

iMin Integrated Printer Developer Documentation

Document Update Description

Version	Date Update	Content Update	Prepared by
V1.0.0	2023/7/31	Initial Release	Xie Huayen
V1.0.1	2024/4/18	1Optimize docking with SDK2.2.4.2 -5.How to obtain printer status using Bluetooth	Xie Huayen
V1.0.2	2024/6/25	1. The vehicle-mounted self-developed printer firmware adds GS1 one-dimensional code printing API. Pdf417, MaxiCode, AztecCode, DataMatrix QR code printing API Note: Only supports PC80 printer firmware 2. setTextBitmapTypeface supports custom font .ttf files and only supports text and image printing.	Xie Huayen
V1.0.3	2024/8/5	IPrinterCallback callback method return value description	Xie Huayen
V1.0.4	2024/9/30	1. Add API related to label printing (4. Definition of label printing interface (new)) 2.Need a new SDK version to use implementation 'com.github.iminsoftware:IminPrinterLibrary:V1.0.0.15'	Xie Huayen

目录

iMin Integrated Printer	1
Developer Documentation	1
Document Update Description	2
Introduction	7
Connect to the printer interface via built-in printing service interface	8
1. Method to connect to the printer via built-in printing service interface	8
2. Bind to printing service initialization utility class	10
3. Interface Definition Description	11
3.1 Printer initialization and configuration	11
3.1.1 Print initialization	11
3.1.2 Initialize, restore printing parameters	12
3.2 To obtain printer-related information	13
3.2.1 To obtain printer board serial number	13
3.2.2 To obtain printer model	14
3.2.3 To obtain printer thermal head model	14
3.2.4 To obtain printer firmware version	15
3.2.5 To obtain printing service version number	16
3.2.6 To obtain printer hardware version	16
3.3 To obtain printer status	17
3.4 Printer configuration-related information	17
3.4.1 To obtain USB connection pid vid	17
3.4.2 To obtain the name of the connected USB devices	18
3.4.3 To obtain the printing density	18
3.4.4 To obtain printing length	18
3.4.8 To obtain the current paper type of the printer	18
3.4.10 To obtain the frequency of paper cutting	19
3.4.11 Configure printer mode	19
3.4.12 To obtain printer current mode	19
3.5 Cash drawer Operation	21
3.5.1 Open cash drawer	21
3.5.2 To obtain current cash drawer status	21
3.5.3 To obtain the number of times cash drawer was opened	21
3.6 Print self-test page	21
3.7 ESC/POS printing command	23
3.8 Paper feeding related	23
3.8.1 Feed one line	23
3.8.2 Self-define height if several lines feeding	23
3.9 Cutter (Paper cutting) related	24
3.9.1 Cut paper (Half-cut)	24
3.9.2 Cut paper (Full cut)	24
3.10 Configure print global alignment method	24
3.11 Text bitmap print related	25
3.11.1 Configure print font face	25

3.11.2	Set the font size for printing text images	25
3.11.3	Configure text bitmap print style	25
3.11.4	Configure text Strikethrough	26
3.11.5	Configure text Underline	26
3.11.6	Configure text line spacing	26
3.11.7	Configure text spacing between letters	26
3.11.8	Configure text Anti-white	27
3.11.9	Text bitmap printing	27
3.11.10	Text bitmap print with alignment	27
3.12	Bitmap printing	28
3.12.1	Bitmap printing	28
3.12.2	Bitmap print with alignment	28
3.12.3	Print multiple bitmap	29
3.12.4	Print multiple bitmap with alignment	29
3.12.5	Single colour bitmap processing and printing	30
3.12.6	Single colour bitmap processing and print with alignment	30
3.13	Print table chart	31
3.13.1	Print table in propotion to width weight	31
3.13.2	Print table according to the width value	31
3.14	1D code print related	32
3.14.1	Configure the width of 1D code	32
3.14.2	Configure the height of 1D code	32
3.14.3	Configure the position of 1D code HRI character	33
3.14.4	Print 1D code	33
3.14.5	1D code print with alignment	34
3.14.6	1D code print with full parameters	34
3.15	QR code print	36
3.15.1	Configure the size of QR code	36
3.15.2	Configure QR code error correction level	36
3.15.3	Print QR code	37
3.15.4	Print QR code with alignment	37
3.15.6	Print PDF417	38
3.15.7	Print MaxiCode	39
3.15.8	Print AztecCode	39
3.15.9	Print DataMatrix	39
3.16	Configure left margin	40
3.17	Print double QR code	40
3.17.1	Configure double QR code size	40
3.17.2	Configure double QR code(QR1) left margin	40
3.17.3	Configure QR code (QR2) left margin	41
3.17.4	Configure double QR code (QR1) error level	41
3.17.5	Configure QR code (QR2) error	41
3.17.6	Configure double QR code (QR1) version	41
3.17.7	Configure double QR code (QR2) version	42

3.17.8 Print double QR code	42
3.18 Transaction printing	42
3.18.1 Enter transaction print mode	43
3.18.2 Submit a transaction	43
3.18.2 Transaction submission callback result	44
3.18.3 Terminate transaction print	44
3.18.4 Terminate transaction print callback result	45
3.19 Printer upgrade	46
3.19.1 To obtain printer upgrade status	46
4.Definition Explanation of Label Printing Interface (New)	47
4.1. Canvas Settings	47
4.2 Draw text content	47
4.3.Draw barcode content	49
4.4. Draw QR code content	50
4.5. Draw an image	50
4.6 Draw special graphics	52
4.7 Print and draw content	53
1. Example	53
2. Printing label skills	55
3.Description of enumeration parameters	56
=====	
=====	
4.Integrate the printer via built-in virtual bluetooth connection	59
2.1 Virtual Bluetooth introduction	59
2.2 Virtual Bluetooth usage	60
5.H5 Web Page integrate with printer through JS Bridge	65
5.1 H5 Web page integrate with printer plug-in example	65
5.2 Integrate with jquery plug-in to print	65
5.2.1、 Simple method to introduce	65
5.2.1.1.Introduce print SDK script	65
5.2.1.2. Initialization	66
5.2.1.3. Integration related	66
5.2.2 Implement vue-cli scarffolding in vue2	66
5.2.2.1. Configure imin-printer package in vue.config.js	66
5.2.2.2. Import package to main.js	67
5.2.2.3. To use it in App.vue or components	67
5.2.3、 Implement vite scarffolding in vue3	68
5.2.3.1. To configure imin-printer package in vite.config.js	68
5.2.3.2. Import package to main.js	68
5.2.3.3. Use it in App.vue or components	69
5.3. Api Description	69
Initialize printer (Only support SPI/USB printing)	69
Obtain printer status	69
Feed paper with one line	69

User-defined paper feed spacing	70
Cut paper	70
Configure alignment	70
Configure print font size	70
Configure print font type	70
Configure text style	71
Configure print text line spacing	71
Configure print paper width (Can ignore this)	71
Print text	71
Print text and add new line	71
Print one row table	72
Configure 1D paper width	72
Configure 1D barcode height	72
Select HRI character print position of 1D barcode	72
18 Print barcode	72
Print barcode and configure alignment method	73
Configure QR code size	73
Configure QR code error level	74
Configure left margin	74
Print QR code	74
Print QR code with alignment	74
Configure paper format (Not recommend to use as the printer will be reset)	74
Print bitmap	75
Trigger cash drawer	75
Configure double QR code size	75
Configure double QR code QR1 error correction level	75
Configure double QR code QR2 error correction level	75
Configure double QR code QR1 left margin	75
Configure double QR code QR2left margin	76
Configure double QR code QR1 version	76
Configure double QR code QR2 version	76
Print double QR code	76

Introduction

iMin devices comes with built-in thermal printer, allowing an application to print thermal receipts directly via iMin SDK. Products with integrated printers are:

Hand-held mobile series —— M2-202、M2-203、M2 Pro、Swift 1 、Swift 1 Pro、Swift2、Swift2 Pro etc.

Tablet series—— M2 Max、D1、D1 Pro、Falcon 1、Falcon 2 etc.

Desktop POS series —— D1w、D4、D4 Pro 、Swan 2 etc

There are two types of specifications for iMin devices' built-in printer:

- 80mm paper width, comes with paper cutter, compatible with 58mm width as well。 For example, Falcon 1 ,D4 Pro, Swan 2,Falcon 2 ,is equipped with this type of printer.
- 58mm paper width, doesn't comes with paper cutter. Products likeD1,D1Pro, M2 Max ,M2-202,M2-203,M2 Pro,Swift 1,Swift 1Pro,Swift2,Swift2 Pro are equipped with this kind of printer
- Label printing only supports Swan 2 label version

Note: Print Products supported by sdk V2.0: D4 Pro , Swift 2 , Swift 2 Pro , Swift 1 Pro, Swan 2, Falcon 2,Swift 2 Ultra

The application developer can integrate with the built-in printer using three different methods:

- **Connect to the printer via built-in printing service interface** —— This method is suitable for developers who develop print-related apps for the first time and have zero knowledge about EPSON commands, to achieve their required printing result through multiple printing interfaces provided by iMin printing services;
- **Connect to the printer via the built-in virtual Bluetooth** —— This method is suitable for developers who have past experiences in integrating Bluetooth, USB printer, or developers who have realized Bluetooth printing in their application, which only need to modify the source code in order to achieve the desired printing effect;
- **Connect to the printer JS SDK for HS web browser** ——This method is suitable for accessing the H5 page of an application;

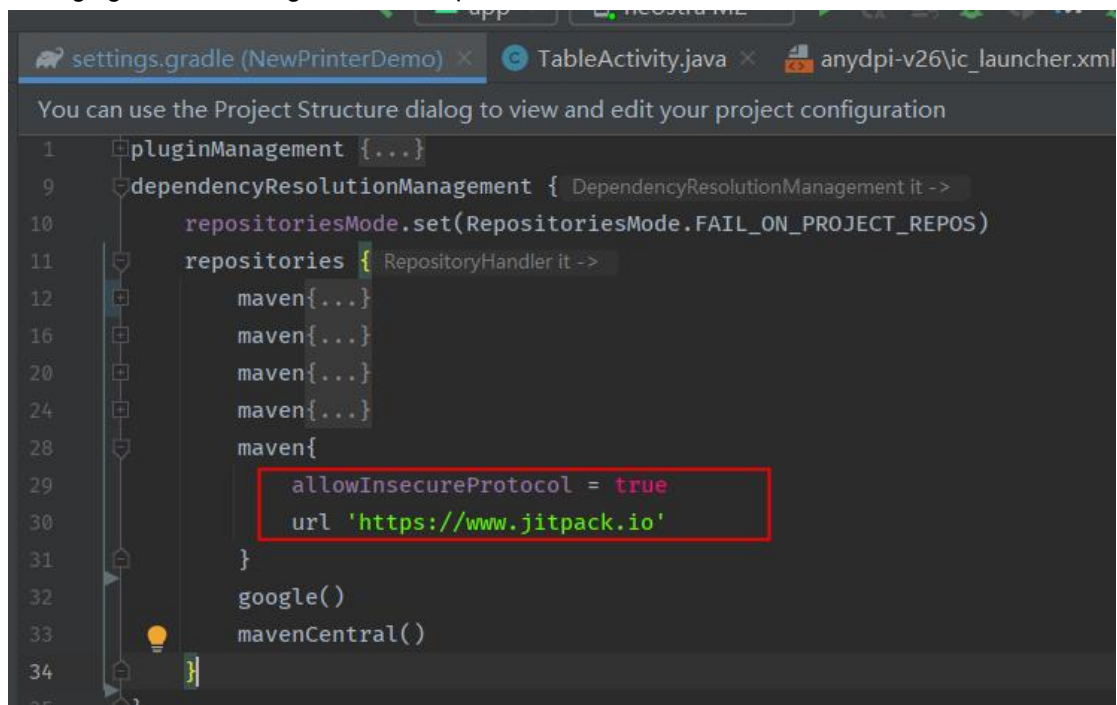
Connect to the printer interface via built-in printing service interface

1. Method to connect to the printer via built-in printing service interface

1.1 Bind to remote library via jitpack implementation method(recommend)

`com.github.iminsoftware:IminPrinterLibrary:V1.0.0.15'` //The latest version published on github shall prevail.

settings.gradle file configuration example:



```
1  pluginManagement {...}
9  dependencyResolutionManagement { DependencyResolutionManagement it ->
10     repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
11     repositories { RepositoryHandler it ->
12         maven{...}
16         maven{...}
20         maven{...}
24         maven{...}
28         maven{
29             allowInsecureProtocol = true
30             url 'https://www.jitpack.io'
31         }
32         google()
33         mavenCentral()
34     }
35 }
```



```
48     }
49     compileOptions {
50         sourceCompatibility JavaVersion.VERSION_1_8
51         targetCompatibility JavaVersion.VERSION_1_8
52     }
53     viewBinding {
54         enabled = true
55     }
56 }
57
58 dependencies {
59
60     implementation 'androidx.appcompat:appcompat:1.5.1'
61     implementation 'com.google.android.material:material:1.7.0'
62     implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
63     implementation 'com.github.iminsoftware:IminPrinterLibrary:V1.0.0.15'
64     testImplementation 'junit:junit:4.13.2'
65     androidTestImplementation 'androidx.test.ext:junit:1.1.5'
66     androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
67     implementation 'com.github.CymChad:BaseRecyclerViewAdapterHelper:3.0.1'
68     implementation "androidx.multidex:multidex:2.0.1"
69
70     //条形码
71     implementation "com.google.zxing:core:3.3.1"
72     implementation 'com.guolindev.permissionx:permissionx:1.7.1'
73 }
```

Note: Only implementation

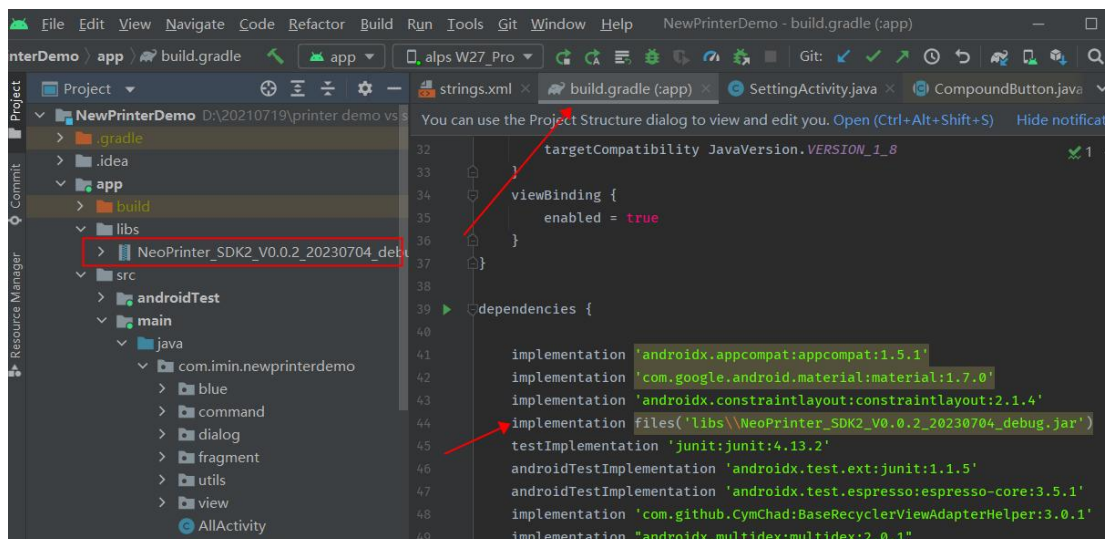
'com.GitHub.iminsoftware: IminPrinterLibrary: V1.0.0.15'

V1.0.0.15 and above versions support label printing

1.2 Via jar package integration method

Download the developer iMinPrinter_SDK2_V1.0.0.jar file from the developers documentation website and import it to the app-libs directory: <https://oss-sg.imin.sg/docs/en/Printer.html>

As shown in the screenshot below:



```
32     targetCompatibility JavaVersion.VERSION_1_8
33
34     viewBinding {
35         enabled = true
36     }
37 }
38
39 dependencies {
40
41     implementation 'androidx.appcompat:appcompat:1.5.1'
42     implementation 'com.google.android.material:material:1.7.0'
43     implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
44     implementation files('libs\\NeoPrinter_SDK2_V0.0.2_20230704_debug.jar')
45     testImplementation 'junit:junit:4.13.2'
46     androidTestImplementation 'androidx.test.ext:junit:1.1.5'
47     androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
48     implementation 'com.github.CymChad:BaseRecyclerViewAdapterHelper:3.0.1'
49     implementation "androidx.multidex:multidex:2.0.1"
```

Integrating the SDK using the two methods above, perform Sync Project clear Project build Project after the configuration of the `build.gradle` file is completed.

2. Bind to printing service initialization utility class

2.1 Initialize the binded printing service

2.1.1 Bind to a service

2.1.1.1 Perform initialization using sdk `PrinterHelper.class` utility class

Function:

```
PrinterHelper.getInstance().initPrinterService(this);
```

Example:

Using `onCreate()` method to reference the binding service in the `MainActivity` implementation class of the APK main interface

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = ActivityMainBinding.inflate(LayoutInflater.from(this));
    setContentView(binding.getRoot());
    PrinterHelper.getInstance().initPrinterService(BaseApplication.mContext.getApplicationContext());
}
```

2.1.1.2 Using `NeoPrinterManager.class` utility class to perform initialization

Related function: `NeoPrinterManager.getInstance().bindService(this,serviceConnectionCallback);`

Initialization configuration source code example:

```
public class MainActivity extends BaseActivity {
```

```
    ActivityMainBinding binding;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        binding = ActivityMainBinding.inflate(LayoutInflater.from(this));
```

```
        setContentView(binding.getRoot());
```

```
        PrinterHelper.getInstance().initPrinterService(BaseApplication.mContext.getApplicationContext());
```

```
    }
```

```
    @Override
```

```
    public void finish() {
```

```
        super.finish();
```

```
        PrinterHelper.getInstance().deInitPrinterService(BaseApplication.mContext.getApplicationContext());
```

```
}  
}
```

2.2 Unbind the service to release resources

```
PrinterHelper.getInstance().deInitPrinterService(this);  
or NeoPrinterManager.getInstance().unBindService(this,serviceConnectionCallback);
```

Example:

Using finish() method to reference the unbind service in the MainActivity implementation class of the APK main interface

```
@Override  
public void finish() {  
    PrinterHelper.getInstance().deInitPrinterService(context: this);  
  
    super.finish();  
}
```

或者

```
NeoPrinterManager.getInstance().unBindService(this,serviceConnectionCallback);
```

3. Interface Definition Description

Call the following interface to achieve own printing function through PrinterHelper. GetInstance () object,

3.1 Printer initialization and configuration

3.1.1 Print initialization

Function: int initPrinter(String packageName, IPrinterCallback callback);

Parameter: packageName ->Current apk package name;

callback -> Initialization result callback

Example:

```
PrinterHelper.getInstance().initPrinter(getPackageName(), new INeoPrinterCallback() {  
    @Override  
    public void onRunResult(boolean b) throws RemoteException {  
        Log.d("printerTest_PrintFragment", b ? "00000 绑定服务成功" : "uuuuu 绑定服务失败");  
    }  
    @Override  
    public void onReturnString(String s) throws RemoteException {
```

```

    }
    @Override
    public void onRaiseException(int i, String s) throws RemoteException {
    }
    @Override
    public void onPrintResult(int i, String s) throws RemoteException {
    }
    });

```

Notes: onRunResult(boolean b) Initialization successful / Failed b=true success; b=false failed

3.1.2 Initialize, restore printing parameters

Function : void initPrinterParams(int fd); Corresponding function in PrinterHelper

: void initPrinterParams();

Parameters: fd-> Each application assigned id.

Notes: The fd is being processed in the jar file by default as we have used PrinterHelper.getInstance() utility to call printing method, the API is called based on PrinterHelper.getInstance() utility tool. Below example for your reference:

Example:

```
PrinterHelper.getInstance().initPrinterParams()
```

Or

//int fd: The id assigned by each application

```
int fd = BaseApplication.getNeoPrinterService().initPrinter(getPackageName(), new INeoPrinterCallback() {
```

```

    @Override
    public void onRunResult(boolean isSuccess) throws RemoteException {
        Log.d(TAG,"initPrinter onRunResult isSuccess = " + isSuccess);
    }

```

```

    @Override
    public void onReturnString(String result) throws RemoteException {
        Log.d(TAG,"initPrinter onReturnString result = " + result);
    }

```

```

    @Override
    public void onRaiseException(int code, String msg) throws RemoteException {
        Log.d(TAG,"initPrinter onRaiseException code = " + code + " msg = " + msg);
    }

```

```

    @Override
    public void onPrintResult(int code, String msg) throws RemoteException {
        Log.d(TAG,"initPrinter onPrintResult code = " + code + " msg = " + msg);
    }

```

```

    });
    } catch (RemoteException e) {
        e.printStackTrace();
    }
}

```

3.2 To obtain printer-related information

3.2.1 To obtain printer board serial number

Function: void getPrinterSerialNumber(IPrinterCallback callback);

Parameters:

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

```

PrinterHelper.getInstance().getPrinterSerialNumber(new INeoPrinterCallback() {
    @Override
    public void onRunResult(boolean isSuccess) throws RemoteException {
        Log.d(TAG," onRunResult =====> "+isSuccess);
    }

    @Override
    public void onReturnString(String result) throws RemoteException {
        Log.d(TAG," onReturnString =====> "+result);
        binding.textSerialNumber.setText(getString(R.string.printer_serial_number,result));
    }

    @Override
    public void onRaiseException(int code, String msg) throws RemoteException {
        Log.d(TAG," onRaiseException =====> "+msg);
    }

    @Override
    public void onPrintResult(int code, String msg) throws RemoteException {

```

```
    }  
    }  
};
```

3.2.2 To obtain printer model

Function: void getPrinterModelName(IPrinterCallback callback);

Parameters: callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

```
PrinterHelper.getInstance().getPrinterModelName(new INeoPrinterCallback() {  
    @Override  
    public void onRunResult(boolean isSuccess) throws RemoteException {  
  
    }  
  
    @Override  
    public void onReturnString(String result) throws RemoteException {  
        Log.d(TAG, " result ==> "+result);  
        binding.textModelName.setText(getString(R.string.printer_model_name,result));  
    }  
  
    @Override  
    public void onRaiseException(int code, String msg) throws RemoteException {  
  
    }  
  
    @Override  
    public void onPrintResult(int code, String msg) throws RemoteException {  
  
    }  
});
```

3.2.3 To obtain printer thermal head model

Function: void getPrinterThermalHead(IPrinterCallback callback);

Parameters:

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

```
PrinterHelper.getInstance().getPrinterThermalHead(new INeoPrinterCallback() {
    @Override
    public void onRunResult(boolean isSuccess) throws RemoteException {
    }
    @Override
    public void onReturnString(String result) throws RemoteException {
        binding.textThermalHead.setText(getString(R.string.printer_thermal_head,result));
    }
    @Override
    public void onRaiseException(int code, String msg) throws RemoteException {
    }
    @Override
    public void onPrintResult(int code, String msg) throws RemoteException {
    }
});
```

3.2.4 To obtain printer firmware version

Function: void getPrinterFirmwareVersion(IPrinterCallback callback);

Parameters: callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

```
PrinterHelper.getInstance().getPrinterFirmwareVersion(new INeoPrinterCallback() {
    @Override
    public void onRunResult(boolean isSuccess) throws RemoteException {
    }
});
```

```

        @Override
        public void onReturnString(String result) throws RemoteException {
            binding.textFirmwareVersion.setText(getString(R.string.printer_firmware_version,result));
        }
        @Override
        public void onRaiseException(int code, String msg) throws RemoteException {
        }
        @Override
        public void onPrintResult(int code, String msg) throws RemoteException {
        }
    });

```

3.2.5 To obtain printing service version number

Function: String getServiceVersion();

Return value description: System printing service current version

Example:

```
PrinterHelper.getInstance().getServiceVersion();
```

3.2.6 To obtain printer hardware version

Function: void getPrinterHardwareVersion(int fd, IPrinterCallback callback);

Return value description: System printer hardware current version

Example:

Using PrinterHelper utility class:

```

PrinterHelper.getInstance().getPrinterHardwareVersion(int fd,new INeoPrinterCallback() {
    @Override
    public void onRunResult(boolean isSuccess) throws RemoteException {
    }
    @Override
    public void onReturnString(String result) throws RemoteException {
        binding.textFirmwareVersion.setText(getString(R.string.printer_firmware_version,result));
    }
    @Override
    public void onRaiseException(int code, String msg) throws RemoteException {
    }
    @Override
    public void onPrintResult(int code, String msg) throws RemoteException {
    }
});

```

Not using PrinterHelper utility class:

```

iNeoPrinterService..getPrinterHardwareVersion(new INeoPrinterCallback() {
    @Override
    public void onRunResult(boolean isSuccess) throws RemoteException {

```



```

    }
    @Override
    public void onReturnString(String result) throws RemoteException {
        binding.textFirmwareVersion.setText(getString(R.string.printer_firmware_version,result));
    }
    @Override
    public void onRaiseException(int code, String msg) throws RemoteException {
    }
    @Override
    public void onPrintResult(int code, String msg) throws RemoteException {
    }
    });

```

3.3 To obtain printer status

Function: `int getPrinterStatus();`

Return value description: Printer current status

-1 -> Not connected to service

3 -> Printer door opened

4 -> Printer head overheated

7 -> Paper missing

0 -> Printer is normal

Example:

```
PrinterHelper.getInstance().getPrinterStatus();
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.getPrinterStatus(int fd);
```

3.4 Printer configuration-related information

3.4.1 To obtain USB connection pid vid

Function: `String getUsbPrinterVidPid();`

Return value description: The pid, vid values of the currently connected USB devices

Example:

```
PrinterHelper.getInstance().getUsbPrinterVidPid();
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.getUsbPrinterVidPid(int fd);
```

3.4.2 To obtain the name of the connected USB devices

Function: `String getUsbDevicesName();`

Return value description: The name of the currently connected USB devices

Example:

```
PrinterHelper.getInstance().getUsbDevicesName();
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.getUsbDevicesName(int fd);
```

3.4.3 To obtain the printing density

Function: `int getPrinterDensity();`

Return value description: Return the current printing density of the printer

```
PrinterHelper.getInstance().getPrinterDensity();
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.getPrinterDensity(int fd);
```

3.4.4 To obtain printing length

Function: `void getPrinterPaperDistance(IPrinterCallback callback);`

Return value description: `callback.onReturnString(String s)` s Unit:cm

Example:

```
PrinterHelper.getInstance().getPrinterPaperDistance(new INeoPrinterCallback() {
```

```
    @Override
```

```
    public void onRunResult(boolean b) throws RemoteException {
```

```
    }
```

```
    @Override
```

```
    public void onReturnString(String s) throws RemoteException {
```

```
        Log.e(TAG,"getPrinterPaperDistance ==> "+s);
```

```
    }
```

```
    @Override
```

```
    public void onRaiseException(int i, String s) throws RemoteException {
```

```
    }
```

```
    @Override
```

```
    public void onPrintResult(int i, String s) throws RemoteException {
```

```
    }
```

```
});
```

3.4.8 To obtain the current paper type of the printer

Function: `int getPrinterPaperType();`

Return parameters: The current paper type 80/58

Example:

```
PrinterHelper.getInstance().getPrinterPaperType();
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.getPrinterPaperType(int fd);
```

3.4.10 To obtain the frequency of paper cutting

Function: void getPrinterCutTimes(int fd, IPrinterCallback callback);

Return parameters: callback.onReturnString(String s) s Unit:n

Example:

```
PrinterHelper.getInstance().getPrinterCutTimes(new INeoPrinterCallback() {  
    @Override  
    public void onRunResult(boolean b) throws RemoteException {  
    }  
    @Override  
    public void onReturnString(String s) throws RemoteException {  
        Log.e(TAG,"getPrinterCutTimes==> "+s);  
    }  
    @Override  
    public void onRaiseException(int i, String s) throws RemoteException {  
    }  
    @Override  
    public void onPrintResult(int i, String s) throws RemoteException {  
    }  
});
```

3.4.11 Configure printer mode (Only supports Swan 2, Swift 2

Ultra supports label printers)

Function: void setPrinterMode(int fd, int mode);

Parameters: mode -> 0 small ticket mode; 1 Label mode

Example:

```
PrinterHelper.getInstance().setPrinterMode(1);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setPrinterMode(int fd,1);
```

3.4.12 To obtain printer current mode(Not Supported)

Function: int getPrinterMode(int fd);

Return value: The current configured printer mode 1: Normal mode

Example:

```
PrinterHelper.getInstance().getPrinterMode();
```

Not using PrinterHelper utility class:

```
INeoPrinterService.getPrinterMode(int fd);
```

3.4.13 To obtain printer current mode(support)(Only supports Swan 2, Swift 2 Ultra supports label printers)

Function: void labelGetPrinterMode(int fd, IPrinterCallback callback);

Return value description: callback-> onReturnString(String result) // "Label" or "Receipt":

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().labelGetPrinterMode(int fd,new INeoPrinterCallback() {  
    @Override  
    public void onRunResult(boolean isSuccess) throws RemoteException {  
    }  
    @Override  
    public void onReturnString(String result) throws RemoteException {  
        //result = "Label" or "Receipt"  
    }  
    @Override  
    public void onRaiseException(int code, String msg) throws RemoteException {  
    }  
    @Override  
    public void onPrintResult(int code, String msg) throws RemoteException {  
    }  
});
```

Not using PrinterHelper utility class:

```
INeoPrinterService.labelGetPrinterMode(new INeoPrinterCallback() {  
    @Override  
    public void onRunResult(boolean isSuccess) throws RemoteException {  
    }  
    @Override  
    public void onReturnString(String result) throws RemoteException {  
        // result ="Label" or "Receipt"  
    }  
    @Override  
    public void onRaiseException(int code, String msg) throws RemoteException {  
    }  
    @Override  
    public void onPrintResult(int code, String msg) throws RemoteException {  
    }  
}
```

```
});
```

3.5 Cash drawer Operation

3.5.1 Open cash drawer

Function: void openDrawer(int fd);

Parameters:

Example:

```
PrinterHelper.getInstance().openDrawer();
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.openDrawer(int fd);
```

3.5.2 To obtain current cash drawer status

Function: boolean getDrawerStatus(int fd);

Return value description: true open cashdrawer; false close cashdrawer

Example:

```
PrinterHelper.getInstance().getDrawerStatus();
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.getDrawerStatus(int fd);
```

3.5.3 To obtain the number of times cash drawer was opened

Function: int getOpenDrawerTimes(int fd);

Return value description: int times Return SDK records of the number of times the cash drawer was opened

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().getOpenDrawerTimes();
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.getOpenDrawerTimes(int fd);
```

3.6 Print self-test page

Function: void printerSelfChecking(int fd, IPrinterCallback callback);

Parameters: fd Package initialization return tag value of the current application; In general, there is no need to transmit callback

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

```
1.PrinterHelper.getInstance().printerSelfChecking(null);
```

```
2.PrinterHelper.getInstance().printerSelfChecking(new INeoPrinterCallback() {
```

```
    @Override
```

```
    public void onRunResult(boolean isSuccess) throws RemoteException {
```

```
    }
```

```
    @Override
```

```
    public void onReturnString(String result) throws RemoteException {
```

```
    }
```

```
    @Override
```

```
    public void onRaiseException(int code, String msg) throws RemoteException {
```

```
    }
```

```
    @Override
```

```
    public void onPrintResult(int code, String msg) throws RemoteException {
```

```
    }
```

```
});
```

Not using PrinterHelper utility class:

```
3.iNeoPrinterService.printerSelfChecking(int fd,null);
```

```
4.iNeoPrinterService.printerSelfChecking(int fd,new INeoPrinterCallback() {
```

```
    @Override
```

```
    public void onRunResult(boolean isSuccess) throws RemoteException {
```

```
    }
```

```
    @Override
```

```
    public void onReturnString(String result) throws RemoteException {
```

```
    }
```

```
    @Override
```

```
    public void onRaiseException(int code, String msg) throws RemoteException {
```

```
    }
```

```
    @Override
```

```
    public void onPrintResult(int code, String msg) throws RemoteException {
```

```
    }
```

```
});
```

3.7 ESC/POS printing command

Function: void sendRAWData(int fd, in byte[] bytes, IPrinterCallback callback);

Parameters: fd -> Package initialization return tag value of the current application

bytes ->ESC/POS command

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().sendRAWData(bytes,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.sendRAWData(int fd,bytes,null);
```

3.8 Paper feeding related

3.8.1 Feed one line

Function: void printAndLineFeed(int fd);

Parameters: fd -> Package initialization return tag value of the current application

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printAndLineFeed();
```

Not using:

```
iNeoPrinterService.printAndLineFeed(this.fd);
```

3.8.2 Self-define height if several lines feeding

Function: void printAndFeedPaper(int fd, int value);

Parameters: fd -> Package initialization return tag value of the current application

value ->0<value<1016 If value is bigger than 101, choose 1016

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printAndFeedPaper(70);
```

Not using:

```
iNeoPrinterService.printAndFeedPaper(this.fd, value);
```

3.9 Cutter (Paper cutting) related

3.9.1 Cut paper (Half-cut)

Function: void partialCut(int fd);

Parameters: fd -> Package name initialization return tag value of the current application

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().partialCut();
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.partialCut(int fd);
```

3.9.2 Cut paper (Full cut)

Function: void fullCut(int fd);

Parameters: fd -> Package name initialization return tag value of the current application

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().fullCut();
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.fullCut(int fd);
```

3.10 Configure print global alignment method

Function: void setCodeAlignment(int fd, int alignmentMode);

Parameters: fd -> Package name initialization return tag value of the current application

alignmentMode -> 0 Align left 1 Center 2 Align right

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setCodeAlignment(0);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setCodeAlignment(int fd,0);
```


3.11 Text bitmap print related

3.11.1 Configure print font face

Function: void setTextBitmapTypeface(int fd, String typeface);

Parameters: fd -> Package name initialization return tag value of current application

typeface ->"Typeface.DEFAULT" configure default font
"Typeface.MONOSPACE" configure Monospace font
"Typeface.DEFAULT_BOLD" configure Bold font
"Typeface.SANS_SERIF" configure Sans Serif font
"Typeface.SERIF" configure Serif font
"shaonvti.ttf" Custom font .ttf file

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setTextBitmapTypeface("Typeface.DEFAULT");
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setTextBitmapTypeface(int fd,"Typeface.DEFAULT");
```

3.11.2 Set the font size for printing text images

Function: void setTextBitmapSize(int fd, int size);

Parameters: fd ->Initialize the package name to return the tag value of the current application

size ->Set text image font size Default28

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setTextBitmapSize(28);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setTextBitmapSize(int fd,28);
```

3.11.3 Configure text bitmap print style

Function: void setTextBitmapStyle(int fd, int style);

Parameters: fd -> Package name initialization return tag value of the current application

style->0 = Normal 1= Bold 2= Italic 3= Bold Italic

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setTextBitmapStyle(0);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setTextBitmapStyle(int fd,0);
```

3.11.4 Configure text Strikethrough

Function: void setTextBitmapStrikeThru(int fd, boolean strikeThru);

Parameters: fd -> Package name initialization return tag value of the current application

strikeThru -> true= configure strikethrough, false= remove strikethrough

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setTextBitmapStrikeThru(false);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setTextBitmapStrikeThru(int fd,false);
```

3.11.5 Configure text Underline

Function: void setTextBitmapUnderline(int fd, boolean haveUnderline);

Parameters: fd -> Package name initialization return value of the current application

haveUnderline -> true= configure underline, false= remove underline

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setTextBitmapUnderline(false);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setTextBitmapUnderline(int fd,false);
```

3.11.6 Configure text line spacing

函数: void setTextBitmapLineSpacing(int fd, float space);

参数: fd -> Package name initialization return tag value of the current application

space -> 1<= space <=255

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setTextBitmapLineSpacing(1);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setTextBitmapLineSpacing(int fd,1);
```

3.11.7 Configure text spacing between letters

Function: void setTextBitmapLetterSpacing(int fd, float space);

Parameters: fd -> Package name initialization return tag value of current application

space -> 1<= space <=255

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setTextBitmapLetterSpacing(1);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setTextBitmapLetterSpacing(int fd,1);
```

3.11.8 Configure text Anti-white

Function: void setTextBitmapAntiWhite(int fd, boolean antiWhite);

Parameters: fd -> Package name initialization return tag value of current application

antiWhite -> true= configure anti-white false = cancel anti-white

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setTextBitmapAntiWhite(false);
```

Not using PrinterHelper class:

```
iNeoPrinterService.setTextBitmapAntiWhite(int fd,false);
```

3.11.9 Text bitmap printing

Function: void printTextBitmap(int fd, String text, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

text -> The content to be printed

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printTextBitmap("text\n",null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printTextBitmap(int fd,"text\n",null);
```

3.11.10 Text bitmap print with alignment

Function: void printTextBitmapWithAli(int fd, String text, int align, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

text -> The content to be printed

align-> Print image alignment method 0= Align left 1= Center 2= Align right

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails

(the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printTextBitmapWithAli("text\n",0,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printTextBitmapWithAli(int fd,"text\n",0,null);
```

3.12 Bitmap printing

3.12.1 Bitmap printing

Function: void printBitmap(int fd, in Bitmap bitmap, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

bitmap- > Bitmap object

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printBitmap(bitmap,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printBitmap(int fd,bitmap,null);
```

3.12.2 Bitmap print with alignment

Function: void printBitmapWithAlign(int fd, in Bitmap bitmap, int alignmentMode, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

bitmap- > Bitmap object

alignmentMode-> Print text alignment mode 0= Align left 1= Center 2= Align right

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails

(the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printBitmapWithAlign(bitmap,0,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printBitmapWithAlign(int fd,bitmap,0,null);
```

3.12.3 Print multiple bitmap

Function: void printMultiBitmap(int fd, in List<Bitmap> bitmaps, IPrinterCallback callback);

Parameters: fd - > Package name initialization return tag value of the current application

bitmaps- > List of multiple bitmap to be printed

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printMultiBitmap(bitmaps,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printMultiBitmap(int fd,bitmaps,null);
```

3.12.4 Print multiple bitmap with alignment

Function: void printMultiBitmapWithAlign(int fd, in List<Bitmap> bitmaps, int alignmentMode, IPrinterCallback callback);

Parameters: fd - > Package name initialization return value of the current application

alignmentMode-> Print text alignment method 0= Align left 1= Center 2= Align right

bitmaps- > List of multiple bitmap to be printed

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying

printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printMultiBitmap(bitmaps,0,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printMultiBitmap(int fd,bitmaps,0,null);
```

3.12.5 Single colour bitmap processing and printing

Function: void printBitmapColorChart(int fd, in Bitmap bitmap, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

alignmentMode-> Print text alignment method 0=Align left 1=Center 2= Align right

bitmaps- >List of multiple bitmap to be printed

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printBitmapColorChart(bitmaps,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printBitmapColorChart(int fd,bitmaps,null);
```

3.12.6 Single colour bitmap processing and print with alignment

Function: void printBitmapColorChartWithAlign(int fd, in Bitmap bitmap,int alignmentMode, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

alignmentMode-> Print text alignment method 0=Align left 1= Center 2= Align right

bitmap- > The bitmap to be printed

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printBitmapColorChartWithAlign(bitmaps,0,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printBitmapColorChartWithAlign(int fd,bitmaps,0,null);
```

3.13 Print table chart

3.13.1 Print table in proportion to width weight

Function : void printColumnsString(int fd, in String[] colsTextArr, in int[] colsWidthArr, in int[] colsAlignArr, in int[] colsSizeArr,IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

colsTextArr -> Array of text strings of each column

colsWidthArr ->The width weight of each column, the ratio of the width

colsAlignArr -> Align method of each column (0 align left, 1 center, 2 align right)

colsSizeArr -> Font size of each column

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Notes: The length of the four arrays must be the same

Example:

```
PrinterHelper.getInstance().printColumnsString(in String[] colsTextArr, in int[] colsWidthArr, in int[] colsAlignArr, in int[] colsSizeArr,IPrinterCallback callback);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printColumnsString(int fd, in String[] colsTextArr, in int[] colsWidthArr, in int[] colsAlignArr, in int[] colsSizeArr,IPrinterCallback callback);
```

3.13.2 Print table according to the width value

Function : void printColumnsText(int fd, in String[] colsTextArr, in int[] colsWidthArr, in int[] colsAlignArr, in int[] colsSizeArr,IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

colsTextArr -> Array of text strings of each column

colsWidthArr- > An array of values for each column width (Based on English characters, each Chinese character is equivalent to two English characters, and each width is greater than 0)

colsAlignArr- > Alignment method of each column (0 align left, 1 center, 2 align right)

colsSizeArr- > The font size of each column

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Notes: The length of the four arrays must be the same

Example:

```
PrinterHelper.getInstance().printColumnsText(in String[] colsTextArr, in int[] colsWidthArr, in int[] colsAlignArr, in int[] colsSizeArr,IPrinterCallback callback);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printColumnsText(int fd, in String[] colsTextArr, in int[] colsWidthArr, in int[] colsAlignArr, in int[] colsSizeArr,IPrinterCallback callback);
```

3.14 1D code print related

3.14.1 Configure the width of 1D code

Function: void setBarcodeWidth(int fd, int width);

Parameters: fd - > Package name initialization return tag value of the current application

width - >The range of 1D code width value 2<= width <= 6

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setBarcodeWidth(2);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setBarcodeWidth(int fd,2);
```

3.14.2 Configure the height of 1D code

Function: void setBarcodeHeight(int fd, int height);

Parameters: fd - > Package name initialization return tag value of current application

height- > The height of 1D code, value range from 24<= height<= 250

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setBarcodeHeight(162);
```

Not using PrinterHelper utility class:


```
iNeoPrinterService.setBarCodeHeight(int fd,162);
```

3.14.3 Configure the position of 1D code HRI character

Function: void setBarCodeContentPrintPos(int fd, int pos);

Parameters: fd -> Package name initialization return tag value of the current application

pos-> HRI character position, value range from 0<= height<= 3 ,

0 do not print, 1 above of the 1D code, 2 below the 1D code, 3 above and below the 1D code

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setBarCodeContentPrintPos(0);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setBarCodeContentPrintPos(int fd,0);
```

3.14.4 Print 1D code

Function: void printBarCode(int fd, String data,int barCodeType, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

data-> The content of the 1D code

barCodeType-> The type of the 1D code

<item>0 UPC-A</item>

<item>1 UPC-E</item>

<item>2 JAN13 (EAN13) </item>

<item>3 JAN8 (EAN8) </item>

<item>4 CODE39</item>

<item>5 ITF</item>

<item>6 CODABAR</item>

<item>7 CODE93</item>

<item>8 CODE128</item>

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printBarCode("123456",4);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printBarCode(int fd,"123456",4);
```

3.14.5 1D code print with alignment

Function: void printBarCodeWithAlign(int fd, String data,int barCodeType, int alignmentMode,IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

data- > The content of the 1D code

barCodeType- > The type of the 1D code

<item>0 UPC-A</item>

<item>1 UPC-E</item>

<item>2 JAN13 (EAN13) </item>

<item>3 JAN8 (EAN8) </item>

<item>4 CODE39</item>

<item>5 ITF</item>

<item>6 CODABAR</item>

<item>7 CODE93</item>

<item>8 CODE128</item>

alignmentMode-> 1D code alignment mode 0=Align left 1=Center 2=Align right

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printBarCodeWithAlign("123456",4,0);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printBarCodeWithAlign(int fd,"123456",4,0);
```

3.14.6 1D code print with full parameters

Function : void printBarCodeWithFull(int fd, String data, int barCodeType, int width, int height, int textposition, int alignmentMode, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

data- > The content of the 1D code

barCodeType- > The type of 1D code

<item>0 UPC-A</item>

<item>1 UPC-E</item>

<item>2 JAN13 (EAN13) </item>

<item>3 JAN8 (EAN8) </item>

<item>4 CODE39</item>

<item>5 ITF</item>

<item>6 CODABAR</item>

<item>7 CODE93</item>

<item>8 CODE128</item>

width - > 1D code width value range from $2 \leq \text{width} \leq 6$

height- > 1D code height, value range from $24 \leq \text{height} \leq 250$

textposition- > HRI characters position, value range from $0 \leq \text{height} \leq 3$

alignmentMode-> 1D code alignment mode 0=Align left 1=Center 2= Align right

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printBarCodeWithFull("123456", 4, 2, 162, 0, 0, null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printBarCodeWithFull(int fd, "123456", 4, 2, 162, 0, 0, null);
```

Notes:

Barcode type (0-6,7,3,8)	Supported barcode Content length	Supported ASCII code range	
0 --> UPC-A	Barcode content length = 11,12	$48 \leq \text{range} \leq 57$	All models
1 --> UPC-E	Barcode content length = 11,12	$48 \leq \text{range} \leq 57$	All models
2 --> JAN13 / EAN13	Barcode content length =12,13	$48 \leq \text{range} \leq 57$	All models
3 --> JAN8 / EAN8	Barcode content length = 7 ,8	$48 \leq \text{range} \leq 57$	All models
4 --> CODE39	Barcode content length ≥ 1	$48 \leq \text{range} \leq 57, 65 \leq \text{range} \leq 90,$ range= 32, 36, 37, 42, 43, 45, 46, 47	All models
5 --> ITF	Barcode content length ≥ 2	$48 \leq \text{range} \leq 57$	All models
6 --> CODABAR	Barcode content length ≥ 2	$48 \leq \text{range} \leq 57,$ $65 \leq \text{range} \leq 68,$ $97 \leq \text{range} \leq 100,$ range = 36, 43, 45, 46, 47, 58	Z2 series don't support printing, other models support

73 -->CODE128, 8-->CODE128	Barcode content length >=2	0≤range≤127	All models
7 -->CODE93	1≤n≤255	1≤n≤255	
9 -> GS1128	条形码内容长度 >=2	0≤range≤127	
10 ->GS1DataBarOmnidirectional	条形码内容长度 =13	0≤range≤127	
11->GS1DataBarTruncated	条形码内容长度 =13	0≤range≤127	
12->GS1DataBarLimited	条形码内容长度 =13	0≤range≤127	
13->GS1DataBarExpanded	条形码内容长度 >=2	0≤range≤127	

Note: 9, 10, 11, 12, 13 only support PC80 printer firmware

3.15 QR code print

3.15.1 Configure the size of QR code

Function: void setQrCodeSize(int fd, int size);

Parameters: fd -> Package name initialization return tag value of the current application

size- > QR code size value range from 1<= size <= 11

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setQrCodeSize(2);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setQrCodeSize(int fd,2);
```

3.15.2 Configure QR code error correction level

Function: void setQrCodeErrorCorrectionLev(int fd, int level);

Parameters: fd -> Package name initialization return tag value of the current application

level- > QR code error correction level value range from 0<= level<= 3

0 ->Error correction level L(7%)

1 ->Error correction level M(15%)

2 ->Error correction level Q(25%)

3 ->Error correction level H(30%)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setQrCodeErrorCorrectionLev(2);
```

Not using PrinterHelper utility class:

```
NeoPrinterService.setQrCodeErrorCorrectionLev(int fd,2);
```

3.15.3 Print QR code

Function: void printQrCode(int fd, String data, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

data -> The content of QR code

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printQrCode("123456",null);
```

Not using PrinterHelper utility class:

```
NeoPrinterService.setQrCodeErrorCorrectionLev(int fd,"123456",null);
```

3.15.4 Print QR code with alignment

Function: void printQrCodeWithAlign(int fd, String data, int alignments, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of the current application

data -> The content of QR code

alignments -> QR code alignment mode 0= align left 1= center 2= align right

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printQrCodeWithAlign("123456",0,null);
```

Not using PrinterHelper utility class:

```
NeoPrinterService.printQrCodeWithAlign(int fd,"123456",null);
```

3.15.5 QR code print with full parameters

Function: void printQRCodeWithFull(int fd, String data, int size, int errorlevel, int alignments, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of current application

data- > The content of the QR code

size- > QR code size value range from 1<= size <= 11

level- >QR code error correction level value range from 0<= level<= 3

alignments -> QR code alignment mode 0= align left 1= center 2= align right

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printQRCodeWithFull("123456",1,2,0,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printQRCodeWithFull(int fd,"123456",1,2,0,null);
```

3.15.6. Print PDF417

Function: void printPDF417(String data, int dataRegionColumns, int rows, int moduleWidth, int rowHeight, int errorLevel, int selectOptions, int alignments)

Parameters: data -> The content of the QR code

dataRegionColumns -> Set the number of columns in the data range Default 0

Rows -> Set the number of rows

ModuleWidth -> Set the width of the module Default 2

RowHeight -> Set row height default 2

errorLevel- > Set error correction level Default m=49 n=1[10%]

SelectOptions -> Please select an option Default 0

Alignments -> QR code alignment mode 0= align left 1= center 2= align right

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printPDF417("122121221",0,0,6,4,1,0,1,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printPDF417(int fd,"122121221",0,0,6,4,1,0,1,null);
```

3.15.7. Print MaxiCode

Function: void printMaxiCode(String data, int modeType, int alignments)

Parameters: data ->The content of the QR code

modeType -> Select mode Default 50

alignments -> QR code alignment mode 0= align left 1= center 2= align right

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printMaxiCode("aaa11223232",50,1,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printMaxiCode(int fd,"aaa11223232",50,1,null);
```

3.15.8. Print AztecCode

Function: void printAztecCode(String data, int modeType, int dataLayers, int moduleSize, int errorLevel, int alignments)

Parameters: data ->The content of the QR code

modeType ,dataLayers -> Set schema type and number of data layers Default

modeType=0,dataLayers=0

moduleSize -> Set module size Default 3

errorLevel -> Set error correction level Default 23

alignments -> QR code alignment mode 0= align left 1= center 2= align right

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printAztecCode("099878676237842dd",0,16,6,23,1,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printAztecCode(int fd,"099878676237842dd",0,16,6,23,1,null);
```

3.15.9. Print DataMatrix

Function: void printDataMatrix(String data, int symbolType, int columns, int rows, int moduleSize, int alignments)

Parameters: data ->The content of the QR code

symbolType -> Set the symbol type, the default 0

columns -> Number of columns rows -> Rows moduleSize -> Set module size Default 3

alignments -> QR code alignment mode 0= align left 1= center 2= align right

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printDataMatrix("9ufdsfkasdf",0,32,32,5,1,null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.printDataMatrix(int fd,"9ufdsfkasdf",0,32,32,5,1,null);
```

3.16 Configure left margin

Function: void setLeftMargin(int fd, int valve);

Parameters: fd -> Package name initialization return tag value of the current application

valve- > Configure left margin value, range from: 0<=valve <=255

Notes: This properties settings is global, the value needs to be manually restored once set

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setLeftMargin(20);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setLeftMargin(int fd,20);
```

3.17 Print double QR code

3.17.1 Configure double QR code size

Function: void setDoubleQRSize(int fd, int size);

Parameters: fd -> Package name initialization return tag value of current application

size ->Double QR code size, value range from 1<= size <= 8

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setDoubleQRSize(1);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setDoubleQRSize(int fd,1);
```

3.17.2 Configure double QR code(QR1) left margin

Function: void setDoubleQR1MarginLeft(int fd, int qr1Left);

Parameters: fd -> Package name initialization return tag value of current application

qr1Left ->Double QR code QR1 left margin, value range from (0<= qr1Left <=255)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setDoubleQR1MarginLeft(10);
```


Not using PrinterHelper utility class:

```
iNeoPrinterService.setDoubleQR1MarginLeft(int fd,10);
```

3.17.3 Configure QR code (QR2) left margin

Function: void setDoubleQR2MarginLeft(int fd, int qr2Left);

Parameters: fd -> Package name initialization return tag value of current application

qr2Left -> Double QR code QR2 left margin, value range from (0<= qr2Left<=255)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setDoubleQR2MarginLeft(200);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setDoubleQR2MarginLeft(int fd,200);
```

3.17.4 Configure double QR code (QR1) error level

Function: void setDoubleQR1Level(int fd, int qr1Level);

Parameters: fd -> Package name initialization return tag value of current application

qr1Level -> Double QR code QR1 error, value range from (0<= qr1Level<=3)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setDoubleQR1Level(1);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setDoubleQR1Level(int fd,1);
```

3.17.5 Configure QR code (QR2) error

Function: void setDoubleQR2Level(int fd, int qr2Level);

Parameters: fd -> Package name initialization return tag value of current application

qr2Level -> Double QR code QR2 error, value range from (0<= qr2Level<=3)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setDoubleQR2Level(1);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setDoubleQR2Level(int fd,1);
```

3.17.6 Configure double QR code (QR1) version

Function: void setDoubleQR1Version(int fd, int qr1Version);

Parameters: fd -> Package name initialization return tag value of current application

qr1Version -> Double QR code QR1 error, value range from (0<= qr1Version<=3)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setDoubleQR1Version(1);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setDoubleQR1Version(int fd,1);
```

3.17.7 Configure double QR code (QR2) version

Function: void setDoubleQR2Version(int fd, int qr2Version);

Parameters: fd -> Package name initialization return tag value of current application

qr2Version ->Double QR code QR2 error, value range from (0<= qr2Version<=3)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().setDoubleQR2Version(1);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setDoubleQR2Version(int fd,1);
```

3.17.8 Print double QR code

Function: void printDoubleQR(int fd, String qr1Data,String qr2Data,IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of current application

qr1Data -> The content of QR1

qr2Data -> The content of QR2

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().printDoubleQR("123456&147","fsdfsdfsdsd144411444&&&&",null);
```

Not using PrinterHelper utility class:

```
iNeoPrinterService.setDoubleQR2Version(int fd, "123456&147", "fsdfsdfsdsd11444&&&&",null);
```

3.18 Transaction printing

The transaction print mode is suitable for those who need to control what content is printed and obtain a return indication of the printed result (whether or not a receipt is printed), This mode is equivalent to creating a transaction queue buffer. When developer enters transaction print mode, it will establish a transaction queue and add additional printing method. At this moment, the printer will not print any content immediately until the transaction is submitted to the printer, then only the printer

will execute the print according to the queue buffer. It will receive a return result when the transaction printing is completed.

Things to take note for transaction printing:

1. When enter buffering mode (transaction), it will return a "success" result when the prints are submitted successfully. However, if the printer encounters abnormal scenario, such as paper missing, overheating, etc., all the instructions and printing tasks that have been submitted will be terminated and return indication showing abnormal status. This also indicates that when the printer is abnormal before or during the execution of a single task, the order will not be printed.
2. When command printing and buffer (transaction) printing are used interchangeably, if the printer is abnormal, the content of the print instruction will not be cleared!
3. When enter transaction printing mode, the printer will not print immediately, the content will be queued and buffered, it will print only when users execute `exitPrinterBuffer()` or `commitPrinterBuffer()`.
4. The transaction print result callback is via the `onPrintResult(int code, String msg)` of the `IPrinterCallback` method (which can be time-consuming. It is not recommended to use transaction printing frequently for a single line, which will affect the printing speed. It is recommended to use transaction printing for the whole receipt.)

Return value code as below:

- a) 0 ! It indicates that the "Transaction print successful!";
- b) 1 ! It indicates that "Transaction print failed!";

3.18.1 Enter transaction print mode

Function: `void enterPrinterBuffer(int fd, boolean clean);`

Parameters: `fd` -> Package name initialization return tag value of current application

`clean` -> Confirm to clear transaction queue data

`true` -> Clear transaction queue data that are not printed yet

`false` -> Do not clear transaction queue data that are not printed. It will be printed during the next transaction submission.

Example:

Using `PrinterHelper` utility class:

```
PrinterHelper.getInstance().enterPrinterBuffer(false);
```

Not using `PrinterHelper` utility class:

```
iNeoPrinterService.enterPrinterBuffer(int fd, false);
```

3.18.2 Submit a transaction

Function: `void commitPrinterBuffer(int fd);`

Parameters: `fd` -> Package name initialization return tag value of current application

Example:

Using `PrinterHelper` utility class:

```
PrinterHelper.getInstance().commitPrinterBuffer();
```

Not using `PrinterHelper` utility class:

```
iNeoPrinterService.commitPrinterBuffer(int fd);
```

3.18.2 Transaction submission callback result

Function: void commitPrinterBufferWithCallback(int fd, IPrinterCallback callback);

Parameters: fd - >Package name initialization return tag value of current application

callback ->onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

->onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

->onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

->onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().commitPrinterBufferWithCallback(callback);
```

Not using PrinterHelper utility class:

```
INeoPrinterService.commitPrinterBufferWithCallback(int fd, callback);
```

```
PrinterHelper.getInstance().commitPrinterBuffer(new INeoPrinterCallback() {
```

```
    @Override
```

```
    public void onRunResult(boolean isSuccess) throws RemoteException {
```

```
        Log.d(TAG, " printTextWithAli onRunResult ==>>>> " +isSuccess);
```

```
    }
```

```
    @Override
```

```
    public void onReturnString(String result) throws RemoteException {
```

```
        Log.d(TAG, " onReturnString ==>>>> " +result);
```

```
    }
```

```
    @Override
```

```
    public void onRaiseException(int code, String msg) throws RemoteException {
```

```
    }
```

```
    @Override
```

```
    public void onPrintResult(int code, String msg) throws RemoteException {
```

```
        Log.d("NeoPrinterSDK_ transaction print callback", " transaction print result ==>>>> » onPrintResult
```

```
==>>>> " + " the code of transaction print => "+code+" , transaction print description ==>> " +msg);
```

```
        binding.tvResult.setText(" transaction print code=> "+code+" , transaction print description ==>>
```

```
 "+msg);
```

```
    }
```

```
}}
```

3.18.3 Terminate transaction print

Function: void exitPrinterBuffer(int fd, boolean commit);

Parameters: fd -> Package name initialization return tag value of current application

commit -> Confirm to remove transaction queue data true clear transaction queue buffer data, false do not clear queue

buffer

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().exitPrinterBuffer(true);
```

Not using PrinterHelper utility class:

```
INeoPrinterService.exitPrinterBuffer(int fd,true);
```

3.18.4 Terminate transaction print callback result

Function: void exitPrinterBufferWithCallback(int fd, boolean commit, IPrinterCallback callback);

Parameters: fd -> Package name initialization return tag value of current application

commit -> Confirm to clear queue buffer data true Clear queue buffer data, false do not clear queue buffer

callback -> onRunResult(boolean isSuccess): Returns the interface execution result (returns the result after the execution and printing is completed true success false failure)

-> onRaiseException(int code, String msg): The specific reason for the exception when the interface execution fails (the method cannot be executed in the code flow)

-> onPrintResult(int code, String msg): Return printer result code=1 success 0 failure (result returned after printing is completed)

-> onReturnString(String result): Returns the interface execution result (string data) (callback result when querying printer-related information)

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().exitPrinterBufferWithCallback(true,callback);
```

Not using PrinterHelper utility class:

```
INeoPrinterService.exitPrinterBufferWithCallback(int fd,true,callback);
```

```
PrinterHelper.getInstance().exitPrinterBufferWithCallback(true,new INeoPrinterCallback() {
```

```
    @Override
```

```
    public void onRunResult(boolean isSuccess) throws RemoteException {
```

```
        Log.d(TAG," printTextWithAli onRunResult ==>>> "+isSuccess);
```

```
    }
```

```
    @Override
```

```
    public void onReturnString(String result) throws RemoteException {
```

```
        Log.d(TAG," onReturnString ==>>> "+result);
```

```
    }
```

```
    @Override
```

```
    public void onRaiseException(int code, String msg) throws RemoteException {
```

```
    }
```

```
    @Override
```

```
    public void onPrintResult(int code, String msg) throws RemoteException {
```

```
        Log.d("NeoPrinterSDK_ transaction print callback"," transaction print result ==>>> ) onPrintResult
```

```
==>>> "+transaction print code=> "+code+" , transaction print description ==>> "+msg);
```

```
binding.tvResult.setText("transaction print code=> "+code+" , transaction print description ==> "+msg);
```

```
}
```

```
}}
```

Whole transaction printing example:

```
enterPrinterBuffer(true) // After entering transaction mode, the subsequent command will not be executed immediately
```

```
    printText(/*something*/)
```

```
    printBitmap(/*bitmap resource*/)
```

```
    // ..... Other printing related method—— Print some content
```

```
    commitPrinterBuffer()/commitPrinterBufferWithCallback(callback)//Submit a transaction , the printer will begin to print, it will then return the print status via callback, whether is it successful or fail.
```

```
    ..... Waiting for the submitted transaction callback
```

```
    printText(/*something*/)
```

```
    printBitmap(/*bitmap resource*/)
```

```
    //..... other printing method —— can choose to wait, or not to wait for the previous submitted transaction callback and continue printing
```

```
    commitPrinterBuffer()/commitPrinterBufferWithCallback(callback)//Continue to submit the next transaction , the printer will continue to print
```

```
    exitPrinterBuffer(true)/exitPrinterBufferWithCallback(true, callback)//Terminate transaction printing mode , if new data is being submitted after the previous submission, printer will continue to print, otherwise the printing task will stop.
```

3.19 Printer upgrade

3.19.1 To obtain printer upgrade status

Function: `int getPrinterIsUpdateStatus(int fd, IPrinterCallback callback);`

Return value description: 0 -> Printer status is normal, 1 -> Manual update is in progress 2 -> Automatic update in progress

Example:

Using PrinterHelper utility class:

```
PrinterHelper.getInstance().getPrinterIsUpdateStatus(null);
```

Not using PrinterHelper utility class:

```
NeoPrinterService.getPrinterIsUpdateStatus(int fd, null);
```

4. Definition Explanation of Label Printing Interface (New)

Only supports Swan 2 label printers

4.1. Canvas Settings

Interface: `void labelInitCanvas(LabelCanvasStyle labelCanvasStyle)`

Function: Specify the canvas size and basic format for printing labels

Note: To render and print content using a canvas, the canvas size to be drawn must be initialized first to ensure final printability;

LabelCanvasStyle explain

Available methods	Method description	Parameter Description	Default value
setWidth	Initialize the canvas width for pre rendering	Can be set to a size not exceeding the width of the printing paper	Default 0 pixels
setHeight	Initialize the height of the pre rendered canvas	Can be set to a size not exceeding the height of the printing paper	Default 0 pixels
setPosX	Specify the starting horizontal axis of the canvas relative to the label	Unit Pixel	Default 0 pixels
setPosY	Specify the starting vertical axis of the canvas relative to the label	Unit Pixel	Default 0 pixels

Note: setWidth and setHeight must specify sizes in order to draw content;

4.2 Draw text content

Interface: `void labelAddText(String text, LabelTextStyle labelTestStyle)`

Function: Draw text content to a designated area on the canvas

String text explain

The printing content drawn in the designated area

LabelTextStyle explain

Available methods	Method description	Parameter Description	Default value
setPosX	Set the text content at the starting horizontal axis position of the canvas	Unit Pixel	0
setPosY	Set the text content to the starting vertical coordinate position of the canvas	Unit Pixel	0
setTextSize	Specify the size of text characters	Effective range 6-96 pixels	24
setTextWidthRatio	Specify the multiple width size	Effective range 0-9	0
setTextHeightRatio	Specify the multiplier size	Effective range 0-9	0
setWidth	Set text width limit	If a width limit is set, exceeding the width will automatically wrap the line	Unrestricted (-1)
setHeight	Set text height limit	If a height limit is set, the excess part will not be displayed	Unrestricted (-1)
setAlign	Set the position of the text content relative to the starting coordinates	Align	Align.DEFAUL
setRotate	Set the direction of text content	Rotate	horizontal direction
setTextSpace	Set text spacing	0~100 pixels	0
setEnableBold	Set text bold	Enable text bolding function	Do not open
setEnableUnderline	Set text underline	Enable text underline function	Do not open
setEnableStrikethrough	Set text strikethrough	Enable text strikethrough function	Do not open
setEnableItalics	Set text italics	Enable text italicization function	Do not open

SetEnAntiColor	Set Text Whitening	Enable text whitewashing function	Do not open
----------------	--------------------	-----------------------------------	-------------

4.3.Draw barcode content

Interface: void labelAddBarCode(String codeData, LabelBarCodeStyle labelBarCodeStyle)

Function: Draw barcode content to a designated area on the canvas

String codeData explain

The content of pre drawn barcodes varies depending on the code range required by different code systems

LabelBarCodeStyle explain

Available methods	Method description	Parameter Description	Default value
setPosX	Set barcode content at the starting horizontal axis position of the canvas	Unit Pixel	0
setPosY	Set barcode content at the starting vertical coordinate position of the canvas	Unit Pixel	0
setDotWidth	Set code block width	1-16 pixels Will affect the total width of the final barcode 1-255 pixels	2
setBarHeight	Set barcode height	Will affect the total height of the final barcode	162
setReadable	Set HRI location	HumanReadable	Not displaying
setSymbology	Set barcode type	Symbology	code128
setAlign	Set the relative starting coordinate	Align	Align.DEFAUL

	position of the barcode		
setRotate	Set barcode rotation direction	Rotate	horizontal direction
setWidth	Specify barcode scaling width	When the scaling width is set, it will force a change in the size of the code content	Do not scale
setHeight	Specify barcode scaling height	When the scaling width is set, it will force a change in the size of the code content	Do not scale

4.4. Draw QR code content

Interface: `labelAddQrCode(String qrData, LabelQrCodeStyle labelQrCodeStyle)`

Function: Draw QR code content to a designated area on the canvas

String qrData explain : Content of pre printed QR code

4.5. Draw an image

Interface: `void labelAddBitmap(Bitmap bitmap, LabelBitmapStyle labelBitmapStyle)`

Function: Draw an image

Bitmap bitmap explain

The bitmap object to be rendered

Note: The image content requires a non transparent background image; Try to keep the image size within the canvas area, otherwise the content will be cropped and displayed

LabelBitmapStyle explain

Available methods	Method description	Parameter Description	Default value
setPosX	Set the image to the starting horizontal axis	Exceeding the canvas size will not print	0

	position on the canvas		
setPosY	Set the starting vertical coordinate position of the image on the canvas	Exceeding the canvas size will not print	0
setAlgorithm	Set image conversion method	ImageAlgorithm	ImageAlgorithm.BINARIZATION
setValue	Set algorithm float value	Floating values vary depending on the specific algorithm	ImageAlgorithm
setWidth	Specify the zoom width of the image	When setting the zoom width, the image size will be forcibly changed	Do not scale
setHeight	Specify the zoom height of the image	When setting the zoom width, the image size will be forcibly changed	Do not scale

LabelQRCodeStyle explain

Available methods	Method description	Parameter Description	Default value
setPosX	Set the QR code content at the starting horizontal axis position of the canvas	Unit Pixel	0
setPosY	Set the QR code content at the starting vertical axis position of the canvas	Unit Pixel	0
setSize	Set QR code block size	1-16 pixels Ultimately, it will affect the size of the QR code	4
setErrorLevel	Set the QR code error correction level	ErrorLevel	ErrorLevel.L
setRotate	Set QR code rotation direction	Rotate	horizontal direction

setWidth	Specify the QR code scaling width	When the scaling width is set, it will force a change in the size of the code content	Do scale	not
setHeight	Specify the QR code zoom height	When the scaling width is set, it will force a change in the size of the code content	Do scale	not

4.6 Draw special graphics

Interface: void labelAddArea(LabelAreaStyle labelAreaStyle)

Function: Draw graphics within the designated area of the canvas

LabelAreaStyle explain

Available methods	Method description	Parameter Description	Default value
setStyle	Set drawing shape Set the width of the graphic	Shape	Shape.RECT_FILL
setWidth	Invalid when the shape is a line segment When the shape is circular, it represents the diameter of the circle Set the height of the graphic	Unit Pixel	50
setHeight	Invalid when the shape is a line segment	Unit Pixel	50
setPosX	Invalid when the shape is circular Set the starting x-coordinate	Unit Pixel	0
setPosY	Set the starting y-coordinate	Unit Pixel	0
setEndX	Set the x-coordinate of the endpoint of the line segment	Unit Pixel	50
setEndY	Set the endpoint y-coordinate of the line segment	Unit Pixel	50
setThick	Set stroke	Unit Pixel	1

4.7 Print and draw content

Interface: void labelPrintCanvas(int count, ILabelPrintResult iLabelPrintResult)

Function: If the canvas content is valid, the specified printer will print the canvas content

count explain

The number of times to print, if it is a label printer, will be printed on each label separately, with a printing quantity greater than 0 sheets

ILabelPrintResult explain

ResultCode : Printing success returns 0, printing failure returns <0

Message : Additional information upon failure

1. Example

Build a complete style template consisting of a title, QR code, text, and barcode for label printing

50mm * 30 mm Label Template



Code implementation:

```
PrinterHelper.getInstance().labelInitCanvas(LabelCanvasStyle.getCanvasStyle())
```

```
.setWidth(50 * 8)
.setHeight(30 * 8)
.setPosX(48));
PrinterHelper.getInstance().labelAddArea(LabelAreaStyle.getAreaStyle()
.setStyle(Shape.BOX)
.setPosX(18)
.setPosY(2)
.setWidth(50 * 8 - 22)
.setHeight(30 * 8 - 24)
.setThick(2));
PrinterHelper.getInstance().labelAddText("POS Label Test", LabelTextStyle.getTextStyle()
.setPosX(57)
.setPosY(43)
.setTextSize(40)
.setEnableBold(true)
.setEnAntiColor(true)
//.setAlign(Align.CENTER)
.setTextSpace(2));
PrinterHelper.getInstance().labelAddArea(LabelAreaStyle.getAreaStyle()
.setStyle(Shape.PATH)
.setPosX(18)
.setPosY(53)
.setEndX(50 * 8 - 4)
.setEndY(53)
.setThick(2));
PrinterHelper.getInstance().labelAddText("Price:",LabelTextStyle.getTextStyle()
.setPosX(26)
.setPosY(88)
.setTextSize(28));
PrinterHelper.getInstance().labelAddText("888",LabelTextStyle.getTextStyle()
.setPosX(96)
.setPosY(88)
.setTextSize(28)
.setEnableStrikethrough(true));
PrinterHelper.getInstance().labelAddText("¥666",LabelTextStyle.getTextStyle()
.setPosX(156)
.setPosY(93)
.setTextSize(36)
.setEnableBold(true));

PrinterHelper.getInstance().labelAddText("Origin:Shenzhen",LabelTextStyle.getTextStyle()
.setPosX(26)
.setPosY(128)
.setTextSize(28));
```

```

PrinterHelper.getInstance().labelAddBarCode("12345678",
LabelBarCodeStyle.getBarCodeStyle()
    .setPosX(26)
    .setPosY(138)
    .setSymbology(Symbology.CODABAR)
    .setDotWidth(2)
    .setBarHeight(45)
    .setReadable(HumanReadable.POS_TWO));
PrinterHelper.getInstance().labelAddQrCode("123456789",
LabelQrCodeStyle.getQrCodeStyle()
    .setPosX(255)
    .setPosY(80)
    .setSize(6)
    .setErrorLevel(ErrorLevel.H));

PrinterHelper.getInstance().labelPrintCanvas(1,null);

```

2. Printing label skills

2-1.Print a blank label

When printing a test blank label, it can be achieved through the following methods

```

PrinterHelper.getInstance().labelInitCanvas(LabelCanvasStyle.getCanvasStyle()
    .setWidth(50 * 8)
    .setHeight(30 * 8));
PrinterHelper.getInstance().labelPrintCanvas(1,null);

```

2-2.Print image labels

When it is necessary to print the image label template generated by the customer, the following interface can be directly called to achieve label printing

Interface: void labelPrintBitmap(Bitmap bitmap, int width, int height, IPrinterCallback callback){

Example:

```

PrinterHelper.getInstance().labelPrintBitmap(picBitmap,picBitmap.getWidth()/8,picBitmap.getHeight() / 8, null);

```

3.Description of enumeration parameters

Align Alignment method

List content	explain
DEFAULT	Default left
LEFT	Align to the left
CENTER	Center Alignment
RIGHT	Right aligned

Rotate Draw direction

List content	explain
ROTATE_0	Level 0 degrees
ROTATE_90	90 degree direction
ROTATE_180	180 degree direction
ROTATE_270	270 degree direction

HumanReadable Barcode content readable location

List content	explain
HIDE	hide
POS_ONE	Bottom on the left
POS_TWO	Bottom centered
POS_THREE	Bottom on the right

SymbologyBarcode type (detailed reference: 3.14.6 one-dimensional code printing transmission full parameters, comments inside)

List content	explain
UPCA	UPC-A Barcode product barcode is a pure number with 11 digits After encoding, an additional checksum is added to form a 12 digit number, mainly used in the United States and Canada
UPCE	UPC-E Barcode product barcode is a pure number, reduced from UPC-A, with 7 digits and the first digit must be 0 After encoding, add a checksum to form an 8-bit number
EAN13	EAN13 The product barcode is a pure number, and the number of digits is 12 After encoding, add a checksum to form a 13 digit number
EAN8	EAN8 product barcode is a pure number, and the number of digits is 7 After encoding, add a checksum to form an 8-bit number
CODE39	Code39 barcode generation character set includes numbers, uppercase letters, and - . \$/+%* Characters such as spaces The '*' is only used to mark the beginning and end
ITF	Interleaved 2 of 5 barcode generation, commonly used in logistics management The character set consists only of numbers and has an even number. If it is an odd number, it will automatically be preceded by a '0'
CODABAR	Codabar barcode generation, character set includes numbers and - \$:/+ And characters such as ABCD ABCD is only used for the beginning or end, and is used as an identifier
CODE93	Code93 barcode generation is in full ASCII mode, which can use all 128 characters of ASCII
CODE128	Combination code 128a code128b、 code128c, Need to dynamically switch according to the code content

ErrorLevel QR code error correction level

List content	explain
L	Error correction level L 7%

M	Error correction level	M 15%
Q	Error correction level	Q 25%
H	Error correction level	H 30%

ImageAlgorithm Image conversion algorithm

List content	explain	Reference floating value
BINARIZATION	Binary algorithm	The binarization algorithm converts different color values into black by adjusting the floating value You can choose a general floating value of 200 based on the color information in the image
DITHERING	Jitter grayscale algorithm	The jitter grayscale algorithm does not need to consider changes in floating values

Shape Special graphics

Special graphics explain

RECT_FILL	Fill the rectangular area with black blocks
RECT_WHITE	Erase the rectangular area into white blocks
RECT_REVERSE	Whitening rectangle: The rectangular area will be whitened
BOX	hollow rectangle
CIRCLE	Hollow circular
OVAL	hollow ellipse
PATH	line segment

4.Integrate the printer via built-in virtual bluetooth connection

2.1 Virtual Bluetooth introduction

You are able to see a paired and ever-present Bluetooth device - **"InnerPrinter"** in the list of bluetooth device, this is the printer device virtualized by the operating system, which does not actually exists. Virtual Bluetooth supports iMin «esc/pos» command, which there are some special commands are self-defined by iMin, for example:

Function	Command
Open cash drawer command	byte [5] : 0x10 0x14 0x00 0x00 0x00
Cut paper command, complete cut	byte [2] : 0x1B 0x69
Cut paper command, leave a little space one the left and do not cut	byte [2] : 0x1d 0x56
Cut paper command, complete cut	: 0x1d, 0x56, 0x00(Only PC80, which is a self-developed printer, supports this command involving continuing D4 Pro, Swan 2, Falcon 2)

```
Half cut : 0x1d, 0x56, 0x01
Half cut : 0x1d, 0x42, 0x00, 0x00 // Walk to the paper tearing position
(recommended to use)
```

2.2 Virtual Bluetooth usage

2.2.1. Establish a connection with the Bluetooth device

2.2.2. Combine the commands and text content and transcode to Bytes

2.2.3. Send to BluetoothPrinter.

2.2.4. The underlying printing service drives the printing device to complete the printing

Note: BluetoothUtel is a type of Bluetooth tool used to connect the virtual Bluetooth device **InnerPrinter** to iMin printing devices.

2.2.4.1. Utility class BluetoothUtil , is the standard Bluetooth utility class

Source code example:

```
public class BluetoothUtil {

    /**
     * Is the Bluetooth enabled?
     */
    public static boolean isBluetoothOn() {
        BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
        if (mBluetoothAdapter != null)
            // Bluetooth is enabled
            if (mBluetoothAdapter.isEnabled())
```

```

        return true;

    }

    return false;
}

/**
 * Select a specific type of device to be shown from the list of paired devices
 * @param deviceClass
 * @return
 */
public static BluetoothDevice getDevice() {
    BluetoothDevice innerprinter_device = null;
    Set<BluetoothDevice> devices = BluetoothAdapter.getDefaultAdapter().getBondedDevices();
    for (BluetoothDevice device : devices) {
        if (device.getAddress().equals(Innerprinter_Address)) {
            innerprinter_device = device;
            break;
        }
    }
    return innerprinter_device;
}

/**
 * Pop-up a dialog requesting to enable Bluetooth
 */
public static void openBluetooth(Activity activity) {
    Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
    activity.startActivityForResult(enableBtIntent, 666);
}

public static BluetoothSocket connectDevice(BluetoothDevice device) {
    //Built-in printer bluetooth 00001101-0000-1000-8000-00805F9B34FB
    BluetoothSocket socket = null;
    try {
        socket = device.createRfcommSocketToServiceRecord(
            UUID.fromString("00001101-0000-1000-8000-00805f9b34fb");//Third-party Bluetooth printer
            00000000-0000-1000-8000-00805f9b34fb 00001101-0000-1000-8000-00805f9b34fb
        );
        socket.connect();
        Log.i("imin_print_cmd_1111", "=====8888888888888888= Bluetooth is connected
        successfully 11111 ");
    } catch (IOException e) {
        Log.i("imin_print_cmd_1111", "=====8888888888888888=====
        e.getMessage()+"e.getMessage());
    }
}

```

```

        socket.close();
    } catch (IOException closeException) {
        return null;
    }
    return null;
}
return socket;
}
}

```

```

private static OutputStreamWriter mWriter = null;
private static OutputStream mOutputStream = null;

```

```

public static void openOutputStream(OutputStream outputStream, String encoding) throws IOException {
    mWriter = new OutputStreamWriter(outputStream, encoding);
    mOutputStream = outputStream;
}

```

```

}

public static void sendData(byte[] bytes, BluetoothSocket socket) throws IOException {
    if(socket != null){
        OutputStream out = socket.getOutputStream();
        out.write(bytes, 0, bytes.length);
        out.close();
    }
}
}
}

```

2.2.4.2. Bluetooth connection print service example

1. To determine whether the system Bluetooth function is enabled

```

if (BluetoothUtil.isBluetoothOn()) {
    //
} else {
    BluetoothUtil.openBluetooth(BluetoothActivity.this);
}
}

```

2. To detect iMin built-in Bluetooth device

```

BluetoothDevice device = BluetoothUtil.getDevice(btAdapter);
if (device == null) {
    Toast.makeText(getBaseContext(), "Please Make Sure Bluetooth have InnterPrinter!",
        Toast.LENGTH_LONG).show();
}

```

```
return;  
}
```

3. Receipt data to be printed

```
byte[] b = null;
```

4. Print receipt data via built-in Bluetooth printer

```
BluetoothSocket socket = null; socket = BluetoothUtil.getSocket(device);  
BluetoothUtil.sendData(data, socket);
```

5. How to obtain printer status using Bluetooth

5.1 Register a broadcast listener within the onCreate() method of the application

```
try {  
    IntentFilter intentFilter = new IntentFilter();  
    intentFilter.addAction(ACTION_PRINTER_STATUS_CHANGE);  
    registerReceiver(mReceiver, intentFilter);  
} catch (Exception e) {  
}
```

5.2 Handle the callback of status value in broadcast mode

```
private static final String ACTION_PRINTER_STATUS = "status";  
    ///Status value  
    public static final int PRINTER_NORMAL = 0; //Normal  
    public static final int PRINTER_UNCAP = 3; ///Printer door open  
    public static final int PRINTER_LOWER_POWER = 4; //Low power  
    public static final int PRINTER_OVER_HEAT = 5; //Overheat  
    public static final int PRINTER_CUTTING_ERROR = 6; //Paper cutter jam  
    public static final int PRINTER_PAPER_OUT = 7; //Paper missing  
    public static final int PRINTER_OTHER_ERROR = 99; //Other errors  
    private TextView tv_status;  
    private BroadcastReceiver mReceiver = new BroadcastReceiver() {  
        @Override  
        public void onReceive(Context context, Intent intent) {  
            int status = intent.getIntExtra(ACTION_PRINTER_STATUS, -1);  
            String PrinterStatus = "";  
            switch (status) {  
                case PRINTER_NORMAL:  
                    PrinterStatus = "Normal";  
                    break;  
                case PRINTER_UNCAP:  
                    PrinterStatus = "Printer door open";  
                    break;
```

```

        case PRINTER_OVER_HEAT:
            PrinterStatus = "Overheat";
            break;
        case PRINTER_LOWER_POWER:
            PrinterStatus = "Low power";
            break;
        case PRINTER_CUTTING_ERROR:
            PrinterStatus = "Paper cutter jam";
            break;
        case PRINTER_PAPER_OUT:
            PrinterStatus = "Paper missing";
            break;
        case PRINTER_OTHER_ERROR:
            PrinterStatus = "Other errors";
            break;
    }
}
};

```

5.3 Register broadcast listener in application onDestroy()

```

    try {
        unregisterReceiver(mReceiver);
    } catch (Exception e) {
    }
}

```

2.2.5. Items to take note

You need to add a Bluetooth permission statement in the Application project in order to use the Bluetooth devices:

```

<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

```


5.H5 Web Page integrate with printer through JS Bridge

5.1 H5 Web page integrate with printer plug-in example

```
$('#btn').click(function() {  
  
    // First method: Native print  
    var mywindow = window.open("", 'PRINT', 'height=400,width=600');  
    mywindow.document.write('<html><head><title>' + document.title + '</title>');  
    mywindow.document.write('</head><body >');  
    mywindow.document.write(document.getElementById('PrintContent').innerHTML);  
    mywindow.document.write('</head><body >');  
    mywindow.print();  
  
})
```

5.2 Integrate with jquery plug-in to print

5.2.1、 Simple method to introduce

5.2.1.1.Introduce print SDK script

```
<script type="text/javascript" src="./imin-printer.min.js"></script>
```

5.2.1.2. Initialization

```
var IminPrintInstance = new IminPrinter();
```

5.2.1.3. Integration related

```
IminPrintInstance.connect().then(async (isConnect) => {
```

```
  if (isConnect) {  
    // Initialize printer  
    IminPrintInstance.initPrinter()  
    // Obtain printer status  
    await IminPrintInstance.getPrinterStatus()  
  }  
})
```

5.2.2 Implement vue-cli scarfolding in vue2

5.2.2.1. Configure imin-printer package in vue.config.js

```
const { defineConfig } = require("@vue/cli-service");  
const path = require('path');  
console.log(path.join(__dirname, 'src/assets/imin-printer.esm.browser.min.js'))  
module.exports = defineConfig({  
  transpileDependencies: true,  
  configureWebpack: {  
    resolve: {  
      alias: {  
        'imin-printer': path.join(__dirname, 'src/assets/imin-printer.esm.browser.min.js')  
      }  
    },  
    plugins: [  
    ]  
  },  
});
```

5.2.2.2. Import package to main.js

```
import Vue from 'vue'
import App from './App.vue'
import IminPrinter from 'imin-printer';

Vue.config.productionTip = false

Vue.use(IminPrinter)

new Vue({
  printer: new IminPrinter(),
  render: h => h(App),
}).$mount('#app')
```

5.2.2.3. To use it in App.vue or components

```
export default {
  name: 'App',
  data() {
    return {
      isConnect: false
    }
  },
  created() {
    this.init();
  },
  methods: {
    async init() {
      this.isConnect = await this.$printer.connect()
    },
    async getPrinterStatus() {
      return await this.$printer.getPrinterStatus()
    },
    handleClick() {
      if (!this.isConnect) return false
      console.log(this.getPrinterStatus());
    }
  },
  beforeDestroy() {
    this.isConnect = false
  }
}
```

```
}
```

5.2.3、 Implement vite scarfolding in vue3

5.2.3.1. To configure imin-printer package in vite.config.js

```
import { fileURLToPath, URL } from 'node:url'

import { defineConfig } from 'vite'
import vue from '@vitejs/plugin-vue'
// https://vitejs.dev/config/
export default defineConfig({
  plugins: [
    vue(),
  ],
  resolve: {
    alias: {
      '@': fileURLToPath(new URL('./src', import.meta.url)),
      'imin-printer': fileURLToPath(new URL('./src/assets/imin-printer.esm.browser.min.js', import.meta.url))
    }
  }
})
```

5.2.3.2. Import package to main.js

```
import IminPrinter from 'imin-printer';
import { createApp } from 'vue'
import App from './App.vue'
const app =createApp(App)
app.config.globalProperties.$printer = new IminPrinter('10.0.21.53');
app.mount('#app')
```

5.2.3.3. Use it in App.vue or components

```
import { getCurrentInstance, ref } from 'vue'
const { proxy } = getCurrentInstance()
const isConnected = ref(false)
const init = async () => {
  isConnected.value = await proxy.$printer.connect()
}
init();
const getPrinterStatus = async() => {
  return await this.$printer.getPrinterStatus()
}
const handleClick = () => {
  if (!isConnect.value) return false
  console.log(getPrinterStatus());
}
```

5.3. Api Description

Initialize printer (Only support SPI/USB printing)

Function: `initPrinter()`

Obtain printer status

Function: `getPrinterStatus(IminPrintInstance.PrintConnectType, callback)`

Printer status description:

- 1 -> Not connected to service
- 3 -> Printer door opened
- 4 -> Printerhead is overheated
- 7 -> Paper is missing
- 0 -> Printer is normal

Example:

```
IminPrintInstance.getPrinterStatus(IminPrintInstance.PrintConnectType.SPI, function (status) {2 console.log('printer status:' + status.value);3})
```

Feed paper with one line

Function: `printAndLineFeed()`

Example:

```
lminPrintInstance.printAndLineFeed();
```

User-defined paper feed spacing

Function: `printAndFeedPaper(value)`

Parameters: `value` paper feeding distance range (0<value<255)

Example:

```
lminPrintInstance.printAndFeedPaper(100);
```

Cut paper

Function: `partialCut()`

Example:

```
lminPrintInstance.partialCut();
```

Configure alignment

Function: `setAlignment(alignment)`

Parameters:

`alignment` ->

0 = Align left

1 = Center

2 = Align right

Default= 0

Example:

```
lminPrintInstance.setAlignment(1);
```

Configure print font size

Function: `setTextSize(size)`

Parameters: `size`, by default is 28

Example:

```
lminPrintInstance.setTextSize(26);
```

Configure print font type

Function: `setTextTypeface(typeface)`

Parameters: `typeface` ->0 Default font 1 Equal width 2 Bold 3 Sans serif 4 Serif

Example:

```
lminPrintInstance.setTextTypeface(0)
```

Configure text style

Function: `setTextStyle(style)`

Parameters: `style` -> 0 standard normal 1 Bold 2 Italic 3 Bold Italic

Example:

```
lminPrintInstance.setTextStyle(1);
```

Configure print text line spacing

Function: `setTextLineSpacing(space)`

Parameters: `space` line spacing, range from $0 < \text{space} < 255$, By default is 1.0f

Example: `lminPrintInstance.setTextLineSpacing(1.0f);`

Configure print paper width (Can ignore this)

Function: `setTextWidth(width)`

Parameters: `width` -> 576 80mm paper width, by default

->384 58mm paper width

Example:

```
lminPrintInstance.setTextWidth(576);
```

Print text

Function: `printText(text)`

Parameters: `text` -> The text content to be printed

Example:

```
lminPrintInstance.printText('test print centent');
```

Print text and add new line

Function: `printText(text, type)`

参数: `text` -> When the content is less than one or more than one lines, you need to add a newline break indication "n" at the end of the content to print immediately, otherwise it will be cached in the buffer

`type` -> Can ignore this

Notes: To change the print text mod (for example the alignment method, font size, font type and etc.), please perform it before `printText` command.

Example:

```
lminPrintInstance.printText('test print centent',0);
```

Print one row table

Function: `printColumnsText(colTextArr, colWidthArr, colAlign, width, size)`

Parameters: `colTextArr`→ Array of column text strings

`colWidthArr`→ Array per column width, based on English characters, each Chinese character is equivalent to two English characters, and each width is greater than 0.

`colAlign`→ alignment method: 0 align left, 1 center, 2 align right

`size`→ The font size of each column string array

`width`→ The total width of one line print (80mm paper width =576, 50mm paper width=384)

Example:

```
lminPrintInstance.printColumnsText(["1","iMin","iMin"],[1,2,1],[1,0,2],[26,26,26],576);
```

Configure 1D paper width

Function: `setBarcodeWidth(int width)`

Parameters: `width`→Barcode width level $2 \leq \text{width} \leq 6$, If the default barcode width level is not set to 3

Example:

```
lminPrintInstance.setBarcodeWidth(4);
```

Configure 1D barcode height

Function: `setBarcodeHeight(height)`

Parameters: `height`→ Barcode height $24 \leq \text{Height} \leq 250$, every 8 dots are equivalent to 1mm

Example:

```
lminPrintInstance.setBarcodeHeight(100);
```

Select HRI character print position of 1D barcode

Function: `setBarcodeContentPrintPos(position)`

Parameters: `position`→ HRI character print position

0→ Do not print

1→ Above the barcode

2→ Below the barcode

3→ Above and below the barcode

Example:

```
lminPrintInstance.setBarcodeContentPrintPos(2);
```

18 Print barcode

Function: `printBarcode(barCodeType, barCodeContent)` throws `UnsupportedEncodingException`

Parameters: `barCodeType`→ barcode type $0 \leq \text{barcode type} \leq 6$ and barcode type =8

barCodeContent-> print barcode string content

Example:

```
lminPrintInstance.printBarCode(8,"0123456789");//Code128
```

Barcode type (0-6,73,8)	Supported barcode content length	Supported ASCII source code range	
0 --> UPC-A	Barcode content length = 11,12	48 ≤range≤ 57	All models
1 --> UPC-E	Barcode content length = 11,12	48 ≤range≤ 57	All models
2 --> JAN13 / EAN13	Barcode content length= 12,13	48 ≤range≤ 57	All models
3 --> JAN8 / EAN8	Barcode content length = 7 ,8	48 ≤range≤ 57	All models
4 --> CODE39	Barcode content length >=1	48≤range≤57, 65≤range≤90, range= 32, 36, 37, 42, 43, 45, 46, 47	All models
5 --> ITF	Barcode content length >=2	48 ≤range≤ 57	All models
6 --> CODABAR	Barcode content length >=2	48≤range≤57, 65≤range≤68, 97≤range≤100, range = 36, 43, 45, 46, 47, 58	Doesn't support printing on Z2 series, others are supported
73 -->CODE128, 8-->CODE128	Barcode content length >=2	0≤range≤127	All models
7 -->CODE93	1≤n≤255	1≤n≤255	

Print barcode and configure alignment method

Function: printBarCode(barCodeType, barCodeContent, alignmentMode) throws UnsupportedOperationException

Parameters: barCodeType-> barcode type 0<= barcode type <=6 and barcode type =73

barCodeContent-> print barcode character content, if it is code128 printing, then have to add {A, {B or {C at the front,

example as below

alignmentMode->0= align left, /1= Center, /2= align right

Example:

```
lminPrintInstance.printBarCode(73 ,"{B0123456789", 1);
```

Configure QR code size

Function: setQrCodeSize(level)

Parameters: level-> QR code size, Unit: dot, 1<= level <=11

Example:

```
lminPrintInstance.setQrCodeSize(2);
```

Configure QR code error level

Function: `setQrCodeErrorCorrectionLev(level)`

Parameters: `level`-> QR code error correction level, value range from $0 \leq \text{level} \leq 3$

- 0 -> error correction level L(7%)
- 1 -> error correction level M(15%)
- 2 -> error correction level Q(25%)
- 3 -> error correction level H(30%)

Example:

```
lminPrintInstance.setQrCodeErrorCorrectionLev(51);
```

Configure left margin

Function: `setLeftMargin(marginValue)`

Parameters: `marginValue`-> left margin value, $0 < \text{marginValue} < 255$

Example:

```
lminPrintInstance.setLeftMargin(100);
```

Print QR code

Function: `printQrCode(qrStr)`

Parameters: `qrStr`-> Qr code content

Example:

```
lminPrintInstance.printQrCode("https://www.imin.sg");
```

Print QR code with alignment

Function: `printQrCode(qrStr, alignmentMode)`

Parameters: `qrStr`-> QR code content

`alignmentMode`-> 0= align left /1= center /2= align right

Example:

```
lminPrintInstance.printQrCode("https://www.imin.sg", 1);
```

Configure paper format

Function: `setPageFormat(style)`

Parameters: `style` -> 0 80mm

1 58mm

Example:

```
lminPrintInstance.setPageFormat(1);
```

Print bitmap

Function: `printSingleBitmap(imgResources)`

Parameters: `imgResources-> Bitmap` (base64 or url)

Example:

```
1.IminPrintInstance.printSingleBitmap("data:image/ico;base64,AAABAAEAICAAAAEAIACoEAAAFgAAACgAAAAGAAAQA  
AAAAEAIAAAAAAAAABAAAAAAAAAAAAAAAAAAAA...");
```

```
2.IminPrintInstance.printSingleBitmap('https://t7.baidu.com/it/u=1517419723,1472324058&fm=193&f=GIF')
```

Trigger cash drawer

Function: `openCashBox()`

Example:

```
IminPrintInstance.openCashBox();
```

Configure double QR code size

Function: `setDoubleQRSize(size)`

Parameters: `size -> 1 <= size <= 8`

Example:

```
IminPrintInstance.setDoubleQRSize(1)
```

Configure double QR code QR1 error correction level

Function: `setDoubleQR1Level(level)`

Parameters: `level -> 1 <= size <= 3`

Example:

```
IminPrintInstance.setDoubleQR1Level(1)
```

Configure double QR code QR2 error correction level

Function: `setDoubleQR2Level(level)`

Parameters: `level -> 1 <= size <= 3`

Example:

```
IminPrintInstance.setDoubleQR2Level(1)
```

Configure double QR code QR1 left margin

Function: `setDoubleQR1MarginLeft(marginValue)`

Parameters: `level -> 0 < level < 255`

Example:

```
IminPrintInstance.setDoubleQR1Level(16)
```

Configure double QR code QR2left margin

Function: setDoubleQR2MarginLeft(marginValue)

Parameters: level -> 0<level<255

Example:

```
IminPrintInstance.setDoubleQR2Level(200)
```

Configure double QR code QR1 version

Function: setDoubleQR1Version(version)

Parameters: version-> 0<=version<=40

Example:

```
IminPrintInstance.setDoubleQR1Version(40)
```

Configure double QR code QR2 version

Function: setDoubleQR2Version(version)

Parameters: version-> 0<=version<=40

Example:

```
IminPrintInstance.setDoubleQR2Version(40)
```

Print double QR code

Function: printDoubleQR(colTextArr)

Parameters: colTextArr-> Array of column text strings

Example:

```
IminPrintInstance.printDoubleQR(["www.iMin.sg", "www.google.com"]);
```