inputRawData(byte[] input, int timeoutMillis)
public int inputRawData(byte[] input, int offset, int length, int timeoutMillis)
```

Argument:

input	Binary data in byte array format that is to be read.
offset	Start offset for the byte array format
length	Maximum number of bytes of data that has been read that can be written
timeoutMillis	Internal control and timeout time in this method

Return value:

Total number of bytes written to the input variable (buffer) is returned.

Remarks:

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

3.33 inputRawData method

1-byte binary data is acquired from the printer port.

Declaration:

```
public int inputRawData(int timeoutMillis)
```

Argument:

timeoutMillis	Internal control and timeout time in this method
---------------	--

Return value: Binary data acquired from the printer port

Remarks::

java.io.IOException If an error occurs in the connected printer port, an exception is notified.

Appendix 1. Printer Status

The printer status refers to the condition of the printer consisting of 4 bytes of data that is returned when the PrinterStatus method is requested.

The host always identifies the status of the printer with all bytes received using this method.

Also, the response values that can be supported by these printer statuses vary depending on the printer model and its configuration.

The printer models and status types that can be supported are shown in this chapter.

SK4-21/31:

PrinterStatus		SK4-21 SK4-31	with Bezel (Bezel Mode=A or C)	with Bezel (Bezel Mode=B)
status1	STATUS_1_ERROR	O	O	O
	STATUS_1_MOVING	O	O	O
	STATUS_1_WAITING	O	O	O
status2	STATUS_2_NORMAL	O	O	O
	STATUS_2_DEVICE_ERROR	O	O	O
	STATUS_2_HEAD_OPEN	O	O	O
	STATUS_2_NEAR_END	O	O	O
	STATUS_2_PAPER_EMPTY	O	O	O
	STATUS_2_PAPER_IN_BEZEL	--	O	--
	STATUS_2_PULL_OUT_THE_PAPER	--	--	O
	STATUS_2_LOWBATTERY	--	--	--
status3	STATUS_3_NO_ERROR	O	O	O
	STATUS_3_PAPER_JAM_ERROR	--	--	O
	STATUS_3_TEMPERATURE_ERROR	O	O	O
	STATUS_3_VOLTAGE_ERROR	O	O	O
status4	0 (Reserved)	O	O	O

SK1-2x1/3x1 (SK1-21H/31H):

PrinterStatus		SK1-2x1 SK1-3x1	with Bezel (Bezel Mode=A or C)	with Bezel (Bezel Mode=B)	with Presenter
status1	STATUS_1_ERROR	O	O	O	O
	STATUS_1_MOVING	O	O	O	O
	STATUS_1_WAITING	O	O	O	O
status2	STATUS_2_NORMAL	O	O	O	O
	STATUS_2_DEVICE_ERROR	O	O	O	O
	STATUS_2_HEAD_OPEN	O	O	O	O
	STATUS_2_NEAR_END	O	O	O	O
	STATUS_2_PAPER_EMPTY	O	O	O	O
	STATUS_2_PAPER_IN_BEZEL	--	O	--	--
	STATUS_2_PULL_OUT_THE_PAPER	--	--	O	O
	STATUS_2_LOWBATTERY	--	--	--	--
status3	STATUS_3_NO_ERROR	O	O	O	O
	STATUS_3_PAPER_JAM_ERROR	--	--	O	O
	STATUS_3_TEMPERATURE_ERROR	O	O	O	O
	STATUS_3_VOLTAGE_ERROR	O	O	O	O
status4	0 (Reserved)	O	O	O	O

SK1-41:

PrinterStatus		SK1-41	with Presenter
status1	STATUS_1_ERROR	O	O
	STATUS_1_MOVING	O	O
	STATUS_1_WAITING	O	O
status2	STATUS_2_NORMAL	O	O
	STATUS_2_DEVICE_ERROR	O	O
	STATUS_2_HEAD_OPEN	O	O
	STATUS_2_NEAR_END	O	O
	STATUS_2_PAPER_EMPTY	O	O
	STATUS_2_PAPER_IN_BEZEL	--	--
	STATUS_2_PULL_OUT_THE_PAPER	--	O
	STATUS_2_LOWBATTERY	--	--
status3	STATUS_3_NO_ERROR	O	O
	STATUS_3_PAPER_JAM_ERROR	--	O
	STATUS_3_TEMPERATURE_ERROR	O	O
	STATUS_3_VOLTAGE_ERROR	O	O
status4	0 (Reserved)	O	O

SD3-21/22:

PrinterStatus		SD3-21	SD3-22
status1	STATUS_1_ERROR	O	O
	STATUS_1_MOVING	O	O
	STATUS_1_WAITING	O	O
status2	STATUS_2_NORMAL	O	O
	STATUS_2_DEVICE_ERROR	O	O
	STATUS_2_HEAD_OPEN	O	O
	STATUS_2_NEAR_END	O	O
	STATUS_2_PAPER_EMPTY	O	O
	STATUS_2_PAPER_IN_BEZEL	--	--
	STATUS_2_PULL_OUT_THE_PAPER	--	--
	STATUS_2_LOWBATTERY	--	--
status3	STATUS_3_NO_ERROR	O	O
	STATUS_3_PAPER_JAM_ERROR	--	--
	STATUS_3_TEMPERATURE_ERROR	O	O
	STATUS_3_VOLTAGE_ERROR	O	O
status4	0 (Reserved)	O	O

SM4-21/31:

PrinterStatus		SM4-21	SM4-31
status1	STATUS_1_ERROR	O	O
	STATUS_1_MOVING	O	O
	STATUS_1_WAITING	O	O
status2	STATUS_2_NORMAL	O	O
	STATUS_2_DEVICE_ERROR	O	O
	STATUS_2_HEAD_OPEN	--	O
	STATUS_2_NEAR_END	--	--
	STATUS_2_PAPER_EMPTY	O	O
	STATUS_2_PAPER_IN_BEZEL	--	--
	STATUS_2_PULL_OUT_THE_PAPER	--	--
	STATUS_2_LOWBATTERY	O	O
status3	STATUS_3_NO_ERROR	O	O
	STATUS_3_PAPER_JAM_ERROR	--	--
	STATUS_3_TEMPERATURE_ERROR	O	O
	STATUS_3_VOLTAGE_ERROR	O	O
status4	0 (Reserved)	O	O

SP1-21:

PrinterStatus		SP1-21
status1	STATUS_1_ERROR	O
	STATUS_1_MOVING	O
	STATUS_1_WAITING	O
status2	STATUS_2_NORMAL	O
	STATUS_2_DEVICE_ERROR	O
	STATUS_2_HEAD_OPEN	O
	STATUS_2_NEAR_END	O
	STATUS_2_PAPER_EMPTY	O
	STATUS_2_PAPER_IN_BEZEL	--
	STATUS_2_PULL_OUT_THE_PAPER	--
	STATUS_2_LOWBATTERY	--
status3	STATUS_3_NO_ERROR	--
	STATUS_3_PAPER_JAM_ERROR	--
	STATUS_3_TEMPERATURE_ERROR	--
	STATUS_3_VOLTAGE_ERROR	--
status4	0 (Reserved)	O

SK5-31:

PrinterStatus		SK5-31
status1	STATUS_1_ERROR	O
	STATUS_1_MOVING	O
	STATUS_1_WAITING	O
status2	STATUS_2_NORMAL	O
	STATUS_2_DEVICE_ERROR	O
	STATUS_2_HEAD_OPEN	O
	STATUS_2_NEAR_END	O
	STATUS_2_PAPER_EMPTY	O
	STATUS_2_LOWBATTERY	--
	STATUS_2_PAPER_IN_BEZEL	--
	STATUS_2_PULL_OUT_THE_PAPER	O
status3	STATUS_3_NO_ERROR	O
	STATUS_3_PAPER_JAM_ERROR	O
	STATUS_3_TEMPERATURE_ERROR	O
	STATUS_3_VOLTAGE_ERROR	O
status4	0 (Reserved)	O

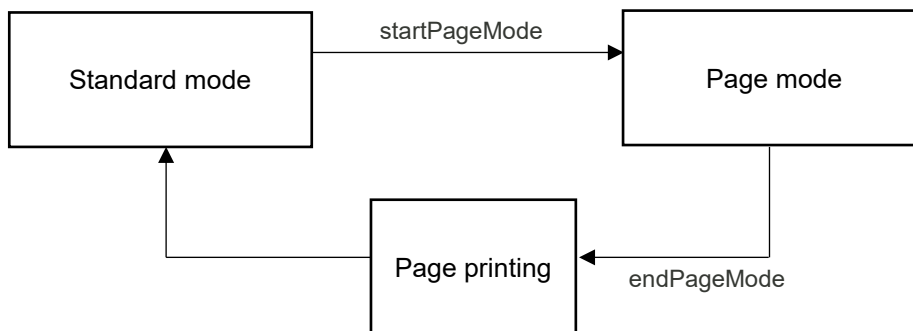
Appendix 2. Page Mode

This printer has two print modes: Standard mode and page mode.

The standard mode (the printer starts from this mode when the power is turned on) is a mode where printing is performed each time a print method is sent.

In page mode, print data is written into the page memory area without performing a print operation even if a print method is sent. By sending the endPageMode method, the page memory area is printed all at once and returns to the standard mode.

The relationship between page mode and standard mode is as follows.



1. Page area

The arguments specified with the setPageAttribute method vary depending on the printer model and print width. The settable range is shown in the table below.

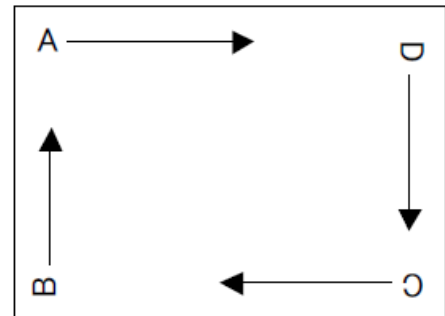
(Value: dot lines / pitch)

Supported model	Print width	x (starting point)	y (starting point)	Width (horizontal width)	Height (height)
SK4-31	72mm	0 – 574	0 – 2398	1 – 575	1 – 2399
SK4-21	54mm	0 – 430	0 – 2398	1 – 431	1 – 2399
SK1-3x1 (SK1-31H)	80mm	0 – 638	0 – 2398	1 – 639	1 – 2399
	72mm	0 – 574	0 – 2398	1 – 575	1 – 2399
SK1-2x1 (SK1-21H)	56mm	0 – 446	0 – 2398	1 – 447	1 – 2399
	54mm	0 – 430	0 – 2398	1 – 431	1 – 2399
SK1-41	104mm	0 – 830	0 – 2798	1 – 831	1 – 2799
SD3-22	54mm	0 – 430	0 – 1598	1 – 431	1 – 1599
SD3-21	48mm	0 – 382	0 – 1598	1 – 383	1 – 1599
SM4-21	48mm	0 – 382	0 – 2398	1 – 383	1 – 2399
SM4-31	72mm	0 – 574	0 – 2398	1 – 575	1 – 2399
SP1-21	48mm	0 – 382	0 – 926	1 – 383	1 – 927
SK5-31	80mm	0 – 638	0 – 2398	1 – 639	1 – 2399
	72mm	0 – 574	0 – 2398	1 – 575	1 – 2399

2. Printing direction and the starting point

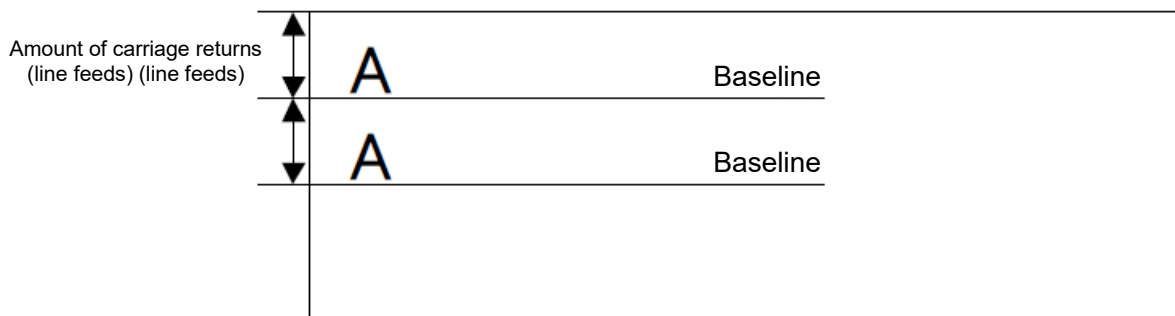
The starting point and the direction of expansion of the print data selected by the PageDirection object are shown as follows.

PageDirection	Start point and expansion direction
Normal: Forward direction	A
Clockwise270: 90-degree rotation to the left	B
Clockwise180: Reverse direction	C
Clockwise90: 90-degree rotation to the right	D

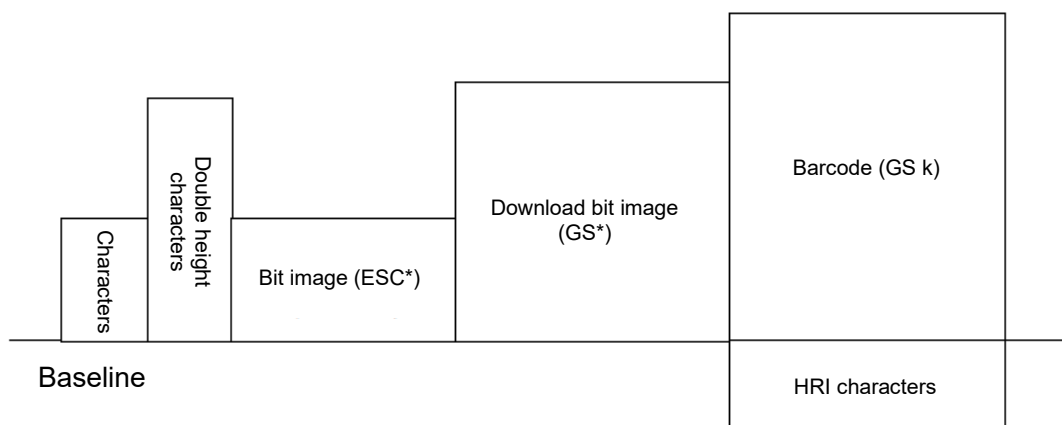


3. Expansion in page mode

The baseline and print data expansion position accompanying carriage returns (line feeds) (line feeds) in printer fonts, bit images, barcodes, etc. in page mode are shown as follows.



Character data expansion position



Print data expansion position

Appendix 3. Confirmation of SDK version

This section describes how to check the SDK version in the program. For checking the SDK version, use the BuildConfig class.

```
String ver = "[" + jp.co.sanei_elec.printerlibrary.BuildConfig.VERSION_NAME + "];
```

Version screen example:

