DPU-S245, DPU-S445,
RP-D10, RP-E10
Print Class Library for Android™
Application Programmer's Guide

U00128560408

Seiko Instruments Inc.

DPU-S245, DPU-S445, RP-D10, RP-E10 Print Class Library for Android
Application Programmer's Guide

U00128560400	August 2012
U00128560401	November 2012
U00128560402	June 2013
U00128560403	November 2013
U00128560404	June 2014
U00128560405	January 2015
U00128560406	April 2015
U00128560407	June 2015
U00128560408	February 2016

Copyright © 2012-2016 by Seiko Instruments Inc.
All rights reserved.

Android™ is a trademark of Google Inc.
Bluetooth® is registered trademark of Bluetooth SIG, Inc.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.
Other names may be trademarks of their respective owners.

Eclipse is a trademark of Eclipse Foundation, Inc.

Seiko Instruments Inc. (hereinafter referred to as ”SII”) has prepared this manual for use by SII personnel, licensees, and customers. The information contained herein is the property of SII and shall not be reproduced in whole or in part without the prior written approval of SII.

Introduction

This document describes the Print Class Library for Android SDK (hereinafter referred to as "the SDK") for the printers provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

Target Printers

This section lists the printers supported by the SDK.

	Description in This Manual	Interface	Printer
Mobile printer	DPU-S245	Bluetooth	DPU-S245-01A-E
			DPU-S245-01B-E
		USB	DPU-S245-0xA-E
			DPU-S245-0xB-E
	DPU-S445	Bluetooth	DPU-S445-01A-E
			DPU-S445-01B-E
		USB	DPU-S445-0xA-E
			DPU-S445-0xB-E
POS printer	RP-D10	Bluetooth	RP-D10-x27J2-B
		USB	RP-D10-x27J1-U
		Ethernet	RP-D10-x27J1-E
	RP-E10	Bluetooth	RP-E10-x3FJ2-B
			RP-E11-x3FJ2-B
		USB	RP-E10-x3FJ1-U
			RP-E11-x3FJ1-U
		Ethernet	RP-E10-x3FJ1-E
			RP-E11-x3FJ1-E

Use main firmware version 1.05 or later, LAN interface firmware version 1.13.01 or later for RP-D10-x27J1-E.

Use main firmware version 1.11 or later, LAN interface firmware version 1.13.01 or later for RP-E10-x3FJ1-E or RP-E11-x3FJ1-E.

Table of Contents

Chapter 1
Product Overview

This chapter describes the product overview of the SDK.

1.1 Function Provided by the SDK

The SII print class library included in the SDK provides Android-enabled applications with the functions to use following SII printers: DPU-S245, DPU-S445 (hereinafter referred to as "Mobile printer" or “Mobile”), RP-D10, RP-E10 (hereinafter referred to as "POS printer" or “POS”).

Moreover, the SDK includes Java Eclipse projects for Android device as a sample.

1.2 SII Print Class Library Overview

1.2.1 SII Print Class Library Configuration

The SII print class library (hereinafter referred to as "the library") and the sample programs included in the SDK are located in the section surrounded by dashed lines in the Android OS configuration diagram (Figure 1-1). The library consists of two classes: the class which produces printer commands, and the class which controls communication port.

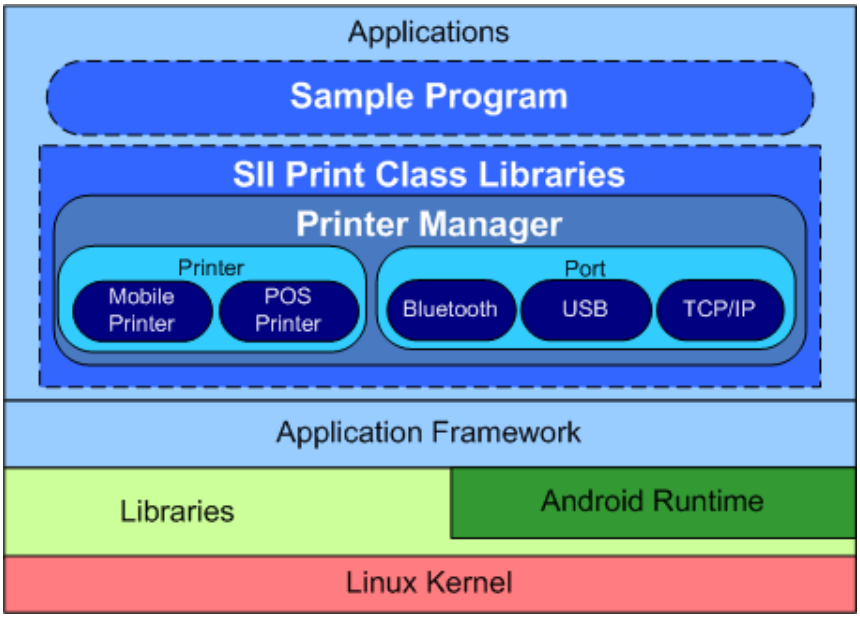

Figure 1-1

1.2.2 Function Provided by the Library

By using the library, applications can easily transmit commands and data to a printer through communication port (Bluetooth, USB, or TCP/IP) on an Android device. Also, applications can retrieve printer status.

The library provides the following functions.

- Connection/disconnection to/from a printer
- Sending data to a printer (print data or commands[*1])
- Barcode print and 2-dimensional barcode print
- Sending a data file to a printer (print data and commands[*1])
- Cut paper
- Obtaining printer status
- Aborting the waiting state of a printer
- Obtaining various responses from a printer
- Printer search by Bluetooth or TCP/IP

(NOTE) *1: Commands that retrieves the response from the printer are not available. In order to obtain responses from a printer, use "Obtaining printer status" or "Obtaining various responses from a printer".

Chapter 2
Product Specification

This chapter describes the product specification of the library.

2.1 Product Specification

2.1.1 Applicable OS versions

Applicable OS versions for the library are shown below.

Bluetooth : Android 2.3.3 (API 10) to Android 6.0 (API 23)

USB, TCP/IP : Android 3.1 (API 12) to Android 6.0 (API 23)

2.1.2 Operating Conditions

This section describes the operating conditions for the library in Table 2-1, Table 2-2, and Table 2-3. Set the Function Setting/Function Selection to the values shown in each table before using the library. See the technical reference of each printer for details about Function Setting/Function Selection.

Table 2-1 Function Setting of the DPU-S245/DPU-S445 When Using Bluetooth Connection

SWDIP	Function	Value	Setting
2-1	Data Input Mode selection	1	Bluetooth/USB
2-2		1	
4-6	Busy Output When Error Occurs	0	Unbusy
4-8	Bluetooth Link Key Selection*1	0/1	Enable/Disable

*1 When connecting with the printer in secure mode, set 1.
When connecting with the printer in insecure mode, set 0.

Table 2-2 Function Setting of the DPU-S245/DPU-S445 When Using USB Connection

SWDIP	Function	Value	Setting
4-6	Busy Output When Error Occurs	0	Unbusy

Table 2-3 Function Setting When Using RP-D10/RP-E10

MS	Function	Value	Setting
5-2	Initialized Response Selection	0	Enable

2.1.3 Precaution

When using TCP/IP connection in this library, the communication port cannot be shared with printer drivers or other libraries.

When using Bluetooth connection in this library, Bluetooth connection needs to be established by SPP (Serial Port Profile).

When using USB connection in this library, Android device needs to support USB host function.

When using TCP/IP connection in this library, wireless LAN access point connected to Android device and POS printer need to be connected to the same network.

Chapter 3
How to Use the Library

This chapter describes development environment for Android application and how to use the library.

3.1 Development Environment for Android Application

In order to develop Android applications, following tools are required. See each of the following URLs for more details.

- Java Development Kit (JDK) 6 (Using JDK 6 is recommended.)

  http://www.oracle.com/technetwork/java/javase/downloads/index.html

- Eclipse Integrated Development Environment (IDE)

  http://www.eclipse.org/downloads/

- Android SDK

  http://developer.android.com/sdk/index.html

- Android Development Toolkit (ADT) for Eclipse

  http://developer.android.com/tools/sdk/eclipse-adt.html

- USB driver for Windows

  http://developer.android.com/tools/extras/oem-usb.html

After this chapter in this reference, it is assumed that there is an environment where you can use each tool.

3.2 Use Developed Application on Android Device

In order to use developed Android applications on the Android device (actual device), configure following settings on the Android device.

(NOTE)	**This procedure is based on the menus of Android 4.0. Menu contents may vary depending on the using Android.**

(a) Select [Settings], [Security], and [Unknown sources].(Figure 3-1)

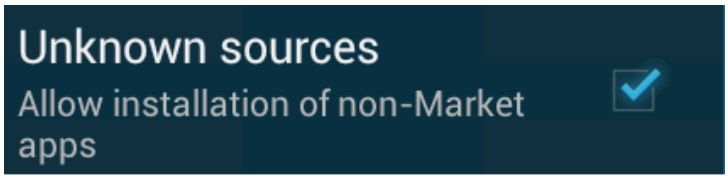

Figure 3-1

(b) Select [Settings], [Developer options], and [USB debugging].(Figure 3-2)

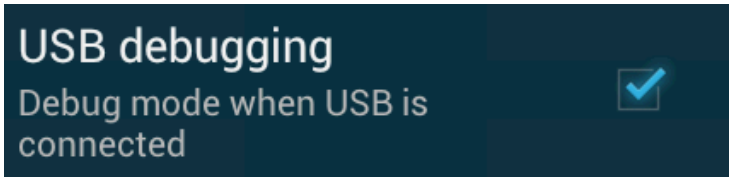

Figure 3-2

3.3 Provided Files

The file configuration of the SDK varies depending on the form it is provided.
The file configuration in CD-ROM is shown in Figure 3-3.
The file configuration in zip format is shown in Figure 3-4.

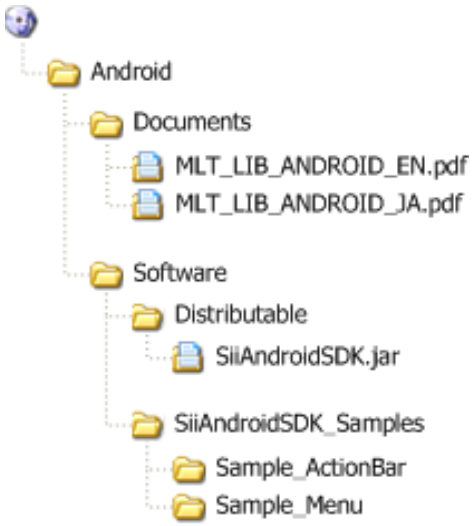

Figure 3-3

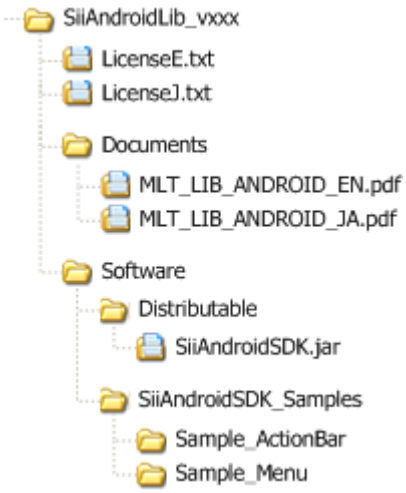

Figure 3-4

The file format of the library is JAR. The file name of the library is **SiiAndroidSDK.jar**.

3.4 Build the Library into Projects

3.4.1 Build the Library into Java Eclipse Projects

Taking the project of the sample program (Sample_ActionBar) included in the SDK as an example, this section describes how to build the library into projects.

See "Chapter 5 Sample Program" for sample programs included in the SDK.

(a) Click [File], [New], and [Folder] to make "libs" folder in the project which is displayed in the [Package Explorer] view in Eclipse.

(b) Copy the library file (SiiAndroidSDK.jar) to the folder (Sample_ActionBar\libs) created in step (a). When you refresh, it looks like Figure 3-5.

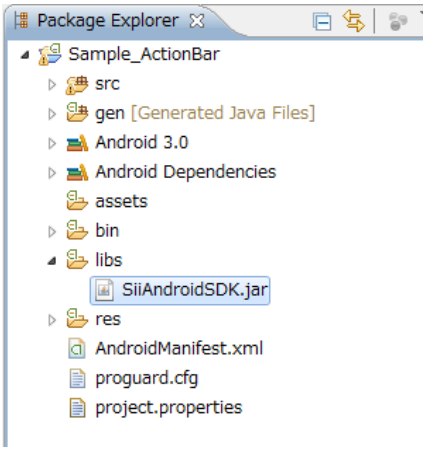

Figure 3-5

(c) Right click the project folder (Sample_ActionBar), and open [Properties] from the menu. Next, click [Java build Path] and display [Libraries] tab. Click [Add JARs] button and select the library deployed in step (b).（Figure 3-6）

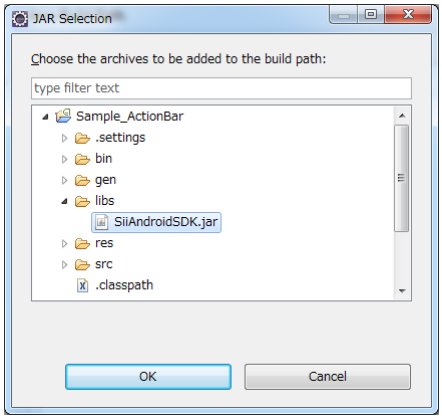

Figure 3-6

(d) After adding the library, it looks like Figure 3-7.

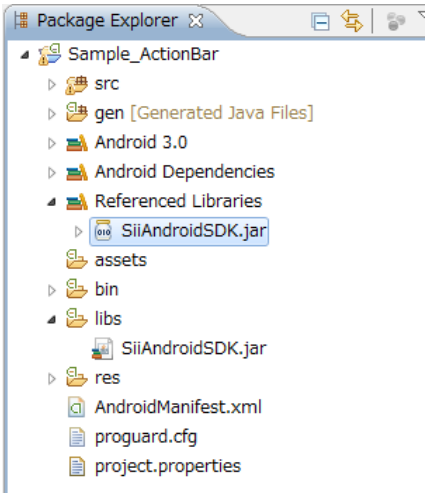

Figure 3-7

(e) Add the following codes at the beginning of the main source file. (In Sample_ActionBar, add these lines at the beginning of the MainActivity.java.)

```
import com.seikoinstruments.sdk.thermalprinter.PrinterEvent;
import com.seikoinstruments.sdk.thermalprinter.PrinterException;
import com.seikoinstruments.sdk.thermalprinter.PrinterInfo;
import com.seikoinstruments.sdk.thermalprinter.PrinterListener;
import com.seikoinstruments.sdk.thermalprinter.PrinterManager;
```

By completing these procedures, functions of the library become available.

Chapter 4
Function of the Library

This chapter describes the APIs for each class implemented in the library.

4.1 Overview of the Library

The library provides Android-enabled applications with the functions to use SII printers.
See Table 4-1 Method of **PrinterManager** Class for details about provided functions.

4.2 Structure of the Library

The file format of the library is JAR. The file name of the library is SiiAndroidSDK.jar.

In order to use the library in Android applications, build the library into projects of applications.

See "Chapter 3 How to Use the Library" for details about how to build the library into projects.

4.3 Package of the Library

The package of the library is com.seikoinstruments.sdk.thermalprinter.
com.seikoinstruments.sdk.thermalprinter includes following classes.

Class Name	Description
PrinterManager	Class that provides the API used for communication with the printer and printing.
PrinterEvent	Class that provides API to obtain the event type proceeded when printer searching is completed.
PrinterListener	Interface that obtains the complete event of printer searching.
PrinterInfo	Class that stores the printer information searched by printer searching method.
PrinterException	Exception class that is thrown at API call.

See "4.4 API Reference" for more details about APIs for each class.
Furthermore, add the following codes at the beginning of the main code to use classes above.

```
import com.seikoinstruments.sdk.thermalprinter.PrinterEvent;
import com.seikoinstruments.sdk.thermalprinter.PrinterException;
import com.seikoinstruments.sdk.thermalprinter.PrinterInfo;
import com.seikoinstruments.sdk.thermalprinter.PrinterListener;
import com.seikoinstruments.sdk.thermalprinter.PrinterManager;
```

4.4 API Reference

This manual describes API in each class contained in this library as follows.

Class Name	Description
PrinterManager	See "4.4.1 **PrinterManager** Class".
PrinterEvent	See "4.4.2 **PrinterEvent** Class".
PrinterListener	See "4.4.3 **PrinterListener** Interface".
PrinterInfo	See "4.4.4 **PrinterInfo** Class".
PrinterException	See "4.4.5 **PrinterException** Class".

4.4.1 PrinterManager Class

(1) Method List

Methods provided by **PrinterManager** class are shown in Table 4-1. Available methods vary depending on the target printer: Mobile printer or POS printer.

Table 4-1 Method of PrinterManager Class

Method	Function Summary	Target	
		Mobile	POS
PrinterManager	Constructor	Supported	Supported
connect	Start communicating with a printer (Bluetooth)	Supported	Supported
connect	Start communicating with a printer (USB)	Supported	Supported
connect	Start communicating with a printer (TCP/IP)	Not supported	Supported
disconnect	Disconnect a printer	Supported	Supported
sendText	Send text data	Supported	Supported
sendTextEx	Send format specified text data	Supported	Not supported
sendTextEx	Send format specified text data	Not supported	Supported
printBarcode	Print barcode	Supported	Supported
printPDF417	Print PDF417	Supported	Supported
printQRcode	Print QR code	Supported	Supported
cutPaper	Cut paper	Not supported	Supported
openDrawer	Open cash drawer	Not supported	Supported
buzzer	Sound buzzer	Not supported	Supported
sendBinary	Send binary data	Supported	Supported
sendDataFile	Send specified file	Supported*	Supported*
getStatus	Obtain printer status	Supported	Supported
abort	Abort the waiting state of a printer	Supported	Supported
registerLogo	Register logo (image) to a printer	Supported	Not supported
registerLogo	Register logo (image) to a printer	Not supported	Supported
printLogo	Print specified logo (image) in printer	Supported	Not supported
printLogo	Print specified logo (image) in printer	Not supported	Supported

Method	Function Summary	Target	
		Mobile	POS
unregisterLogo	Delete specified logo (image) in a printer	Supported	Not Supported
unregisterLogo	Delete specified logo (image) in a printer	Not supported	Supported
registerStyleSheet	Register style sheet to a printer	Not supported	Supported
unregisterStyleSheet	Delete specified style sheet in a printer	Not supported	Supported
resetPrinter	Printer hardware reset	Supported	Supported
getPrinterResponse	Obtain various responses from a printer	Supported*	Supported*
startDiscoveryPrinter	Start printer search (Bluetooth)	Supported	Supported
startDiscoveryPrinter	Start printer search (TCP/IP)	Not supported	Supported
cancelDiscoveryPrinter	Cancel printer search	Supported	Supported
getFoundPrinter	Obtain searched printer information	Supported	Supported
getSendTimeout	Obtain send timeout period	Supported	Supported
setSendTimeout	Set send timeout period	Supported	Supported
getReceiveTimeout	Obtain receive timeout period	Supported	Supported
setReceiveTimeout	Set receive timeout period	Supported	Supported
getInternationalCharacter	Obtain international character set	Supported	Supported
setInternationalCharacter	Set international character set	Supported	Supported
getCodePage	Obtain codepage	Supported	Supported
setCodePage	Set codepage	Supported	Supported
getPrinterModel	Obtain printer model	Supported	Supported
getPortType	Obtain connection port type	Supported	Supported
isConnect	Verify connection state with a printer	Supported	Supported
getSocketKeepingTime	Obtain socket keeping time	Not supported	Supported
setSocketKeepingTime	Set socket keeping time	Not supported	Supported

*: Provided function varies depending on the target printer.

(2) Constant List

(a) International Character Set Constant

Constants used when setting/obtaining international character set are shown in Table 4-2.

Table 4-2 International Character Set Constant

Constant Name	Description	Value	Target	
			Mobile	POS
COUNTRY_USA	USA	0	Supported	Supported
COUNTRY_FRANCE	France	1	Supported	Supported
COUNTRY_GERMANY	Germany	2	Supported	Supported
COUNTRY_ENGLAND	United Kingdom	3	Supported	Supported
COUNTRY_DENMARK_1	Denmark I	4	Supported	Supported
COUNTRY_SWEDEN	Sweden	5	Supported	Supported
COUNTRY_ITALY	Italy	6	Supported	Supported
COUNTRY_SPAIN	Spain	7	Supported	Supported
COUNTRY_JAPAN	Japan	8	Supported	Supported
COUNTRY_NORWAY	Norway	9	Supported	Supported
COUNTRY_DENMARK_2	Denmark II	10	Supported	Supported
COUNTRY_SPAIN_2	Spain II	11	Supported	Supported
COUNTRY_LATIN_AMERICA	Latin America	12	Supported	Supported
COUNTRY_ARABIA	Arabia	17	Not supported	Supported

(b) Codepage Constant

Constants used when setting/ obtaining codepage are shown in Table 4-3.

Table 4-3 Codepage Constant

Constant Name	Description	Value	Target	
			Mobile	POS
CODE_PAGE_KATAKANA	Katakana character set	1	Supported	Supported
CODE_PAGE_1252	Codepage 1252 (Latin)	16	Supported	Supported
CODE_PAGE_864[*1]	Codepage 864 (Arabic)	37	Not supported	Supported
CODE_PAGE_1250	Codepage 1250 (Central European)	45	Not supported	Supported
CODE_PAGE_1251	Codepage 1251 (Cyrillic)	46	Not supported	Supported
CODE_PAGE_1253	Codepage 1253 (Greek)	47	Not supported	Supported
CODE_PAGE_1254	Codepage 1254 (Turkish)	48	Not supported	Supported

Note: This library does not support other than above codepage.
*1: When **CODE_PAGE_864** is specified, Euro symbol is not printed.

(c) Constant for barcode or PDF417

Constants used for printing barcode or PDF417 are shown in Table 4-4.

Table 4-4 Constant for Barcode or PDF417

Constant Name	Description	Value	Target	
			Mobile	POS
BARCODE_HEIGHT_DEFAULT	Default value of barcode height	162	Supported	Supported
PDF417_MODULE_HEIGHT_DEFAULT	Default value of PDF417 height	10	Supported	Supported
PDF417_ROW_AUTO	Automatic selection of the number of rows	0	Supported	Supported
PDF417_COLUMN_AUTO	Automatic selection of the number of columns	0	Supported	Supported

(d) Printer Model Constant

Constants used when obtaining printer model are shown in Table 4-5.

Table 4-5 Printer Model Constant

Constant Name	Description	Value	Target	
			Mobile	POS
PRINTER_MODEL_DPU_S245	DPU-S245	284	Supported	Not supported
PRINTER_MODEL_DPU_S445	DPU-S445	281	Supported	Not supported
PRINTER_MODEL_RP_D10	RP-D10	295	Not supported	Supported
PRINTER_MODEL_RP_E10	RP-E10	291	Not supported	Supported
PRINTER_MODEL_DEFAULT	Printer model default	284	Supported	Supported

(e) Port Type Constant

Constants used when obtaining connection port type are shown in Table 4-6.

Table 4-6 Port Type Constant

Constant Name	Description	Value	Target	
			Mobile	POS
PRINTER_TYPE_BLUETOOTH	Bluetooth	0	Supported	Supported
PRINTER_TYPE_USB	USB	1	Supported	Supported
PRINTER_TYPE_TCP	TCP/IP	2	Not supported	Supported

(f) Response Type Constant

Constants used when obtaining various responses from a printer are shown in Table 4-7.

Table 4-7 Response Type Constant

Constant Name	Description	Value	Target	
			Mobile	POS
PRINTER_RESPONSE_ REQUEST	Execution response request	0	Supported	Supported
PRINTER_RESPONSE_ USER_AREA	Remaining user area response	1	Supported	Supported
PRINTER_RESPONSE_ ARRANGE_USER_AREA	Remaining user area after defragment response	2	Not supported	Supported
PRINTER_RESPONSE_NV_ GRAPHICS	NV graphics memory capacity response	3	Not supported	Supported
PRINTER_RESPONSE_ KEY_CODE	The key code list defined NV graphics	4	Not supported	Supported
PRINTER_RESPONSE_ BATTERY_STATUS	Battery voltage status	5	Supported	Not supported
PRINTER_RESPONSE_ EXTERNAL_RAM	Remaining RAM capacity response	6	Supported	Not supported

(3) Constant List of Enumerated Type

(a) Bold print (**CharacterBold**)

Enumerated type constants used for bold print are shown in Table 4-8.

Table 4-8 Bold Print (CharacterBold)

Constant Name	Description	Target	
		Mobile	POS
BOLD_CANCEL	Release bold print	Supported	Supported
BOLD	Specify bold print	Supported	Supported

(b) Underline (**CharacterUnderline**)

Enumerated type constants used for underline are shown in Table 4-9.

Table 4-9 Underline (CharacterUnderline)

Constant Name	Description	Target	
		Mobile	POS
UNDERLINE_CANCEL	Release underline print.	Supported	Supported
UNDERLINE_1	Specify 1 dot width underline print	Supported	Supported
UNDERLINE_2	Specify 2 dots width underline print	Supported	Supported

(c) Reverse print (**CharacterReverse**)

Enumerated type constants used for reverse print are shown in Table 4-10.

Table 4-10 Reverse Print (CharacterReverse)

Constant Name	Description	Target	
		Mobile	POS
REVERSE_CANCEL	Release reverse print.	Not supported	Supported
REVERSE	Specify reverse print	Not supported	Supported

(d) Character font (**CharacterFont**)

Enumerated type constants used for Character font are shown in Table 4-11.

Table 4-11 Character Font (CharacterFont)

Constant Name	Description	Target	
		Mobile	POS
FONT_A	Font A (24×12)	Supported	Supported
FONT_B	Font A (16×8)	Supported	Supported

(e) Character Scale (**CharacterScale**)

Enumerated type constants used for character scale are shown in Table 4-12.

Table 4-12 Character Scale (CharacterScale)

Constant Name	Description	Target	
		Mobile	Mobile
VARTICAL_1_HORIZONTAL_1	Height × 1 and width × 1	Supported	Supported
VARTICAL_1_HORIZONTAL_2	Height × 1 and width × 2	Supported	Supported
VARTICAL_1_HORIZONTAL_3	Height × 1 and width × 3	Not supported	Supported
VARTICAL_1_HORIZONTAL_4	Height × 1 and width × 4	Not supported	Supported
VARTICAL_2_HORIZONTAL_1	Height × 2 and width × 1	Supported	Supported
VARTICAL_2_HORIZONTAL_2	Height × 2 and width × 2	Supported	Supported
VARTICAL_2_HORIZONTAL_3	Height × 2 and width × 3	Not supported	Supported
VARTICAL_2_HORIZONTAL_4	Height × 2 and width × 4	Not supported	Supported
VARTICAL_2_HORIZONTAL_6	Height × 2 and width × 6	Not supported	Supported
VARTICAL_3_HORIZONTAL_1	Height × 3 and width × 1	Not supported	Supported
VARTICAL_3_HORIZONTAL_2	Height × 3 and width × 2	Not supported	Supported
VARTICAL_3_HORIZONTAL_3	Height × 3 and width × 3	Not supported	Supported
VARTICAL_3_HORIZONTAL_4	Height × 3 and width × 4	Not supported	Supported
VARTICAL_4_HORIZONTAL_1	Height × 4 and width × 1	Not supported	Supported
VARTICAL_4_HORIZONTAL_2	Height × 4 and width × 2	Not supported	Supported
VARTICAL_4_HORIZONTAL_3	Height × 4 and width × 3	Not supported	Supported
VARTICAL_4_HORIZONTAL_4	Height × 4 and width × 4	Not supported	Supported
VARTICAL_4_HORIZONTAL_6	Height × 4 and width × 6	Not supported	Supported
VARTICAL_4_HORIZONTAL_8	Height × 4 and width × 8	Not supported	Supported
VARTICAL_6_HORIZONTAL_2	Height × 6 and width × 2	Not supported	Supported
VARTICAL_6_HORIZONTAL_4	Height × 6 and width × 4	Not supported	Supported
VARTICAL_6_HORIZONTAL_6	Height × 6 and width × 6	Not supported	Supported
VARTICAL_6_HORIZONTAL_8	Height × 6 and width × 8	Not supported	Supported
VARTICAL_8_HORIZONTAL_4	Height × 8 and width × 4	Not supported	Supported
VARTICAL_8_HORIZONTAL_6	Height × 8 and width × 6	Not supported	Supported
VARTICAL_8_HORIZONTAL_8	Height × 8 and width × 8	Not supported	Supported

(f) Alignment (**PrintAlignment**)

Enumerated type constants used for alignment are shown in Table 4-13.

Table 4-13 Alignment (PrintAlignment)

Constant Name	Description	Target	
		Mobile	POS
ALIGNMENT_LEFT	Align left	Supported	Supported
ALIGNMENT_CENTER	Centered	Supported	Supported
ALIGNMENT_RIGHT	Align right	Supported	Supported

(g) Barcode symbol (**BarcodeSymbol**)

Enumerated type constants used for barcode symbol are shown in Table 4-14.

Table 4-14 Barcode Symbol (BarcodeSymbol)

Constant Name	Description	Type	Target	
			Mobile	POS
BARCODE_SYMBOL_UPC_A	UPC-A	(a)	Supported	Supported
BARCODE_SYMBOL_UPC_E	UPC-E	(a)	Supported	Supported
BARCODE_SYMBOL_EAN13	EAN13	(a)	Supported	Supported
BARCODE_SYMBOL_JAN13	JAN13	(a)	Supported	Supported
BARCODE_SYMBOL_EAN8	EAN8	(a)	Supported	Supported
BARCODE_SYMBOL_JAN8	JAN8	(a)	Supported	Supported
BARCODE_SYMBOL_CODE39	CODE39	(a), (b)	Supported	Supported
BARCODE_SYMBOL_CODE93	CODE93	(c)	Not supported	Supported
BARCODE_SYMBOL_CODE128	CODE128	(c)	Supported	Supported
BARCODE_SYMBOL_ITF	ITF	(a), (b)	Supported	Supported
BARCODE_SYMBOL_CODABAR	CODABAR	(a), (b)	Supported	Supported
BARCODE_SYMBOL_EAN13_ADDON	EAN13 add-on	(a)	Not supported	Supported
BARCODE_SYMBOL_JAN13_ADDON	JAN13 add-on	(a)	Not supported	Supported

For the type, see "**printBarcode**" in 4.4.1 (1) Method List.

(h) Module size (**ModuleSize**)

Enumerated type constants used for module size are shown in Table 4-15.

Table 4-15 Module Size (ModuleSize)

Constant Name	Description	Using Method	Target	
			Mobile	POS
BARCODE_MODULE_WIDTH_2	Fine element 2 dots Module width 0.250 mm	**printBarcode**	Supported	Supported
BARCODE_MODULE_WIDTH_3	Fine element 3 dots Module width 0.375 mm		Supported	Supported
BARCODE_MODULE_WIDTH_4	Fine element 4 dots Module width 0.500 mm		Supported	Supported
BARCODE_MODULE_WIDTH_5	Fine element 5 dots Module width 0.625 mm		Not supported	Supported
BARCODE_MODULE_WIDTH_6	Fine element 6 dots Module width 0.750 mm		Not supported	Supported
PDF417_MODULE_WIDTH_2	Nominal fine element width 2 dots	**printPDF417**	Supported	Supported
PDF417_MODULE_WIDTH_3	Nominal fine element width 3 dots		Supported	Supported
PDF417_MODULE_WIDTH_4	Nominal fine element width 4 dots		Supported	Supported
PDF417_MODULE_WIDTH_5	Nominal fine element width 5 dots		Supported	Not supported
PDF417_MODULE_WIDTH_6	Nominal fine element width 6 dots		Supported	Not supported
PDF417_MODULE_WIDTH_7	Nominal fine element width 7 dots		Supported	Not supported
PDF417_MODULE_WIDTH_8	Nominal fine element width 8 dots		Supported	Not supported
QR_MODULE_SIZE_2	2 dots	**printQRcode**	Supported	Supported
QR_MODULE_SIZE_3	3 dots		Supported	Supported
QR_MODULE_SIZE_4	4 dots		Supported	Supported
QR_MODULE_SIZE_5	5 dots		Supported	Supported
QR_MODULE_SIZE_6	6 dots		Supported	Supported
QR_MODULE_SIZE_7	7 dots		Supported	Supported
QR_MODULE_SIZE_8	8 dots		Supported	Supported
QR_MODULE_SIZE_9	9 dots		Supported	Supported
QR_MODULE_SIZE_10	10 dots		Supported	Supported
QR_MODULE_SIZE_11	11 dots		Supported	Supported

(i) HRI character print position (**HriPosition**)

Enumerated type constants used for HRI character print position are shown in Table 4-16.

Table 4-16 HRI Character Print Position (HriPosition)

Constant Name	Description	Target	
		Mobile	POS
HRI_NONE	Do not print	Supported	Supported
HRI_POSITION_ABOVE	Above barcode	Supported	Supported
HRI_POSITION_BELOW	Below barcode	Supported	Supported
HRI_POSITION_ABOVE_ BELOW	Above and below barcode (both)	Supported	Supported

(j) N:W ratio (**NwRatio**)

Enumerated type constants used for N:W ratio are shown in Table 4-17.

Table 4-17 N:W Ratio (NwRatio)

Constant Name	Description	Target	
		Mobile	POS
WIDE_WIDTH_1	Wide width type 1	Supported	Not supported
WIDE_WIDTH_2	Wide width type 2	Supported	Not supported
WIDE_WIDTH_3	Wide width type 3	Supported	Not supported
WIDE_WIDTH_4	Wide width type 4	Supported	Not supported
NWRATIO_1TO2	1:2	Not supported	Supported
NWRATIO_1TO2_5	1:2.5	Not supported	Supported
NWRATIO_1TO3	1:3	Not supported	Supported

(k) Error correction level (**ErrorCorrection**)

Enumerated type constants used for error correction level are shown in Table 4-18.

Table 4-18 Error Correction Level (ErrorCorrection)

Constant Name	Description	Using Method	Target	
			Mobile	POS
PDF417_ERROR_ CORRECTION_0	Error correction level 0	**printPDF417**	Supported	Supported
PDF417_ERROR_ CORRECTION_1	Error correction level 1		Supported	Supported
PDF417_ERROR_ CORRECTION_2	Error correction level 2		Supported	Supported
PDF417_ERROR_ CORRECTION_3	Error correction level 3		Supported	Supported
PDF417_ERROR_ CORRECTION_4	Error correction level 4		Supported	Supported
PDF417_ERROR_ CORRECTION_5	Error correction level 5		Supported	Supported
PDF417_ERROR_ CORRECTION_6	Error correction level 6		Supported	Supported
PDF417_ERROR_ CORRECTION_7	Error correction level 7		Supported	Supported
PDF417_ERROR_ CORRECTION_8	Error correction level 8		Supported	Supported
QR_ERROR_ CORRECTION_L	Error correction level L	**printQRcode**	Supported	Supported
QR_ERROR_ CORRECTION_M	Error correction level M		Supported	Supported
QR_ERROR_ CORRECTION_H	Error correction level H		Supported	Supported
QR_ERROR_ CORRECTION_Q	Error correction level Q		Supported	Supported

(l) PDF417 symbol (**Pdf417Symbol**)

Enumerated type constants used for PDF417 symbol are shown in Table 4-19.

Table 4-19 PDF417 Symbol (Pdf417Symbol)

Constant Name	Description	Target	
		Mobile	POS
PDF417_STANDARD	Standard PDF417	Supported	Supported
PDF417_COMPACT	Compact PDF417	Supported	Supported

(m) QR code model (**QrModel**)

Enumerated type constants used for QR code model symbol are shown in Table 4-20.

Table 4-20 QR Code Model (QrModel)

Constant Name	Description	Target	
		Mobile	POS
QR_MODEL_1	QR code model 1	Supported	Supported
QR_MODEL_2	QR code model 2	Supported	Supported

(n) Cutting method (**CuttingMethod**)

Enumerated type constants used for cutting method are shown in Table 4-21.

Table 4-21 Cutting Method (CuttingMethod)

Constant Name	Description	Target	
		Mobile	POS
CUT_FULL	Full cut	Not supported	Supported
CUT_PARTIAL	Partial cut	Not supported	Supported

(o) Drawer number (**DrawerNum**)

Enumerated type constants used for drawer number are shown in Table 4-22.

Table 4-22 Drawer Number (DrawerNum)

Constant Name	Description	Target	
		Mobile	POS
DRAWER_1	Drawer 1	Not supported	Supported
DRAWER_2	Drawer 2	Not supported	Supported

(p) Activation pulse width (**PulseWidth**)

Enumerated type constants used for activation pulse width are shown in Table 4-23.

Table 4-23 Activation Pulse Width (PulseWidth)

Constant Name	Description	Target	
		Mobile	POS
ON_OFF_TIME_100	ON/OFF time 100 millisecond	Not supported	Supported
ON_OFF_TIME_200	ON/OFF time 200 millisecond	Not supported	Supported
ON_OFF_TIME_300	ON/OFF time 300 millisecond	Not supported	Supported
ON_OFF_TIME_400	ON/OFF time 400 millisecond	Not supported	Supported
ON_OFF_TIME_500	ON/OFF time 500 millisecond	Not supported	Supported
ON_OFF_TIME_600	ON/OFF time 600 millisecond	Not supported	Supported
ON_OFF_TIME_700	ON/OFF time 700 millisecond	Not supported	Supported
ON_OFF_TIME_800	ON/OFF time 800 millisecond	Not supported	Supported

(4) Method Details

PrinterManager — Constructor

Target: Mobile printer/POS printer

Syntax: public **PrinterManager**()

Description: Constructor for **com.seikoinstruments.sdk.thermalprinter.PrinterManager** class.

connect — Start communicating with a printer (Bluetooth)

Target: Mobile printer/POS printer

Syntax:

(a) public void **connect**(int *printerModel*,
String *address*) throws **PrinterException**

(b) public void **connect**(int *printerModel*,
String *address*,
boolean *secure*) throws **PrinterException**

Parameter:

printerModel	Printer model constant for Bluetooth connection	
address	Bluetooth address Example: "00:11:22:AA:BB:CC"	
secure	true	Connect to a printer in secure mode
	false	Connect to a printer in insecure mode

Description: This method starts communication between Android device and a printer through Bluetooth connection. Call this method before using other methods of this class.

This method connects to the Bluetooth address specified by *address* parameter. Also, printer default settings are executed based on *printerModel* parameter specified at the connection. See Table 4-5 Printer Model Constant for available printer model constant.

The method in syntax (a) always connects to a printer in secure mode. The method in syntax (b) connects to a printer by specifying secure mode or insecure mode based on the value of *secure* parameter. Normally, it is recommended that the connection in secure mode be used.

In order to operate a printer properly, printer settings may be changed at the connection in this method.

Error: **PrinterException** may be thrown when the method is called.

connect — Start communicating with a printer (USB)

Target	Mobile printer/POS printer
Syntax	public void **connect**(int *printerModel*, Context *context*) throws **PrinterException**
Parameter	*printerModel* Printer model constant for USB connection *context* Specify application context to call this method. Example: `MainActivity.this`
Description	This method starts communication between Android device and a printer through USB connection. Call this method before using other methods of this class. This method connects to a printer specified by *printerModel* parameter. Also, printer default settings are executed based on *printerModel* parameter specified at the connection. See Table 4-5 Printer Model Constant for available printer model constant. In order to operate a printer properly, printer settings may be changed at the connection in this method.
Error	**PrinterException** may be thrown when the method is called.

connect — Start communicating with a printer (TCP/IP)

Target	POS printer
Syntax	public void **connect**(int *printerModel*, String *address*) throws **PrinterException**
Parameter	*printerModel* Printer model constant for Ethernet connection *address* IP address Example: "192.168.0.1"
Description	This method is valid only for POS printer. This method starts communication between Android device and a printer connected to the same network through TCP/IP connection. Call this method before using other methods of this class. This method connects to the IP address specified by *address* parameter. TCP port number 9100 is used for communication. Also, printer default settings are executed based on *printerModel* parameter specified at the connection. See Table 4-5 Printer Model Constant for available printer model constant. In order to operate a printer properly, printer settings may be changed at the connection in this method. **<Introduction to creating/deleting a socket during connection with TCP/IP in the Library>** During connection with TCP/IP, the Library creates a socket when data transmission/reception is necessary, and deletes one when unnecessary. On the basis of data transmission completion time to a printer, a created socket is kept until socket keeping time elapses that is set with **setSocketKeepingTime** method. Do not connect the same printer from other Android application that uses the Library until socket keeping time elapses. After socket keeping time elapses, the socket is deleted temporarily but a socket is created again when data is transmitted by method execution.

The Library deletes socket and disconnects the printer and TCP/IP connection when executing **disconnect** method.

Error **PrinterException** may be thrown when the method is called.

disconnect — Disconnect a printer

Target Mobile printer/POS printer

Syntax public void **disconnect**() throws **PrinterException**

Description This method disconnects communication with a printer.

Error **PrinterException** may be thrown when the method is called.

sendText — Send text data

Target Mobile printer/POS printer

Syntax public void **sendText**(String *text*) throws **PrinterException**

Parameter *text* Text data sent to a printer

Description This method sends the text data specified by *text* parameter to a printer. Data size that can be specified at a time is 16K bytes (16384 bytes).

This method encodes the specified text data to printable text data based on the settings of international character set and codepage, and then sends it to a printer.

In this method, the line spacing code is not added to the end of text data.

Error **PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

sendTextEx — Send format specified text data

Target: Mobile printer

Syntax:
public void **sendTextEx**(String *text*,
CharacterBold *bold*,
CharacterUnderline *underline*,
CharacterFont *font*,
CharacterScale *scale*) throws **PrinterException**

Parameter:

text	Text data to send to a printer
bold	Bold print
underline	Underline
font	Character font
scale	Character scale

Description: This method is valid only for Mobile printer. This method encodes format specified text data to printable text data based on the settings of international character set or codepage, and then sends it to a printer. Data size that can be specified at one time is 16K bytes (16384 bytes).

See Table 4-8 Bold Print (**CharacterBold**) for available setting in *bold* parameter.

See Table 4-9 Underline (**CharacterUnderline**) for available setting in *underline* parameter.

See Table 4-11 Character Font (**CharacterFont**) for available setting in *font* parameter.

See Table 4-12 Character Scale (**CharacterScale**) for available setting in *scale* parameter.

In this method, the line spacing code is not added to the end of text data.

Error: **PrinterException** may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

sendTextEx — Send format specified text data

Target: POS printer

Syntax:
public void **sendTextEx**(String *text*,
CharacterBold *bold*,
CharacterUnderline *underline*,
CharacterReverse *reverse*,
CharacterFont *font*,
CharacterScale *scale*,
PrintAlignment *alignment*) throws **PrinterException**

Parameter:

text	Text data to send to a printer
bold	Bold print
underline	Underline
reverse	Reverse print
font	Character font
scale	Character scale
alignment	Alignment

Description:

This method is valid only for POS printer. This method encodes format specified text data to printable text data based on the settings of international character set or codepage, and then sends it to a printer. Data size that can be specified at one time is 16K bytes (16384 bytes).

See Table 4-8 Bold Print (**CharacterBold**) for available setting in *bold* parameter.

See Table 4-9 Underline (**CharacterUnderline**) for available setting in *underline* parameter.

See Table 4-10 Reverse Print (**CharacterReverse**) for available setting in *reverse* parameter.

See Table 4-11 Character Font (**CharacterFont**) for available setting in *font* parameter.

See Table 4-12 Character Scale (**CharacterScale**) for available setting in *scale* parameter.

See Table 4-13 Alignment (**PrintAlignment**) for available setting in *alignment* parameter.

In this method, the line spacing code is not added to the end of text data.

Error:

PrinterException may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

printBarcode — Print barcode

Target: Mobile printer/POS printer

Syntax:

(a) public void **printBarcode**(BarcodeSymbol *barcodeSymbol*,
String *text*,
ModuleSize *moduleWidth*,
int *moduleHeight*,
HriPosition *hriPosition*,
CharacterFont *hriFont*,
PrintAlignment *alignment*) throws **PrinterException**

(b) public void **printBarcode**(BarcodeSymbol *barcodeSymbol*,
String *text*,
ModuleSize *moduleWidth*,
int *moduleHeight*,
HriPosition *hriPosition*,
CharacterFont *hriFont*,
PrintAlignment *alignment*,
NwRatio *nwRatio*) throws **PrinterException**

(c) public void **printBarcode**(BarcodeSymbol *barcodeSymbol*,
byte[] *data*,
ModuleSize *moduleWidth*,
int *moduleHeight*,
HriPosition *hriPosition*,
CharacterFont *hriFont*,
PrintAlignment *alignment*) throws **PrinterException**

Parameter:

barcodeSymbol	Barcode symbol
text (data)	Text data to send to a printer
moduleWidth	Barcode width
moduleHeight	Barcode height
hriPosition	HRI character print position
hriFont	HRI character font
alignment	Alignment
nwRatio	N:W ratio

Description: This method executes barcode print.

See Table 4-14 Barcode Symbol (**BarcodeSymbol**) for available setting in *barcodeSymbol* parameter.

See Table 4-15 Module Size (**ModuleSize**) for available setting in *moduleWidth* parameter.

The valid range of *moduleHeight* parameter is from 1 to 255.

See Table 4-16 HRI Character Print Position (**HriPosition**) for available setting in *hriPosition* parameter.

See Table 4-11 Character Font (**CharacterFont**) for available setting in *hriFont* parameter.

See Table 4-13 Alignment (**PrintAlignment**) for available setting in *alignment* parameter.

See Table 4-17 N:W Ratio (**NwRatio**) for available setting in *nwRatio* parameter.

Depending on the relationship between *moduleWidth* parameter and *nwRatio* parameter, the wide element width is set in the following tables. See "Table 4-24 N:W Ratio for Mobile Printer" for Mobile printer, and "Table 4-25 N:W Ratio for POS Printer" for POS printer.

Table 4-24 N:W Ratio for Mobile Printer

moduleWidth	*nwRatio*			
	WIDE_WIDTH_1	**WIDE_WIDTH_2**	**WIDE_WIDTH_3**	**WIDE_WIDTH_4**
BARCODE_MODULE_WIDTH_2	0.625 mm (5 dots)	0.750 mm (6 dots)	0.750 mm (6 dots)	0.750 mm (6 dots)
BARCODE_MODULE_WIDTH_3	0.875 mm (7 dots)	1.000 mm (8 dots)	1.125 mm (9 dots)	1.125 mm (9 dots)
BARCODE_MODULE_WIDTH_4	1.125 mm (9 dots)	1.250 mm (10 dots)	1.375 mm (11 dots)	1.500 mm (12 dots)

Table 4-25 N:W Ratio for POS Printer

moduleWidth	*nwRatio*		
	NWRATIO_1TO2	**NWRATIO_1TO2_5**	**NWRATIO_1TO3**
BARCODE_MODULE_WIDTH_2	0.500 mm (4 dots)	0.625 mm (5 dots)	0.750 mm (6 dots)
BARCODE_MODULE_WIDTH_3	0.750 mm (6 dots)	1.000 mm (8 dots)	1.125 mm (9 dots)
BARCODE_MODULE_WIDTH_4	1.000 mm (8 dots)	1.250 mm (10 dots)	1.500 mm (12 dots)
BARCODE_MODULE_WIDTH_5	1.250 mm (10 dots)	1.625 mm (13 dots)	1.875 mm (15 dots)
BARCODE_MODULE_WIDTH_6	1.500 mm (12 dots)	1.875 mm (15 dots)	2.250 mm (18 dots)

Error

PrinterException may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

Target Mobile printer/POS printer

Syntax (a) public void **printPDF417**(String *text*,
ErrorCorrection *errorCorrection*,
int *row*,
int *column*,
ModuleSize *moduleWidth*,
int *moduleHeight*,
PrintAlignment *alignment*,
Pdf417Symbol *pdf417Symbol*) throws **PrinterException**

(b) public void **printPDF417**(String *text*,
ErrorCorrection *errorCorrection*,
int *row*,
int *column*,
ModuleSize *moduleWidth*,
int *moduleHeight*,
PrintAlignment *alignment*) throws **PrinterException**

Parameter

text	Barcode data to send a printer
errorCorrection	Error correction level
row	The number of row
column	The number of columns in data area
moduleWidth	Nominal fine element width
moduleHeight	Module height
alignment	Alignment
pdf417Symbol	Symbol of PDF417

Description This method prints PDF417. *pdf417Symbol* parameter for syntax (b) is fixed to standard PDF417.

See Table 4-18 Error correction level (**ErrorCorrection**) for available setting in *errorCorrection* parameter.

The valid range of *row* parameter is from 0 to 90. When 0 is specified, the number of row is automatically set.

The valid range of *column* parameter is from 0 to 30. When 0 is specified, the number of column in the data area is automatically set.

See Table 4-15 Module Size (**ModuleSize**) for available setting in *moduleWidth* parameter.

The valid range of *moduleHeight* parameter is from 2 to 127. When make the module height smaller, barcode reader may not read it. Specify 3 or higher in the normal use.

See Table 4-13 Alignment (**PrintAlignment**) for available setting in *alignment* parameter.

See Table 4-19 PDF417 Symbol (**Pdf417Symbol**) for available setting in *pdf417Symbol* parameter.

Error **PrinterException** may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

printQRcode — Print QR code

Target: Mobile printer/POS printer

Syntax:

(a) public void **printQRcode**(String *text*,
ErrorCorrection *errorCorrection*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

(b) public void **printQRcode**(String *text*,
ErrorCorrection *errorCorrection*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*,
QrModel *model*) throws **PrinterException**

Parameter:

text	Barcode data to send a printer
errorCorrection	Error correction level
moduleSize	Module Size
alignment	Alignment
model	QR code model

Description: This method prints QR code. The type (a) is a QR code model 2 fixed.

Also the version for either type (a) or (b) is automatically set depends on the number of data specified in *text* parameter.

See Table 4-18 Error correction level (**ErrorCorrection**) for available setting in *errorCorrection* parameter.

See Table 4-15 Module Size (**ModuleSize**) for available setting in *moduleSize* parameter.

See Table 4-13 Alignment (**PrintAlignment**) for available setting in *alignment* parameter.

See Table 4-20 QR Code Model (**QrModel**) for available setting in *model* parameter.

Error: **PrinterException** may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

cutPaper — Cut paper

Target: POS printer

Syntax: public void **cutPaper**(CuttingMethod *cuttingMethod*) throws **PrinterException**

Parameter:

cuttingMethod	Cutting method

Description: This method is valid only for POS printer. This method cuts paper. Cutting paper is executed after certain paper feed to avoid cutting the print data.

See Table 4-21 Cutting Method (**CuttingMethod**) for available setting in *cuttingMethod* parameter.

Error: **PrinterException** may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

openDrawer — Open cash drawer

Target: POS printer

Syntax: public void **openDrawer**(DrawerNum *drawerNum*,
PulseWidth *onOffTime*) throws **PrinterException**

Parameter:

drawerNum	Drawer number
onOffTime	pulse width

Description: This method is valid only for POS printer. This method opens the specified cash drawer.

See Table 4-22 Drawer Number (**DrawerNum**) for available setting in *drawerNum* parameter.

See Table 4-23 Activation Pulse Width (**PulseWidth**) for available setting in *onOffTime* parameter.

Error: **PrinterException** may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

buzzer — Sound buzzer

Target: POS printer

Syntax: public void **buzzer**(int *onTime*, int *offTime*) throws **PrinterException**

Parameter:

onTime	Buzzer On time (millisecond)
offTime	Buzzer Off time (millisecond)

Description: This method is valid only for POS printer. This method sounds the buzzer.

The valid range of *onTime* and *offTime* parameter is from 0 to 510.

Error: **PrinterException** may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

sendBinary — Send binary data

Target: Mobile printer/POS printer

Syntax: public void **sendBinary**(byte [] *binary*) throws **PrinterException**

Parameter:

binary	Binary data sent to a printer

Description: This method sends the binary data specified by *binary* parameter to a printer. Data size that can be specified at a time is 16K bytes (16384 bytes).

In this method, specified binary data is sent to a printer without conversion.

By sending printer command as binary data with this method, printer functions which are not supported in the library become available. However, this method does not support commands which obtain responses from a printer.

Error: **PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

sendDataFile — Send specified file

Target: Mobile printer/POS printer

Syntax: public void **sendDataFile**(String *fileName*) throws **PrinterException**

Parameter:

fileName	Name of data file sent to a printer

Description: This method determines data format based on the file extension specified by *fileName* parameter, convert it to printer-enabled data format, and send it to a printer. The maximum file size that can be specified is 1M byte (1048576 bytes).

When file extension is .bmp, .jpg, or .jpeg:
Send data to a printer as image data. When that image data is colored one, it is converted to monochrome image by binarization, and sent to a printer.

When file extension is .txt:
Send data to a printer as text data. Text data format supports UTF-8. Just like **sendText** method, encodes the data to printable text data based on the settings of international character set and codepage, and then sends it to a printer. Also, in this method, the line spacing code is not added to the end of text data.

When file extension is .bin or .dat:
Data is sent to a printer as binary data without conversion.

When file extension is .htm or .html:
Only valid for POS printer. Data is sent to a printer as html data without conversion.

Error: **PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

sendDataFile — Send specified file

Target: POS printer

Syntax: public void **sendDataFile**(String *fileName*,
PrintAlignment *alignment*) throws **PrinterException**

Parameter:

fileName	Name of data file sent to a printer
alignment	Alignment

Description: This method is valid only for POS printer. This method determines data format based on the file extension specified by *fileName* parameter, converts it to printer-enabled data format, and sends it to a printer. The maximum file size that can be specified is 1M byte (1048576 bytes).

See Table 4-13 Alignment (**PrintAlignment**) for available setting in *alignment* parameter.

When file extension is .bmp, .jpg, or .jpeg:
Send data to a printer as image data. When that image data is colored one, it is converted to monochrome image by binarization, and sent to a printer.

When file extension is .txt:
Send data to a printer as text data. Text data format supports UTF-8. Same as **sendText** method, encodes the data to printable text data based on the settings of international character set and codepage, and then sends it to a printer. Also, in this method, the line spacing code is not added to the end of text data.

When file extension is .bin or .dat:
Data is sent to a printer as binary data without conversion.

When file extension is .htm or .html:
Data is sent to a printer as html data without conversion.

Error: **PrinterException** may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

getStatus — Obtain printer status

Target Mobile printer/POS printer

Syntax public void **getStatus**(int [] *buf*) throws **PrinterException**

Parameter *buf* Status obtained from a printer

Description This method obtains printer status. Status obtained from a printer is stored to an integer array.
Contents of status vary between Mobile printer and POS printer.
When *printerModel* parameter specified while executing **connect** method is **PRINTER_MODEL_DPU_S245** or **PRINTER_MODEL_DPU_S445**, see contents of Mobile printer status.
When *printerModel* parameter is **PRINTER_MODEL_RP-D10** or **PRINTER_MODEL_RP-E10**, see POS printer status.

Mobile printer status is shown in Table 4-26.

Table 4-26 Printer Status (Mobile Printer)

Bit	Function	Value	
		0	**1**
0	Out-of-paper error	OK	Error
1	Head up error	OK	Error
2	Vp voltage malfunction	OK	Error
3	Thermal head temperature error	OK	Error
4	Function Setting error	OK	Error
5	Battery voltage state	See table below	
6			
7	Reserved	-	Fixed
8 to 31	Reserved	Fixed	-

Bit 6	Bit 5	Battery Voltage State
0	0	8.0 V or higher
0	1	7.5 V to 8.0 V
1	0	7.0 V to 7.5 V
1	1	Lower than 7.0 V

Status of POS printer is shown in Table 4-27.

Table 4-27 Printer Status (POS Printer)

Bit	Function	Value	
		0	1
0	VP voltage error	OK	Error
1	Hardware error	OK	Error
2	Head temperature error	OK	Error
3	Autocutter error	OK	Error
4	Out-of-paper error	OK	Error
5	Paper-near-end sensor error*1	OK	Error
6	Paper jam error while detecting mark*1	OK	Error
7	Cover open error	OK	Error
8	FEED Switch status	OFF	ON
9	Reserved	Fixed	-
10	Paper feed status	Stop	Operating
11	Return-waiting status	No	Yes
12	Reserved	Fixed	-
13	Reserved	-	Fixed
14	Reserved	-	Fixed
15	Drawer switch input status	Low	High
16	FLASH memory rewriting	No	Yes
17	Peripheral device selection	Printer	Other
18 to 31	Reserved	-	Fixed

*1: Supported only in RP-E10. In RP-D10, it is always OK (value: 0).

Error **PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending or receiving. See **isConnect** method for verifying connection state.

abort — Abort the waiting state of a printer

Target: Mobile printer/POS printer

Syntax: public void **abort**() throws **PrinterException**

Description: When sending of image data by **sendDataFile** method is aborted, a printer does not accept other processes until specified image data is received completely. (Method or sent data are misinterpreted and recognized as part of the image data.)
To solve this situation, use this method to abort the waiting state of a printer.
Note that when executing this method, a part of unprocessed image data may be printed.

Error: **PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

registerLogo — Register logo (image) to a printer

Target: Mobile printer

Syntax: public void **registerLogo**(String *fileName*, int *id*) throws **PrinterException**

Parameter:

fileName	File name of image data to register as logo
id	Logo ID to register

Description: This method is valid only for Mobile printer. This method registers image data specified by *fileName* parameter to a printer as a logo.

File extension for image data supported by *fileName* parameter is .bmp, .jpg, or .jpeg.
You can specify values from 0 to 127 for *id* parameter.

Error: **PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

registerLogo — Register logo (image) to a printer

Target: POS printer

Syntax: public void **registerLogo**(String *fileName*, String *id*) throws **PrinterException**

Parameter:

fileName	File name of image data to register as logo
id	Logo ID to register

Description: This method is valid only for POS printer. This method registers image data specified by *fileName* parameter to a printer as a logo.

File extension for image data supported by *fileName* parameter is .bmp, .jpg, or .jpeg.
You can specify two characters for *id* parameter.
Subsequent characters are ignored.
Furthermore, valid characters are ASCII character codes 20h (space) to 7Eh (tilde) such as alphanumeric ('0' to '9', 'A' to 'Z', 'a' to 'z').

Error: **PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

printLogo — Print specified logo (image) in printer

Target: Mobile printer

Syntax: public void **printLogo**(int *id*) throws **PrinterException**

Parameter:

id	Logo ID to print

Description: This method is valid only for Mobile printer. This method print the logo (image) registered by **registerLogo** method. Specify registered logo ID in *id* parameter. The valid range of *id* parameter is from 0 to 127.

Error: **PrinterException** may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

printLogo — Print specified logo (image) in printer

Target: POS printer

Syntax: public void **printLogo**(String *id*, PrintAlignment *alignment*) throws **PrinterException**

Parameter:

id	Logo ID to print
alignment	Alignment

Description: This method is valid only for POS printer. This method print the logo (image) registered by **registerLogo** method. Specify registered logo ID in *id* parameter. The valid range of *id* parameter is 2 characters. The valid characters are ASCII character code from 20h (space) to 7Eh (tilde) such as alphanumeric ('0' to '9', 'A' to 'Z', 'a' to 'z').

See Table 4-13 Alignment (**PrintAlignment**) for available setting in *alignment* parameter.

Error: **PrinterException** may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

unregisterLogo — Delete specified logo (image) in a printer

Target: Mobile printer

Syntax: public void **unregisterLogo**(int *id*) throws **PrinterException**

Parameter:

id	Logo ID to delete

Description: This method is valid only for Mobile printer.
This method deletes the logo (image) registered by **registerLogo** method (for Mobile printer).
Specify the registered logo ID for *id* parameter. You can specify values from 0 to 127 for *id* parameter.

Error: **PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

unregisterLogo — Delete specified logo (image) in a printer

Target: POS printer

Syntax: public void **unregisterLogo**(String *id*) throws **PrinterException**

Parameter:

id	Logo ID to delete

Description: This method is valid only for POS printer.
This method deletes the logo (image) registered by **registerLogo** method (for POS printer).
Specify the registered logo ID for *id* parameter.
You can specify two characters for *id* parameter. Subsequent characters are ignored.
Furthermore, valid characters are ASCII character codes 20h (space) to 7Eh (tilde) such as alphanumeric ('0' to '9', 'A' to 'Z', 'a' to 'z').

Error: **PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

registerStyleSheet — Register style sheet to a printer

Target: POS printer

Syntax: public void **registerStyleSheet**(String *fileName*,
int *num*) throws **PrinterException**

Parameter:

fileName	CSS file name to register as style sheet
num	Style sheet number to register

Description: This method is valid only for POS printer. This method registers CSS file specified by *fileName* parameter to a printer. Maximum number of registerable style sheet is four.

Style sheet supported by *fileName* parameter is that style sheet language is written in CSS (cascading style sheets), and that file extension is .css.
Maximum number of style registerable to one CSS file is 64.
Also, you can specify values from 1 to 4 for *num* parameter.

See "6.5.13 Tag Processing Mode" in "RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" or "RP-E10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for more details about style sheet.

Error: **PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

unregisterStyleSheet — Delete specified style sheet in a printer

Target	POS printer
Syntax	public void **unregisterStyleSheet**(int *num*) throws **PrinterException**
Parameter	*num* Style sheet number to delete
Description	This method is valid only for POS printer. This method deletes the style sheet registered by **registerStyleSheet** method. Specify the registered style sheet number for *num* parameter. You can specify values from 1 to 4 for *num* parameter.
Error	**PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending. See **isConnect** method for verifying connection state.

resetPrinter — Printer hardware reset

Target	Mobile printer/POS printer
Syntax	public void **resetPrinter**() throws **PrinterException**
Description	The available connecting method varies between Mobile printer and POS printer. For Mobile printer, this method is valid only when a printer communication is conducted through USB connection. For POS printer, this method is valid when a printer communication is conducted through any of Bluetooth connection, USB connection, or TCP/IP connection. For Bluetooth connection, this method resets a connected printer by using printer command. For USB connection, this method resets a connected printer by using the SOFT_RESET function in USB printer class. For TCP/IP connection, this method resets a connected printer by using the SII original command (reset command) to TCP port number 26100. After executing this method, however, connection to a printer will be maintained.
Error	**PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending. See **isConnect** method for verifying connection state.

getPrinterResponse — Obtain various responses from a printer

Target: Mobile printer/POS printer

Syntax: public void **getPrinterResponse**(int *id*, Object *buf*) throws **PrinterException**

Parameter:

id	Response type constant
buf	Buffer that stores obtained response data (Buffer type varies depending on a response type constant.)

Description: This method stores the response data specified by *id* parameter to the object specified by *buf* parameter. See Table 4-7 Response Type Constant for details about available constants for *id* parameter. Contents of obtainable response from a printer vary between Mobile printer and POS printer.

When *printerModel* parameter specified while executing **connect** method is **PRINTER_MODEL_DPU_S245** or **PRINTER_MODEL_DPU_S445**, see contents of obtained response for Mobile printer. When *printerModel* parameter is **PRINTER_MODEL_RP-D10** or **PRINTER_MODEL_RP-E10**, see contents of obtained response for POS printer.

Contents of obtained response for Mobile printer are shown in Table 4-28.

Table 4-28 Contents of Obtained Response (Mobile Printer)

Constant Name	Description
PRINTER_RESPONSE_ REQUEST	Obtains execution response request. For *buf* parameter, specify integer array whose length is 1. Specify 0 to 15 (00h to 0Fh) for *buf[0]*. When responses are obtained successfully, 80 to 95 (50h to 5Fh) are stored to *buf[0]*.
PRINTER_RESPONSE_ EXTERNAL_RAM	Obtains remaining RAM capacity response. For *buf* parameter, specify integer array whose length is 1. When responses are obtained successfully, remaining RAM capacity is stored to *buf[0]* in byte value.
PRINTER_RESPONSE_ USER_AREA	Obtains remaining user area response. For *buf* parameter, specify integer array whose length is 1. When responses are obtained successfully, remaining user area is stored to *buf[0]* in byte value.
PRINTER_RESPONSE_ BATTERY_STATUS	Obtains battery voltage status. For *buf* parameter, specify integer array whose length is 1. When responses are obtained successfully, the battery status value is stored to *buf[0]*. Meanings of battery status values are as follows.

Battery Status Value	Battery Voltage State
0	8.0 V or higher
1	7.5 V to 8.0 V
2	7.0 V to 7.5 V
3	Lower than 7.0 V

Contents of obtained responses for POS printer are shown in Table 4-29.

Table 4-29 Contents of Obtained Response (POS Printer)

Constant Name	Description
PRINTER_RESPONSE_ REQUEST	Obtains execution response request. For *buf* parameter, specify integer array whose length is 1. Specify 0 to 15 (00h to 0Fh) for *buf[0]*. When responses are obtained successfully, 128 to 143 (80h to 8Fh) are stored to *buf[0]*.
PRINTER_RESPONSE_ USER_AREA	Obtains remaining user area response. For *buf* parameter, specify integer array whose length is 1. When responses are obtained successfully, remaining user area is stored to *buf[0]* in byte value.
PRINTER_RESPONSE_ ARRANGE_USER_AREA	Obtains remaining memory response after user area defragment. For *buf* parameter, specify integer array whose length is 1. When responses are obtained successfully, remaining memory after user area defragment is stored to *buf[0]* in byte value.
PRINTER_RESPONSE_ NV_GRAPHICS	Obtains NV graphics memory capacity response. For *buf* parameter, specify integer array whose length is 1. When responses are obtained successfully, NV graphics memory capacity is stored to *buf[0]* in byte value.
PRINTER_RESPONSE_ KEY_CODE	Obtains the key code list defined NV graphics. For *buf* parameter, specify ArrayList<String> type array. When responses are obtained successfully, the key code of NV graphics is stored to *buf* parameter in string array. Example: *buf.size()* = 3, *buf[0]* = "22", *buf[1]* = "23", *buf[2]* = "24" etc.

Error **PrinterException** may be thrown when the method is called. The printer may be disconnected when **PrinterException** is thrown during data sending or receiving. See **isConnect** method for verifying connection state.

startDiscoveryPrinter — Start printer search (Bluetooth)

Target: Mobile printer/POS printer

Syntax: public void **startDiscoveryPrinter**(PrinterListener *listener*, Context *context*) throws **PrinterException**

Parameter:

listener	PrinterListener Instance described in the following
context	Context of Application.

Description: This method searches the printer connected to Bluetooth. Completion or cancellation of the printer search is notified to user application as event by **finishEvent** method through instance specified in *listener* parameter. This method may discover other printers besides SII printer. In addition, the printers established the Bluetooth connection by this library or other applications are not searched. The searched printer information is stored in **PrinterInfo** class described in the following. Do not call this method from main thread of application.

Error: **PrinterException** may be thrown when this method is called.

startDiscoveryPrinter — Start printer search (TCP/IP)

Target: POS printer

Syntax: public void **startDiscoveryPrinter**(PrinterListener *listener*, int *retry*, int *timeout*) throws **PrinterException**

Parameter:

listener	**PrinterListener** Instance described in the following
retry	The number of retries.
timeout	Search timeout period.

Description: This method is valid only for POS printer. This method sends local broadcast packet and searches SII printers. Completion or cancellation of the printer search is notified to user application as event by **finishEvent** through instance specified in *listener* parameter. This method sends local broadcast packet by *retry* parameter setting times and waits the response from the printer by time out period specified in *timeout* parameter. The searched printer information is stored in **PrinterInfo** class described in the following.

LAN interface F/W version 1.12.01 or higher is needed in order to use this method.

Error: **PrinterException** may be thrown when the method is called.

cancelDiscoveryPrinter — Cancel printer search

Target: Mobile printer/POS printer

Syntax: public void **cancelDiscoveryPrinter**()

Description: This method cancels **startDiscoveryPrinter** under execution. Cancel searching is notified to user application as event by **finishEvent** method through instance specified in the *listener* parameter of **startDiscoveryPrinter** method.

getFoundPrinter — Obtain searched printer information

Target	Mobile printer/POS printer
Syntax	public ArrayList<**PrinterInfo**> **getFoundPrinter**()
Description	This method obtains the printer information searched by the printer search in ArrayList of **PrinterInfo** class.
Return value	ArrayList of **PrinterInfo** class.

getSendTimeout — Obtain send timeout period

Target	Mobile printer/POS printer
Syntax	public int **getSendTimeout**()
Description	This method obtains the timeout period when data is sent. This method is obtainable whether a printer is connected or not. Obtained timeout period is expressed in msec (millisecond) value.
Return value	Send timeout period in msec (millisecond)

setSendTimeout — Set send timeout period

Target	Mobile printer/POS printer
Syntax	public void **setSendTimeout**(int *sendTimeout*)
Parameter	*sendTimeout* Send timeout period
Description	This method sets the timeout period when data is sent in msec (millisecond). This method is configurable whether a printer is connected or not. However, it is not until sending data next time that the configured timeout period is enabled. Moreover, the default is set when you configure values outside the scope of the valid range.
	Default 10000 msec (10 seconds)
	Effective range 100 msec to 90000 msec (90 secconds)

getReceiveTimeout — Obtain receive timeout period

Target	Mobile printer/POS printer
Syntax	public int **getReceiveTimeout**()
Description	This method obtains the timeout period when data is received. This method is obtainable whether a printer is connected or not. Obtained timeout period is expressed in msec (millisecond) value.
Return value	Receive timeout period in msec (millisecond)

setReceiveTimeout — Set receive timeout period

Target: Mobile printer/POS printer

Syntax: public void **setReceiveTimeout**(int *receiveTimeout*)

Parameter: *receiveTimeout* Receive timeout period

Description: This method sets the timeout period when data is received in msec (millisecond). This method is configurable whether a printer is connected or not. However, it is not until receiving data next time that the configured timeout period is enabled. Moreover, the default is set when you configure values outside the scope of the valid range.

Default: 10000 msec (10 seconds)

Effective range: 100 msec to 90000 msec (90 seconds)

getInternationalCharacter — Obtain international character set

Target: Mobile printer/POS printer

Syntax: public int **getInternationalCharacter**()

Description: This method obtains setting values of international character set. When you send text data by using **sendText** method, **sendTextEx** method, or **sendDataFile** method with setting values of international character set, following character codes are printed differently. See "Appendix A Character Sets (Character Code Table)" for details about printed characters.

Character codes whose print result varies depending on the international character set configuration:
0x23, 0x24, 0x40, 0x5B, 0x5C, 0x5D, 0x5E, 0x60, 0x7B, 0x7C, 0x7D, 0x7E

Return value: See Table 4-2 International Character Set Constant for details.

setInternationalCharacter — Set international character set

Target: Mobile printer/POS printer

Syntax: public void **setInternationalCharacter**(int *internationalCharacter*)

Parameter: *internationalCharacter* International character set constant

Description: This method sets international character set. See Table 4-2 International Character Set Constant for details about configurable values. When international character set is not configured, it is initialized to following state depending on a language setting of an Android device. Also, when you set an invalid value to *internationalCharacter* parameter, the following values are used as well.

When a language setting of an Android device is Japanese:
COUNTRY_JAPAN

When a language setting of an Android device is other languages than Japanese:
COUNTRY_USA

getCodePage — Obtain codepage

Target: Mobile printer/POS printer

Syntax: public int **getCodePage**()

Description: This method obtains setting values of codepage. The encoder used when you send text data by using **sendText** method, **sendTextEx** method, or **sendDataFile** method is changed by the codepage setting. See "Appendix A Character Sets (Character Code Table)" for details about printed characters.

Return value: See Table 4-3 Codepage Constant for details.

setCodePage — Set codepage

Target: Mobile printer/POS printer

Syntax: public void **setCodePage**(int *codePage*)

Parameter: *codePage* Codepage constant

Description: This method sets codepage. See Table 4-3 Codepage Constant for details about configurable values. When a codepage is not configured, it is initialized to following state depending on a language setting of an Android device. Also, when you set an invalid value to *codePage* parameter, it will be ignored.

When a language setting of an Android device is Japanese:
CODE_PAGE_KATAKANA

When a language setting of an Android device is other languages than Japanese:
CODE_PAGE_1252

getPrinterModel — Obtain printer model

Target: Mobile printer/POS printer

Syntax: public int **getPrinterModel**()

Description: This method obtains a model value for the connected printer. When a printer is not connected, the default returns. Even though a printer is not connected, when **connect** method has been succeeded once, the printer model value previously connected returns.

Return value: See Table 4-5 Printer Model Constant for details.

Default: **PRINTER_MODEL_DEFAULT**

getPortType — Obtain connection port type

Target	Mobile printer/POS printer
Syntax	public int **getPortType**()
Description	This method obtains the port type of connecting printer in use during connection with a printer. When a printer is not connected, the default returns. Even though a printer is not connected, when **connect** method has been succeeded once, the port type value previously connected returns.
Return value	See Table 4-6 Port Type Constant for details.
Default	**PRINTER_TYPE_BLUETOOTH**

isConnect — Verify connection state with a printer

Target	Mobile printer/POS printer
Syntax	public boolean **isConnect**()
Description	This method verifies connection state with a printer. Returns true when a printer is connected and false when the printer is disconnected. When **PrinterException** is thrown during data sending and is disconnected from printer, the false returns in this method. If false, it is necessary to connect with the printer again in **connect** method.
Return value	Following values are returned depending on the connection state with a printer.

true	Connected to a printer
false	Disconnected to a printer

getSocketKeepingTime — Obtain socket keeping time

Target	POS printer
Syntax	public int **getSocketKeepingTime**()
Description	This method is valid only for POS printer. This method obtains socket keeping time during TCP/IP connection. This method is obtainable whether a printer is connected or not. Obtained time is expressed in msec (millisecond) value.
Return value	Socket keeping time in msec (millisecond)

setSocketKeepingTime — Set socket keeping time

Target: POS printer

Syntax: public void **setSocketKeepingTime**(int *socketKeepingTime*)

Parameter

socketKeepingTime	Socket keeping time

Description: This method is valid only for POS printer. This method obtains socket keeping time during TCP/IP connection. Obtained timeout period is expressed in msec (millisecond) value. For socket keeping time, set the same time as Network Printer Receive Timeout of the printer to be connected. See "RP-E10 SERIES LAN INTERFACE TECHNICAL REFERENCE" for more details about Network Printer Receive Timeout.
This method is configurable whether a printer is connected or not. However, it is not until executing **connect** method (TCP/IP) next time that the configured socket keeping time is enabled. Moreover, the default is set when you configure values outside the scope of the valid range.

Default	300000msec (5 minutes)
Effective range	60000 to 300000msec (5 minutes)

4.4.2 PrinterEvent Class

(1) Method List

This class obtains the event type proceeded when printer searching is completed.
Methods provided by **PrintEvent** class are shown in Table 4-30.

Table 4-30 Method of PrinterEvent Class

Method	Function Summary	Target	
		Mobile	POS
getEventType	Obtain event type	Supported	Supported

(2) Constant List

Constants used for obtaining event type are shown in Table 4-31.

Table 4-31 Event Type Constant

Method	Function Summary	Value	Target	
			Mobile	POS
EVENT_FINISHED_DISCOVERY	Complete printer searching	1	Supported	Supported
EVENT_CANCELED_DISCOVERY	Cancel printer searching	2	Supported	Supported

(3) Method Details

getEventType — Obtain event type

Target: Mobile printer/POS printer

Syntax: public int **getEventType**()

Description: This method obtains event class proceeded after the printer search completion.
This method determines whether the printer search is completed or cancelled depending on the obtained event type. Even when the printer was not discovered, **EVENT_FINISHED_DISCOVERY** is returned.

Return value: See Table 4-31 Event Type Constant for details.

4.4.3 PrinterListener Interface

(1) Method List

This interface obtains the complete event of printer searching.
Methods of **PrinterListener** Interface are shown in Table 4-32.

Table 4-32 Method of PrinterListener Interface

Method	Function Summary	Target	
		Mobile	POS
finishEvent	Finish event of the printer search	Supported	Supported

(2) Method Details

finishEvent — Finish event of the printer search

Target: Mobile printer/POS printer

Syntax: public void **finishEvent**(PrinterEvent *event*)

Parameter: *event* Printer event

Description: This method is for interface with no implementation.
This method is called when the printer search is completed or cancelled.
Specify the above mentioned **PrinterEvent** class for *event* parameter. Implement this method to the user application for receiving notice of completion of printer search or cancelled event, and determine the completed event type by **getEventType** method of **PrinterEvent** class.

4.4.4 PrinterInfo Class

(1) Method List

This class stores the printer information searched by printer searching method.
Printer model name, Bluetooth address, MAC address, and IP address can be obtained by searched printer information. Methods of **PrinterInfo** class are shown in Table 4-33.
The available method varies depending on the target printer either of Mobile printer or POS printer.

Table 4-33 Method of PrinterInfo Class

Method	Function Summary	Target	
		Mobile	POS
getPrinterModelName	Obtain printer model name	Supported	Supported
getBluetoothAddress	Obtain Bluetooth address	Supported	Supported
getMacAddress	Obtain MAC address	Not supported	Supported
getIpAddress	Obtain IP address	Not supported	Supported

(2) Method Details

getPrinterModelName — Obtain printer model name

Target: Mobile printer/POS printer

Syntax: public String **getPrinterModelName**()

Description: This method obtains the character string of printer model name from the printer information searched by the printer searching.

Return value: Printer model name.

getBluetoothAddress — Obtain Bluetooth address

Target: Mobile printer/POS printer

Syntax: public String **getBluetoothAddress**()

Description: This method obtains the character string of Bluetooth address from the printer information searched by the printer search.

Return value: Bluetooth address.

getMacAddress — Obtain MAC address

Target: POS printer

Syntax: public String **getMacAddress**()

Description: This method is valid only for POS printer. This method obtains the character string of MAC address from the printer information searched by the printer search.

LAN interface F/W version 1.12.01 or higher is needed in order to use this method.

Return value: MAC address

getIpAddress — Obtain IP address

Target	POS printer
Syntax	public String **getIpAddress**()
Description	This method is valid only for POS printer. This method obtains the character string of IP address from printer information searched by the printer search. LAN interface F/W version 1.12.01 or higher is needed in order to use this method.
Return value	IP address

4.4.5 PrinterException Class

(1) Method List

The list of methods provided by **PrinterException** class is shown in Table 4-34.

Table 4-34 Method of the PrinterException Class

Method	Function Summary	Target	
		Mobile	POS
PrinterException	Constructor	Supported	Supported
getErrorCode	Obtain error codes	Supported	Supported

(2) Constant List

(a) Constants used for obtaining error codes are shown in Table 4-35.

Table 4-35 Error Codes List

Constant Name	Description	Value	Target	
			Mobile	POS
ERROR_ACCESS_DENIED	Failed to obtain the handle.[*1]	-1	Supported	Supported
	Unavailable port specified.		Supported	Supported
ERROR_SHARING_ VIOLATION	Already opened port specified.	-11	Supported	Supported
ERROR_PORT_NOT_ OPENED	Port not opened.	-12	Supported	Supported
ERROR_DEVICE_NOT_ CONNECTED	Specified Bluetooth address printer does not exist.	-21	Supported	Supported
	No printer having the specified printer model constants exist in USB connection.		Supported	Supported
	Specified IP address printer does not exist.		Not supported	Supported
ERROR_DEVICE_ INITIALIZE_FAILED	Failed to modify the printer setting. It might have happened that data sending to the printer was not completed within send time out period or data reception from the printer was not completed within receive timeout period.	-31	Supported	Supported
ERROR_DATA_SIZE_ZERO	0-byte data specified.	-101	Supported	Supported
ERROR_OVER_MAX_ DATA_SIZE	The data size exceeds the maximum value.	-102	Supported	Supported
ERROR_ENCODE_FAILED	Error occurred in encoding text data.[*1]	-111	Supported	Supported
ERROR_TIMEOUT	Send timeout happened.	-201	Supported	Supported
	Receive timeout happened.		Supported	Supported
ERROR_FILE_NOT_ FOUND	Specified file not found.	-301	Supported	Supported

Constant Name	Description	Value	Target	
			Mobile	POS
ERROR_FILE_USED	The process cannot access the file because it is being used by another process.	-302	Supported	Supported
ERROR_FILE_INVALID	Invalid file specified.	-303	Supported	Supported
ERROR_LOW_MEMORY	Insufficient memory when loading image data file.	-311	Supported	Supported
ERROR_OVER_ MAX_IMAGE	Either width or height of image data, or both of them exceed the number of printable maximum dots.	-312	Supported	Supported
ERROR_LOGO_NOT_ DEFINED	Specified the image of key code is not registered.	-313	Not supported	Supported
ERROR_LOW_USER_AREA	Insufficient remaining user area.	-401	Supported	Supported
ERROR_LOW_EXTERNAL_ RAM	Insufficient remaining RAM capacity.	-402	Supported	Not supported
ERROR_INVALID_NO*	Inappropriate value specified for style sheet number.	-501	Not supported	Supported
ERROR_OVER_STYLE_ NUM	The number of style registered in the specified file exceeds rated value (64).	-502	Not supported	Supported
ERROR_INVALID_PARAM	Invalid parameter specified.	-9999	Supported	Supported

* Hereafter, do not use **ERROR_INVALIT_NO** defined before. The function will be abolished when the version is updated.

*1 Abnormality processing might have happened.

(3) Method Details

PrinterException — Constructor

Target	Mobile printer/POS printer
Syntax	public **PrinterException**(int *code*, String *message*)
Description	Constructor for **com.seikoinstruments.sdk.thermalprinter.PrinterException** class.

PrinterException — Constructor

Target	Mobile printer/POS printer
Syntax	public **PrinterException**(int *code*, String *message*, String *detail*)
Description	Constructor for **com.seikoinstruments.sdk.thermalprinter.PrinterException** class.

getErrorCode — Obtain error codes

Target	Mobile printer/POS printer
Syntax	public int **getErrorCode**()
Description	This method obtains error code for thrown exception.
Return value	See Table 4-35 Error Codes List for details.

Chapter 5
Sample Program

This chapter describes the sample programs provided by the SDK.

5.1 Sample Program Overview

5.1.1 Sample Programs for Java Eclipse Project

The SDK includes two sample programs that are Java Eclipse project format. Between these two sample programs, implemented functions for using the library is the same, but the ways to display settings screen are different.

(a) The sample program which uses buttons on the ActionBar to display settings screen. ActionBar is a new UI introduced in Android 3.x (Honeycomb).
(Sample_ActionBar project, Figure 5-1)

(b) The sample program which uses menu buttons on the device to display settings screen for non-ActionBar-compliant Android devices such as Android 2.3.x (Gingerbread).
(Sample_Menu project, Figure 5-2)

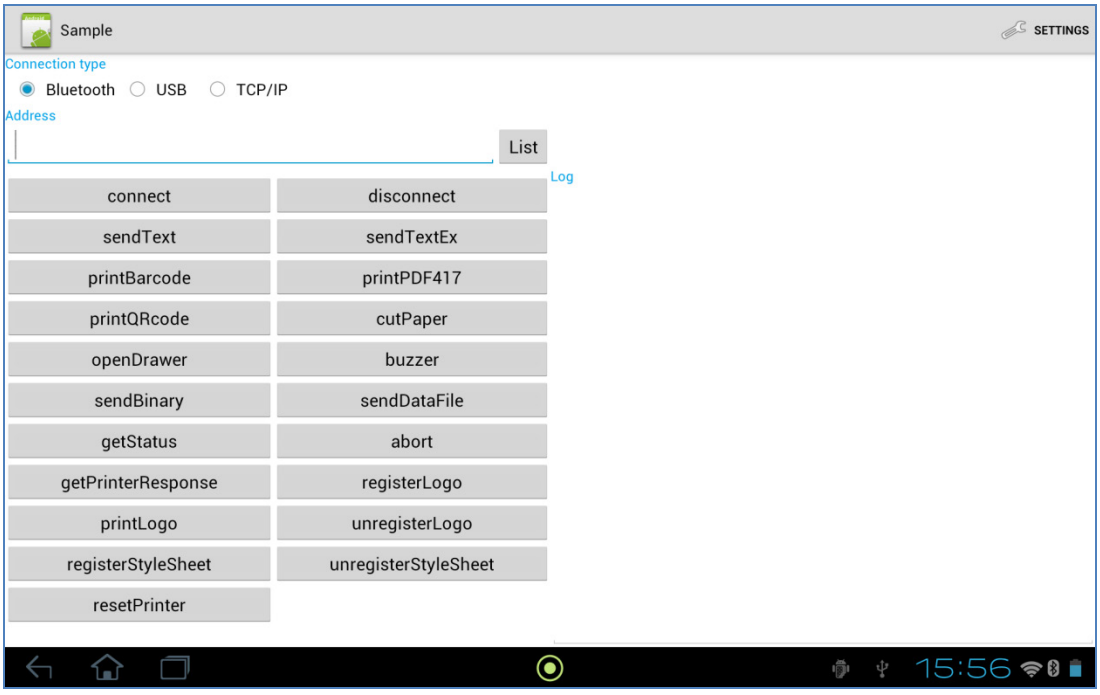

Figure 5-1 Screen of Sample_ActionBar (Sample Screen in Android 4.0.3 Tablet)

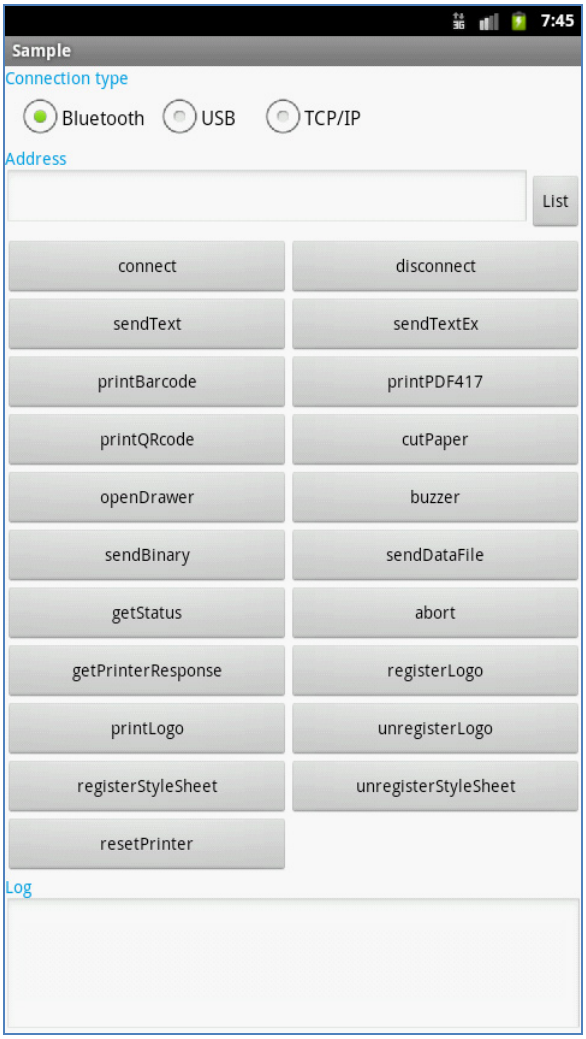

Figure 5-2 Screen of Sample_Menu (Sample Screen in Android 2.3.3 Smartphone)

5.2 How to Use Sample Programs

Ensure that the environment for developing Android application is prepared.
See "Chapter 3 How to Use the Library" for details about required development environment.

5.2.1 Add Sample Project Programs to Eclipse

This section describes the way to add a sample program projects to Eclipse by using the Sample_ActionBar project included in the SDK as an example. Procedures are shown below.

(a) Click [File] and [Import]. (Figure 5-3)

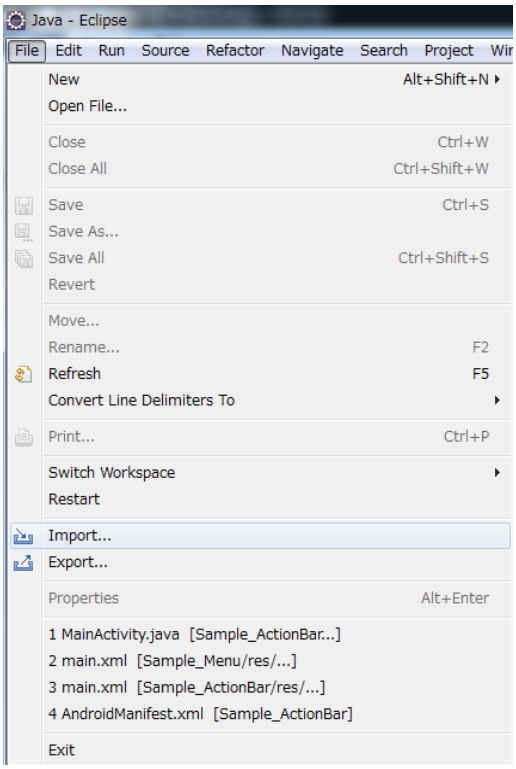

Figure 5-3

(b) Select [General] and [Existing Projects into Workspace], and then click [Next].(Figure 5-4)

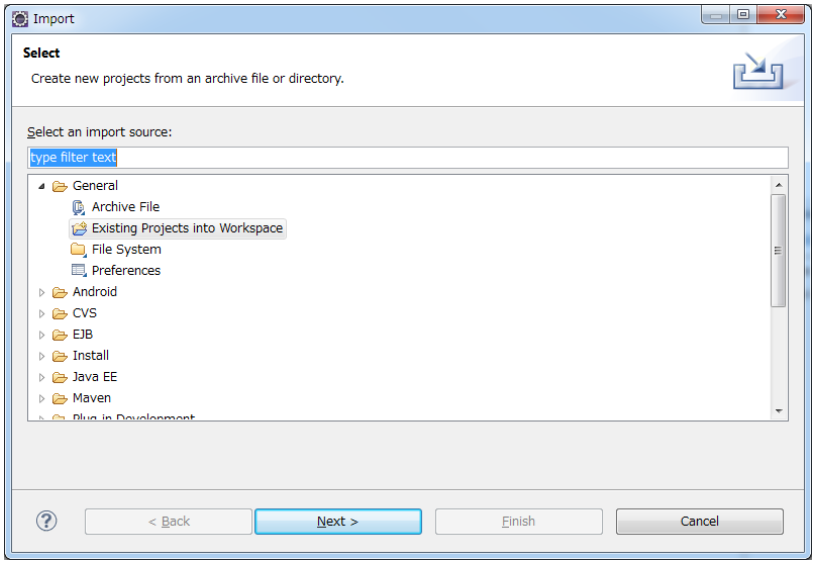

Figure 5-4

(c) Browse the folder in which the Sample_ActionBar project is extracted, and click [Finish]. (Figure 5-5)

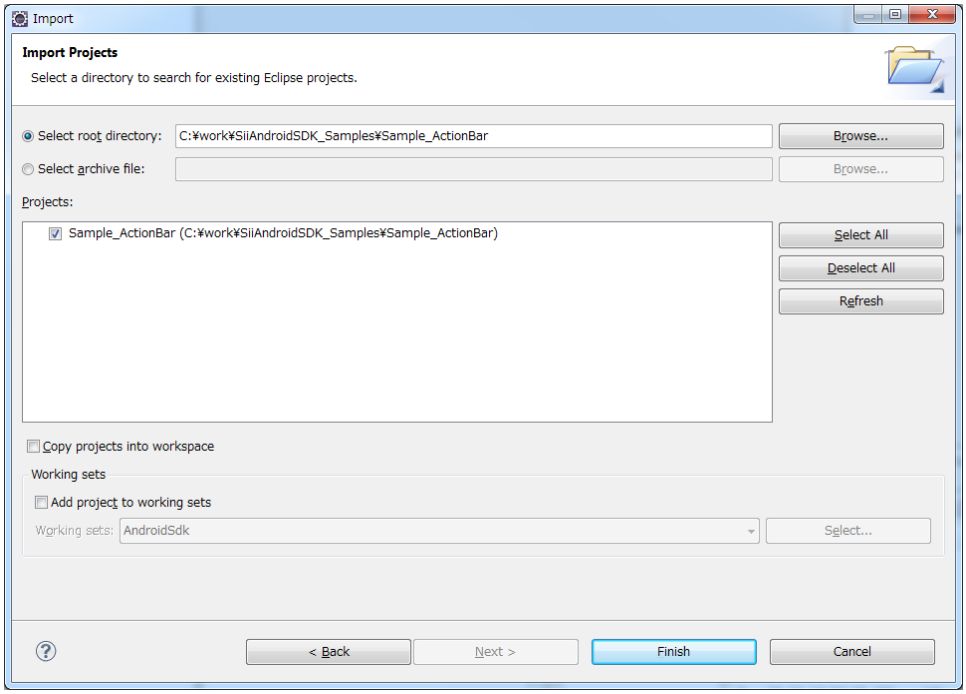

Figure 5-5

(d) The Sample_ActionBar project is added to the [Package Explorer] view. (Figure 5-6)

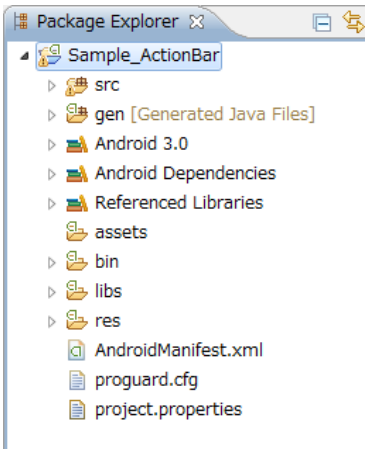

Figure 5-6

5.2.2 Set Debug Mode in a Sample Program Project

This section describes the way to set debug mode in a project which is added to Eclipse. Procedures are shown below.

(a) In [Package Explorer] view, click [AndroidManifest.xml]. (Figure 5-7)

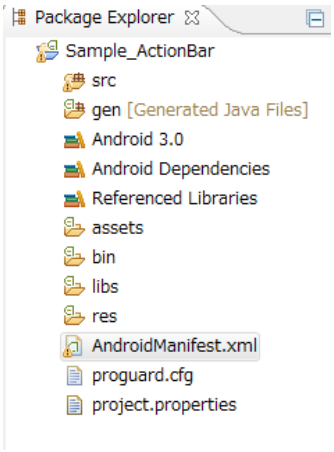

Figure 5-7

(b) [Sample_ActionBar Manifest] is displayed in Java editor view, click [Application] tab at the bottom of the screen. Set true for [Debuggable] field in [Application Attributes].(Figure 5-8)

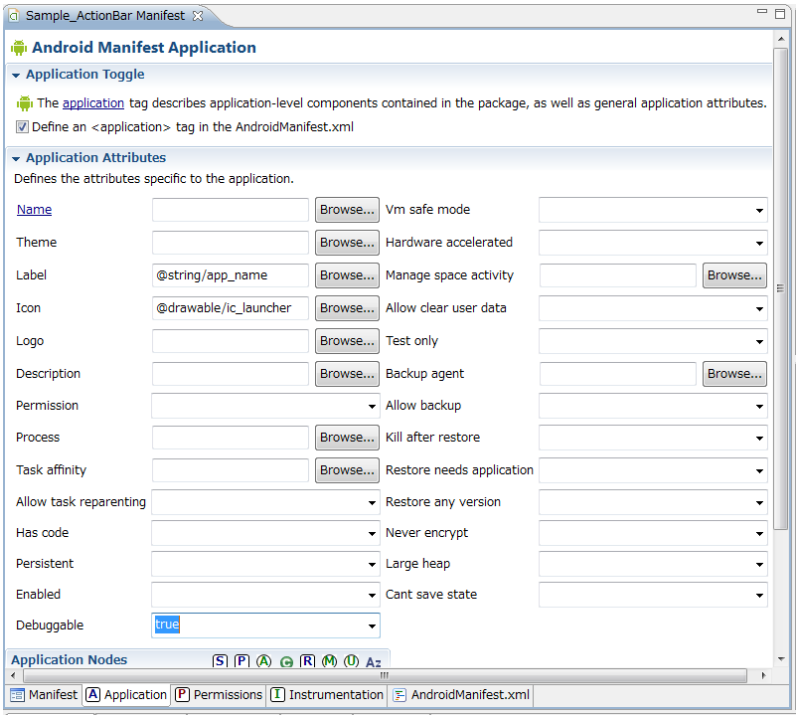

Figure 5-8

5.2.3 Execute Sample Program

In order to execute the project in sample program which is added to Eclipse, select the top of the package in [Package Explorer] view, then click [Run] and [Run], or [Run] and [Debug].

(NOTE) Introduction to Using Android Virtual Device (AVD：Android emulator)

- **You cannot connect to a printer through Bluetooth communication or USB communication.**
- **No guarantee of proper operation is provided for printer connection through TCP/IP communication.**

5.3 Sample Program Function

This section describes the functions of the sample program by using the Sample_ActionBar project screen (Figure 5-9) as an example.

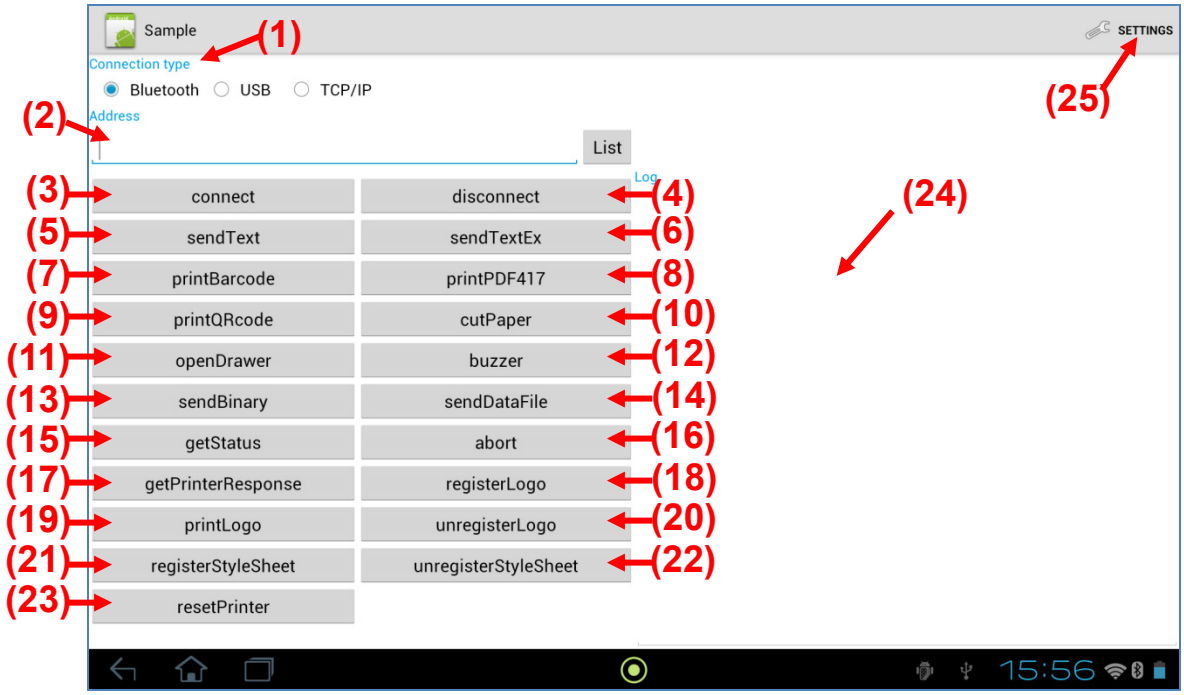

Figure 5-9 Functions of the Program

Functions of the sample program are shown in Table 5-1.

Table 5-1 Functions of the Sample Program

No.	Description	Remarks
(1)	Connection type	Selects connection form to a printer. Select [Bluetooth] or [USB] for Mobile printer. Select [Bluetooth], [USB], or [TCP/IP] for POS printer.
(2)	Address	Specifies address for a printer. If you use Bluetooth connection, enter Bluetooth address for a printer. By tapping [List] button, the list of devices searched by **startDiscoveryPrinter (Bluetooth)** method is displayed. By selecting a printer to connect from the list, Bluetooth address can be entered. Example: "00:11:22:AA:BB:CC" If you use [TCP/IP] connection, enter IP address for a printer. Example: "192.168.0.1"
(3)	connect	Executes **connect** method.
(4)	disconnect	Executes **disconnect** method.
(5)	sendText	Executes **sendText** method.
(6)	sendTextEx	Executes **sendTextEx** method.
(7)	printBarcode	Executes **printBarcode** method.
(8)	printPDF417	Executes **printPDF417** method.

No.	Description	Remarks
(9)	printQRcode	Executes **printQRcode** method.
(10)	cutPaper	Executes **cutPaper** method.
(11)	openDrawer	Executes **openDrawer** method.
(12)	buzzer	Executes **buzzer** method.
(13)	sendBinary	Executes **sendBinary** method.
(14)	sendDataFile	Executes **sendDataFile** method.
(15)	getStatus	Executes **getStatus** method.
(16)	abort	Executes **abort** method.
(17)	getPrinterResponse	Executes **getPrinterResponse** method.
(18)	registerLogo	Executes **registerLogo** method.
(19)	printLogo	Executes **printLogo** method.
(20)	unregisterLogo	Executes **unregisterLogo** method.
(21)	registerStyleSheet	Executes **registerStyleSheet** method.
(22)	unregisterStyleSheet	Executes **unregisterStyleSheet** method.
(23)	resetPrinter	Executes **resetPrinter** method.
(24)	Log	Displays method execution logs from (3) to (23).
(25)	SETTING	Displays settings screen shown in Figure 5-10 [SETTING] Screen. See Table 5-2 for configurable options. In order to go back to the screen in Figure 5-9, tap  on the screen

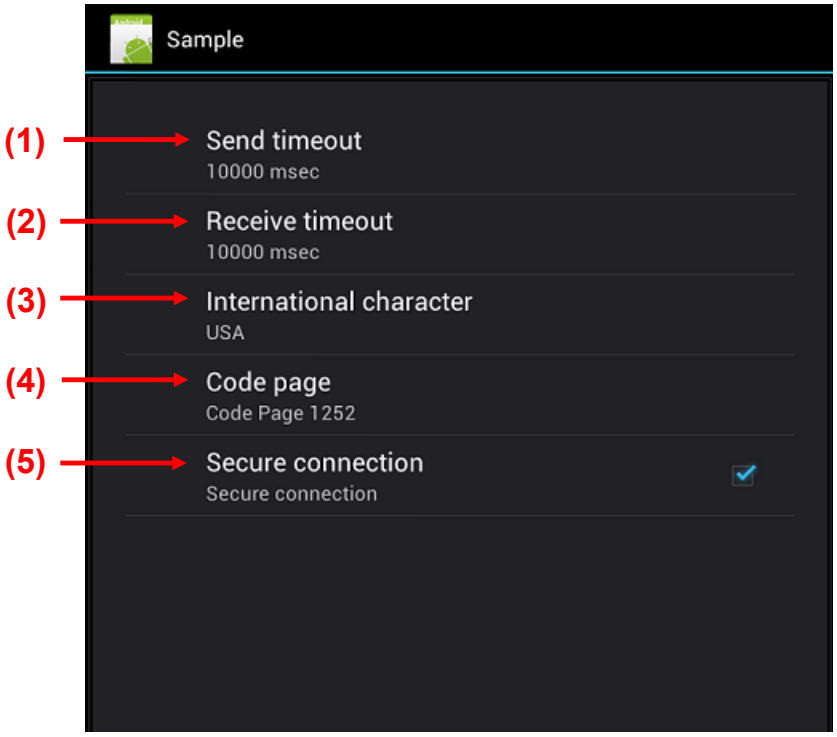

Figure 5-10 [SETTING] Screen

Functions for settings screen are shown in Table 5-2.

Table 5-2 Settings Screen

No.	Description	Remarks
(1)	Send timeout	Sets the time until send timeout happens in msec (millisecond). The text entry screen is displayed in another window.
(2)	Receive timeout	Sets the time until receive timeout happens in msec (millisecond). The text entry screen is displayed in another window.
(3)	International character	Sets international character set. Figure 5-11 is displayed in another window. This is the international character set configuration for printer side when printing texts (when text files are specified by **sendText** method, **sendTextEx** method, or **sendDataFile** method).
(4)	Code page	Sets codepage. Figure 5-12 is displayed in another window. This is the encoding and the character set configuration value for printer side when printing texts (when text files are specified by **sendText** method, **sendTextEx** method, or **sendDataFile** method).
(5)	Secure connection	Sets whether secure mode or insecure mode is used at Bluetooth connection. When the checkbox is selected, secure mode is used. When the checkbox is cleared, insecure mode is used.

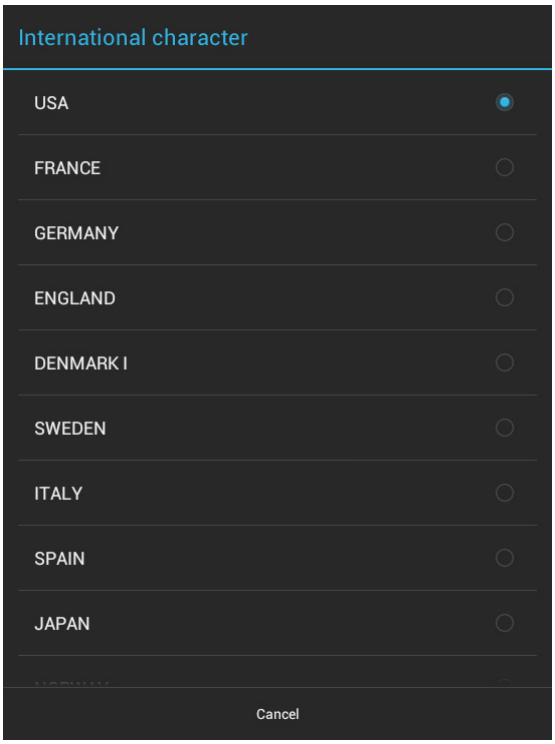

Figure 5-11

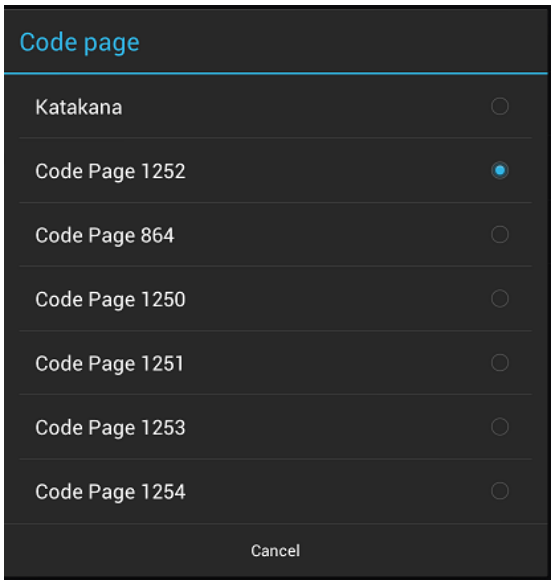

Figure 5-12

5.4 Precaution

No guarantee of proper operation and support are provided for sample programs.

Sample programs are subject to change without notice.

Chapter 6
Disclaimer

We closely monitor the development of this software in order to avoid problems. However, we are not responsible for any damages arising out of the use of this software.

Appendix A
Character Sets (Character Code Table)

A.1 Character Code Table (Codepage)

It is possible to use the following codepages when Mobile printer is used.
(When **COUNTRY_USA** is set in the setting of international character set)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	”	#	$	%	&	’	(	)	*	+	,	−	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	
80																
90																
A0		。	「	」	、	・	ヲ	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
B0	−	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
C0	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
D0	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	゛	゜
E0																
F0																

Figure A-1 CODE_PAGE_KATAKANA

Figure A-2 CODE_PAGE_1252 (Latin)

It is possible to use the following codepages when POS printer is used.
(When **COUNTRY_USA** is set in the setting of international character set)

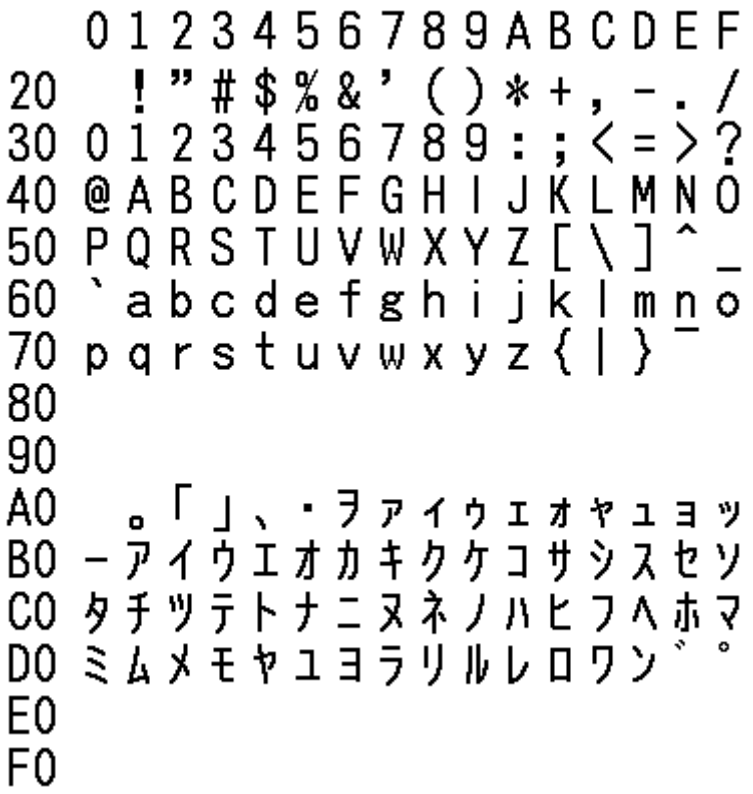

Figure A-3 CODE_PAGE_KATAKANA

Figure A-4 CODE_PAGE_1252 (Latin)

Figure A-5 CODE_PAGE_864 (Arabic)

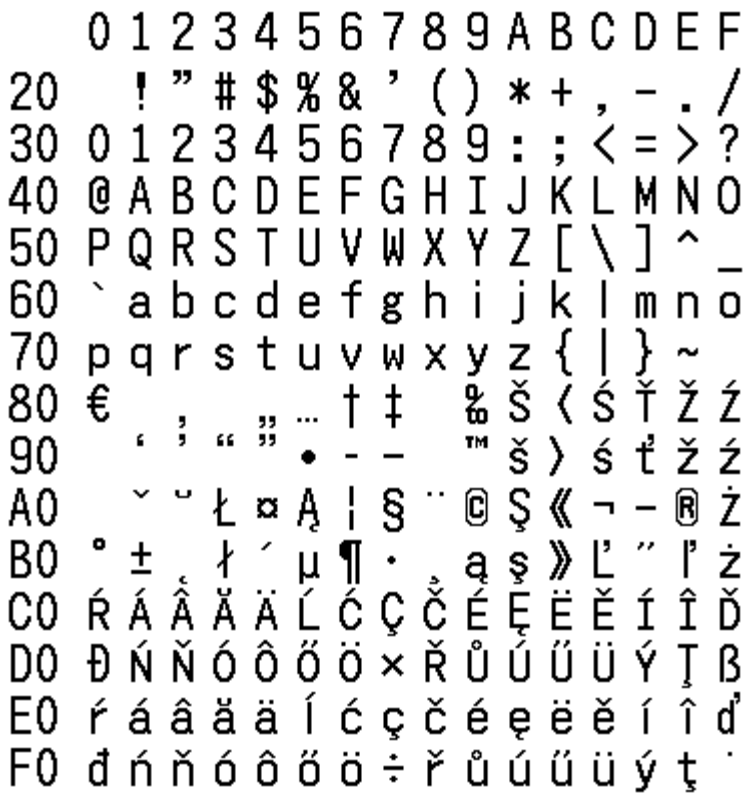

Figure A-6 CODE_PAGE_1250 (Central European)

Figure A-7 CODE_PAGE_1251 (Cyrillic)

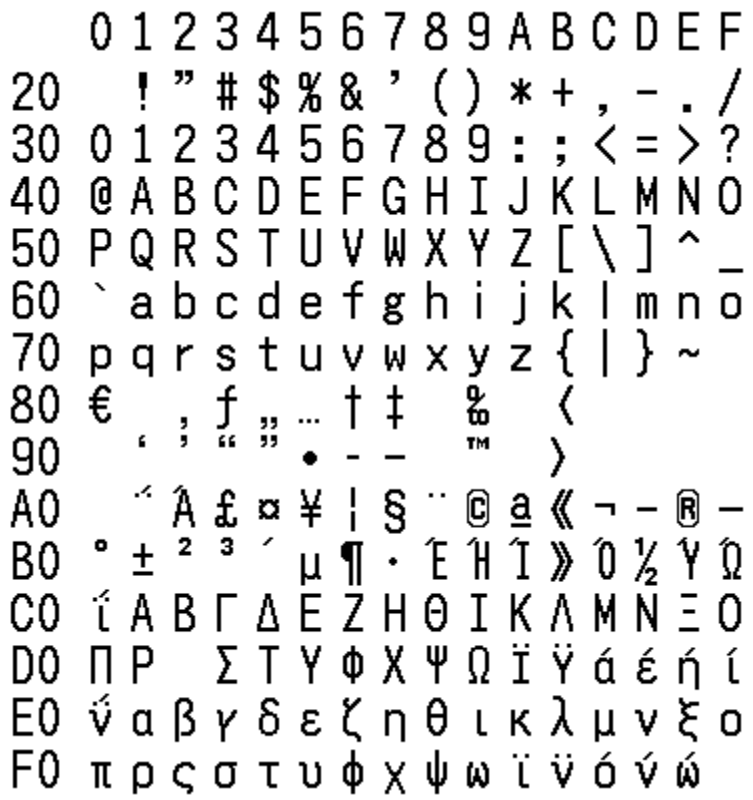

Figure A-8 CODE_PAGE_1253 (Greek)

Figure A-9 CODE_PAGE_1254 (Turkish)

A.2 International Character Set

The following codes differ depending on the specified international character set.

	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
COUNTRY_USA	#	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_FRANCE	#	$	à	°	ç	§	^	`	é	ù	è	¨
COUNTRY_GERMANY	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
COUNTRY_ENGLAND	£	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_DENMARK_1	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
COUNTRY_SWEDEN	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
COUNTRY_ITALY	#	$	@	°	\	é	^	ù	à	ò	è	ì
COUNTRY_SPAIN	₧	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~
COUNTRY_JAPAN	#	$	@	[	¥	]	^	`	{	|	}	~
COUNTRY_NORWAY	#	¤	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_DENMARK_2	#	$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
COUNTRY_SPAIN_2	#	$	á	¡	Ñ	¿	é	`	í	ñ	ó	ú
COUNTRY_LATIN_AMERICA	#	$	á	¡	Ñ	¿	é	ü	í	ñ	ó	ú
COUNTRY_ARABIA	#	$	@	[	\	]	^	`	{	|	}	~

Figure A-10 International Character Set