

ESC SDK v1.0.03Interface documentation

Document change history

一.Directions for use

- 1.1 Project configuration description
- 1.2 Obfuscated configurations
- 1.3 Initialize

二.Sample column demonstration

- 2.1Connect Printer Process
 - 2.1.1 Bluetooth Search Printer
 - 2.1.2 Connect Printer

三.Interface Description

PrinterHelper Introduction

- 3.1.2 Bluetooth connection
- 3.1.3 WIFI connection
- 3.1.4 USB connection
- 3.1.5 Serial port connection
- 3.1.6 Connection status
- 3.1.7 Disconnect
- 3.2.1 Get Printer Flags
- 3.2.2 Set the SDK character encoding
- 3.2.3 Send print data to the printer
- 3.2.4 Add data to the printer
- 3.2.5 Read data from the printer
- 3.2.6 Add Text
- 3.2.7 Add Text Data of Table Type
- 3.2.8 Add Alignment
- 3.2.9 Add Font Style
- 3.2.10 Font Enlargement
- 3.2.11 Add Picture

- 3.2.12 Add Barcode
- 3.2.13 Add QR Code
- 3.2.14 Add PDF417
- 3.2.15 Add Page Mode
- 3.2.16 Adding Clear Page Mode Cache Data
- 3.2.17 Adding a Page Mode Print Area
- 3.2.18 Adding Page Mode Print Orientation
- 3.2.19 Add Page Mode Absolute Position
- 3.2.20 Add Page Mode Printing
- 3.2.21 Add Positioning
- 3.2.22 Add paper feed
- 3.2.23 Adding Line Spacing
- 3.2.24 Adding Printer Codes
- 3.2.25 Initializing the Printer
- 3.2.25 Adding printer density
- 3.2.26 Adding Print Speed
- 3.2.27 Add paper cutter
- 3.2.28 Add Open Cash Box
- 3.2.29 Add open buzzer
- 3.2.30 Add Left Margin
- 3.2.31 Get Printer Status
- 3.2.32 Obtaining the Printer Serial Number
- 3.2.33 Get Printer Power
- 3.2.34 Setting the Print End Switch
- 3.2.35 Acquisition Print End
- 3.2.36 Two Color Print
- 3.2.37 Set Print Width
- 3.2.38 Get Print Width
- 3.2.39 LAN Search Printer
- 3.2.40 LAN Configuration Ethernet Parameters
- 3.2.41 LAN configuration WIFI AP mode parameters
- 3.2.42 LAN configuration WIFI STA mode parameters

- 4.1.1 Coding Table
- 4.1.2 Barcode Type
- 4.1.3 PDF417 Error Correction Level
- 4.1.4 Turn on the SDK to send data logs

Document change history

serial number	Version number	Modifications	Modified by	Date modified
01	v1.0.0	Documentation builds	QX	2024-11-15
02	v1.0.01	Add dual-color printing	QX	2025-03-03
03	v1.0.02	Add Print Width	QX	2025-04-21
04	v1.0.03	Add network config	QX	2025-06-13

—Directions for use

1.1 Project configuration description

1. The SDK must run on the Android system
2. Put the ESC_SDK_Vx.x.x.jar and xxxx.so 、udplibrary-release.aar into the project libs directory, and add the jar aar to build.gradle

▼ build.gradle config Java |

1 implementation fileTree(include: ['*.jar','*.aar'], dir: 'libs')

3. Bluetooth permission configuration

▼ AndroidManifest.xml config

XML |

```
1 <uses-permission android:name="android.permission.BLUETOOTH" />
2 <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

3 <uses-permission android:name="android.permission.BLUETOOTH_CONNECT"/>
4 <uses-permission android:name="android.permission.BLUETOOTH_SCAN"/>
5 <uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE"/>
6 </uses-permission>
```

▼ Dynamically configure targeting permissions in your code

Kotlin |

```
1 disposable = RxPermissions(this).request(
2     Manifest.permission.BLUETOOTH,
3     Manifest.permission.ACCESS_FINE_LOCATION,
4     Manifest.permission.ACCESS_COARSE_LOCATION
5 )
6 .subscribe {
7     if (it) {
8     }}
```

1.2 Obfuscated configurations

1.proguard-rules.pro to prevent the SDK from being obfuscated

▼ Don't confuse the classes below

Kotlin |

```
1 -keep class com.prt.esc.PrinterHelper{*;}
2 -keep class com.prt.esc.printer.Printer{*;}
```

1.3 Initialize

1.Application call the following code in

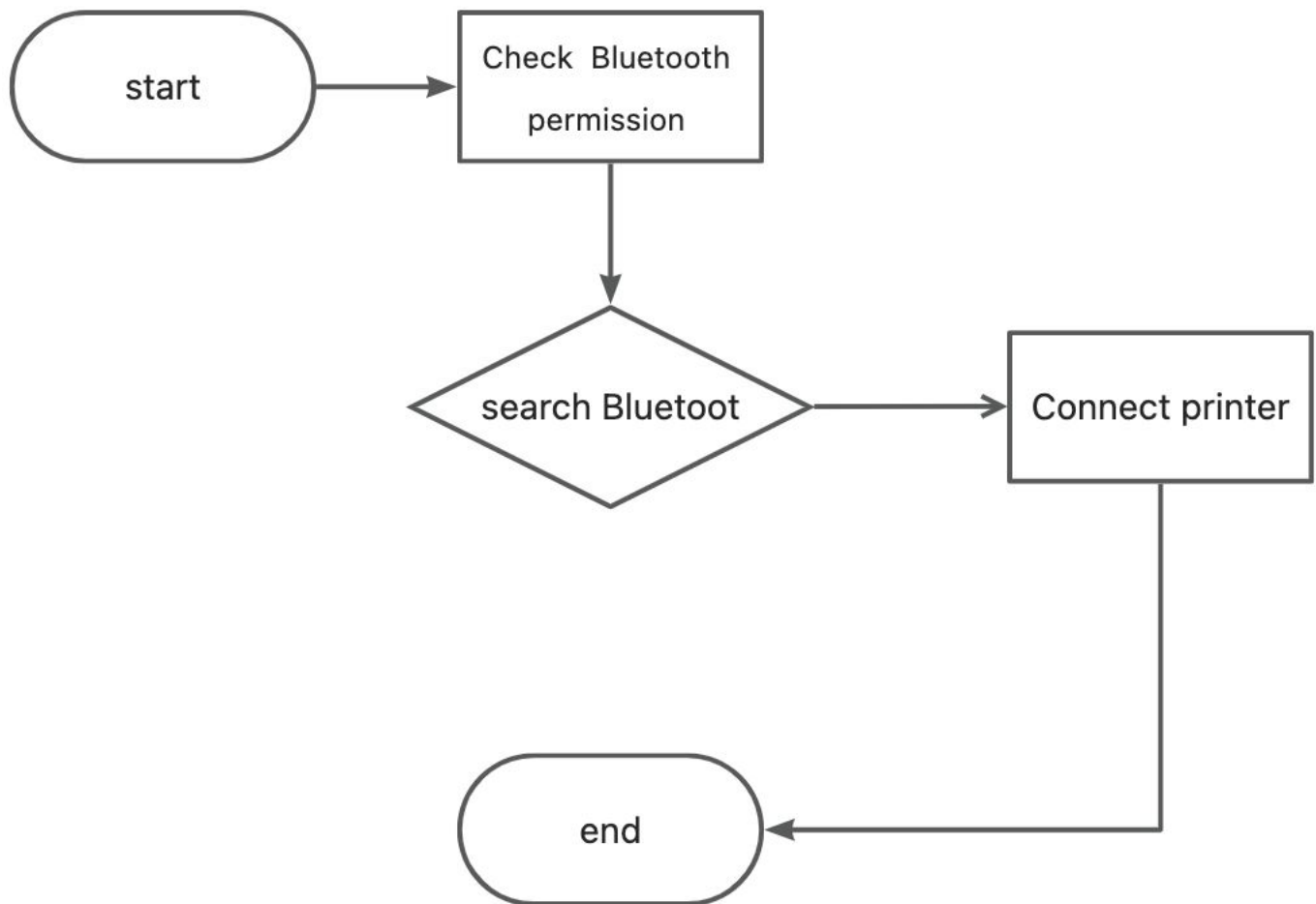
▼ initialize

Java |

```
1 PrinterHelper.init(this);
```

二.Sample column demonstration

2.1Connect Printer Process



2.1.1 Bluetooth Search Printer

▼ register Broadcast

Java |

```
1 private void registerBroadcast() {  
2     IntentFilter intent = new IntentFilter();  
3     intent.addAction(BluetoothDevice.ACTION_FOUND);  
4     intent.addAction(BluetoothDevice.ACTION_BOND_STATE_CHANGED);  
5     intent.addAction(BluetoothAdapter.ACTION_DISCOVERY_FINISHED);  
6     context.registerReceiver(mReceiver, intent);  
7 }  
8
```

```
1 RxPermissions rxPermissions = new RxPermissions((AppCompatActivity) context);
2 rxPermissions.request(Manifest.permission.BLUETOOTH_ADMIN,
3                       Manifest.permission.BLUETOOTH,
4                       Manifest.permission.ACCESS_FINE_LOCATION)
5 .subscribe(aBoolean -> {
6     if (aBoolean) {
7         if (null == mBluetoothAdapter) {
8             mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
9         }
10        if (!mBluetoothAdapter.isEnabled()) {
11            mBluetoothAdapter.enable();
12        }
13        if (mBluetoothAdapter.isDiscovering()) {
14            mBluetoothAdapter.cancelDiscovery();
15        }
16        mBluetoothAdapter.startDiscovery();//开启搜索
17    } else {
18        Utility.show(context, "no bluetooth permission");
19    }
20 });
```

```
1 private final BroadcastReceiver mReceiver = new BroadcastReceiver() {
2     @SuppressWarnings("MissingPermission")
3     @Override
4     public void onReceive(Context context, Intent intent) {
5         String action = intent.getAction();
6         if (BluetoothDevice.ACTION_FOUND.equals(action)) {
7             BluetoothDevice device =
8                 intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
9             if (device.getBluetoothClass().getMajorDeviceClass() == 1536)
10             {
11                 //The representative is a printer
12             }
13             break;
14         }
15     };
```

2.1.2 Connect Printer

▼ Bluetooth connection

Java |

```
1  Printer printer = PrinterHelper.connectBT(btMac);
```

三.Interface Description

PrinterHelper Introduction

Initialize PrinterHelper at APP startup

▼ Initialize code

Java |

```
1  public class App extends Application {  
2      @Override  
3      public void onCreate() {  
4          super.onCreate();  
5          PrinterHelper.init(this);  
6      }  
7  }
```

3.1.2 Bluetooth connection

- Pay attention to permission configuration

Bluetooth permission configuration

Java

```

1  //added in AndroidManifest.xml file
2  <uses-permission android:name="android.permission.BLUETOOTH" />
3  <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
4  <uses-permission android:name="android.permission.BLUETOOTH_CONNECT"/>
5  <uses-permission android:name="android.permission.BLUETOOTH_SCAN"/>
6  <uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE"/>
7
8  //Then get permissions dynamically in your code
9  RxPermissions rxPermissions = new RxPermissions(this);
10 rxPermissions.request(Manifest.permission.BLUETOOTH_ADMIN,
11                      Manifest.permission.BLUETOOTH,
12                      Manifest.permission.BLUETOOTH_CONNECT,
13                      Manifest.permission.BLUETOOTH_SCAN,
14                      Manifest.permission.ACCESS_FINE_LOCATION)
15 .subscribe(aBoolean -> {
16     if (aBoolean) {
17         //The permission is successfully obtained
18     }
19 });

```

• description

Connect printer via Bluetooth mac address

Bluetooth connection

Java

```

1  public static Printer connectBT(String mac)

```

• parameter

parameter	description
mac	Bluetooth address (uppercase)

• example

▼ Example of a Bluetooth connection

Java |

```
1  Printer printer = PrinterHelper.connectBT(btMac);
2  if(printer == null){
3      //The connection failed
4  }else{
5      //connection successful, all print-related interfaces in the Printer
6  }
```

3.1.3 WIFI connection

- Note Configuration Permissions

▼ wifi Permissions

Java |

```
1  //added in AndroidManifest.xml file
2  <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
3  <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
4  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
5  <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
6  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"
7  />
8  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
9  //You need to get dynamic permissions in your code
10 RxPermissions rxPermissions = new RxPermissions(this);
11 rxPermissions.request(Manifest.permission.ACCESS_COARSE_LOCATION,
12                      Manifest.permission.ACCESS_FINE_LOCATION)
13 .subscribe(aBoolean -> {
14     if (aBoolean) {
15         //The permission was obtained
16     }
17 });
```

- description

Connect through the IP address of the printer

▼ WiFi connection interface

Java |

```
1  public static Printer connectWifi(String address)
```

- parameter

parameter	description
address	printer ip(example: 192.168.1.100)

- **example**

▼ Example of a wifi connection

Java |

```

1  Printer printer = PrinterHelper.connectWifi(ip);
2  if(printer == null){
3      //The connection failed
4  }else{
5      //connection successful, all print-related interfaces in the Printer
6  }

```

3.1.4 USB connection

- **Pay attention to permission configuration**

▼ USB permission configuration

Java |

```

1  //added in AndroidManifest.xml file
2  <uses-feature android:name="android.hardware.usb.host" />
3  <uses-permission android:name="android.permission.USB_PERMISSION" />
4  <uses-permission android:name="android.permission.ACCESS_USB_DEVICE" />
5  <meta-data android:name="android.hardware.usb.action.USB_DEVICE_ATTACHED" /
>

```

- **description**

Connecting the printer via usbdevice

▼ USB connection interface

Java |

```

1  public static Printer connectUSB(UsbDevice usbdevice)

```

- **parameter**

parameter	description
-----------	-------------

- example

- ▼ Example USB connection

Java |

```
1  Printer printer = PrinterHelper.connectUSB(device);
2  if(printer == null){
3      //connection failed
4  }else{
5      //connection successful, all print-related interfaces in the Printer
6  }
```

- ▼ Get USB device

Java |

```
1  //device:You can refer to the demo or refer to the following cod
2  private void connectUSB() {//Go through all the USB devices in the system
3      mUsbManager = (UsbManager) thisCon.getSystemService(Context.USB_SERVICE);
4      HashMap<String, UsbDevice> deviceList = mUsbManager.getDeviceList();
5      Iterator<UsbDevice> deviceIterator = deviceList.values().iterator();
6
7      boolean HavePrinter = false;
8      while (deviceIterator.hasNext()) {
9          device = deviceIterator.next();
10         int count = device.getInterfaceCount();
11         for (int i = 0; i < count; i++) {
12             UsbInterface intf = device.getInterface(i);
13             if (intf.getInterfaceClass() == 7) {//printer type
14                 HavePrinter = true;
15                 if (mPermissionIntent != null) {
16                     //Request USB permissions
17                     mUsbManager.requestPermission(device, mPermissionIntent);
18                 }
19             }
20         }
21     }
22 }
```

▼ Register USB System Permission Broadcast

Java

```
1 Intent intent = new Intent(ACTION_USB_PERMISSION);
2 intent.setPackage(thisCon.getPackageName());
3 if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
4     mPermissionIntent = PendingIntent.getBroadcast(thisCon, 0, intent, FLAG_MUTABLE);
5 } else {
6     mPermissionIntent = PendingIntent.getBroadcast(thisCon, 0, intent, 0);
7 }
8 IntentFilter filter = new IntentFilter(ACTION_USB_PERMISSION);
9 filter.addAction(UsbManager.ACTION_USB_DEVICE_DETACHED);
10 filter.addAction(BluetoothDevice.ACTION_ACL_DISCONNECTED);
11 filter.addAction(BluetoothAdapter.ACTION_STATE_CHANGED);
12 if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
13     thisCon.registerReceiver(mUsbReceiver, filter, RECEIVER_EXPORTED);
14 } else {
15     thisCon.registerReceiver(mUsbReceiver, filter);
16 }
```

▼ Receive broadcasts

Java

```
1 private BroadcastReceiver mUsbReceiver = new BroadcastReceiver() {
2     public void onReceive(Context context, Intent intent) {
3         String action = intent.getAction();
4         if (ACTION_USB_PERMISSION.equals(action)) {
5             synchronized (this) {
6                 UsbDevice device =
7                     (UsbDevice) intent.getParcelableExtra(UsbManager.EXTRA_DEVICE);
8                 boolean result =
9                     intent.getBooleanExtra(UsbManager.EXTRA_PERMISSION_GRANTED, false)
10                 if (result) {
11                     Printer printer = PrinterHelper.connectUSB(device);
12                     if (printer == null) {
13                         //The connection failed
14                         return;
15                     } else {
16                         //The connection is successful
17                     }
18                 } else {
19                     return;
20                 }
21             }
22         }
23     }
24 };
```

3.1.5 Serial port connection

- description

Connect the printer through the connection node and baud rate serial port

Serial port connection interface

Java

```
1 public static Printer connectSerial(String node, int baudRate)
```

- parameter

parameter	description
node	Connection node (example:/dev/ttyS1)
baudRate	Baud rate (example: 9600)

- example

Example of a serial port connection

Java

```
1 Printer printer = PrinterHelper.connectSerial(port, Integer.valueOf(baudRate));
2 if(printer == null){
3     //The connection failed
4 }else{
5     //connection successful, all print-related interfaces in the Printer
6 }
```

3.1.6 Connection status

- description

Determine whether the printer is connected

Connection status interface

Java

```
1 public boolean isConnected()
2 //This interface cannot obtain the real-time status,
3 //and Bluetooth can be obtained through the system Bluetooth broadcast,
4 //refer to the demo
```

- example

▼ Example of connection status

Java |

```
1 printer.isConnect()  
2 //true: Connected, false: disconnected  
3 //The printer is returned via the connection interface
```

3.1.7 Disconnect

- description

Disconnect a connected device

▼ Disconnect the interface

Java |

```
1 public boolean closeOperator()
```

- example

▼ Example of disconnecting an interface

Java |

```
1 printer.closeOperator()  
2 //The printer is returned via the connection interface
```

3.2.1 Get Printer Flags

- description

Get some identifications of connected printers. Bluetooth: returns Bluetooth address, WIFI: Returns IP address, USB: returns device node, serial port: returns serial port node

▼ Get the printer tag interface

Java |

```
1 public String getPrinterTAG()
```

- parameter

NULL

- example

▼ Get an example of a printer tag

Java |

```
1 //The printer is returned via the connection interface
2 printer.getPrinterTAG()
```

3.2.2 Set the SDK character encoding

- description

Set the character encoding required by the SDK, for example, set GB2312 for Chinese

▼ Configure the SDK character encoding API

Java |

```
1 public void setLanguageEncode(String languageEncode)
```

- parameter

parameter	description
languageEncode	Code (example: GBK) details in 4.1.1

- example

▼ Example for setting SDK character encoding

Java |

```
1 //The printer is returned via the connection interface
2 printer.setLanguageEncode("GB2312");
```

3.2.3 Send print data to the printer

- description

Send data stored in the Printer class to the Printer

▼ Interface for sending print data

Java |

```
1 public boolean print()
```

- parameter

NULL

- example

```
▼ Java |
1 //The printer is returned via the connection interface
2 printer.addText("text",1, false, false, false, false, 1, 1);
3 boolean result = printer.print();
4 //The data stored in the printer is sent to the print
5 //false: failed to send, true: sent
```

3.2.4 Add data to the printer

- description

Add data to the Printer class and send it to the Printer through the print interface.

```
▼ Add a data interface Java |
1 public void addData(byte[] data)//Add byte array to printer
2 public void addData(String data)//Add string data to printer
```

- parameter

parameter	description
data	Data that needs to be sent to the printer

- example

```
▼ Java |
1 //The printer is returned via the connection interface
2 printer.addData(new byte[]{0x0d,0x0a});
3 printer.addData("test");
4 printer.print();
```

3.2.5 Read data from the printer

- description

Read the data returned by the printer to APP. If no data is read within the extra time, a byte array with a length of 0 will be returned.

▼ Read data interface

Java |

```
1 public byte[] readData(long timeout)
```

- parameter

parameter	description
timeout	Timeout (MS)

- example



Java |

```
1 //The printer is returned via the connection interface
2 byte[] data = printer.readData(2000)
3 //data:Data read from the printer
```

3.2.6 Add Text

- description

Add text data to the Printer class, which can set alignment, font style, font scaling, and send it to the Printer through the print interface.

▼ Add a text interface

Java |

```
1 public void addText(String data, int alignment, boolean isFontSmall,
2                     boolean isBold, boolean isUnderline,
3                     boolean isTurnWhite, int widthMultiplier,
4                     int heightMultiplier)
```

- parameter

parameter	description
data	Text data
alignment	Alignment (0: Left, 1: Center, 2: Right)

isFontSmall	Whether to use small font (small font: 9x 17, large font: 12x 24)
isBold	Bold or not
isUnderline	Whether to use underscore
isTurnWhite	Whether anti-White
widthMultiplier	Font horizontal magnification (0–8) maximum multiple follows the printer
heightMultiplier	Font vertical magnification (0–8) maximum multiple follows the printer

- example

▼

Java |

```

1 //The printer is returned via the connection interface,
2 //Add text data to the Printer
3 printer.addText("text",1, false, false, false, false, 1, 1);
4 boolean result = printer.print();//Send data

```

3.2.7 Add Text Data of Table Type

- description

The interface is also a way to add text, and the style is more like a table.

```

1  public void addTable(Table table)
2
3  public class Table { // Tabular classes
4      // column: title bar data (e.g., product name; Quantity; Unit price; amount)
5      //          data are used"; "Do the separation
6      // regularExpression: a separator (e.g. ; )
7      // columnWidth: the byte length of each column, each Chinese is 2 byte length
8      h,
9      //          each number and letter is 1 byte length,
10     //          e.g.(new int[]{6, 6, 6, 6}) 6 means fit 3 Chinese or 6 letters
11     public Table(String column, String regularExpression, int[] columnWidth)
12     // Set the alignment for each column (0: left, 1: center, 2: right)
13     public void setColumnAlign(int align)
14     // Add data without a row (e.g., plastic bag; 10.00; 1; 10.00)
15     public void addRow(String row)
16 }

```

- parameter

parameter	description
table	Table Class Object

- example

```

1  // The printer is returned via the connection interface
2  String column = "Product name; Quantity; Unit price; Amount";
3  Table table = new Table(column, ";", new int[]{6, 6, 6, 6});
4  table.setColumnAlign(1);
5  table.addRow("Crisper Bag" + "; 10.00; 1; 10.00");
6  table.addRow("Wire hook" + "; 5.00; 2; 10.00");
7  table.addRow("Umbrella" + "; 5.00; 3; 15.00");
8  printer.addTable(table);
9  boolean result = printer.print(); // Send the data to the printer

```

3.2.8 Add Alignment

- description

This interface is used to set the overall alignment, but for some interfaces, it is invalid to set the alignment (for example, addText, additive, addBarCode, addQRCode, etc.)

▼ Add an alignment interface Kotlin |

```
1 public void addAlignment(int alignment)
```

- parameter

parameter	description
alignment	Alignment (0: Left, 1: Center, 2: Right)

- example

▼ Java |

```
1 //The printer is returned via the connection interface
2 printer.addAlignment(1)
3 printer.writeData("test");//Center the text test to print
4 printer.print()
```

3.2.9 Add Font Style

- description

This interface is used to add a font style to text data, but it is invalid for the addText interface. addText already has a parameter that sets the font style.

▼ Add a font style interface Java |

```
1 public void addFontStyle(boolean isFontSmall, boolean isBold,
2                          boolean isUnderline)
```

- parameter

parameter	description
isFontSmall	Whether to use small font (small font: 9x 17, large font: 12x 24)

isBold	Bold or not
isUnderline	Whether to use underscore

- example

▼ Java

```

1 //The printer is returned via the connection interface
2 printer.addFontStyle(true, true, true);
3 printer.addData("test");
4 printer.print();

```

3.2.10 Font Enlargement

- description

This interface is used to enlarge the font of text data, but the addText interface is invalid, addText itself has set the font enlargement parameter.

▼ Font enlargement interface Java

```

1 public void addFontMultiplier(int widthMultiplier, int heightMultiplier)

```

- parameter

parameter	description
widthMultiplier	Font horizontal magnification (0–8) maximum multiple follows the printer
heightMultiplier	Font vertical magnification (0–8) maximum multiple follows the printer

- example

```

1 //The printer is returned via the connection interface
2 printer.addFontMultiplier(1, 1);
3 printer.addData("test");
4 printer.print();

```

3.2.11 Add Picture

- description

This interface can add picture printing, set alignment, picture algorithm, width and height to be printed, and whether to compress data

200dpi printer 8 points = 1mm

300dpi printer 11.8 point = 1mm

- Add an image API

```

1 public void addImage(int alignment, int type, int imageWidth,
2                     int imageHeight, boolean isCompress, Bitmap bitmap)

```

- parameter

parameter	description
alignment	Alignment (0: Left, 1: Center, 2: Right)
type	Image Algorithm (0: Binary, 1: Dither)
imageWidth	Width to be printed (in dots)
imageHeight	Height to be printed (in dots)
isCompress	Compressed or not (supported by some printers)
bitmap	Picture

- example

```

1 //The printer is returned via the connection interface
2 printer.addImage(0, 0, 100, 100, false, bitmapPrint);
3 boolean result = printer.print();

```

3.2.12 Add Barcode

- description

This interface is used to print bar codes, which can print UPC-A,UPC-E,JAN13/EAN13, JAN8/EAN8,CODE39,ITF,CODABAR(NW-7),CODE93,CODE128 type bar codes

▼ Add a barcode interface

```

1 public void addBarCode(int bcType, String bcData, int width, int height,
2                       int hriPosition, int alignment)

```

- parameter

parameter	description
bcType	Barcode type (see Table 4.1.2 for details)
bcData	Barcode content
width	Bar code width (1–6 grades)
height	Bar code height (1–255 level)
hriPosition	Text Position (0: not printing, 1: above barcode, 2: below barcode, 3: Above and below barcode)
alignment	Alignment (0: Left, 1: Center, 2: Right)

- example

```

1 //The printer is returned via the connection interface
2 printer.addBarCode(65, "12345678901", 2, 50, 2, 1);
3 boolean result = printer.print();

```

3.2.13 Add QR Code

- description

The interface is used to print two-dimensional code

▼ Add a QR code interface

Java |

```
1 public void addQRCode(String qrData, int sizeOfModule, int errorLevel,  
2                       int alignment)
```

- parameter

parameter	description
qrData	QR code content
sizeOfModule	Size (Grade 1–16)
errorLevel	Error correction level (48–51, to L,M,Q,R)
alignment	Alignment (0: Left, 1: Center, 2: Right)

- example

▼

Java |

```
1 //The printer is returned via the connection interface  
2 printer.addQRCode("Test print QRCode",6, 48, 1);  
3 boolean result = printer.print();
```

3.2.14 Add PDF417

- description

This interface is used to print PDF417 code

▼ Add the PDF417 interface

Java

```
1 public void addPDF417(String data, int dataColumns, int dataRows,
2                       int moduleWidth, int rowHeight, int errorMode,
3                       int errorLevel, int options)
```

- parameter

parameter	description
data	Data Content
dataColumns	Set the number of columns (0–30)0: automatic processing, according to the data to determine the number of columns
dataRows	Set the number of rows (0,3–90)0: automatic processing, according to the data to determine the number of rows
moduleWidth	Set module width (2–8)
rowHeight	Set module height (2–8)
errorMode	Error correction mode (48: Rank mode, 49: Rate mode)
errorLevel	Error correction level (see 4.1.3 for details)
options	Mode (0: standard mode, 1: Compressed mode)

- example



Java

```
1 //The printer is returned via the connection interface
2 printer.addPDF417("123456",0, 0, 3, 3, 49, 1, 0);
3 boolean result = printer.print();
```

3.2.15 Add Page Mode

- description

this interface is to put the printer into page mode



Java |

```
1 public void addSelectPageMode()
```

- parameter
none
- example



Java |

```
1 //The printer is returned via the connection interface
2 printer.addSelectPageMode();//Enter page mode
3 printer.addClearPageModePrintAreaData();//Clear the page mode cache data
4 printer.addPageModePrintArea(0,0,200,200);//Set the print area
5 printer.addPageModePrintDirection(0);//Set the print orientation
6 printer.addPageModeAbsolutePosition(0,0);//Set the print start coordinates
7 printer.addQRCode("Test print QRCode",6, 48, 1);//print QR code
8 printer.addPrintPageModeData();//set print
9 printer.print();//Send Printer
```

3.2.16 Adding Clear Page Mode Cache Data

- description

this interface clears the last page-mode cached data, preferably before creating a page-mode print area



Clear the page-mode cached data interface

Java |

```
1 public void addClearPageModePrintAreaData()
```

- example

```

1 //The printer is returned via the connection interface
2 printer.addSelectPageMode();//Enter page mode
3 printer.addClearPageModePrintAreaData();//Clear the page mode cache data
4 printer.addPageModePrintArea(0,0,200,200);//Set the print area
5 printer.addPageModePrintDirection(0);//Set the print orientation
6 printer.addPageModeAbsolutePosition(0,0);//Set the print start coordinates
7 printer.addQRCode("Test print QRCode",6, 48, 1);//print QR code
8 printer.addPrintPageModeData();//set print
9 printer.print();//Send Printer

```

3.2.17 Adding a Page Mode Print Area

- description

this interface is used to create a canvas in page mode. You can set the starting position and width and height.

Page mode print area interface

```

1 public void addPageModePrintArea(int horizontal, int vertical, int width,
2                                 int height)

```

- parameter

parameter	description
horizontal	starting point abscissa (unit: Point)
vertical	starting point ordinate (unit: Point)
width	area width (unit: Point)
height	area height (unit: Point)

- example

```

1 //The printer is returned via the connection interface
2 printer.addSelectPageMode();//Enter page mode
3 printer.addClearPageModePrintAreaData();//Clear the page mode cache data
4 printer.addPageModePrintArea(0,0,200,200);//Set the print area
5 printer.addPageModePrintDirection(0);//Set the print orientation
6 printer.addPageModeAbsolutePosition(0,0);//Set the print start coordinates
7 printer.addQRCode("Test print QRCode",6, 48, 1);//print QR code
8 printer.addPrintPageModeData();//set print
9 printer.print();//Send Printer

```

3.2.18 Adding Page Mode Print Orientation

- description

this interface is used to set the printing direction in page mode.

Page mode print orientation interface

```

1 public void addPageModePrintDirection(int direction)

```

- parameter

parameter	description
direction	direction (0: from left to right, 1: From bottom to top, 2: from right to left, 3: From top to bottom)

- example

```

1 //The printer is returned via the connection interface
2 printer.addSelectPageMode();//Enter page mode
3 printer.addClearPageModePrintAreaData();//Clear the page mode cache data
4 printer.addPageModePrintArea(0,0,200,200);//Set the print area
5 printer.addPageModePrintDirection(0);//Set the print orientation
6 printer.addPageModeAbsolutePosition(0,0);//Set the print start coordinates
7 printer.addQRCode("Test print QRCode",6, 48, 1);//print QR code
8 printer.addPrintPageModeData();//set print
9 printer.print();//Send Printer

```

3.2.19 Add Page Mode Absolute Position

- description

this interface is used to set the absolute position of the print point in page mode. After setting the interface, follow a print content, then the content can be printed at this position.

▼ Added a page mode absolute position interface

Java |

```
1 public void addPageModeAbsolutePosition(int xPosition, int yPosition)
```

- parameter

parameter	description
xPosition	print abscissa
yPosition	print ordinate

- example

▼

Java |

```
1 //The printer is returned via the connection interface
2 printer.addSelectPageMode();//Enter page mode
3 printer.addClearPageModePrintAreaData();//Clear the page mode cache data
4 printer.addPageModePrintArea(0,0,200,200);//Set the print area
5 printer.addPageModePrintDirection(0);//Set the print orientation
6 printer.addPageModeAbsolutePosition(0,0);//Set the print start coordinates
7 printer.addQRCode("Test print QRCode",6, 48, 1);//print QR code
8 printer.addPrintPageModeData();//set print
9 printer.print();//Send Printer
```

3.2.20 Add Page Mode Printing

- description

this interface is used to let the printer know the data stored in the print page mode

▼ Added a page mode print interface

Java |

```
1 public void addPrintPageModeData()
```

- parameter
none

- example

▼

Java |

```
1 //The printer is returned via the connection interface
2 printer.addSelectPageMode();//Enter page mode
3 printer.addClearPageModePrintAreaData();//Clear the page mode cache data
4 printer.addPageModePrintArea(0,0,200,200);//Set the print area
5 printer.addPageModePrintDirection(0);//Set the print orientation
6 printer.addPageModeAbsolutePosition(0,0);//Set the print start coordinates
7 printer.addQRCode("Test print QRCode",6, 48, 1);//print QR code
8 printer.addPrintPageModeData();//set print
9 printer.print();//Send Printer
```

3.2.21 Add Positioning

- description
this interface can be used for positioning function when printing labels or black marks

▼ Add a positioning interface

Java |

```
1 public void addPositioning()
```

- parameter
none

- example

▼

Java |

```
1 //The printer is returned via the connection interface
2 printer.addPositioning();
3 printer.print();
```

3.2.22 Add paper feed

- description

this interface allows the printer to run the paper according to the number of lines you set.

▼ Add a paper feed interface

Java |

```
1 public void addFeedLine(int lines)
```

- parameter

parameter	description
lines	line number of paper walks

- example



Java |

```
1 //The printer is returned via the connection interface
2 printer.addFeedLine(100);
3 printer.print();
```

3.2.23 Adding Line Spacing

- description

this interface is used to set the line spacing, in rows.

▼ Add a line spacing interface

Java |

```
1 public void addLineSpace(int lineSpace)
```

- parameter

parameter	description
-----------	-------------

lineSpace	line Spacing (Unit: Line)
-----------	---------------------------

- example

▼

Java |

```

1 //The printer is returned via the connection interface
2 printer.addFeedLine(2);
3 printer.print();

```

3.2.24 Adding Printer Codes

- description
this interface can be used to set the printer code

▼

Add a printer encoding interface

Java |

```

1 public void addPrinterCharacter(int character)

```

- parameter

parameter	description
character	printer code (see 4.1.1 for details)

- example

▼

Java |

```

1 //The printer is returned via the connection interface
2 printer.addPrinterCharacter(1);
3 printer.print();

```

3.2.25 Initializing the Printer

- description
the interface can initialize the printer, clear the font style, alignment and so on.

▼ Initialize the printer interface

Java |

```
1 public void addInitialize()
```

- parameter
none

- example

▼

Kotlin |

```
1 //The printer is returned via the connection interface
2 printer.addInitialize();
3 printer.print();
```

3.2.25 Adding printer density

- description

used to add printer concentration, but different models will have different concentration ranges, please consult customer service for details

▼ Add the Printer Concentration interface

Java |

```
1 public void addPrintDensity(int density)
```

- parameter

parameter	description
density	print density, range depending on model

- example

▼

Java |

```
1 //The printer is returned via the connection interface
2 printer.addPrintDensity(0);
3 printer.print();
```

3.2.26 Adding Print Speed

- description

used to add printing speed, but different models will have different speed range, please consult customer service for details

▼ Add a print speed interface

Java |

```
1 public void addPrintSpeed(int speed)
```

- parameter

parameter	description
speed	print speed, range depending on model

- example



Java |

```
1 //The printer is returned via the connection interface
2 printer.addPrintSpeed(0);
3 printer.print();
```

3.2.27 Add paper cutter

- description

the interface can be used for cutting after paper

▼ Add a paper feeder interface

Java |

```
1 public void addCutterPaperFeeding(int cutMode, int distance)
```

- parameter

parameter	description
-----------	-------------

cutMode	cutter mode (0: full cut, 1: half cut)
distance	paper distance (0–255 single point)

- example

▼	Java
<pre> 1 //The printer is returned via the connection interface 2 printer.addCutterPaperFeeding(0, 100); 3 printer.print(); </pre>	

3.2.28 Add Open Cash Box

- description

this interface is used to open the cash box connected to the printer.

▼	Add the open cash drawer interface	Java
<pre> 1 public void addOpenCashDrawer(int openMode) </pre>		

- parameter

parameter	description
openMode	0: open Cashbox No. 1, 1: Open Cashbox No. 2, 2: Open all cashboxes

- example

▼	Java
<pre> 1 //The printer is returned via the connection interface 2 printer.addOpenCashDrawer(0); 3 printer.print(); </pre>	

3.2.29 Add open buzzer

- description

this interface is used to make the buzzer sound

▼ Added the interface to turn on the buzzer

Java |

```
1 public void addBeepBuzzer(int times, int t1, int t2)
```

- parameter

parameter	description
times	number of times the buzzer sounds
t1	sound time (unit: 100ms)
t2	stop time (unit: 100ms)

- example

▼

Java |

```
1 //The printer is returned via the connection interface
2 printer.addBeepBuzzer(1, 2, 2);
3 printer.print();
```

3.2.30 Add Left Margin

- description

this interface can set the left margin of the printed content

▼ Add left margin interface

Java |

```
1 public void addLeftMargin(int margin)
```

- parameter

parameter	description
margin	left Margin (Unit: Points)

- example

▼

Java |

```

1 //The printer is returned via the connection interface
2 printer.addLeftMargin(100);
3 printer.addText("text",1, false, false, false, false, 1, 1);
4 printer.print();

```

3.2.31 Get Printer Status

- description

this interface can obtain the opening and closing cover and paper status of the printer

▼

API for obtaining printer status

Java |

```

1 public int getPrinterStatus(int model)

```

- parameter

parameter	description
model	2: Acquire the state of opening and closing cover, 4: Acquire the state of paper

- return

parameter	description
-1	failed to send
other	printer Status

- example

```
1 //The printer is returned via the connection interface
2 int status = printer.getPrinterStatus(2);
3 if ((status & 4) == 4) {
4     //Open the lid
5 } else {
6     //Close the lid
7 }
8 status = printer.getPrinterStatus(4);
9 if ((status & 96) == 96) {
10     //Paperless
11 } else {
12     //There is paper
13 }
14 if ((status & 12) == 12) {
15     //The paper will be exhausted
16 } else {
17     //Paper is plentiful
18 }
```

3.2.32 Obtaining the Printer Serial Number

- description

this interface can be used to obtain the serial number of the printer

API for obtaining the printer serial number

```
1 public String getPrinterSN()
```

- parameter

none

- example

```
1 //The printer is returned via the connection interface
2 String strSN = printer.getPrinterSN();
```

3.2.33 Get Printer Power

- description

this interface obtains the percentage of printer power. This interface is only applicable to some printers with batteries.

▼ Get printer power interface

Java |

```
1 public int getPrinterQuantity()
```

- parameter

none

- return

parameter	description
-1	send failed (only supported by some printers)
other	electricity

- example

▼

Java |

```
1 //The printer is returned via the connection interface
2 int quantity = printer.getPrinterQuantity();
```

3.2.34 Setting the Print End Switch

- description

this interface is used to set the switch of whether to return app when printing is completed. Only some printers support it.

▼ Set the print end switch interface

Java |

```
1 public boolean setPrintEndSwitch(boolean isOpen)
```

- parameter

parameter	description
isOpen	true: On, false: Off

- example

▼

Java

```

1 //The printer is returned via the connection interface
2 printer.setPrintEndSwitch(true);
3 printer.addText("Test print", 1, false, false, false, false, 0, 0);
4 printer.addQRCode("Test print QRCode",6, 48, 1);
5 printer.addCutterPaperFeeding(0, 100);
6 boolean result = printer.print();
7 int printResult = printer.getPrintResult(3000);

```

3.2.35 Acquisition Print End

- description

this interface is used to receive the response sent back by the printer to app when printing is completed. First, the interface setPrintEndSwitch needs to be set to true, which is only supported by some printers.

▼

Receives the interface when printing is complete

Java

```

1 public int getPrintResult(long outTime)

```

- parameter

parameter	description
outTime	get timeout time

- return

parameter	description
-----------	-------------

-1	failed to send
-2	get timeout (only supported by some printers)
0	print complete

- example

▼

Java |

```

1 //The printer is returned via the connection interface
2 printer.setPrintEndSwitch(true);
3 printer.addText("Test print", 1, false, false, false, false, 0, 0);
4 printer.addQRCode("Test print QRCode",6, 48, 1);
5 printer.addCutterPaperFeeding(0, 100);
6 boolean result = printer.print();
7 int printResult = printer.getPrintResult(3000);

```

3.2.36 Two Color Print

- description

This interface is used for printing images in two colors. It requires two images of different colors to be passed as parameters, and the width and height of both images must be the same. This interface is only supported by some printers.

▼

Two Color Print

Java |

```

1 public boolean printDoubleColor(int type, Bitmap blackBitmap, Bitmap redBit
  map)

```

- parameter

parameter	description
type	Image algorithm (0: Binary, 1: Dithering)
blackBitmap	black object in the image
redBitmap	red object in the image

- **return**

parameter	description
true	Send successful
false	Send failed

- **example**

▼ Example of two-color print	Java
<pre> 1 InputStream inputStream = getResources().getAssets().open("black02.png"); 2 Bitmap bitmapBlack = BitmapFactory.decodeStream(inputStream); 3 inputStream = getResources().getAssets().open("red07.png"); 4 Bitmap bitmapRed = BitmapFactory.decodeStream(inputStream); 5 boolean result = printer.printDoubleColor(0, bitmapBlack, bitmapRed); </pre>	

3.2.37 Set Print Width

- **description**

This interface is used to set the width that the printer can print, and the content that exceeds the width will be automatically changed to the next line, which is measured in points. A 200dpi printer with 8 points is equal to 1 mm, and a 300dpi printer is 11.8 is equal to 1 mm

▼ set print width	Java
<pre> 1 public boolean setPrintWidth(int width) </pre>	

- **parameter**

parameter	description
width	Printable width in dots. 8 dots is equal to 1 mm for a 200dpi printer, 11.8 is equal to 1 mm for a 300dpi printer)

- **return**

parameter	description
-----------	-------------

true	Send successful
false	Send failed

- **example**

▼ Example of setting the print width

Java |

```
1 boolean result = printer.setPrintWidth(Integer.valueOf(data));
```

3.2.38 Get Print Width

- **description**

This interface is used to get the width that the printer can print in dots. A 200dpi printer with 8 points is equal to 1 mm, and a 300dpi printer is 11.8 is equal to 1 mm

▼ get print width

Java |

```
1 public int getPrintWidth()
```

- **parameter**

parameter	description
null	null

- **return**

parameter	description
-1	Send failed
other	print width

- **example**

▼ Get an example of the print width

Java |

```
1 int result = printer.getPrintWidth();
```

3.2.39 LAN Search Printer

- **Description**

this interface is used to search for printers in the local area network and obtain some printer information. This interface exists in the LanConfigUtil tool class

▼ Java

```
1 public static void searchPrinter(SearchPrinterListener listener)
```

- **Parameters**

parameters	description
listener	callback Parameters

- **Return**

parameters	description
none	none

- **Examples**

▼ Java

```
1  ▼ LanConfigUtil.searchPrinter(printer -> {
2      //printer.getIp() Printer IP address
3      //printer.getStatus() Printer status
4      //printer.getPrintWidth() Printer print width
5      //printer.getSn() Printer SN
6      //printer.getMac() Printer MAC
7      //printer.getPrinterName() Printer name
8      //printer.getVersion() Printer version number
9      //printer.getDpi() Printer resolution
10     //printer.getType() Printer network type
11     //printer.getDhcp() Specifies whether DHCP is enabled for the printer
12     //printer.getMask() The printer's subnet mask
13     //printer.getGateway() Printer gateway
14 });
```

3.2.40 LAN Configuration Ethernet Parameters

- **Description**

this interface is used to configure Ethernet parameters in the LAN. This interface exists in the LanConfigUtil tool class

▼

Java

```

1  public static void setEthernetConfig(Printer printer,
2                                     EthernetData ethernetData,
3                                     SetConfigListener listener)
4
5  public EthernetData(boolean isDHCP, String ip,
6                      String subnetMask, String gateway) {
7  }

```

- **Parameters**

parameters	description
printer	printer information, obtained from the 3.2.39 interface
ethernetData	<p>ethernet information that needs to be configured:</p> <p>isDHCP: whether to enable DHCP</p> <p> true: The following parameters do not need to be configured.</p> <p> false: The following parameters need to be configured.</p> <p>ip: The IP address to be configured.</p> <p>subnetMask: the subnet mask.</p> <p>gateway: gateway.</p>
listener	<p>set Result Callback</p> <p>0: indicates success,-1: failure,-2: Parameter error</p>

- **Return**

parameters	description
none	none

- Examples

```

1  LanConfigUtil.setEthernetConfig(
2      list.get(position), new EthernetData(config.dhcp, config.ip,
3      config.mask, config.gateway), result ->
4      Log.d("NetworkConfig", "result: " + result));

```

3.2.41 LAN configuration WIFI AP mode parameters

- Description

this interface is used to configure parameters in WIFI AP mode in the local area network. This interface exists in the LanConfigUtil tool class

```

1  public static void setWifiApConfig(Printer printer,
2                                     WifiAPData wifiAPData,
3                                     SetConfigListener listener)
4
5  public WifiAPData(String ssid, boolean isNoPassword, String password) {
6  }

```

- Parameters

parameters	description
printer	printer information, obtained from the 3.2.39 interface
wifiAPData	need to configure the WIFI AP information: ssid: Printer wifi name in AP mode isNoPassword: whether a password is required true: No password is required. false: A password is required password: The password.
listener	set Result Callback 0: indicates success,-1: failure,-2: Parameter error

- Return

parameters	description
none	none

- Examples

▼

Java

```

1  LanConfigUtil.setWifiApConfig(
2      list.get(position),
3      new WifiAPData(config.ssid, !config.passwordRequired, config.passwor
d),
4      result -> {Log.d("NetworkConfig", "result: " + result);
5      });

```

3.2.42 LAN configuration WIFI STA mode parameters

- Description

this interface is used to configure parameters in WIFI STA mode in the local area network. This interface exists in the LanConfigUtil tool class

▼

Java

```

1  public static void setWifiStaConfig(Printer printer,
2                                     WifiSTAData wifiStaData,
3                                     SetConfigListener listener)
4
5  public WifiSTAData(String ssid, String encryption, String password,
6                     boolean isDHCP, String ip, String gateway, String netmas
k)

```

- Parameters

parameters	description
printer	printer information, obtained from the 3.2.39 interface

wifiStaData	<p>need to configure the WIFI STA information:</p> <p>ssid: wifi name to be configured</p> <p>encryption: the encryption method ("OPEN", "WEP", "WPA2-PSK", "WPA/WPA2-PSK", "WPA-PSK", "WPA", "WPA2", "SAE", "WPA3-WPA2-PSK-MIX", "unknown").</p> <p>password: The password.</p> <p>isDHCP: whether DHCP is enabled</p> <p> true: The following parameters do not need to be configured.</p> <p> false: The following parameters need to be configured.</p> <p>ip: The IP address to be configured.</p> <p>netmask: The subnet mask.</p> <p>gateway: gateway.</p>
listener	<p>set Result Callback</p> <p>0: indicates success,-1: failure,-2: Parameter error</p>

- Return

parameters	description
none	none

- Examples

▼

Java |

```

1  LanConfigUtil.setWifiStaConfig(
2      list.get(position),
3      new WifiSTADData(config.ssid, config.encryption, config.password,
4                          config.dhcp, config.ip, config.gateway, config.mask),
5      result -> {
6          Log.d("NetworkConfig", "result: " + result);
7      });

```


4.1.1 Coding Table

Name	Printer Code	SDK Encoding
Default	0	gb2312
Chinese Simplified	0	gb2312
Chinese Traditional	0	big5
PC437(USA)	0	iso8859-1
KataKana	1	Shift_JIS
PC850(Multilingual)	2	iso8859-3
PC860(Portuguese)	3	iso8859-6
PC863(Canadian-French)	4	iso8859-1
PC865(Nordic)	5	iso8859-1
PC857(Turkish)	13	IBM857
PC737(Greek)	14	iso8859-7
ISO8859-7(Greek)	15	iso8859-7
WCP1252	16	iso8859-1
PC866(Cyrillic #2)	17	iso8859-5
PC852(Latin 2)	18	iso8859-2
PC858(Euro)	19	iso8859-15
KU42	20	ISO8859-11
TIS11(Thai)	21	ISO8859-11
TIS18(Thai)	26	ISO8859-11

PC720	32	iso8859-6
WPC775	33	iso8859-1
PC855(Cyrillic)	34	iso8859-5
PC862(Hebrew)	36	iso8859-8
PC864(Arabic)	37	iso8859-6
ISO8859-2(Latin2)	39	iso8859-2
ISO8859-15(Latin9)	40	iso8859-15
WPC1250	45	iso8859-2
WPC1251(Cyrillic)	46	iso8859-5
WPC1253	47	iso8859-7
WPC1254	48	iso8859-3
WPC1255	49	iso8859-8
WPC1256	50	Windows-1256
WPC1257	51	iso8859-1
WPC1258	52	bg2312
MIK(Cyrillic/Bulgarian)	54	iso8859-15
CP755	55	iso8859-5
Iran	56	iso8859-6
Iran II	57	iso8859-6
Latvian	58	iso8859-4
ISO-8859-1(West Europe)	59	iso8859-1

ISO-8859-3(Latin 3)	60	iso8859-3
ISO-8859-4(Baltic)	61	iso8859-4
ISO-8859-5(Cyrillic)	62	iso8859-5
ISO-8859-6(Arabic)	63	iso8859-6
ISO-8859-8(Hebrew)	64	iso8859-8
ISO-8859-9(Turkish)	65	iso8859-9
PC856	66	iso8859-8
ABICOIM	67	iso8859-15

4.1.2 Barcode Type

bcType	name	length	data Range
65	UPC-A	11-12	numbers
66	UPC-E	11-12	number (first number must be 0)
67	JAN13/EAN13	12-13	numbers
68	JAN8/EAN8	7-8	numbers
69	CODE39	1-255	numbers, letters, some symbols
70	ITF	2-254	numbers
71	CODABAR(NW-7)	2-255	number, part letter, part symbol (start and end must be A,B,C,D)
72	CODE93	1-255	numbers, letters, symbols

73	CODE128	2–255	numbers, letters, symbols (select character set first, {A: ASCII characters 00h to 5FH {B: ASCII characters 20h to 7FH {C: 100 digits of 00–99)
----	---------	-------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.1.3 PDF417 Error Correction Level

value	mode	grade	number of error correction code words
n = 48	level mode	0	2
n = 49	level mode	1	4
n = 50	level mode	2	8
n = 51	level mode	3	16
n = 52	Level mode	4	32
n = 53	level mode	5	64
n = 54	level mode	6	128
n = 55	level mode	7	256
n = 56	level mode	8	512
data code word x n x 0.1 = (0–3)	ratio mode	1	4
data code word x n x 0.1 = (4–10)	ratio mode	2	8
data code word x n x 0.1 = (11–20)	ratio mode	3	16
data code word x n x 0.1 = (21–45)	ratio mode	4	32
data code word x n x 0.1 = (46–100)	ratio mode	5	64
data code word x n x 0.1 = (101– 200)	ratio mode	6	128

data code word $\times n \times 0.1 = (201-400)$	ratio mode	7	256
data code word $\times n \times 0.1 = (400 \text{ or more})$	ratio mode	8	512

4.1.4 Turn on the SDK to send data logs

- **description**

This function is used to record the data sent by the SDK to the printer and to determine whether the data is correct.

The path to log is here

/sdcard/Android/data/app package/files/Documents/SDK_log.txt

app package is the package name of your project.

Opening method:

▼

Java |

```

1  Constants.isWriteLog = true
2  Constants.isHex = true

```