Printer SDK for Android API Reference Guide POS Printer

Ver.1.0.14

http://www.winpos.com.tw

1. Index

2. About This Manual

This manual has been prepared to provide the information required to configure and design Android applications using WinPOS's printers. WinPOS constantly makes improvements to the functions and quality and the specifications of the product and the contents of this manual are subject to change without prior notice for this reason.

3. WP Printer SDK for Android Overview

3.1 Features

This SDK has been designed to make mobile printing and POS printing with WP printer easier using Android applications. Android applications can easily check the status of the printer with this SDK.

3.1.1 Functions

- Print Settings (Alignment / Page Mode)
- Charter Data Settings (Code Page / Device Font Type)
- Character Style Settings (Bold / Reverse / Underline)
- Image Printing (Raster Bit / NV Graphics)
- One-Dimensional Barcode Printing (Refer to the specifications of each printer for supported barcode types)
- Two-Dimensional Barcode Printing (Refer to the specifications of each printer for supported barcode types)
- Cash Register Open Function / Melody Box Function
- Printer Command Transmission
- Printer Response Reception (Printing Result / Printer Status)

4. Operation Environment

4.1 Android Version

- Printing over Bluetooth or Wireless LAN: Android 2.2 (Froyo) or higher
- Printing over USB: Android 3.1 (Honeycomb) or higher

4.2 Printer Interface

- Bluetooth
- Wireless LAN
- LAN
- USB

5. Development Environment

5.1 System Requirement

5.1.1 Operating System

- Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)
- Mac OS X 10.5.8 or later (x86 only)
- Linux (tested on Ubuntu Linux, Lucid Lynx)
- GNU C Library (glibc) 2.7 or later is required.
- On Ubuntu Linux, version 8.04 or later is required.
- 64-bit distributions must be capable of running 32-bit applications.

5.1.2 Eclipse IDE

Eclipse 3.6.2 (Helios) or greater

5.1.3 JDK 6 (JRE alone is not sufficient)

- Android SDK
- ADT(Android Development Tools) plugin
- Reference: http://developer.android.com/sdk/index.html

5.1.4 Android Studio

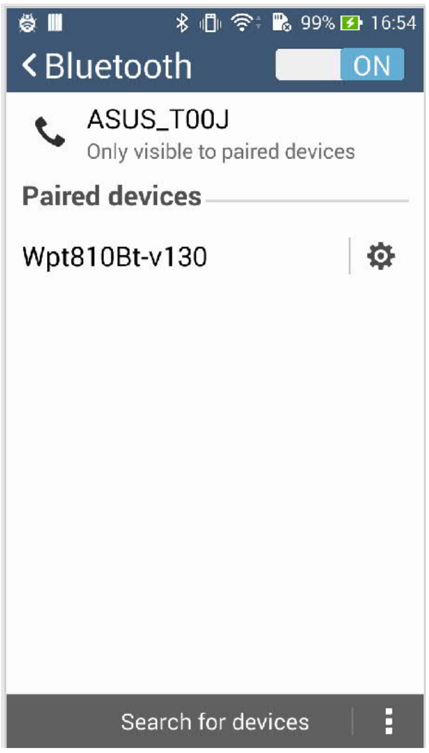
- Android Studio 2.1.1 or later
- Reference: http://tools.android.com/welcome-to-android-studio

5.2 Connecting Android Device

The following screen was captured from an Android 4.2 smart phone. The screen and field names might be different for different Android operating systems or devices.

5.2.1 Bluetooth

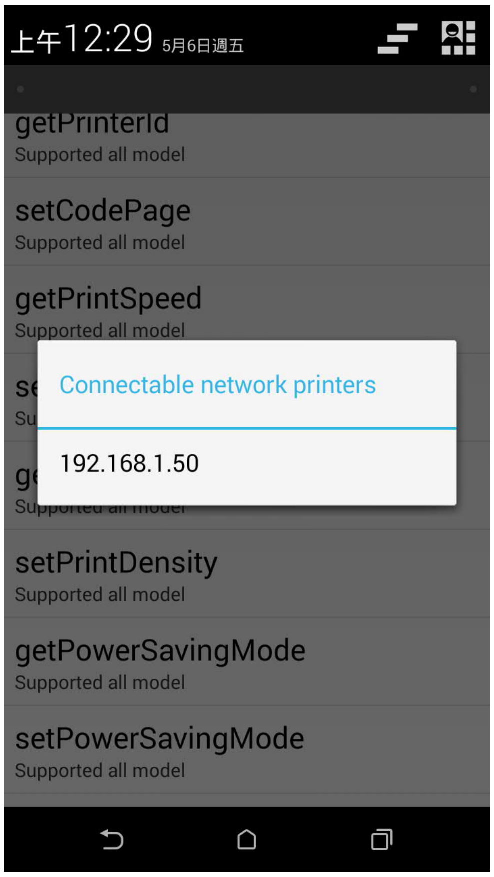
- Select [Settings].
- Bluetooth should be enabled and the printer power should be on.
- Select [Bluetooth] for settings.



- Select Scan. Search the printer to connect and perform pairing operation.

5.2.2 Network (LAN / WLAN)

- Connect the printer to the network AP (Access Point) and assign an IP address or obtain one using DHCP. Since WP printer is configured with ad-hoc network from the factory, network should be configured at least once using the Net Configuration Tool that is included in the master CD. The Net Configuration Tool can be downloaded from WP's website.
- Select [Settings].
- Wi-Fi should be turned on.
- Connect the device to the same network that the WP printer is connected to.


上午12:29 5月6日週五

getPrinterId
Supported all model

setCodePage
Supported all model

getPrintSpeed
Supported all model

Connectable network printers

192.168.1.50

setPrintDensity
Supported all model

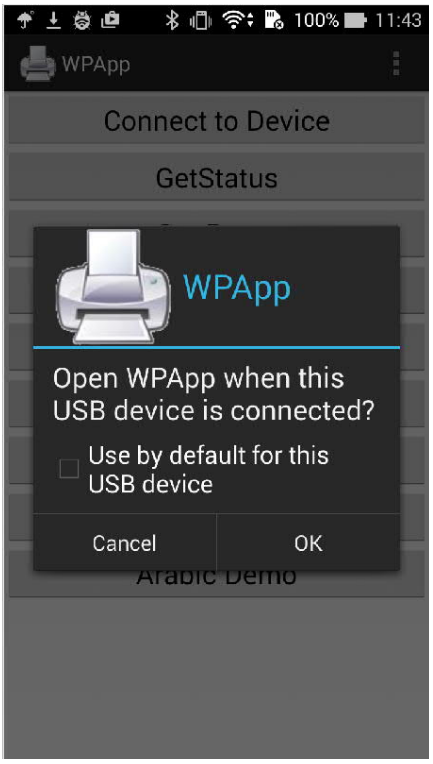
getPowerSavingMode
Supported all model

setPowerSavingMode
Supported all model

1. Additional setting is not required to connect the Android device to the TCP/IP port of printer.

5.2.3 USB

1. Android device can be connected to USB peripheral devices using OS version 3.1 or higher.
2. Special driver or printer software of WP does not have to be installed in the Android device.
3. Type of required USB cable depends on the type of smart phone or tablet device.
4. Most Android devices do not come with A to B USB cable and mini/micro USB cable or adapter/dock might be required. Check whether the cable works with the Android device to be used.
5. The following message may be displayed depending on Android device whenWP printer is connected first time.
6. The following code should be entered to AndroidManifest.xml and res/xml/device_filter.xml in order to connect USB peripheral devices.



Open WPApp when this USB device is connected?

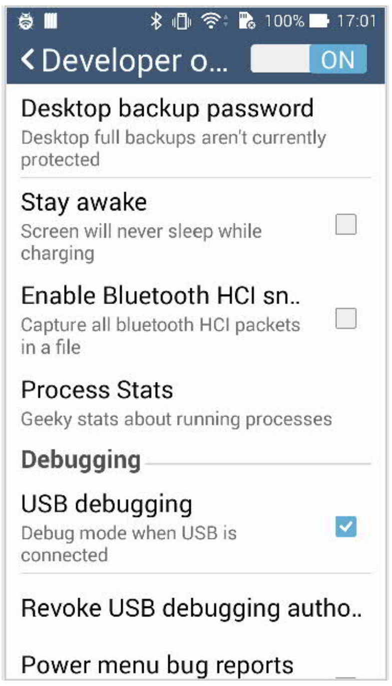
Use by default for this USB device

```
[AndroidManifest.xml]

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

```
[device_filter.xml]

<?xml version="1.0" encoding="utf-8"?>
<resources>
<usb-device class="10" protocol="0" subclass="0" vendor-id="04d8" />
</resources>
```

5.2.4 Setting Android Device Developer Options

1. Select [Settings].
2. Select [Developer options].
3. Activate [USB debugging].


< Developer o... ON

Desktop backup password
Desktop full backups aren't currently protected

Stay awake ☐
Screen will never sleep while charging

Enable Bluetooth HCI sn.. ☐
Capture all bluetooth HCI packets in a file

Process Stats
Geeky stats about running processes

Debugging

USB debugging ☑
Debug mode when USB is connected

Revoke USB debugging autho..

Power menu bug reports

5.3 Importing Library and Running Sample Application

ADT (Android Development Tool) plug in should be installed in Eclipse.

5.3.1 How to import Android project in Eclipse

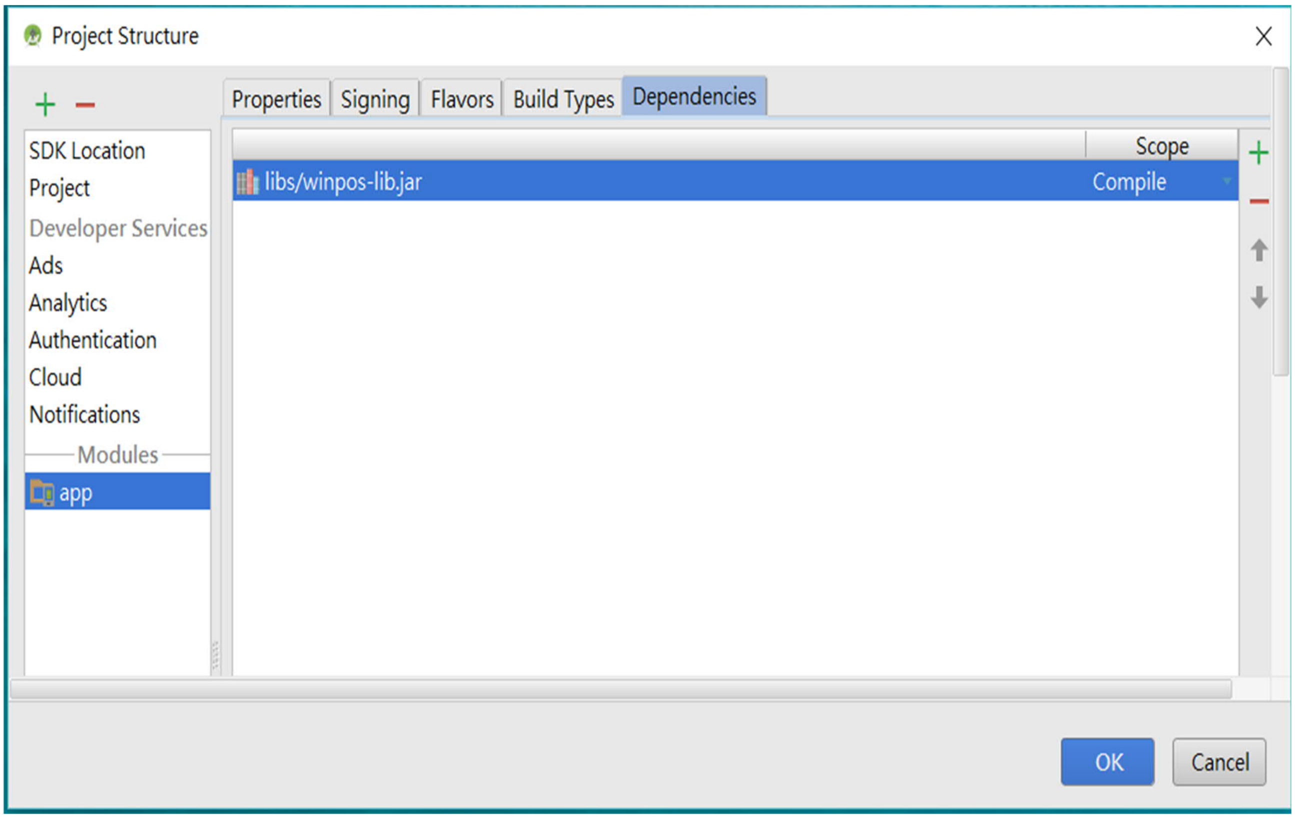
1. Run Eclipse.
2. Select [File → Import].
3. Select [General –> Existing Project into Workspace].
4. Select [Browse] and specify the path to the WPPrinterSample.

5.3.2 How to run/debug the project in Eclips

1. Select [Project – Run/Debug].

5.3.3 How to import Winpos SDK library into Android studio

1. Copy the winpos-lib.jar into your App\libs directory
2. Setting the Dependencies in Project Structure


Project Structure

Properties | Signing | Flavors | Build Types | Dependencies

SDK Location
Project
Developer Services
Ads
Analytics
Authentication
Cloud
Notifications
Modules
app

	Scope
libs/winpos-lib.jar	Compile

OK | Cancel

6. Package Contents

6.1 Package

1. doc/ POS Printer Android SDK API Reference Guide_Rev_1.0.1.pdf:
2. libs/Winpos-lib.jar: Printer library
3. sample/app: Sample program project folder

7. Sample Program

This chapter explains how to use the sample program. (WPPrinterSample) The sample is provided as an Android application project using Android Studio.

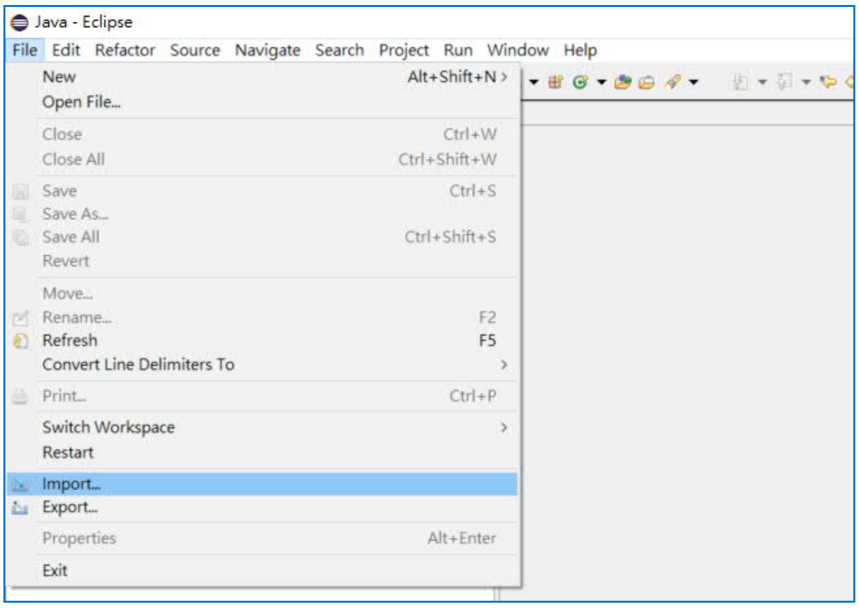
7.1 Demo Functions

- Search printer
- GetStatus
- Cut Paper
- Print Text
- 1D Barcode
- Print Receipt (列印電子發票)
- Print BitMap image
- Other Function Test
- Arabic Demo

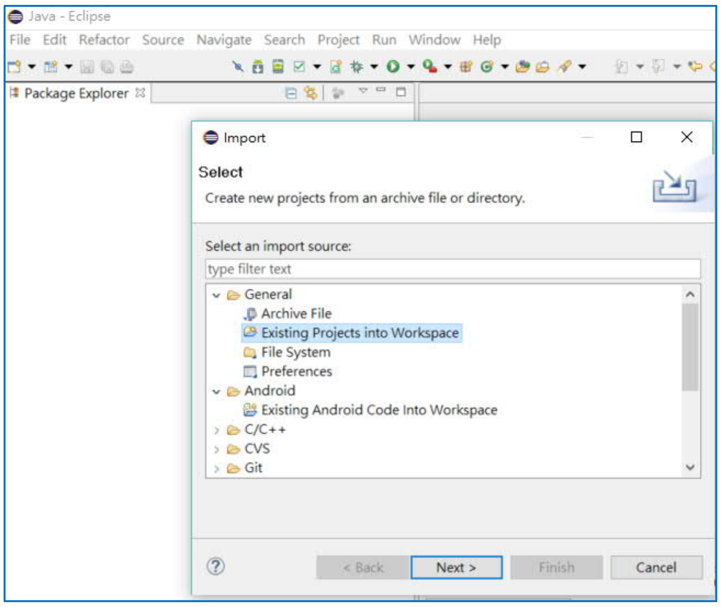
7.2 Environment Configuration

7.3 For Eclipse:

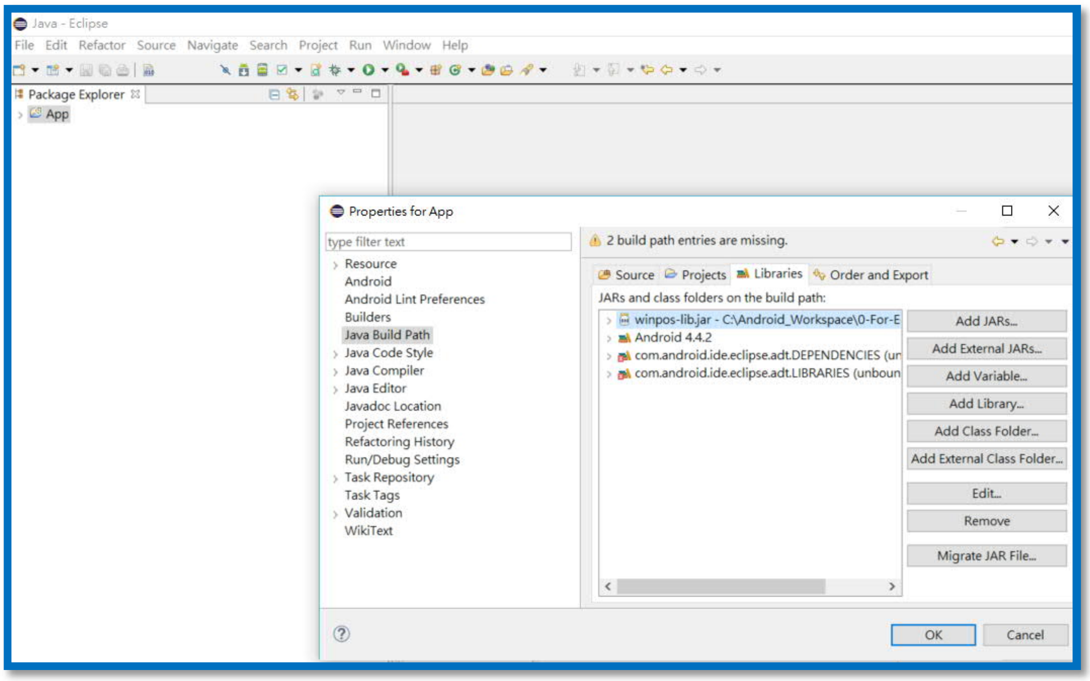
1. Decompress the WP Printer SDK for Android package into any desired target location.
2. Select [File → Import] from Eclipse



3. Select [General → Existing Project into Workspace] from the corresponding dialogue.


Java - Eclipse
File Edit Refactor Source Navigate Search Project Run Window Help
Package Explorer
Import
Select
Create new projects from an archive file or directory.
Select an import source:
type filter text
General
Archive File
Existing Projects into Workspace
File System
Preferences
Android
Existing Android Code Into Workspace
C/C++
CVS
Git
< Back
Next >
Finish
Cancel

4. Select [Browse] and set the path where App is located.
5. Right-click on WPPrinterSample in the Package Explorer and select [Properties].
6. Select [Add JARs] from the [Libraries] tab of the Java Build Path and select lib/



winpos-lib.jar

7. Right-click on App in the Package Explorer and select [Run As –> Android Application].
8. When the sample program is installed in the target Android device, select the App application and run the program.

7.4 For Android Studio:

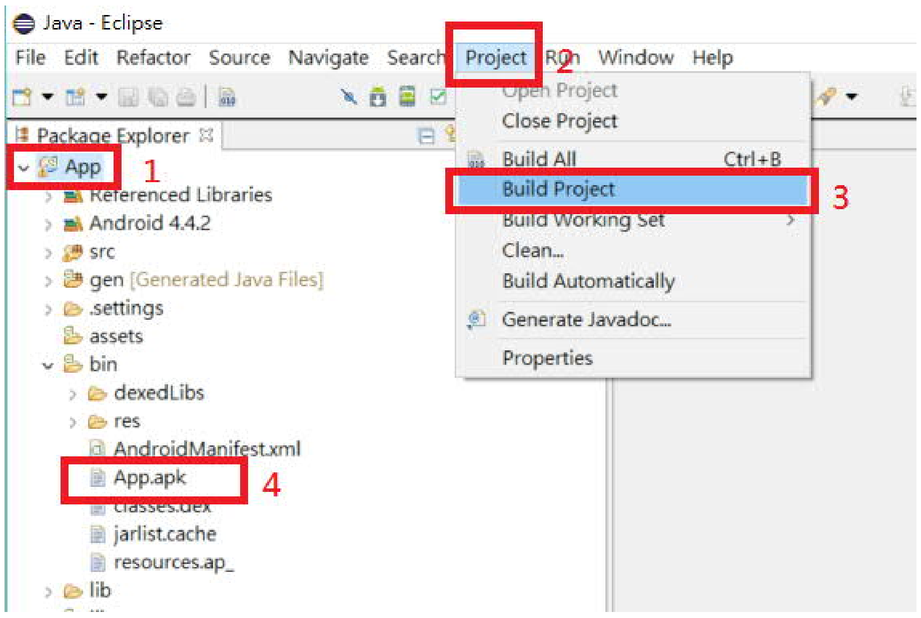
1. Unzip the winpos-lib.jar into App\libs directory
2. Select [File → Project Structure] , Select [Modules → app] , choice Tab : Dependencies → + i, to import the jar library.
3. Select [Build → Rebuild Project] to compile the project.
4. When the sample program is installed in the target Android device, select the WPPrinterSample application and run the program.

8. How to Use Sample Program

8.1 Build Apk

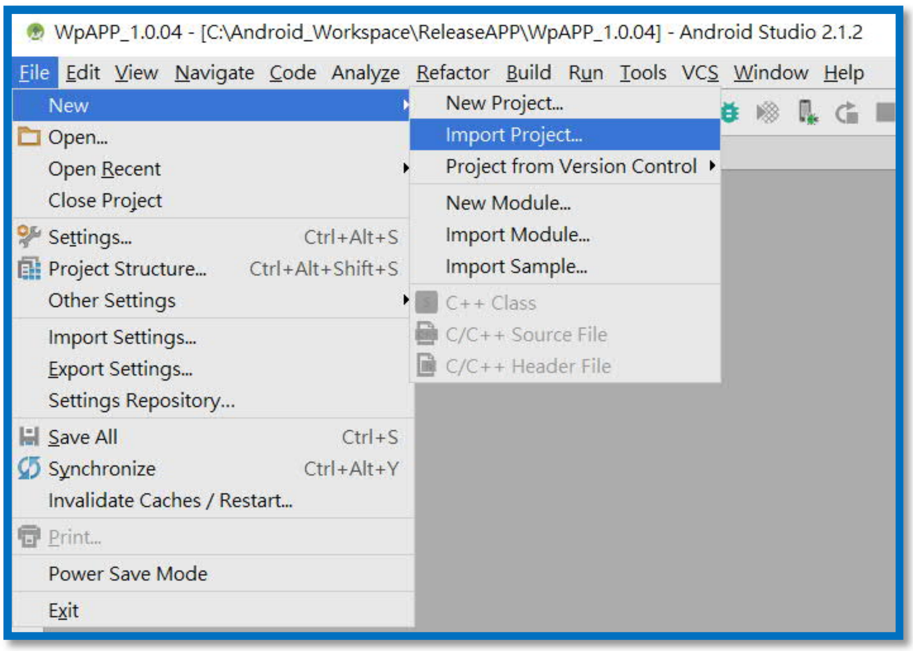
8.1.1 For Eclipse

- Select Project → Build Project
- After Eclipse finish compile the project, the apk will show on the bin directory

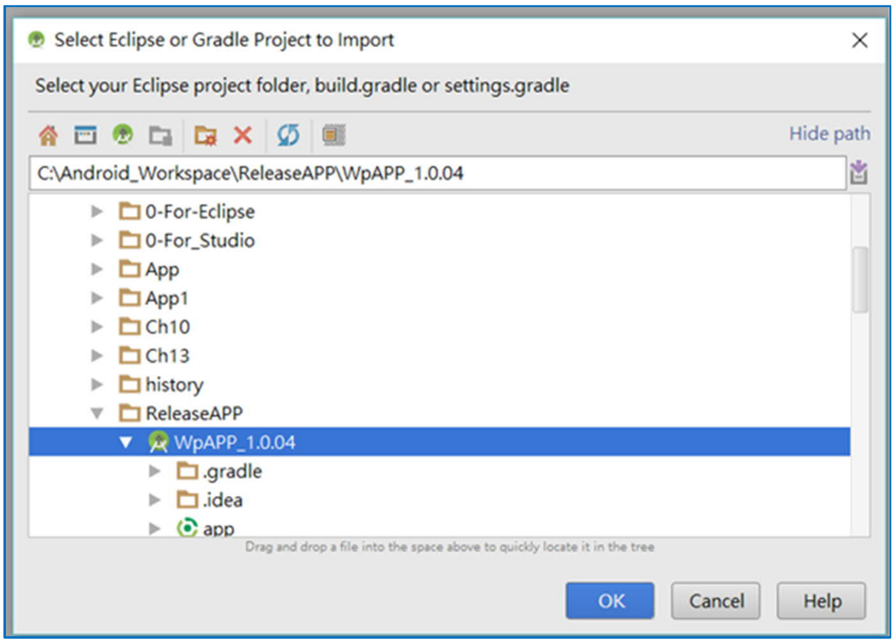


8.1.2 For Android Studio

- Import Project



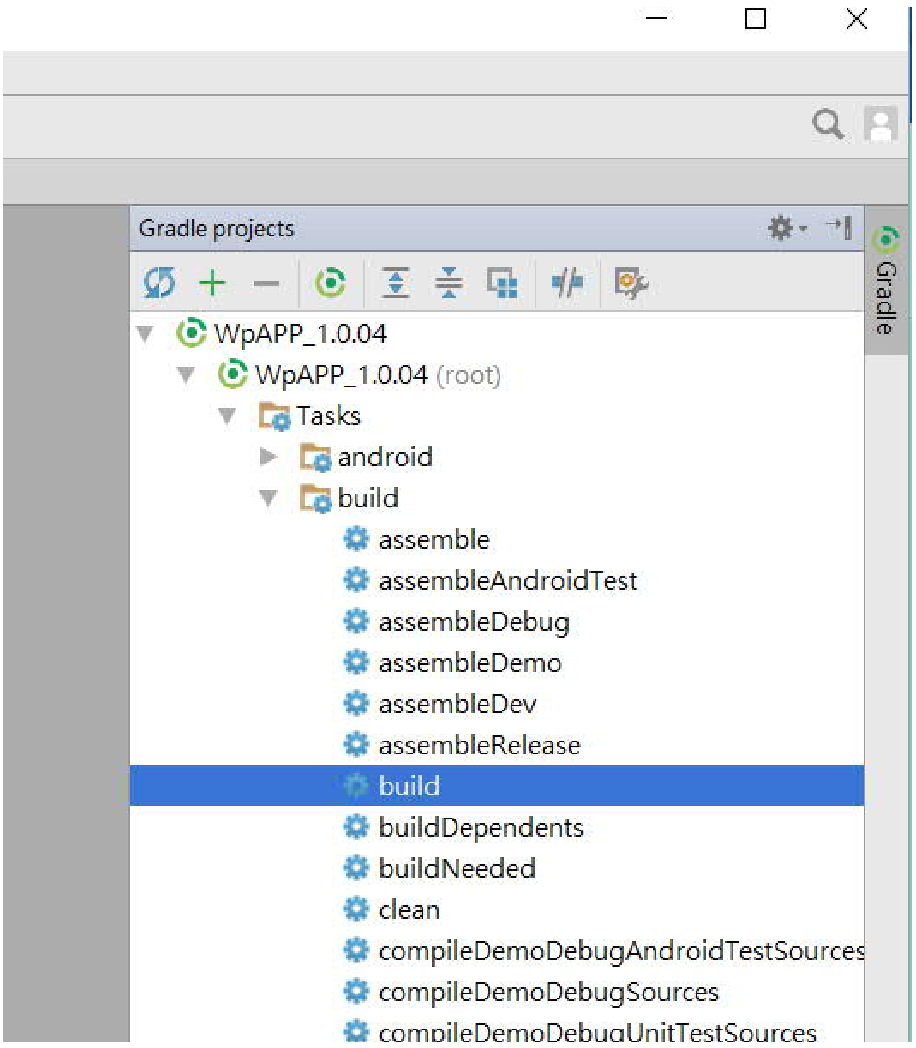
- Select project to import


Select Eclipse or Gradle Project to Import

Select your Eclipse project folder, build.gradle or settings.gradle

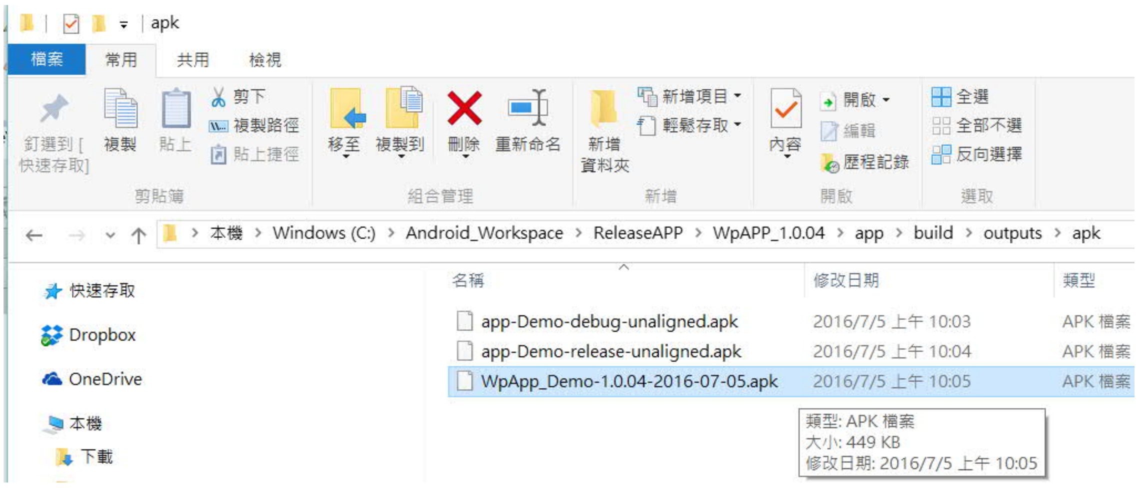
C:\Android_Workspace\ReleaseAPP\WpAPP_1.0.04

- Build the project form the Gradle → build folder → build



- Generate the output APK

  The APK file will be built under the directory → WpAPP_1.0.04\app\build\outputs\apk



本機 > Windows (C:) > Android_Workspace > ReleaseAPP > WpAPP_1.0.04 > app > build > outputs > apk

名稱	修改日期	類型
app-Demo-debug-unaligned.apk	2016/7/5 上午 10:03	APK 檔案
app-Demo-release-unaligned.apk	2016/7/5 上午 10:04	APK 檔案
WpApp_Demo-1.0.04-2016-07-05.apk	2016/7/5 上午 10:05	APK 檔案

類型: APK 檔案
大小: 449 KB
修改日期: 2016/7/5 上午 10:05

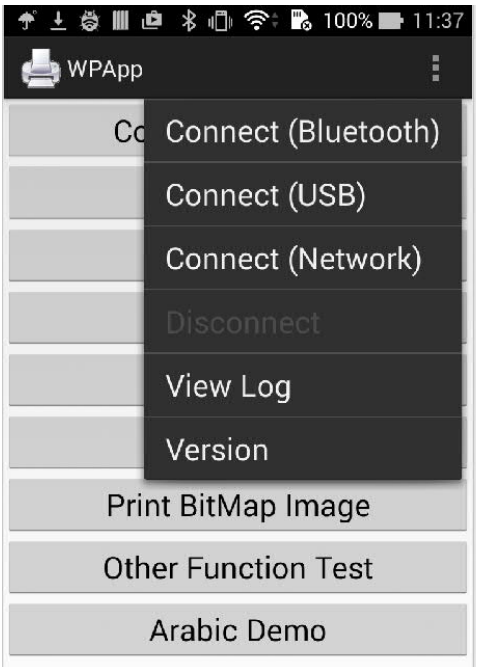
- Then you can install the Application APK file into your Android device.

8.1.3 Debug your App.

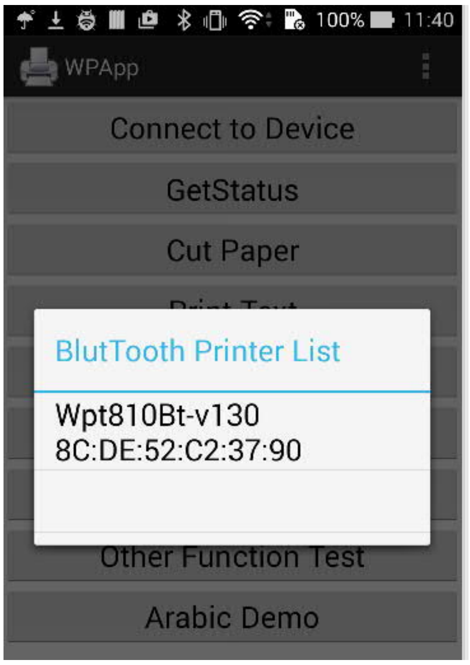
You can debug your App by following the Android studio:
https://developer.android.com/studio/debug/index.html

8.2 Search and Connect Printer

1. Run the sample program.
2. Select the [Option] menu and choose one of Bluetooth, Network, or USB interface to make connection.

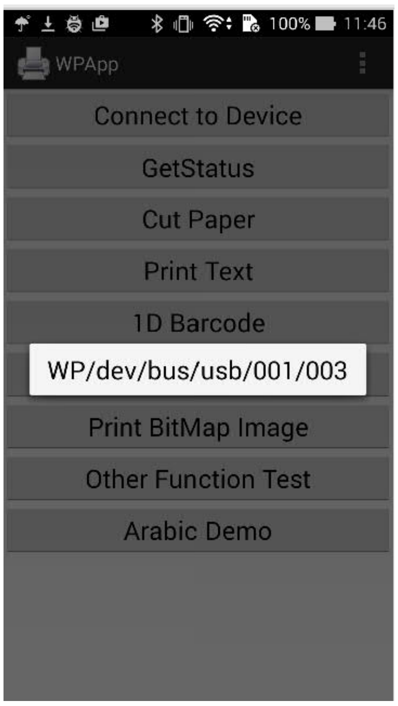


3. If Bluetooth is selected, a dialog box containing the list of paired printer MAC addresses will be displayed. Please touch the listed Blue Tooth device to select the printer.



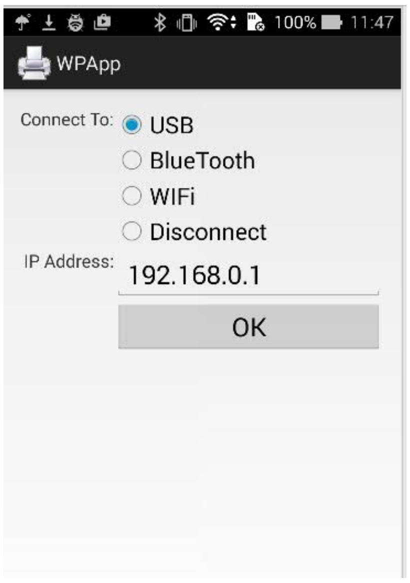
4. If Network is selected, a dialog box containing the list of printer IP addresses that can be connected will be displayed.

8.3 If USB is selected, a dialog box containing the list of



device information of printer connected with Android device through USB connection will be displayed.

5. Or you can touch the [Connect to Device] button to select the connection interface.



Connect To: [✓ USB] BlueTooth WIFi Disconnect

IP Address: 192.168.0.1

6. Select the printer to connect from the device list dialog box.
7. Connection is established if "connected to [Printer Model Name or USB Device ID]" appears in the Title area.
8. Printer function list is activated when connection is established. Refer to the Appendix for details about functions supported for each model.

9. API Reference

This chapter describes the API provided by the Printer SDK.

9.1 Create printer instance

9.1.1 Constructor

Create an instance of WpPrinter to use the printer.Methods implemented in the WpPrinter class are configured for asynchronous operation. When a response is to be received from the printer, other methods can be executed only after reception of the response is completed.When multiple printers are connected, a separate instance should be created for each printer.

- Syntax

public WPPrinter(Context context, Handler handler, Looper looper)

- Parameters

- context: is a context instance used to access Wi-Fi service, USB service, and file system of Android.
- handler: is application handler to connect and receive print message.
- looper: Set main looper in application to create WpPrinter instance in a separate thread in order to avoid the collision between the handler in application and the internal handler in printer library. Set it to null to create WpPrinter instance from the main thread.

- Example

```
public class MainActivity extends Activity {
    …
    static WpPrinter mWpPrinter;

    …
    public void onCreate(Bundle savedInstanceState) {
        …
        mWpPrinter = new WPPrinter(this, mHandler, null);
        …
    }

    private final Handler mHandler = new Handler(new Handler.Callback() {
        public Boolean handleMessage(Message msg) {
            …
        }
    };
}
```

9.2 Search Printer

9.2.1 findBluetoothPrinters

This method obtains the information of paired Bluetooth device and passes the MESSAGE_BLUETOOTH_DEVICE_SET message to the application handler.The message includes the information of the paired Bluetooth device. The return value will be null if no paired Bluetooth device is found.

- Syntax

public void findBluetoothPrinters()

- Example

.

```
public class MainActivity extends Activity {
    …
    private WpPrinter mWPPrinter;
    …
    public void onCreate(Bundle savedInstanceState) {
        …

        mWpPrinter = new WPPrinter(this, mHandler, null);
        mWpPrinter.findBluetoothPrinters();
        …
    }
    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
         case WPPrinter.MESSAGE_BLUETOOTH_DEVICE_SET: Set<BluetoothDevice>
                    bluetoothDeviceSet =  (Set<BluetoothDevice>) msg.obj;
                    break;
            }
            return true;
        }
    };
}
```

9.2.2 findNetworkPrinters

This method searches the printers connected to the same network as Android device that runs the application. It sends the MESSAGE_NETWORK_DEVICE_SET message to the application handler when the search operation is completed. The message includes the IP addresses of the printers that can be connected. The return value will be null if no printer is found.

- Syntax

public void findNetworkPrinters(int timeout)

- Parameters

- timeout: timeout in searching printer (millisecond)

- Example

```
public class MainActivity extends Activity {
    …
  private WpPrinter mWPPrinter;
    …
    public void onCreate(Bundle savedInstanceState) {
        …
        mWpPrinter = new WPPrinter(this, mHandler, null);
        mWPPrinter.findNetworkPrinters(5000);
        …
    }
    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case WPPrinter.MESSAGE_NETWORK_DEVICE_SET:
                    if (msg.obj != null) {
                        Set<String> ipAddressSet = (Set<String>) msg.obj;
                        for (String ipAddress : ipAddressSet) {
                            if (ipAddress.equals("192.168.0.100")) {
                                // TODO: Connect printer break;
                            }
                        }
                    }
                    break;
            }
            return true;
        }
    };
}
```

9.2.3 findUsbPrinters

This method obtains the information of the printer connected by USB with the Android device that runs the application, and it sends the MESSAGE_USB_DEVICE_SET message to the application handler. The message includes the information of the printer connected by USB. The return value will be null if no USB printer is found.

- Syntax

public void findUsbPrinters()

- Example

```
public class MainActivity extends Activity {
        …
        private WpPrinter mWPPrinter;
        …
        public void onCreate(Bundle savedInstanceState) {
        …
        mWpPrinter = new WPPrinter(this, mHandler, null);
        mWPPrinter.findUsbPrinters();
        …
}

private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
              switch (msg.what) {
                  case WPPrinter.MESSAGE_USB_DEVICE_SET:
                         Set<UsbDevice> usbDeviceSet = (Set<UsbDevice>) msg.obj;
                           break;
                   }
              return true;
          }
 };
```

9.2.4 findUsbPrintersBySerial

This method obtains the information of the USB printer connected with Android device that runs the application, and it sends the MESSAGE_USB_SERIAL_SET message to the application handler. This method is used to identify a specific printer using USB serial number when more than one printer of same model are connected.Only the printer models with unique USB serial numbers can be used. The message includes the USB serial numbers of the connected USB printers. This method returns null if no USB printer is found.

- Syntax

public void findUsbPrintersBySerial()

- Example

```
public class MainActivity extends Activity {
    ...
    private WpPrinter mWPPrinter;
    ...
    public void onCreate(Bundle savedInstanceState) {
        ...
        mWpPrinter = new WPPrinter(this, mHandler, null);
        mWPPrinter.findUsbPrintersBySerial();
        ...
    }
    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case WPPrinter.MESSAGE_USB_SERIAL_SET:
                    Set<String> usbSerialSet = (Set<String>) msg.obj; for (String
                      usbSerial : usbSerialSet) {
                    .........
        }
    };
}
```

9.3 Connect Printer

9.3.1 connect

This method opens the printer port of USB printer and enables communication. It sends the MESSAGE_STATE_CHANGE that includes STATE_CONNECTING in arg1 when connection process is initiated and MESSAGE_STATE_CHANGE with STATE_CONNECTED in arg1 when the connection is completed to the application handler. The messages are transmitted in the following order when this method is called.

- If connection is successful
  1. MESSAGE_STATE_CHANGE (arg1: STATE_CONNECTING): Connection is being established.
  2. MESSAGE_DEVICE_NAME: Connection to printer port is successful (USB device name of the printer recognized in Android device is transmitted.)
  3. MESSAGE_STATE_CHANGE (arg1: STATE_CONNECTED): Communication with the printer is enabled.
- If connection fails
  1. MESSAGE_TOAST: “Unable to connect device” message is sent.
  2. MESSAGE_STATE_CHANGE (arg1: STATE_NONE): Printer is not connected.

■ Syntax

```
public void connect()
```

Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect();
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          switch (msg.arg1) {
            case WPPrinter.STATE_CONNECTING:
              // TODO
            case WPPrinter.STATE_CONNECTED:
              // TODO
            case WPPrinter.STATE_NONE:
              // TODO: Processing
          }
          break;
        case WPPrinter.MESSAGE_DEVICE_NAME:
          String connectedDeviceName = msg.getData().getString(
              WPPrinter.KEY_STRING_DEVICE_NAME);
          break;
        case WPPrinter.MESSAGE_TOAST: Toast.makeText(getApplicationContext(),
            msg.getData().getString(WPPrinter.KEY_STRING_TOAST),
Toast.LENGTH_SHORT).show();
          break;
      }
      return true;
    }
  };
}
```

9.3.2 connect

This method opens the port of paired Bluetooth printer and enables communication.It sends the MESSAGE_STATE_CHANGE message with STATE_CONNECTING in arg1 when connection process is initiated and MESSAGE_STATE_CHANGE with STATE_CONNECTED in arg1 when connection is completed to the application handler. Messages are transmitted in the following order when this method is called.

- If connection is successful

1. MESSAGE_STATE_CHANGE (arg1: STATE_CONNECTING): Connection is being established.
2. MESSAGE_DEVICE_NAME: Connection to printer port is successful. (Bluetooth device name of the printer recognized by Android device is transmitted.)
3. MESSAGE_STATE_CHANGE (arg1: STATE_CONNECTED): Communication with printer is enabled.

- If connection fails

1. MESSAGE_TOAST: “Unable to connect device” message is sent.
2. MESSAGE_STATE_CHANGE (arg1: STATE_NONE): Printer is not connected.

■ Syntax

```
public void connect(String address)
```

■ Parameters

address: Bluetooth MAC address of the printer to connect (it can also be checked with Self-Test function.) If this parameter is set to null, then connection will be tried with the first searched printer among all paired printers.

Example

```
public class MainActivity extends Activity {
  private WPPrinter mWPPrinter;
   ...
   public void onCreate(Bundle savedInstanceState) {
      ...
      String address = "01:23:45:67:89:ab";
      mWPPrinter = new WPPrinter(this, mHandler, null);
      mWPPrinter.connect(address);
   }
   private final Handler mHandler = new Handler(new Handler.Callback() {
      public boolean handleMessage(Message msg) {
         switch (msg.what) {
            case WPPrinter.MESSAGE_STATE_CHANGE:
               switch (msg.arg1) {
                  case WPPrinter.STATE_CONNECTING:
                     // TODO: Processing when connection
               }
         }
         return true;
      }
   };
}
```

9.3.3 connect

This method opens the port of printer connected with LAN or Wireless LAN and enables communication. It sends the MESSAGE_STATE_CHANGE message with STATE_CONNECTING in arg1 when connection process is initiated and MESSAGE_STATE_CHANGE with STATE_CONNECTED in arg1 when connection is completed to the application handler.Messages are transmitted in the following order when this method is called.

- If connection is successful
  1. MESSAGE_STATE_CHANGE (arg1: STATE_CONNECTING): Connection is being established
  2. MESSAGE_DEVICE_NAME: Connection to printer port is successful. (Device name of the printer recognized by Android device is transmitted.)
  3. MESSAGE_STATE_CHANGE (arg1: STATE_CONNECTED): Communication with printer is enabled.
- If connection fails
  1. MESSAGE_TOAST: “Unable to connect device” message is sent.
  2. MESSAGE_STATE_CHANGE (arg1: STATE_NONE): Printer is not connected.

■ Syntax

```
public void connect(String host, int port, int timeout)
```

■ Parameters

- host: IP address of the printer to connect
- port: Port number of the printer to connect (only 9100 is allowed.)
- timeout: Timeout in connecting printer (millisecond)

Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    String address = “01:23:45:67:89:ab”;
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(address);
    ...
  }

  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          switch (msg.arg1) {
            case WPPrinter.STATE_CONNECTING:
            case WPPrinter.STATE_CONNECTED:
            case WPPrinter.STATE_NONE:
          }
          break;
        case WPPrinter.MESSAGE_DEVICE_NAME:
          String connectedDeviceName =  msg.getData().getString(
              WPPrinter.KEY_STRING_DEVICE_NAME);
          break;
        case WPPrinter.MESSAGE_TOAST:
          Toast.makeText(getApplicationContext(),
              msg.getData().getString(WPPrinter.KEY_STRING_TOAST),
                          Toast.LENGTH_SHORT).show();
          break;
      }
      return true;
    }
  };
}
```

9.3.4 connect

This method opens the port of printer connected over USB and enables communication.It sends the MESSAGE_STATE_CHANGE message with STATE_CONNECTING in arg1 when connection process is initiated and MESSAGE_STATE_CHANGE with STATE_CONNECTED in arg1 when connection is completed to the application handler. Messages are transmitted in the following order when this method is called.

- If connection is successful
  1. MESSAGE_STATE_CHANGE (arg1: STATE_CONNECTING): Connection is being established
  2. MESSAGE_DEVICE_NAME: Connection to printer port is successful. (Device name of the printer recognized by Android device is transmitted.)
  3. MESSAGE_STATE_CHANGE (arg1: STATE_CONNECTED): Communication with printer is enabled.
- If connection fails
  1. MESSAGE_TOAST: “Unable to connect device” message is sent.
  2. MESSAGE_STATE_CHANGE (arg1: STATE_NONE): Printer is not connected.

■ Syntax

```
public void connect(UsbDevice device)
```

■ Parameters

- device: UsbDevice instance of the device to connect. It can be received through the message received as a response to findUsbPrinters().

Example

```
public class MainActivity extends Activity {
   private WpPrinter mWPPrinter;
   ...
   public void onCreate(Bundle savedInstanceState) {
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.findUsbPrinters();
      ...
   }
   private final Handler mHandler = new Handler(new Handler.Callback() {
      public boolean handleMessage(Message msg) {
         switch (msg.what) {
            case WPPrinter.MESSAGE_USB_DEVICE_SET:
               Set<UsbDevice> usbDeviceSet = (Set<UsbDevice>) msg.obj;
               return true;
            case WPPrinter.MESSAGE_STATE_CHANGE:
               switch (msg.arg1) {
                  case WPPrinter.STATE_CONNECTING:
                  case WPPrinter.STATE_CONNECTED:
                  case WPPrinter.STATE_NONE:
               }
               break;
            case WPPrinter.MESSAGE_DEVICE_NAME:
               String connectedDeviceName = msg.getData().
                                    getString(WPPrinter.KEY_STRING_DEVICE_NAME);
               break;
```

.

9.3.5 connectUsb

This method opens the port of printer connected over USB and enables communication.It sends the MESSAGE_STATE_CHANGE message with STATE_CONNECTING in arg1 when connection process is initiated and MESSAGE_STATE_CHANGE with STATE_CONNECTED in arg1 when connection is completed to the application handler. Messages are transmitted in the following order when this method is called.

- If connection is successful
  1. MESSAGE_STATE_CHANGE (arg1: STATE_CONNECTING): Connection is being established
  2. MESSAGE_DEVICE_NAME: Connection to printer port is successful. (Device name of the printer recognized by Android device is transmitted.)
  3. MESSAGE_STATE_CHANGE (arg1: STATE_CONNECTED): Communication with printer is enabled.
- If connection fails
  1. MESSAGE_TOAST: “Unable to connect device” message is sent.
  2. MESSAGE_STATE_CHANGE (arg1: STATE_NONE): Printer is not connected.

■ Syntax

public void connectUsb(String serial)

■ Parameters

- device: USB serial number of the printer to connect. It can be received through the message received as a response to findUsbPrintersBySerial().

Example

```
public class MainActivity extends Activity {
    ...
    private WpPrinter mWPPrinter;
    ...
    public void onCreate(Bundle savedInstanceState) {
        ...
        mWpPrinter = new WPPrinter(this, mHandler, null);
        mWPPrinter.findUsbPrintersBySerial();
        ...
    }

    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case WPPrinter.MESSAGE_USB_SERIAL_SET:
                    Set<String> usbSerialSet = (Set<String>) msg.obj;
                    for (String serial : usbSerialSet) {
                        if (serial.equals("xxxxxxxxxxxx")) { mWPPrinter.connectUsb(serial); break;
                        }
                    }
                    return true;
                case WPPrinter.MESSAGE_STATE_CHANGE:
                    switch (msg.arg1) {
                        case WPPrinter.STATE_CONNECTING:
                            //TODO
                        case WPPrinter.STATE_CONNECTED:
                            //TODO
                        case WPPrinter.STATE_NONE:
                            //TODO
                    }
                    break;
                case WPPrinter.MESSAGE_DEVICE_NAME:
                    String connectedDeviceName =
                            msg.getData().getString(WPPrinter.KEY_STRING_DEVICE_NAME);
                    break;
                case WPPrinter.MESSAGE_TOAST: Toast.makeText(getApplicationContext(),
                        msg.getData().getString(WPPrinter.KEY_STRING_TOAST),
Toast.LENGTH_SHORT).show();
                    break;
            }
            return true;
        }
    };
```

9.3.6 disconnect

The method closes the port of connected printer and terminates the connection.When connection is terminated, it sends the MESSAGE_STATE_CHANGE messge (arg1: STATE_NONE) to application handler.

- Syntax

public void disconnect()

- Example

```
public class MainActivity extends Activity {
    ...
    private WpPrinter mWPPrinter;
    ...
    public void onCreate(Bundle savedInstanceState) {
        ...
        mWpPrinter = new WPPrinter(this, mHandler, null);
        mWPPrinter.connect(null);
        ...
    }
    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case WPPrinter.MESSAGE_STATE_CHANGE:
                    switch (msg.arg1) {
                        case WPPrinter.STATE_CONNECTING:
                            // TODO:
                        case WPPrinter.STATE_CONNECTED:
                            mWPPrinter.disconnect();
                            break;
                        case WPPrinter.STATE_NONE:
                            Toast.makeText(getApplicationContext(),
                            "Printer is disconnected", Toast.LENGTH_SHORT).show();
                            break;
                    }
                    break;
            }
            return true;
        }
    };
}
```

9.4 Print

9.4.1 lineFeed

This method feeds the paper by the specified number of lines.

- Syntax

public void lineFeed(int lines, boolean getResponse)

- Parameters

- lines: number of lines to feed
- getResponse: A message is sent to the application handler upon completion of feeding if this parameter is set to True, and message is not sent if it is set to False.

- Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case MESSAGE_STATE_CHANGE:
          if (msg.arg1 == STATE_CONNECTED) {
            mWPPrinter.lineFeed(5, true);
          }
          break;
        case MESSAGE_PRINT_COMPLETE: mWPPrinter.disconnect(); break;
      }
      return true;
    }
  };
}
```

9.4.2 print1dBarcode

This method prints one dimensional barcode.

- Syntax

public void print1dBarcode(String data, int barCodeSystem, int alignment, int width, int height, int characterPosition, boolean getResponse)

- Parameters

- data: barcode data to print
- barCodeSystem: barcode system

Code	Value	Description
BAR_CODE_UPC_A	65	UPC-A
BAR_CODE_UPC_E	66	UPC-E
BAR_CODE_EAN13	67	EAN13
BAR_CODE_EAN8	68	EAN8
BAR_CODE_CODE39	69	CODE93
BAR_CODE_ITF	70	ITF
BAR_CODE_CODABAR	71	CODABAR
BAR_CODE_CODE93	72	CODE93
BAR_CODE_CODE128	73	CODE128

- alignment: barcode alignment
- width: width of barcode (1 ~ 6)

Code	Value	Description
ALIGNMENT_LEFT	0	Align to left
ALIGNMENT_CENTER	1	Align to center
ALIGNMENT_RIGHT	2	Align to right

- height: height of barcode (1 ~ 255)
- characterPosition: position to print barcode data string
- getResponse: A message is sent to the application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

Code	Value	Description
HRI_CHARACTER_NOT_PRINTED	0	Character string is not printed
HRI_CHARACTER_ABOVE_BAR_CODE	1	Character string is printed above barcode
HRI_CHARACTER_BELOW_BAR_CODE	2	Character string is printed below barcode
HRI_CHARACTER_ABOVE_AND_BELOW_BAR_CODE	3	Character string is printer above and below barcode

Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
               mWPPrinter.print1dBarcode("012345678905",
                WPPrinter.BAR_CODE_UPC_A, WPPrinter.ALIGNMENT_LEFT,3, 162,
                WPPrinter.HRI_CHARACTER_ABOVE_BAR_CODE, true);
          }
          break;
        case WPPrinter.MESSAGE_PRINT_COMPLETE:
          mWPPrinter.disconnect();
          break;
      }
      return true;
    }
  };
}
```

9.4.3 printBitmap

This method converts Bitmap instance to black and white image and prints the image.

- Syntax

public void printBitmap(Bitmap bitmap, int alignment, int width, int level, boolean getResponse)

- Parameter

- bitmap: Bitmap instance to print
- alignment: Image alignment

Code	Value	Description
ALIGNMENT_LEFT	0	Align to left
ALIGNMENT_CENTER	1	Align to center
ALIGNMENT_RIGHT	2	Align to right

- width: width of image to print

Code	Value	Description
BITMAP_WIDTH_FULL	-1	Image is enlarged or reduced to the maximum printing width and printed.
BITMAP_WIDTH_NONE	0	Image is printed without resizing or reduced to the maximum printing width if the image is wider than the maximum width.
Integer		Enter integer number directly

- level: brightness level of the image to print (13 ~ 88)
- getResponse: A message is sent to the application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

Example

```
public class MainActivity extends Activity {

  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            BitmapDrawable drawable = (BitmapDrawable)
             getResources().getDrawable(R.drawable.logo);
            Bitmap bitmap = drawable.getBitmap();
            mWPPrinter.printBitmap(bitmap,WPPrinter.ALIGNMENT_LEFT,
                WPPrinter.BITMAP_WIDTH_FULL, 50, true);
          }
          break;
        case WPPrinter.MESSAGE_PRINT_COMPLETE:
          mWPPrinter.disconnect();
          break;
      }
      return true;
    }
  };
}
```

9.4.4 printBitmap

This method converts Bitmap instance to black and white image and prints the image

- Syntax

public void printBitmap(Bitmap bitmap, int alignment, int width, int level, boolean dither, Boolean compress, boolean getResponse)

- Parameter

- bitmap: Bitmap instance to print
- alignment: Image alignment

Code	Value	Description
ALIGNMENT_LEFT	0	Align to left
ALIGNMENT_CENTER	1	Align to center
ALIGNMENT_RIGHT	2	Align to right

- width: width of image to print

Code	Value	Description
BITMAP_WIDTH_FULL	-1	Image is enlarged or reduced to the maximum printing width and printed.
BITMAP_WIDTH_NONE	0	Image is printed without resizing or reduced to the maximum printing width if the image is wider than the maximum width.
Integer		Enter integer number directly

- level: brightness level of the image to print (13 ~ 88)
- dither: true to try bit dithering
- compress: true to compress bitmap to transfer
- getResponse: A message is sent to the application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

Example

```
public class MainActivity extends Activity {

    ...
    private WpPrinter mWPPrinter;
    ...
    public void onCreate(Bundle savedInstanceState) {
        ...
        mWpPrinter = new WPPrinter(this, mHandler, null);
        mWPPrinter.connect(null);
        ...
    }
    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case WPPrinter.MESSAGE_STATE_CHANGE:
                    if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
                BitmapDrawable drawable = (BitmapDrawable)
                  getResources().getDrawable(R.drawable.logo);
                        Bitmap bitmap = drawable.getBitmap();
                mWPPrinter.printBitmap(bitmap,WPPrinter.ALIGNMENT_LEFT,
                                        WPPrinter.BITMAP_WIDTH_FULL,
                                           50, true, true, true);
                    }
                    break;
                case WPPrinter.MESSAGE_PRINT_COMPLETE:
                    mWPPrinter.disconnect();
                    break;
            }
            return true;
        }
    };
}
```

9.4.5 printBitmap

This method prints black and white image data.

- Syntax

public void printBitmap(byte[] pixels, int alignment, int width, int height,boolean getResponse)

- Parameter

  - pixels: image data to print
  - alignment: Image alignment
  - width: width of image to print
  - height: height of image to print
  - getResponse: A message is sent to the application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

- Example

```
public class MainActivity extends Activity {
  ...
  private WPPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWPPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
        BitmapDrawable drawable = (BitmapDrawable) getResources().
                                         getDrawable(R.drawable.logo);
            Bitmap bitmap = drawable.getBitmap();
            mWPPrinter.getMonoPixels(bitmap, WPPrinter.BITMAP_WIDTH_FULL, 50);
          }
          break;
        }
      return true;
    }
  };
}
```

9.4.6 printBitmap

This method converts the image file located in the specified path to black and white image and prints the image.

- Syntax

public void printBitmap(String pathName, int alignment, int width, int level, boolean getResponse)

- Parameter

  - pathName: absolute path of the image file to print
  - alignment: Image alignment
  - width: width of image to print. Height of the image is automatically adjusted in proportion to the width.
  - level: brightness level of the image to print (13 ~ 88)
  - getResponse: A message is sent to the application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            String pathName =  Environment.getExternalStorageDirectory().getAbsolutePath() +
                          “/logo.png”;
            mWPPrinter.printBitmap(pathName, WPPrinter.ALIGNMENT_LEFT,WPPrinter.
                                   BITMAP_WIDTH_FULL, 50, true);
          }
          break;
        case WPPrinter.MESSAGE_PRINT_COMPLETE:
          mWPPrinter.disconnect();
          break;
      }
      return true;
    }
  };
}
```

9.4.7 printBitmap

This method converts the image file located in the specified path to black and white image and prints the image.

- Syntax

public void printBitmap(String pathName, int alignment, int width, int level, boolean dither, Boolean compress, boolean getResponse)

- Parameter

- pathName: absolute path of the image file to print
- alignment: Image alignment

Code	Value	Description
ALIGNMENT_LEFT	0	Align to left
ALIGNMENT_CENTER	1	Align to center
ALIGNMENT_RIGHT	2	Align to right

- width: width of image to print
- level: brightness level of the image to print (13 ~ 88)

Code	Value	Description
BITMAP_WIDTH_FULL	-1	Image is enlarged or reduced to the maximum printing width and printed.
BITMAP_WIDTH_NONE	0	Image is printed without resizing or reduced to the maximum printing width if the image is wider than the maximum width.
Integer		Enter integer number directly

- dither: true to try bit dithering
- compress: true to compress bitmap to transfer
- getResponse: A message is sent to the application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            String pathName = Environment.getExternalStorageDirectory().getAbsolutePath()
+"/logo.png";
            mWPPrinter.printBitmap(pathName, WPPrinter.ALIGNMENT_LEFT,
                WPPrinter.BITMAP_WIDTH_FULL, 50, true, true, true);
          }
          break;
        case WPPrinter.MESSAGE_PRINT_COMPLETE:
          mWPPrinter.disconnect();
          break;
      }
      return true;
    }
  };
}
```

9.4.8 printDotMatrixBitmap

This method converts Bitmap instance to black and white image data and prints the image on dot matrix type printer.

■ Syntax

```
public void printDotMatrixBitmap(Bitmap bitmap, int alignment, int width, int level, boolean getResponse)
```

■ Parameter

- bitmap: Bitmap instance to print
- alignment: Image alignment

Code	Value	Description
ALIGNMENT_LEFT	0	Align to left
ALIGNMENT_CENTER	1	Align to center
ALIGNMENT_RIGHT	2	Align to right

- width: width of image to print. Height of the image is automatically adjusted in proportion to the width.

Code	Value	Description
BITMAP_WIDTH_FULL	-1	Image is enlarged or reduced to the maximum printing width and printed.
BITMAP_WIDTH_NONE	0	Image is printed without resizing or reduced to the maximum printing width if the image is wider than the maximum width.
Integer		Enter integer number directly

- level: brightness level of the image to print (13 ~ 88)
- getResponse: A message is sent to the application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            BitmapDrawable drawable =(BitmapDrawable)
getResources().getDrawable(R.drawable.logo);
            Bitmap bitmap = drawable.getBitmap();
            mWPPrinter.printDotMatrixBitmap(bitmap, WPPrinter.ALIGNMENT_LEFT,
                WPPrinter.BITMAP_WIDTH_FULL, 50, true);
          }
          break;
        case WPPrinter.MESSAGE_PRINT_COMPLETE:
          mWPPrinter.disconnect();
          break;
      }
      return true;
    }
  };
}
```

9.4.9 printDotMatrixBitmap

This method converts image file in the specified path to black and white image data and prints the image on dot matrix type printer.

■ Syntax

public void printDotMatrixBitmap(String pathName, int alignment, int width, int level, boolean getResponse)

■ Parameters

- pathName: absolute path of the image file to print
- alignment: Image alignment. Refer to the following table for possible options.

Code	Value	Description
ALIGNMENT_LEFT	0	Align to left
ALIGNMENT_CENTER	1	Align to center
ALIGNMENT_RIGHT	2	Align to right

- width: width of image to print. Height of the image is automatically adjusted in proportion to the width. Refer to the following table for possible options.

Code	Value	Description
BITMAP_WIDTH_FULL	-1	Image is enlarged or reduced to the maximum printing width and printed.
BITMAP_WIDTH_NONE	0	Image is printed without resizing or reduced to the maximum printing width if the image is wider than the maximum width.
Integer		Enter integer number directly

- level: brightness level of the image to print (13 ~ 88)
- getResponse: A message is sent to the application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
             String pathName =  Environment.getExternalStorageDirectory().getAbsolutePath()
              +"/logo.png";
            mWPPrinter.printDotMatrixBitmap(pathName, WPPrinter.ALIGNMENT_LEFT,
                WPPrinter.BITMAP_WIDTH_FULL, 50, true);
          }
          break;
        case WPPrinter.MESSAGE_PRINT_COMPLETE:
          mWPPrinter.disconnect();
          break;
      }
      return true;
    }
  };
}
```

9.4.10 printDotMatrixText

This method prints character string on Dot Matrix type printer.

Syntax

public void printDotMatrixText(String text, int alignment, int attribute, int size, boolean getResponse)

Parameter

- text: character string to print
- alignment: character string alignment
- attribute: attributes of character string printing. When more than one parameter is configured, each option should be combined through bitwise OR operation. Refer to the following table for possible options.

Code	Value	Description
TEXT_ATTRIBUTE_FONT_A	0	Font type A
TEXT_ATTRIBUTE_FONT_B	1	Font type B
TEXT_ATTRIBUTE_UNDERLINE1	4	Underline with 1 dot thickness
TEXT_ATTRIBUTE_UNDERLINE2	8	Underline with two dots thickness
TEXT_ATTRIBUTE_EMPHASIZED	16	Bold font

- size: size of character string to print. The width and hight parameters should be
- combined through bitwise OR operation. Refer to the following table for possible options.

Code	Value	Description
TEXT_SIZE_HORIZONTAL1	0	1X magnification on width
TEXT_SIZE_HORIZONTAL2	16	2X magnification on width
TEXT_SIZE_VERTICAL1	0	1X magnification in height
TEXT_SIZE_VERTICAL2	1	1X magnification in height

- getResponse: A message is sent to the application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            mWPPrinter.printDotMatrixText("printText\n",
                    WPPrinter.ALIGNMENT_LEFT, WPPrinter.TEXT_ATTRIBUTE_FONT_A |
                      WPPrinter.TEXT_ATTRIBUTE_UNDERLINE1,
                      WPPrinter.TEXT_SIZE_HORIZONTAL1 |
                     WPPrinter.TEXT_SIZE_VERTICAL1, true);
          }
          break;
        case WPPrinter.MESSAGE_PRINT_COMPLETE:
          mWPPrinter.disconnect();
          break;
      }
      return true;
    }
  };
}
```

9.4.11 printQrCode

This method prints QR Code.

■ Syntax

public void printQrCode(String data, int alignment, int model, int size, boolean getResponse)

■ Parameter

- data: barcode data to print
- alignment: barcode printing alignment
- model: QR Code model to print

Code	Value	Description
ALIGNMENT_LEFT	0	Align to left
ALIGNMENT_CENTER	1	Align to center
ALIGNMENT_RIGHT	2	Align to right

- size: size of barcode to print (1 ~ 8)

Code	Value	Description
QR_CODE_MODEL1	48	QR Code model 1
QR_CODE_MODEL1	49	QR Code model 2

- getResponse: A message is sent to the application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

Example

```
public class MainActivity extends Activity {
   ...
   private WpPrinter mWPPrinter;
   ...
   public void onCreate(Bundle savedInstanceState) {
      ...
      mWpPrinter = new WPPrinter(this, mHandler, null);
      mWPPrinter.connect(null);
      ...
   }
   private final Handler mHandler = new Handler(new Handler.Callback() {
      public boolean handleMessage(Message msg) {
         switch (msg.what) {
            case WPPrinter.MESSAGE_STATE_CHANGE:
               if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            mWPPrinter.printQrCode(“www.WP.com”, WPPrinter.ALIGNMENT_LEFT,
                                                 WPPrinter.QR_CODE_MODEL2, 8, true);
               }
               break;
            case WPPrinter.MESSAGE_PRINT_COMPLETE:
               mWPPrinter.disconnect();
               break;
         }
         return true;
      }
   };
}
```

9.4.12 printQrCode

This method prints QR Code.

■ Syntax

public void printQrCode(String data, int alignment, int model, int size, int errorCorrectionLevel, boolean getResponse)

■ Parameter

- data: barcode data to print
- alignment: barcode printing alignment

Code	Value	Description
ALIGNMENT_LEFT	0	Align to left
ALIGNMENT_CENTER	1	Align to center
ALIGNMENT_RIGHT	2	Align to right

- model: QR Code model to print

Code	Value	Description
QR_CODE_MODEL1	48	QR Code model 1
QR_CODE_MODEL1	49	QR Code model 2

- size: size of barcode to print (1 ~ 8)
- errorCorrectionLevel: error correction level of barcode to print

Code	Value	Description
QR_CODE_ERROR_CORRECTION_LEVEL_L	48	Error correction level L
QR_CODE_ERROR_CORRECTION_LEVEL_M	49	Error correction level M
QR_CODE_ERROR_CORRECTION_LEVEL_Q	50	Error correction level Q
QR_CODE_ERROR_CORRECTION_LEVEL_H	51	Error correction level H

- getResponse: A message is sent to the application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

■ Example

```
public class MainActivity extends Activity {
  …
  private WpPrinter mWPPrinter;
  …
  public void onCreate(Bundle savedInstanceState) {
    …
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    …
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
      mWPPrinter.printQrCode(“www.WP.com”, WPPrinter.ALIGNMENT_LEFT,
                              WPPrinter.QR_CODE_MODEL2, 8,
                              WPPrinter.QR_CODE_ERROR_CORRECTION_LEVEL_L, true);
          }
          break;
        case WPPrinter.MESSAGE_PRINT_COMPLETE:
          mWPPrinter.disconnect();
          break;
      }
      return true;
    }
  };
}
```

9.4.13 printSelfTest

This page prints Self-Test page. Printer settings and current code page are printed.

■ Syntax

public void printSelfTest(boolean getResponse)

■ Example

```
public class MainActivity extends Activity {
    ...
    private WpPrinter mWPPrinter;
    ...
    public void onCreate(Bundle savedInstanceState) {
        ...
        mWpPrinter = new WPPrinter(this, mHandler, null);
        mWPPrinter.connect(null);
        ...
    }
    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case WPPrinter.MESSAGE_STATE_CHANGE:
                    if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
                        mWPPrinter.printSelfTest(true);
                    }
                    break;
                case WPPrinter.MESSAGE_PRINT_COMPLETE:
                    mWPPrinter.disconnect();
                    break;
            }
            return true;
        }
    };
}
```

9.4.14 printText

This method prints character string.

- Syntax

public void printText(String text, int alignment, int attribute, int size, boolean getResponse)

- Parameters

- text: character string to print
- alignment: character string alignment

Code	Value	Description
ALIGNMENT_LEFT	0	Align to left
ALIGNMENT_CENTER	1	Align to center
ALIGNMENT_RIGHT	2	Align to right

- attribute: attributes of the character string. When more than one parameter is configured, each option should be combined through bitwise OR operation.

Code	Value	Description
TEXT_ATTRIBUTE_FONT_A	0	font type A (12X24)
TEXT_ATTRIBUTE_FONT_B	1	font type B (9X17)
TEXT_ATTRIBUTE_FONT_C	2	font type C (9X24)
TEXT_ATTRIBUTE_UNDERLINE1	4	Underline with 1 dot thickness
TEXT_ATTRIBUTE_UNDERLINE2	8	Underline with 2 dots thickness
TEXT_ATTRIBUTE_EMPHASIZED	16	Bold
TEXT_ATTRIBUTE_REVERSE	32	Reversed

- size: size of character string to print. The width and height parameters should be combined through bitwise OR operation.

Code	Value	Description
TEXT_SIZE_HORIZONTAL1	0	1X magnification on width
TEXT_SIZE_HORIZONTAL2	16	2X magnification on width
TEXT_SIZE_HORIZONTAL3	32	3X magnification on width
TEXT_SIZE_HORIZONTAL4	48	4X magnification on width
TEXT_SIZE_HORIZONTAL5	64	5X magnification on width
TEXT_SIZE_HORIZONTAL6	80	6X magnification on width
TEXT_SIZE_HORIZONTAL7	96	7X magnification on width
TEXT_SIZE_HORIZONTAL8	112	8X magnification on width
TEXT_SIZE_VERTICAL1	0	1X magnification on height
TEXT_SIZE_VERTICAL2	1	2X magnification on height
TEXT_SIZE_VERTICAL3	2	3X magnification on height
TEXT_SIZE_VERTICAL4	3	4X magnification on height
TEXT_SIZE_VERTICAL5	4	5X magnification on height
TEXT_SIZE_VERTICAL6	5	6X magnification on height
TEXT_SIZE_VERTICAL7	6	7X magnification on height
TEXT_SIZE_VERTICAL8	7	8X magnification on height

- getResponse: A message is sent to the application handler upon completion ofprinting if this parameter is set to True, and message is not sent if it is set to False.

Example

```
public class MainActivity extends Activity {
    ...
    private WpPrinter mWPPrinter;
    ...
    public void onCreate(Bundle savedInstanceState) {
        ...
        mWpPrinter = new WPPrinter(this, mHandler, null);
        mWPPrinter.connect(null);
        ...
    }
    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case WPPrinter.MESSAGE_STATE_CHANGE:
                    if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
                        mWPPrinter.printText("printText\n",
                                     WPPrinter.ALIGNMENT_LEFT,
                                    WPPrinter.TEXT_ATTRIBUTE_FONT_A |
                                                 WPPrinter.TEXT_ATTRIBUTE_UNDERLINE1,
                                                WPPrinter.TEXT_SIZE_HORIZONTAL1 |
                                     WPPrinter.TEXT_SIZE_VERTICAL1, true);
                    }
                    break;

                case WPPrinter.MESSAGE_PRINT_COMPLETE:
                    mWPPrinter.disconnect();
                    break;
            }
            return true;
        }
    };
}
```

9.4.15 StringToBitMap

This Method pring string to bitmap

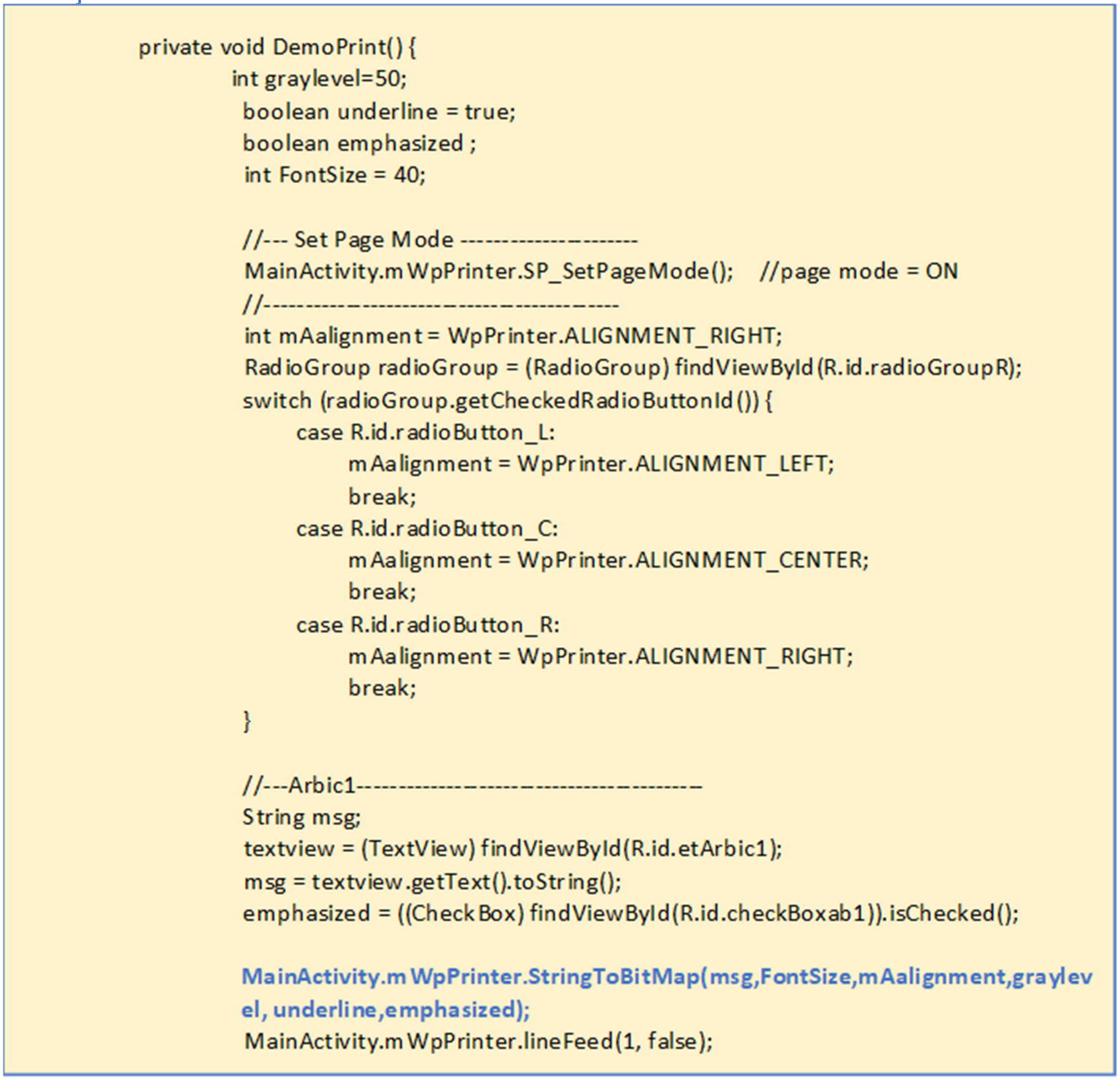
- Syntax

public void StringToBitMap(String string,int fontSize,int alignment,int graylevel, boolean underline,boolean Emphasized)

- Parameters

- string: Multi language string text
- fontSize: String font size
- alignment:
- graylenvel:
- underline:
- Emphasized:

- Example



```
private void DemoPrint() {
        int graylevel=50;
         boolean underline = true;
         boolean emphasized ;
         int FontSize = 40;

         //--- Set Page Mode -------------------
         MainActivity.mWpPrinter.SP_SetPageMode();    //page mode = ON
         //------------------------------------
         int mAalignment = WpPrinter.ALIGNMENT_RIGHT;
         RadioGroup radioGroup = (RadioGroup) findViewById(R.id.radioGroupR);
         switch (radioGroup.getCheckedRadioButtonId()) {
              case R.id.radioButton_L:
                   mAalignment = WpPrinter.ALIGNMENT_LEFT;
                   break;
              case R.id.radioButton_C:
                   mAalignment = WpPrinter.ALIGNMENT_CENTER;
                   break;
              case R.id.radioButton_R:
                   mAalignment = WpPrinter.ALIGNMENT_RIGHT;
                   break;
         }

         //---Arbic1------------------------------------
         String msg;
         textview = (TextView) findViewById(R.id.etArbic1);
         msg = textview.getText().toString();
         emphasized = ((CheckBox) findViewById(R.id.checkBoxab1)).isChecked();

         MainActivity.mWpPrinter.StringToBitMap(msg,FontSize,mAalignment,graylev
         el, underline,emphasized);
         MainActivity.mWpPrinter.lineFeed(1, false);
```

9.4.16 printText_THAI

This method prints unicode THAI string.

- Syntax

public void printText_THAI(String text, int alignment, int attribute, int size, boolean getResponse)

- Parameters

- text: character string to print
- Others is the same as printText()

- Example

```
msg = "ใบเสร็จรับเงิน" ;

int alignment = WpPrinter.ALIGNMENT_CENTER;
int attribute = WpPrinter.TEXT_ATTRIBUTE_FONT_A ;
int size = WpPrinter.TEXT_SIZE_HORIZONTAL2 |

           WpPrinter.TEXT_SIZE_VERTICAL2 ;

mWpPrinter.printText_THAI(data,alignment, attribute, size , false);
```

9.5 Receive Printer Response

9.5.1 automateStatusBack

This method enables or disables automatic status check function of printer. When it is activated, the MESSAGE_READ message (arg1: PROCESS_AUTO_STATUS_BACK) is sent to the application handler whenever there is any change in the printer status. The arg2 includes the following values if there is any error in printer.

Code	Value	Description
AUTO_STATUS_COVER_OPEN	0x20	Printer cover is open.
AUTO_STATUS_NO_PAPER	0x0c	No printer paper

- Syntax

public void automateStatusBack(boolean isEnable)

9.5.2 getBatteryStatus

This method checks the battery status of mobile printer. The MESSAGE_READ message (arg1: PROCESS_GET_BATTERY_STATUS) is sent to the application handler when battery status check is completed. The arg2 of this message includes the following values to indicate the battery status.

Code	Value	Description
STATUS_BATTERY_FULL	48	Battery is full.
STATUS_BATTERY_HIGH	49	Battery high high.
STATUS_BATTERY_MIDDLE	50	Battery is middle.
STATUS_BATTERY_LOW	51	Battery is low.

- Syntax

public void getBatteryStatus()

Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect();
    ...
  }
  public void onDestroy() {
    ...
    if (mWpPrinter != null) { mWPPrinter.automateStatusBack(false); mWPPrinter.disconnect();
    }
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            mWPPrinter.automateStatusBack(true);
          }
          break;
        case WPPrinter.MESSAGE_READ: StringBuffer buffer = new StringBuffer(0);
          if (msg.arg1 == WPPrinter.PROCESS_AUTO_STATUS_BACK) {
             if ((msg.arg2 & WPPrinter.AUTO_STATUS_COVER_OPEN) ==
               WPPrinter.AUTO_STATUS_COVER_OPEN) {
              buffer.append("Cover is open.\n");
            }
             if ((msg.arg2 & WPPrinter.AUTO_STATUS_NO_PAPER) ==
              WPPrinter.AUTO_STATUS_NO_PAPER) {
              buffer.append("Paper end sensor: no paper present.\n");
            }
            if (buffer.capacity() > 0) {
               Toast.makeText(getApplicationContext(), buffer.toString(),
               Toast.LENGTH_SHORT).show();
            }
            break;
          }
          Toast.makeText(getApplicationContext(), "No error.", Toast.LENGTH_SHORT).show();
          return true;
      }
    };
```

9.5.3 getPrinterId

This method checks the printer information. When the information is available, MESSAG_READ message (arg1: PROCESS_GET_PRINTER_ID) is sent to the application handler.

- Syntax

public void getPrinterId(int idType)

- Parameters

- idType: Information of the printer to check

ID Type	Value	Description
PRINTER_ID_FIRMWARE_VERSION	65	Firmware version
PRINTER_ID_MANUFACTURER	66	Manufacturer
PRINTER_ID_PRINTER_MODEL	67	Printer model name
PRINTER_ID_PRODUCT_SERIAL	68	Print Serial Number

- Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;

  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            mWPPrinter.getPrinterId(WPPrinter.PRINTER_ID_FIRMWARE_VERSION);
          }
          break;
        case WPPrinter.MESSAGE_READ:
          if (msg.arg1 == WPPrinter.PROCESS_GET_PRINTER_ID) {
            Bundle data = msg.getData(); Toast.makeText(getApplicationContext(),
                data.getString(WPPrinter.KEY_STRING_PRINTER_ID),
                    Toast.LENGTH_SHORT).show();
          }
          break;
      }
      return true;
    }
  };
}
```

9.5.4 getStatus

This method gets the status of printer. When the status is obtained, the MESSAGE_READ message (arg1: PROCESS_GET_STATUS) is sent to the application handler. The following values can be returned in arg2.

Code	Value	Description
STATUS_NORMAL	0	Normal
STATUS_COEVER_OPEN	4	Cover is open
STATUS_PAPER_NEAR_END	12	Printer paper reaches to near end state
STATUS_PAPER_NOT_PRESENT	96	No printer paper

- Syntax

public void getStatus()

- Example

```
public class MainActivity extends Activity {
  .....
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            mWPPrinter.getStatus();
          }
          break;
        case WPPrinter.MESSAGE_READ:
          ........
      }
    };
  }
}
```

9.6 NV Image

9.6.1 printNvImage

This method prints image stored in the non-volatile memory area of printer.

■ Syntax

public void printNvImage(int keyCode, boolean getResponse)

■ Parameters

- keyCode: address code of NV image to print
- getResponse: MESSAGE_PRINT_COMPLETE A message is sent to the
- application handler upon completion of printing if this parameter is set to True, and message is not sent if it is set to False.

■ Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ....
  private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            mWPPrinter.getDefinedNvImageKeyCodes();
          }
          break;
        case WPPrinter.MESSAGE_READ:
          if (msg.arg1 == WPPrinter.PROCESS_GET_NV_IMAGE_KEY_CODES) {
            Bundle data = msg.getData();
            int[] keyCodes =false;
            data.getintArray(WPPrinter.NV_IMAGE_KEY_CODES);
            if (keyCodes != null) {
              for (int i = 0; i < keyCodes.length; i++) {
                mWPPrinter.printNvImage(keyCodes[i],false);
              }
            }
          }
          break;
      }
      return true;
    }
  };
}
```

9.7 Page Mode

9.7.1 setAbsolutePrintPosition

This method sets the horizontal position in the page mode.

- Syntax

public void setAbsolutePrintPosition(int position)

- Parameters

- position: horizontal position to set

9.7.2 setAbsoluteVerticalPrintPosition

This method sets the vertical position in the page mode.

- Syntax

public void setAbsoluteVerticalPrintPosition(int position)

- Parameters

- position: vertical position to set

9.7.3 setPageMode

This function switches the mode to page mode.

- Syntax

public void setPageMode()

9.7.4 setPrintArea

This function sets the printing area in the page mode.

- Syntax

public void setPrintArea(int x, int y, int width, int height)

- Parametes

- x: origin of the horizontal printing area
- y: origin of the vertical printing area
- width: width of printing area
- height: height of printing area

9.7.5 setPrintDirection

This method sets the printing direction in the page mode.

- Syntax

public void setPrintDirection(int direction)

- Parameter

  - direction: direction of printing. Refer to the following table for possible options.

Code	Value	Description
DIRECTION_0_DEGREE_ROTATION	0	Print without rotation
DIRECTION_90_DEGREE_ROTATION	1	Print clockwise 90˚ rotated image
DIRECTION_180_DEGREE_ROTATION	2	Print clockwise 180˚ rotated image
DIRECTION_270_DEGREE_ROTATION	3	Print clockwise 270˚ rotated image

9.7.6 setStandardMode

This method closes the page mode and switches to the standard mode.

- Syntax

public void setStandardMode()

- Example

```
public class MainActivity extends Activity {
    private WpPrinter mWPPrinter;
   ....
    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case MESSAGE_STATE_CHANGE:
                    if (msg.arg1 == STATE_CONNECTED) {
                        mWPPrinter.setPageMode();
                        mWPPrinter.setPrintDirection(WPPrinter.DIRECTION_180_DEGREE_ROTATION);
                        mWPPrinter.setPrintArea(0, 0, 384, 840);
                        mWPPrinter.setAbsoluteVerticalPrintPosition(100);
                mWPPrinter.printBitmap(bitmap, WPPrinter.ALIGNMENT_CENTER,
                                                    WPPrinter.BITMAP_WIDTH_FULL, 88,false);
                        mWPPrinter.formFeed(true);
                        mWPPriner.setStandardMode();
                    }
                    break;
                ...
            }
            return true;
        }
    };
}
```

9.8 Settings

9.8.1 getBsCodePage

This method obtains the current code page. When code page information is obtained, the MESSAGE_READ message (arg1: PROCESS_GET_BS_CODE_PAGE) is sent to the application handler. The code page is returned in arg2 of the message as follows.

Code	Value	Description
CODE_PAGE_437_USA	0	Page 0 437(USA standard Europe)
CODE_PAGE_KATAKANA	1	Page 1 Katakana
CODE_PAGE_850_MULTILINGUAL	2	Page 2 850 (Multilingual)
CODE_PAGE_860_PORTUGUESE	3	Page 3 860 (Portuguese)
CODE_PAGE_863_CANADIAN_FRENCH	4	Page 4 863 (Canadian-French)
CODE_PAGE_865_NORDIC	5	Page 5 865 (Nordic)
CODE_PAGE_1252_LATIN1	16	Page 16 1252 (Latin I)
CODE_PAGE_866_CYRILLIC2	17	Page 17 866 (Cyrillic #2)
CODE_PAGE_852_LATIN2	18	Page 18 852 (Latin 2)
CODE_PAGE_858_EURO	19	Page 19 858 (Euro)
CODE_PAGE_862_HEBREW_DOS_CODE	21	Page 21 862 (Hebrew DOS code)
CODE_PAGE_864_ARABIC	22	Page 22 864 (Arabic)
CODE_PAGE_THAI42	23	Page 23 Thai42
CODE_PAGE_1253_GREEK	24	Page 24 1253 (Greek)
CODE_PAGE_1254_TURKISH	25	Page 25 1254 (Turkish)
CODE_PAGE_1257_BALTIC	26	Page 26 1257 (Baltic)
CODE_PAGE_FARSI	27	Page 27 Farsi
CODE_PAGE_1251_CYRILLIC	28	Page 28 1251 (Cyrillic)
CODE_PAGE_737_GREEK	29	Page 29 737 (Greek)
CODE_PAGE_775_BALTIC	30	Page 30 775 (Baltic)
CODE_PAGE_THAI14	31	Page 31 Thai14
CODE_PAGE_1255_HEBREW_NEW_CODE	33	Page 33 1255 (Hebrew Newcode)
CODE_PAGE_THAI11	34	Page 34 Thai11
CODE_PAGE_THAI18	35	Page 35 Thai18
CODE_PAGE_855_CYRILLIC	36	Page 36 855 (Cyrillic)
CODE_PAGE_857_TURKISH	37	Page 37 857 (Turkish)
CODE_PAGE_928_GREEK	38	Page 38 928 (Greek)
CODE_PAGE_THAI16	39	Page 39 Thai16
CODE_PAGE_1256_ARABIC	40	Page 40 1256 (ARB)
CODE_PAGE_1258_VIETNAM	41	Page 41 1258 (Vietnam)
CODE_PAGE_KHMER_CAMBODIA	42	Page 42 Khmer (Cambodia)
CODE_PAGE_1250_CZECH	43	Page 47 1250 (Czech)

■ Syntax

public void getBsCodePage()

Example

```
public class MainActivity extends Activity {
    ...
    private WpPrinter mWPPrinter;
    ...
    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case WPPrinter.MESSAGE_STATE_CHANGE:
                    if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
                        mWPPrinter.getBsCodePage();
                    }
                    break;

                case WPPrinter.MESSAGE_READ:
                    if (msg.arg1 == WPPrinter.PROCESS_BS_CODE_PAGE) {
                        data = msg.getData();
                        Toast.makeText(getApplicationContext(),
                                data.getString(WPPrinter.KEY_STRING_CODE_PAGE),
Toast.LENGTH_SHORT).show();
                    }
                }
                break;
            }
            return true;
        }
    };
```

9.8.2 setDoubleByteFont

This method downloads the double byte font (KS5601, BIG5, GB2312, SHIFT-JIS) to printer. When download is completed, the MESSAGE_WRITE message (arg1: PROCESS_SET_DOUBLE_BYTE_FONT) is sent to the application handler. In this case, the downloaded fonts will be applied after printer is rebooted after receiving this message.

- Syntax

public void setDoubleByteFont(int codePage)

- Parameters

  - codePage: code page to set

Code	Value	Description
DOUBLE_BYTE_FONT_KS5601	124	Korean (KS5601)
DOUBLE_BYTE_FONT_BIG5	125	Chinese (BIG5)
DOUBLE_BYTE_FONT_GB2312	126	Chinese (GB2312)
DOUBLE_BYTE_FONT_SHIFT_JIS	127	Japanese (SHIFT-JIS)

- Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
   private final Handler mHandler = new Handler(new Handler.Callback() {
      public boolean handleMessage(Message msg) {
         switch (msg.what) {
            case WPPrinter.MESSAGE_STATE_CHANGE:
               if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
                  mWPPrinter.setDoubleByteFont( WPPrinter.DOUBLE_BYTE_FONT_KS5601);
               }
               break;
            case WPPrinter.MESSAGE_WRITE:
         mWPPrinter.disconnect();
               break;
         }
         return true;
      }
  };}
```

9.8.3 setSingleByteFont

This method sets the single byte font.

- Syntax

public void setSingleByte(int codePage)

- Parameters

  - codePage: code page to set

- Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    ...
  }
  public void onDestroy() {
    ...
    if (mWpPrinter != null) {
      mWPPrinter.disconnect();
    }
    ...
  }
    private final Handler mHandler = new Handler(new Handler.Callback() {
    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            mWPPrinter.setSingleByteFont(WPPrinter.CODE_PAGE_437_USA);
          }
          break;
      }
      return true;
    }
  };
}
```

9.9 Miscellaneous Functions

9.9.1 cutPaper

This method cuts paper.

- Syntax

public void cutPaper(boolean getResponse)

- Paramters

- getResponse: When this parameter is set to True,
- the MESSAGE_PRINT_COMPLETE message is sent to the application handler when paper cut operation is completed. If it is set to False, message is not sent to the application handler after paper is cut.

- Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect(null);
    ...
  }
  public void onDestroy() {
    ...
    if (mWpPrinter != null) {
      mWPPrinter.disconnect();
    }
    ...
  }
   public boolean handleMessage(Message msg) {
      switch (msg.what) {
         case WPPrinter.MESSAGE_STATE_CHANGE:
            if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
               // TODO:
               mWPPrinter.cutPaper(true);
            }
            break;
         .....
      }
      return true;
    }
  };
}
```

9.9.2 cutPaper

This method feeds the paper by the specified number of lines and cut the paper.

- Syntax

public void cutPaper(int feeds, Boolean getResponse)

- Paramters

  - feeds: number of lines to feed before cutting paper
  - getResponse: When this parameter is set to True,the MESSAGE_PRINT_COMPLETE message is sent to the application handler when paper cut operation is completed. If it is set to False, message is not sent to the application handler after paper is cut.

- Example

```
public class MainActivity extends Activity {
    ...
    private WpPrinter mWPPrinter;
    ...
    public void onCreate(Bundle savedInstanceState) {
        ...
        mWpPrinter = new WPPrinter(this, mHandler, null);
        mWPPrinter.connect(null);
        ...
    }
    public void onDestroy() {
        ...
        if (mWpPrinter != null) {
            mWPPrinter.disconnect();
        }
        ...
    }
    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case WPPrinter.MESSAGE_STATE_CHANGE:
                    if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
                     // TODO:
                      mWPPrinter.cutPaper(10, true);
                    }
                case WPPrinter.MESSAGE_PRINT_COMPLETE:
                    mWPPrinter.disconnect();
                    break;
            }
            return true;
        }
    };
}
```

9.9.3 executeDirectIo

This method sends command directly to printer. If response is to be received from the printer, the MESSAGE_READ message (arg1: PROCESS_EXECUTE_DIRECT_IO)should be sent to the application handler.

- **Syntax**

public void executeDirectIo(byte[] command, boolean hasResponse)

- **Parameters**

  - command: command to send to the printer
  - hasResponse: Set this True if response to the command is to be received from the printer, or False if not.

- **Example**

```
public class MainActivity extends Activity {
    private WpPrinter mWPPrinter;
    ...

    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case WPPrinter.MESSAGE_STATE_CHANGE:
                    if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
                        byte[] command = new byte[] {0x10, 0x04, 0x02};
                        mWPPrinter.executeDirectIo(command, true);
                    }
                    break;
                case WPPrinter.MESSAGE_READ:
                    if (msg.arg1 == WPPrinter.PROCESS_EXECUTE_DIRECT_IO) {
                        Bundle data = msg.getData();
                        byte[] response =data.getByteArray(WPPrinter.KEY_STRING_DIRECT_IO);
                        // TODO: response time
                    }
                    break;
            }
            return true;
        }
    };
```

9.9.4 getMacAddress

This method obtains and returns network MAC address of the printer connected with LAN or Wireless LAN.

- Syntax

public String getMacAddress()

- Example

```
public class MainActivity extends Activity {
    private WpPrinter mWPPrinter;
    ...
    private final Handler mHandler = new Handler(new Handler.Callback() {
        public boolean handleMessage(Message msg) {
            switch (msg.what) {
                case WPPrinter.MESSAGE_STATE_CHANGE:
                    if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
                        String macAddress = mWPPrinter.getMacAddress();
                        Toast.makeText(getApplicationContext(), Toast.LENGTH_SHORT).show();
                    }
                    break;
            }
            return true;
        }
    };
}
```

9.9.5 getUsbSerial

This method returns USB serial number of the printer connected over USB.

- Syntax

public String getUsbSerial()

- Example

```
public class MainActivity extends Activity {
    private WpPrinter mWPPrinter;
    ...
   private final Handler mHandler = new Handler(new Handler.Callback() {
       public boolean handleMessage(Message msg) {
           switch (msg.what) {
               case WPPrinter.MESSAGE_STATE_CHANGE:
                   if (msg.arg1 == STATE_CONNECTED) {
                       String usbSerial = mWPPrinter.getUsbSerial();
                       Toast.makeText(getApplicationContext(), usbSerial,Toast.LENGTH_SHORT).show();
                   }
                   break;
           }
           return true;
       }
    };
}
```

9.9.6 initialize

This method initializes the printer settings to a state the same as after booting. The data in the printer buffer is initialized but the data in the printer receive buffer is not. NV image stored in the printer is not initialized. If the printer is in the page mode, all data in the print area is removed and the printer is initialized to the standard mode.

- Syntax

public void initialize()

- Example

```
public class MainActivity extends Activity {
  ...
  private WpPrinter mWPPrinter;
  ...
  public void onCreate(Bundle savedInstanceState) {
    ...
    mWpPrinter = new WPPrinter(this, mHandler, null);
    mWPPrinter.connect();
    ...
  }
  public void onDestroy() {
    ...
    if (mWpPrinter != null) {
      mWPPrinter.disconnect();
    }
    ...
  }
  private final Handler mHandler = new Handler(new Handler.Callback() {

    public boolean handleMessage(Message msg) {
      switch (msg.what) {
        case WPPrinter.MESSAGE_STATE_CHANGE:
          if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
            mWPPrinter.initialize();
          }
          break;
      }
      return true;
    }
  };
}
```

9.9.7 kickOutDrawer

This method opens cash drawer or run melody box.

- Syntax

public void kickOutDrawer(int connectorPin)

- Parameter

- connectorPin: connector pin of cash register or melody box

Code	Value	Description
DRAWER_CONNECTOR_PIN2	0	Connector pin 2
DRAWER_CONNECTOR_PIN5	1	Connector pin 5

- Example

```
public class MainActivity extends Activity {
   ...
   private WpPrinter mWPPrinter;
   ...
   public void onCreate(Bundle savedInstanceState) {
      ...
      mWpPrinter = new WPPrinter(this, mHandler, null);
      mWPPrinter.connect();
      ...
   }
   public void onDestroy() {
      ...
      if (mWpPrinter != null) {
         mWPPrinter.disconnect();
      }
      ...
   }
   private final Handler mHandler = new Handler(new Handler.Callback() {
      public boolean handleMessage(Message msg) {
         switch (msg.what) {
            case WPPrinter.MESSAGE_STATE_CHANGE:
               if (msg.arg1 == WPPrinter.STATE_CONNECTED) {
                  mWPPrinter.kickOutDrawer(DRAWER_CONNECTOR_PIN5);
               }
               break;
         }
         return true;
      }
   };
}
```

9.9.8 shutDown

This method terminates the connection with the printer and releases all resources. WpPrinter instance is not available anymore after executing this method.

- Syntax

public void shutDown()

- Example

```
public class MainActivity extends Activity {
    ...
    private WpPrinter mWPPrinter;
    ...
    public void onCreate(Bundle savedInstanceState) {
        ...
        mWpPrinter = new WPPrinter(this, mHandler, null);
        mWPPrinter.connect();
        ...
    }
    public void onDestroy() {
      if (mWpPrinter != null) {
           mWPPrinter.shutDown();
        }
    };
}
```

9.9.9 get_SDK_Version

This method will get the SDK library version.

- Syntax

public String get_SDK_Version()

- Example

```
public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.item1:
                mWpPrinter.findBluetoothPrinters();
                break;
            case R.id.itemInfo:
                String SDK_Ver = WpPrinter.get_SDK_Version();
                Toast.makeText(getApplicationContext(), "App version: "+
                            versionName +versionNumber +
                            "\r\nSDK version: "+SDK_Ver, Toast.LENGTH_SHORT).show();
                return true;
              case R.id.itemLog:
                DialogManager.showLogDialog(this, mWpPrinter.getLog());;
                return true;
        }
        return false;
}
```

9.9.10 is_connected

This method will get the printer connection status.

- Syntax

public boolean is_connected()

- Example

```
public void onCreate(Bundle savedInstanceState) {
            // get Button Getstatus
            Button mybtn = (Button) findViewById(R.id.btSGetStat);
            mybtn.setOnClickListener(new View.OnClickListener() {
             public void onClick(View v) {
                         if( WpPrinter.is_connected() == true) {
                                     mWpPrinter.getStatus();
                        }
                        else
                           Toast.makeText(getApplicationContext(),AskConnectString,  Toast.LENGTH_SHORT).show();
                         }
            });
}
```

9.9.11 is_DrawerOpen

This method will check the Drawer status.

- Syntax

public boolean is_DrawerOpen ()

9.9.12 is_PrinterOffline

This method will check the printer offline status.

- Syntax

public boolean is_PrinterOffline ()

9.9.13 is_PrinterWaitforRecover

This method will check the printer WaitforRecover status.

- Syntax

public boolean is_PrinterWaitforRecover ()

9.9.14 is_PaperFedByButton

This method will check the printer fed by button status.

- Syntax

public boolean is_PaperFedByButton ()

9.9.15 is_CoverOpen

This method will check the printer cover open/close status.

- Syntax

public boolean is_CoverOpen ()

9.9.16 is_StopByPaperEnd

This method will check the printer is stopped at paper end status.

- Syntax

public boolean is_StopByPaperEnd ()

9.9.17 is_ErrorOccurred

This method will check the printer is stopped at paper end status.

- Syntax

public boolean is_ErrorOccurred ()

9.9.18 is_AutoCutterError

This method will check the printer cutter status.

- Syntax

public boolean is_ErrorOccurred ()

9.9.19 is_RollPaperNearEnd

This method will check the printer roll paper near end status.

- Syntax

public boolean is_AutoRecoverableError ()

9.9.20 is_PaperNotPresent

This method will check the printer paper status.

- Syntax

public boolean is_PaperNotPresent ()

9.9.21 is_is_AnyErrors

This method will check the printer has any erroe status.

- Syntax

public boolean is_AnyErrors ()

Example

```
private void dispatchMessage(Message msg) {
       Bundle data;
       StringBuffer buffer;
       Intent intent;
       switch (msg.arg1) {

        case WpPrinter.PROCESS_GET_KIOSK_STATUS:   {
                    if( WpPrinter.is_PrinterOffline() ) {
                          Toast.makeText(getApplicationContext(), "Printer off-line.",
                          Toast.LENGTH_SHORT).show();
                     }
                     if( WpPrinter.is_PaperFedByButton() ) {
                            Toast.makeText(getApplicationContext(), "Paper is being fed by the paper
                             feed button.", Toast.LENGTH_SHORT).show();
                       }
                       if(WpPrinter.is_CoverOpen()) {
                            Toast.makeText(getApplicationContext(), "Cover is Opened.",
                             Toast.LENGTH_SHORT).show();
                       }
                       if( WpPrinter.is_PaperFeedByButton_A()  )    {
                            Toast.makeText(getApplicationContext(), "Paper is being fed by the paper
                             feed button A", Toast.LENGTH_SHORT).show();
                       }
                       if( WpPrinter.is_StopByPaperEnd()  )  {
                            Toast.makeText(getApplicationContext(), "Printing stops due to a paper-
                             end.", Toast.LENGTH_SHORT).show();
                         }
                         if( WpPrinter.is_ErrorOccurred()  )    {
                            Toast.makeText(getApplicationContext(), "Error occurred.",
                             Toast.LENGTH_SHORT).show();
                         }
                         if( WpPrinter.is_AutoCutterError()  ){
                               Toast.makeText(getApplicationContext(), "Autocutter error occurred.",
                             Toast.LENGTH_SHORT).show();
                         }
                         if( WpPrinter.is_RollPaperNearEnd()  )         {
                            Toast.makeText(getApplicationContext(), "Paper near end.",
                             Toast.LENGTH_SHORT).show();
                         }
                         if( WpPrinter.is_PaperNotPresent()  )          {
                            Toast.makeText(getApplicationContext(), "Paper not present.",
                             Toast.LENGTH_SHORT).show();
                         }
                        if( WpPrinter.is_AnyErrors() == false)
                           Toast.makeText(getApplicationContext(), "No Error.",
                           Toast.LENGTH_SHORT).show();
                         break;
                         }
```

9.10 Special Function

9.10.1 SP_cutPaper

This method is setting the cut mode and feed line numbers

- Syntax

void SP_cutPaper(byte feeds, CutType cutMode, final boolean toStart, final boolean getResponse)

- Parameter

- Feeds: set feed line numbers
- cutMode: set cut mode : Full /Partial Cut
- toStart: set to start position after cut
- getResponse: waiting for response

9.10.2 SP_printBig5

This method is printing the Big5 string

- Syntax

void SP_printBig5(String str, final int alignment,final int attribute,final int size,final boolean getResponse)

- Parameter

- str: Big5 string
- alignment:
- attribute:
- size : font size
- getResponse: waiting for response

9.10.3 SP_SetPageMode

This method set the printer page mode start

- Syntax

void SP_SetPageMode ()

9.10.4 SP_PrintPageMode

This method set the printer page mode stop and print the buffer data

- Syntax

void SP_PrintPageMode ()

9.10.5 SP_PrintFeedPaperDot

This method is to set feed dot numbers

- Syntax

void SP_PrintFeedPaperDot (byte size)

- Parameter

- size : the number of dots

9.10.6 SP_PrintNVimage

This method is to print the inside image of NV ram

- Syntax

  void SP_PrintNVimage(byte index, byte mode,int alignment, final boolean getResponse)

- Parameter

- index : the NV image ID (1 ~ 10)
- mode : 0
- alignment:
- getResponse : waiting for response

9.10.7 SP_SetPaperBackFed

This method is to set printer start position up to n dots.

- Syntax

  void SP_SetPaperBackFed(byte position)

- Parameter

- position : the enumber of dots up

9.10.8 SP_SetDotposition

This method is to set printer start position offset from the left.

- Syntax

  void SP_SetDotposition(int position)

- Parameter

- position : the enumber of dots from left

9.10.9 SP_SetQRcodeVersion

This method is to set the size of QR code .

- Syntax

  void SP_SetQRcodeVersion(byte version)

- Parameter

version: the size of QR code module (1~ 16)

9.10.10 SP_SelectCorrectionLevel

This method is to set the size of QR code correction level.

- Syntax

void SP_SelectCorrectionLevel(byte Level)

- Parameter

Level: the sQR code correction level

level	Function	Recovery Capapity %
‘0’	Selects Error correction level L	7%
‘1’	Selects Error correction level M	15%
‘2’	Selects Error correction level Q	25%
‘3’	Selects Error correction level H	30%

9.10.11 getDefaultCodePage()

This method is to get the printer default code page setting

- Syntax

void getDefaultCodePage (byte Level)

- example:

```
Button btn2 = (Button) findViewById(R.id.btnClear);
btn2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
            MainActivity.mWpPrinter.getDefaultCodePage();
    }
 });
```

9.10.12 SelectCodePage

This method is to set the selected code page to printer.

- Syntax

  void SelectCodePage(int index)

- example

```
public void onClick(DialogInterface dialog, int which) {
     switch (which) {
          case 0:
             MainActivity.mWpPrinter.SelectCodePage( WpPrinter.CODE_PAGE_437_USA);
             MainActivity.mWpPrinter.printText(CODE_PAGE_ITEMS[which] ,
                       WpPrinter.ALIGNMENT_CENTER,
                       WpPrinter.TEXT_ATTRIBUTE_FONT_A |
                       WpPrinter.TEXT_ATTRIBUTE_EMPHASIZED,
                       WpPrinter.TEXT_SIZE_HORIZONTAL1 | WpPrinter.TEXT_SIZE_VERTICAL1,
                       false);
             MainActivity.mWpPrinter.lineFeed(1,false);
             ………..

      }
  }
```

9.10.13 SP_printText

This method is to print the text by using code page setting

- Syntax

  void SP_printText(String codepage,String str, final int alignment,final int attribute,final int size,final boolean getResponse)

- Parameter

  - codepage
  - str: text string
  - alignment:
  - attribute:
  - size : font size
  - getResponse: waiting for response

- example

```
  //---Print Text1------------------------
alignment = WpPrinter.ALIGNMENT_CENTER;
attribute = WpPrinter.TEXT_ATTRIBUTE_FONT_A ;
size = WpPrinter.TEXT_SIZE_HORIZONTAL2 | WpPrinter.TEXT_SIZE_VERTICAL2 ;
mDataEdit = (EditText) findViewById(R.id.editText1);
data = mDataEdit.getText().toString();
MainActivity.mWpPrinter.SP_printBig5(data,alignment,attribute,size ,false);
MainActivity.mWpPrinter.SP_printText("BIG-5",data,alignment,attribute,size ,false);
MainActivity.mWpPrinter.SP_printText("GBK",data,alignment,attribute,size ,false);
MainActivity.mWpPrinter.SP_printText("GB18030",data,alignment,attribute,size ,false);
```

9.10.14 Net_connect

This method is connecting the printer by using network interface (TCPIP)

- Syntax

```
void Net_connect (final int type,final String host, final int port,
                                        final int timeout,final int interval)
```

- Parameter

- Type : TYPE_USB , TYPE_BLUETOOTH, TYPE_TCP
- host: IP addrsss string , “192.168.1.1”
- port: data port (9100)
- timeout: connection timeout time ,(3000ms)
- interval : Task delay time interval (50ms)

- example

```
mybtn = (Button) findViewById(R.id.btcutPaper);
mybtn.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
if( MainActivity.mDevice_Type != WpPrinter.TYPE_TCP) {
        MainActivity.mWpPrinter.SP_cutPaper((byte)10, WpPrinter.CutType.PART_CUT_MODE, false, false);
}
else
{
    byte count=0;
  while(Net_connect(MainActivity.mDevice_Type,WpPrinter.SelectDevice, 9100, 1000,50) == true) {
   try {Thread.sleep(MainActivity.THREAD_DLY_TIME);}
   catch (InterruptedException e) {e.printStackTrace();}
   if(count++ > (MainActivity.Host_TIMEOUT/MainActivity.THREAD_DLY_TIME))
     {return ; }
   }
   MainActivity.mWpPrinter.SP_cutPaper((byte)10, WpPrinter.CutType.PART_CUT_MODE, false, false);
   Net_disconnect(mDevice_Type);
   }
 }
});
```

9.10.15 Net_disconnect

This method is to disconnect the network interface

- Syntax

void Net_disconnect (final int type)

- Parameter

  - Type : TYPE_USB , TYPE_BLUETOOTH, TYPE_TCP

- example

```
mybtn = (Button) findViewById(R.id.btcutPaper);
mybtn.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
if( MainActivity.mDevice_Type != WpPrinter.TYPE_TCP) {
        MainActivity.mWpPrinter.SP_cutPaper((byte)10, WpPrinter.CutType.PART_CUT_MODE, false, false);
}
else
{
    byte count=0;
   while(Net_connect(MainActivity.mDevice_Type,WpPrinter.SelectDevice, 9100, 1000,50) == true) {
   try {Thread.sleep(MainActivity.THREAD_DLY_TIME);}
   catch (InterruptedException e) {e.printStackTrace();}
   if(count++ > (MainActivity.Host_TIMEOUT/MainActivity.THREAD_DLY_TIME))
    {return ; }
   }
   MainActivity.mWpPrinter.SP_cutPaper((byte)10, WpPrinter.CutType.PART_CUT_MODE, false, false);
   Net_disconnect(mDevice_Type);
   }
  }
});
```

9.10.16 ReleaseUSB()

This method is to send the Relese command to the connected printer when APP using the USB interface

- Syntax

void ReleaseUSB (void)

- Parameter

  - NA

- Example

```
public void onDestroy() {
        finish();
        unregisterReceiver(mUsbReceiver);

        super.onDestroy();

        try {
                if(this.mConnectedDeviceName != null)
                 {
                        mWpPrinter.ReleaseUSB();
                         mWpPrinter.disconnect();
                }
        }
        catch (IllegalArgumentException e)
        {
                 e.printStackTrace();
        }
  }
```

9.10.17 pageDataPrint()

This method set the printer start printing the page area data

- Syntax

void pageDataPrint (void)

- Parameter

  - NA

- Example

```
public void StringBitmapPrint () {
             ……
            ……
            //--- Set Page Mode ----------------------
                    MainActivity.mWpPrinter.setPageMode();           //1b 4c  (ESC L)
                    MainActivity.mWpPrinter.setPrintArea(0,0,576,300);//(0,0) to(576,300)
                    MainActivity.mWpPrinter.setPrintDirection(0);
            msg = "1234566778"
            MainActivity.mWpPrinter.setAbsolutePrintPosition(start_posX);        //set H position
            MainActivity.mWpPrinter.setAbsoluteVerticalPrintPosition(start_posY); //set V position
             MainActivity.mWpPrinter.StringToBitMap(msg,FontSize,mAalignment,graylevel,
                        underline,emphasized);
             MainActivity.mWpPrinter.pageDataPrint();
             MainActivity.mWpPrinter.cutPaper(6, true);
             MainActivity.mWpPrinter.setStandardMode();
   }
```

10. Programming

10.1 Programming Flow

Applications should be programmed in the following sequence.

1. Search printer
2. Open printer port
3. Send print data
4. Close printer port

10.2 Search Printer

Search available printers. This step can be skipped if the printer to connect is manually specified. Refer to the following codes.

10.3 Open Printer Port

Use the connect method to create printer instance and open the port. Refer to the following codes.

10.4 Send Print Data

Application handler sends print data to the printer after receiving connection completion message. Refer to the following codes.

10.5 Close Printer Port

Close the printer connection when application is shutdown if connection with the printer is maintained while application is running. If printer is to be connected on demand, close the connection after receiving print completion message. Refer to the following codes.

```
  public class MainActivity extends Activity {
    ...
    private WpPrinter mWPPrinter;
    ...
    public void onCreate(Bundle savedInstanceState) {
      ...
      mWpPrinter = new WPPrinter(this, mHandler, null);
     mWPPrinter.findBluetoothPrinters();
// mWPPrinter.findNetworkPrinters("192.168.1.50", 9100, 3000);
// mWPPrinter.findUsbPrinters();
 }
   }
```

11. Appendix

11.1 Connection related methods

Support printer / Method	WP-T612	WP-T810	WP-K835	WP-K837					
Connection related									
void initialize()	O	O	O	O					
void connect()	O	O	O	O					
void connect(final String address)	O	O	O	O					
void connect(final String host, final int port, final int timeout)	O	O	O	O					
void connect(final UsbDevice device)	O	O	O	O					
void connectUsb(final String serial)	O	O	O	O					
void disconnect()	O	O	O	O					
void shutDown()	O	O	O	O					
void findBluetoothPrinters()	X	O	O	O					
void findNetworkPrinters(final String ipAddr, final int port, final int timeout)	X	O	X	X					
findUsbPrinters()	O	O	O	O					
void findUsbPrintersBySerial()	O	O	O	O					
String getMacAddress()	X	O	X	X					
String getUsbSerial()	O	O	O	O					

11.2 Print related methods

Support printer / Method	WP-T612	WP-T810	WP-K835	WP-K837					
Print related									
void printBitmap(final Bitmap bitmap, final int alignment, int width, final int level, final boolean getResponse)	O	O	O	O					
void printBitmap(final String pathName, final int alignment, final int width, final int level, final boolean dither, final boolean compress, final boolean getResponse)	O	O	O	O					
void printBitmap(final Bitmap bitmap, final int alignment, int width, final int level, final boolean dither, final boolean compress, final boolean getResponse)	O	O	O	O					
void printBitmap(final Bitmap bitmap, final int alignment, int width, final int level, final boolean getResponse)	O	O	O	O					
void printBitmap(final byte[] pixels, final int alignment, final int width, final int height, final boolean getResponse)	O	O	O	O					
void print1dBarcode(final String data, final int barCodeSystem, final int alignment, final int width, final int height, final int characterPosition, final boolean getResponse)	O	O	O	O					
void printDotMatrixBitmap(final String pathName, final int alignment, final int width, final int level, final boolean getResponse)	O	O	O	O					
void printDotMatrixBitmap(final Bitmap bitmap, final int alignment, int width, final int level, final boolean getResponse)	O	O	O	O					
void printDotMatrixText(final String text, final int alignment, final int attribute, final int size, final boolean getResponse)	O	O	O	O					
void printQrCode(final String data, final int alignment, final int model, final int size, final boolean getResponse)	X	O	O	O					
void printQrCode(final String data, final int alignment, final int model, final int size, final int errorCorrectionLevel, final boolean getResponse)	X	O	O	O					
void printSelfTest(final boolean getResponse)	O	O	O	O					

void printText(String text, final int alignment, final int attribute, final int size, final boolean getResponse)	O	O	O	O					
void StringToBitMap(String string,int fontSize,int aligment,int graylevel, boolean underline,boolean Emphasized)	O	O	O	O					
void printNvImage(final int keyCode, final boolean getResponse)	O	O	O	O					
void lineFeed(int lines, boolean getResponse)	O	O	O	O					
void SP_PrintNVimage(byte index byte mode,int alignment, final boolean getResponse)	O	O	O	O					
void SP_PrintPageMode ()	O	O	O	O					
void SP_PrintFeedPaperDot (byte size)	O	O	O	O					
void SP_PrintNVimage(byte range, byte mode,int alignment, final boolean getResponse)	O	O	O	O					
void SP_SelectCorrectionLevel(byte Level)	O	O	O	O					
void SP_SetPageMode()	O	O	O	O					

11.3 Setting related methods

Method \ Support printer	WP-T612	WP-T810	WP-K835	WP-K837					
Setting related									
void executeDirectIo(byte[] command, boolean hasResponse)	O	O	O	O					
void setAbsolutePrintPosition(final int position)	O	O	O	O					
void setAbsoluteVerticalPrintPosition(final int position)	O	O	O	O					
void setPageMode()	O	O	O	O					
void setDoubleByteFont(int codePage)	X	X	X	X					
void setSingleByte(int codePage)	X	X	X	X					
void setPrintArea(final int x, final int y, final int width, final int height)	O	O	O	O					
void setPrintDirection(final int direction)	O	O	O	O					
void setStandardMode()	O	O	O	O					
void getBsCodePage()	O	O	O	O					
void selectCodePage(final int codePage)	O	O	O	O					
void SelectCharSet(int index)	O	O	O	O					
void cutPaper(boolean getResponse)	O	O	O	O					
void cutPaper(int feeds, Boolean getResponse)	O	O	O	O					
void SP_cutPaper(byte feeds, CutType cutMode, final boolean toStart, final boolean getResponse)	X	O	O	O					
void kickOutDrawer(int connectorPin)	X	O	O	O					
void getDefaultCodePage()	O	O	O	O					
void getDefaultCharSet()	O	O	O	O					

11.4 Status related methods

Support printer / Method	WP-T612	WP-T810	WP-K835	WP-K837					
Status related									
boolean is_connected()	O	O	O	O					
boolean is_DrawerOpen ()	O	O	O	O					
boolean is_PrinterOffline ()	O	O	O	O					
boolean is_PrinterWaitforRecover ()	O	O	O	O					
boolean is_PaperFedByButton ()	O	O	O	O					
boolean is_CoverOpen ()	O	O	O	O					
boolean is_StopByPaperEnd ()	O	O	O	O					
boolean is_ErrorOccurred ()	O	O	O	O					
boolean is_AutoCutterError ()	X	O	O	O					
boolean is_AutoRecoverableError ()	O	O	O	O					
boolean is_RollPaperNearEnd()	O	O	O	O					
boolean is_PaperNotPresent ()	O	O	O	O					
boolean is_AnyErrors ()	O	O	O	O					
String get_SDK_Version()	O	O	O	O					
void automateStatusBack(final boolean isEnable)	O	O	O	O					
void getBatteryStatus()	X	X	X	X					
void getPrinterId(final int idType)	O	O	O	O					
void getStatus()	O	O	O	O					

11.5 Support Code Page Table

Code Page	index
CODE_PAGE_437_USA	0
CODE_PAGE_KATAKANA	1
CODE_PAGE_850_MULTILINGUAL	2
CODE_PAGE_860_PORTUGUESE	3
CODE_PAGE_863_CANADIAN_FRENCH	4
CODE_PAGE_865_NORDIC	5
CODE_PAGE_851_GREEK	6
CODE_PAGE_853_TURKISH	7
CODE_PAGE_857_TURKISH	8
CODE_PAGE_737_GREEK	9
CODE_PAGE_ISO8859_7_GREEK	10
CODE_PAGE_1252_LATIN1	11
CODE_PAGE_866_CYRILLIC2	12
CODE_PAGE_852_LATIN2	13
CODE_PAGE_858_EURO	14
CODE_PAGE_KU42_THAI42	15
CODE_PAGE_TIS11_THAI11	16
CODE_PAGE_TIS13_THAI13	17
CODE_PAGE_THAI14	18
CODE_PAGE_THAI16	19
CODE_PAGE_THAI17	20
CODE_PAGE_THAI18	21
CODE_PAGE_TCVN_3_VIETNAM	22
CODE_PAGE_TCVN_3_VIETNAM_II	23
CODE_PAGE_PC720_ARABIC	24
CODE_PAGE_775_BALTIC	25

CODE_PAGE_PC855_CYRILLIC	26
CODE_PAGE_PC861_ICELAND	27
CODE_PAGE_PC862_HEBREW	28
CODE_PAGE_864_ARABIC	29
CODE_PAGE_PC869_GREEK	30
CODE_PAGE_ISO8859_2_EURO	31
CODE_PAGE_ISO8859_15_LATIN9	32
CODE_PAGE_PC1098_FARSI	33
CODE_PAGE_PC1118_LITHUANIAN	34
CODE_PAGE_PC1119_LITHUANIAN	35
CODE_PAGE_PC1125_UKRAINIAN	36
CODE_PAGE_WPC1250_EURO	37
CODE_PAGE_WPC1251_CYRILLIC	38
CODE_PAGE_WPC1253_CREEK	39
CODE_PAGE_WPC1254_TURKISH	40
CODE_PAGE_WPC1255_HEBREW	41
CODE_PAGE_WPC1256_ARABIC	42
CODE_PAGE_WPC1257_BALTIC	43
CODE_PAGE_WPC1258_VIETNAMESE	44
CODE_PAGE_KZ1048_KAZAKHSTAN	45

11.6 Internaltion Character Set

index	Country
0	USA
1	FRANCE
2	GERMANY
3	UK
4	DENMARK_I
5	SWEDEN
6	ITALY
7	SPAIN_I
8	JAPAN
9	NORWAY
10	DENMARK_II
11	SPAIN_II
12	LATIN_AMERICAN
13	KOREA

11.7 Text Font Attribute

Font Attribute	Value
TEXT_ATTRIBUTE_FONT_A	0
TEXT_ATTRIBUTE_FONT_B	1
TEXT_ATTRIBUTE_FONT_C	2
TEXT_ATTRIBUTE_UNDERLINE1	4
TEXT_ATTRIBUTE_UNDERLINE2	8
TEXT_ATTRIBUTE_EMPHASIZED	16
TEXT_ATTRIBUTE_REVERSE	32
TEXT_ATTRIBUTE_REVERSE_ORDER	64

11.8 Text Size Attribute

Horizontal Size Attribute	Value
TEXT_SIZE_HORIZONTAL1	0
TEXT_SIZE_HORIZONTAL2	16
TEXT_SIZE_HORIZONTAL3	32
TEXT_SIZE_HORIZONTAL4	48
TEXT_SIZE_HORIZONTAL5	64
TEXT_SIZE_HORIZONTAL6	80
TEXT_SIZE_HORIZONTAL7	96
TEXT_SIZE_HORIZONTAL8	112

Vertical Size Attribute	Value
TEXT_SIZE_VERTICAL1	0
TEXT_SIZE_VERTICAL2	1
TEXT_SIZE_VERTICAL3	2
TEXT_SIZE_VERTICAL4	3
TEXT_SIZE_VERTICAL5	4
TEXT_SIZE_VERTICAL6	5
TEXT_SIZE_VERTICAL7	6
TEXT_SIZE_VERTICAL8	7

11.9 Bar Code Type

Bar Code Type	Value
BAR_CODE_UPC_A	65
BAR_CODE_UPC_E	66
BAR_CODE_EAN13	67
BAR_CODE_EAN8	68
BAR_CODE_CODE39	69
BAR_CODE_ITF	70
BAR_CODE_CODABAR	71
BAR_CODE_CODE93	72
BAR_CODE_CODE128	73

11.10 Bar Code Correction Level

Bar Code Correction Level	Value
QR_CODE_ERROR_CORRECTION_LEVEL_L	48
QR_CODE_ERROR_CORRECTION_LEVEL_M	49
QR_CODE_ERROR_CORRECTION_LEVEL_Q	50
QR_CODE_ERROR_CORRECTION_LEVEL_H	51

11.11 Printer Status Define

Bar Code Correction Level	Value
QR_CODE_ERROR_CORRECTION_LEVEL_L	48
QR_CODE_ERROR_CORRECTION_LEVEL_M	49
QR_CODE_ERROR_CORRECTION_LEVEL_Q	50
QR_CODE_ERROR_CORRECTION_LEVEL_H	51

11.12 Print Status

Printer Status	Value
STATUS_NORMAL	0x00
STATUS_COVER_OPEN	0x04
STATUS_PAPER_FED	0x08
STATUS_PRINTING_STOPPED	0x20
STATUS_ERROR_OCCURRED	0x40
STATUS_PAPER_NEAR_END	0x0c
STATUS_PAPER_NOT_PRESENT	0x60
STATUS_TPH_OVER_HEATING	0x04
STATUS_SMPS_MODE	0x40
STATUS_BATTERY_LOW_VOLTAGE	0x20
STATUS_BATTERY_FULL	0x30
STATUS_BATTERY_HIGH	0x31
STATUS_BATTERY_MIDDLE	0x32
STATUS_BATTERY_LOW	0x33

11.13 Auto Back Status

Auto Back Status	Value
AUTO_STATUS_OFF_LINE	0x08
AUTO_STATUS_COVER_OPEN	0x20
AUTO_STATUS_PAPER_FED	0x40
AUTO_STATUS_UNRECOVERABLE_ERROR	0x02
AUTO_STATUS_AUTO_RECOVERABLE_ERROR	0x04
AUTO_STATUS_NO_PAPER	0x0c

11.14 Connection Type

Connection Type	Value
TYPE_USB	0
TYPE_BLUETOOTH	1
TYPE_TCP	2

12. Revision History

Date	Version	Comments
2016/6/22	1.0.2	First SDK Release
2016/6/24	1.0.3	First APP release
2016/6/26	1.0.4	Add SDK library protection flow control
2016/7/7	1.0.5	Add CodePage Demo Activity
2016/7/28	1.0.6	Support Android JerryBean MR1 (API level 17)
2016/10/18	1.0.7	Support network intetface connection (WiFi / TCPIP)
		Supoort network multi client connection
		Setting Launch mode = "singleTask" in AndroidManifest.xml
		Enhance USB connection compatability in some Android platforms
		Enhance Bluetooth connection compatability in some Android platforms
		Add new command: ReleaseUSB()
2016/12/12	1.0.8	Fixed kickoutDrawer SDK library
		Add KickoutDrawer demo in APK
		Add drawer status update
2017/5/31	1.0.9	Performance improve in StringToBitMap() function.
		Add pageDataPrint() function is SDK library
		Add demo code for repeat pring
2017/8/16	1.0.10	Modify PrintReceipt2 function to improve Qrcode compitability
2018/01/01	1.0.11	Modify Printer status handler description to get paper status
2018/06/06	1.0.12	1. Bug Fix for Android 7 or above , open network interface will cause exception at main thread in ConnectSelectActivity.java. 2. Replace library Httpclient-4.5.2.jar with Httpclient-4.5.5.jar
2018/06/22	1.0.13	1. Add new API function which support unicode THAI string printing
2018/08/02	1.0.14	1. Modify USB auto bindig function for option (boolean AutoBindUSB = false;) 2. Modify USB port listening thread with timeout function