
Android Library Manual

Bluetooth, Wi-Fi, USB, Serial

Mobile, Thermal Printer

Rev. 1.111

CONTENTS

1. Instruction.
2. Method.

1. Instruction

This Android Port Package Manual describes the method which is connecting using Bluetooth or Wi-Fi.

2. Method

Defined in the BluetoothPort Class and the WiFiPort Class and USBPort Class.

2.1 Bluetooth

2.1.0 GetInstance

public static BluetoothPort getInstance()

Returns an instance of the BluetoothPort.

2.1.1 Connect

public void connect(String address) throws IOException

public void connectSecure(String address) throws IOException

Connect the interface to the destination address.

- address

the target host address to connect to.

Address format is 12 characters with delimiter.

(Example: 12:34:56:78:90:AB)

2.1.2 Connect

public void connect(BluetoothDevice device) throws IOException

public void connectSecure(BluetoothDevice device) throws IOException

Connect the interface to the specific bluetooth device.

- device

Bluetooth device object to connect to.

2.1.3 Disconnect

public void disconnect() throws IOException

public void disconnect(int waitTimeout) throws IOException

Disconnect the current connection.

- waitTimeout

Waiting time for disconnect. [Milliseconds].

2.1.4 GetInputStream

public InputStream getInputStream() throws IOException

Returns an input stream for this connection.

2.1.5 GetOutputStream

public OutputStream getOutputStream() throws IOException

Returns an output stream for this connection.

2.1.6 isConnected

public boolean isConnected()

Returns a status that the interface were connected (true or false).

2.1.7 SetMacFilter

public void SetMacFilter(Boolean filter)

Set Bluetooth Mac Address Filtering Option.

- filter

 true : using Filtering(Default is true)

 false : not using Filtering

2.2 Wi-Fi

2.2.0 GetInstance

public static WiFiPort getInstance()

Returns an instance of the WiFiPort.

2.2.1 Connect

public void connect(String address) throws IOException

Connect the interface to the destination address.

- address

 the target host address to connect to.

 (Example: 192.168.0.192)

2.2.2 Connect

public void connect(String ip, int port) throws IOException

Connect the interface to the specific IP address and Port number

- ip

 the target host IP address to connect to.

- port

 the port on the target host to connect to.

2.2.3 Disconnect

public void disconnect() throws IOException

public void disconnect(int waitTimeout) throws IOException

Disconnect the current connection.

- waitTimeout

Waiting time for disconnect. [Milliseconds].

2.2.4 GetInputStream

public InputStream getInputStream() throws IOException

Returns an input stream for this connection.

2.2.5 GetOutputStream

public OutputStream getOutputStream() throws IOException

Returns an output stream for this connection.

2.2.6 isConnected

public boolean isConnected()

Returns a status that the interface were connected (true or false).

2.3 RequestHandler

2.3.1 How to use RequestHandler

RequestHandler is implemented Runnable Interface.

If connection established with BluetoothPort or WiFiPort, RequestHandler Thread must executed like below. **(Except for using USBPortConnection and WiFiPortConnection)**

It must be running until connection closed.

When the connection close, Thread must terminated by using Interrupt().

disconnection routine or OnDestroy() that whole Application terminated part.

If you want more the details, see the sample code in the SDK.

```

Thread handlerThread;
.....
// Execution routine.
handlerThread = new Thread(new RequestHandler());
handlerThread.run();
.....

// Termination routine. To Avoid NullPointerException.
if ( ( handlerThread != null ) && ( handlerThread.isAlive() ) )
    handlerThread.interrupt();
.....

```

*** If you connect to the printer by using USB, WiFiPortConnection Class, you should not to run RequestHandler.**

2.4 USB

This class support to connecting to the printer by USB OTG(USB On The Go).

You must have a USB OTG cable though, a regular micro USB to USB adapter will not work.

Also your Android device support OTG function.

2.4.1 USBPort Class

public USBPort(UsbManager usbManager)

- Constructor method.

[Parameter]

- UsbManager

public USBPortConnection connect_device(int model)

- Connect to UsbDevice Object.

[Parameter]

- model : Printer Model. (*POS_PRINTER* and *MOBILE_PRINTER* of USBPort Class.)

[Return]

- USBPortConnection : USBPortConnection Object.

public int searchPrinterType()

- Search the type of connected printer.

[Parameter]

- None

[Return]

- POS_PRINTER(0)
LABEL_PRINTER(1)
MOBILE_PRINTER(2)
MECHA_POS_PRINTER(3)

2.4.2 USBPortConnection Class

USBPortConnection Object will be created when USB connection established.

This Object used for parameter of Constructor for ESCPOSPrinter, CPCLPrinter, AndroidMSR Class.

public void close()

- Close the connection.

2.5 WiFiPortConnection

This class suport to connecting multiple via WiFi interface.

WiFiPortConnection's instance created by open() method of WiFiPort class.

2.5.1 WiFiPort Class

public static WiFiPort getInstance()

- Returns an instance of the WiFiPort.

public WiFiPortConnection open(String address)

- Returns an instance of the WiFiPortConnection.

[Parameter]

- address : the target host IP address to connect to
(Example: 192.168.0.192)

2.5.2 WiFiPortConnection Class

public void close()

- Close the connection.

public boolean isConnected()

- Returns a status that the interface were connected (true or false).

2.6 Serial

2.6.1 Connect

public long connect(String port, int baudrate) throws IOException

Connect to the interface via port and baudrate.

[Parameter]

- port : Using Port path
- baudrate : The value set in the printer baudrate

[Return]

- 0 : Success
- 2 : failed open port

2.6.2 Disconnect

public void disconnect() throws IOException

Disconnect the current connection.

2.6.3 GetInputStream

public InputStream getInputStream() throws IOException

Returns an input stream for this connection.

2.6.4 GetOutputStream

public OutputStream getOutputStream() throws IOException

Returns an output stream for this connection.

2.6.5 isConnected

public boolean isConnected()

Returns a status that the interface were connected (true or false).