



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

U00139060109

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
U00128560409	December 2016
U00139060100	February 2017
U00139060101	March 2017
U00139060102	April 2017
U00139060103	January 2018
U00139060104	March 2019
U00139060105	July 2019
U00139060106	August 2019
U00139060107	October 2019
U00139060108	March 2020
U00139060109	June 2020

Copyright © 2012-2020 by Seiko Instruments Inc.
All rights reserved.

Android™ is a trademark of Google LLC.
Bluetooth® is a 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.

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
			DPU-S245-01C-E
		USB	DPU-S245-0xA-E
			DPU-S245-0xB-E
			DPU-S245-0xC-E
	DPU-S445	Bluetooth	DPU-S445-01A-E
			DPU-S445-01B-E
			DPU-S445-01C-E
		USB	DPU-S445-0xA-E
			DPU-S445-0xB-E
			DPU-S445-0xC-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	1-1
1.1	Function Provided by the SDK.....	1-1
1.2	SII Print Class Library Overview	1-1
1.2.1	SII Print Class Library Configuration	1-1
1.2.2	Function Provided by the Library.....	1-2
Chapter 2	Product Specification	2-1
2.1	Product Specification	2-1
2.1.1	Applicable OS versions	2-1
2.1.2	Operating Conditions.....	2-2
2.1.3	Precaution	2-2
Chapter 3	How to Use the library	3-1
3.1	Development Environment for Android Application	3-1
3.2	Provided Files	3-2
3.3	Add the Library to Android Studio Projects	3-3
3.4	Use Developed Android Application on Android Device	3-5
Chapter 4	Function of the Library	4-1
4.1	Standard Mode and Page Mode	4-1
4.1.1	Basic Operation.....	4-1
4.1.2	Text Data Printing in Standard Mode	4-3
4.1.3	Mapping Position of Print Data in Page Mode.....	4-4
4.1.4	Print Data Process at Out of Print Area of Page Mode	4-6
4.2	Package of the Library.....	4-7
4.3	API Reference	4-8
4.3.1	PrinterManager Class.....	4-8
	(a) Common method to standard mode and page mode	4-26
	PrinterManager Constructor	4-26
	connect Start communicating with a printer (Bluetooth) ..	4-26
	connect Start communicating with a printer (USB)	4-27
	connect Start communicating with a printer (TCP/IP)	4-27
	disconnect Disconnect a printer.....	4-28
	cutPaper Cut paper	4-28
	openDrawer Open cash drawer	4-28
	buzzer Sound buzzer.....	4-29
	externalBuzzer Sound external buzzer.....	4-29
	getStatus Obtain printer status	4-29
	setCallbackFunctionListener	
	Start/End callback of printer status change	4-31

abort	Abort the waiting state of a printer	4-31
registerLogo	Register logo (image) to a printer	4-32
registerLogo	Register logo (image) to a printer	4-32
unregisterLogo	Delete specified logo (image) in a printer	4-33
unregisterLogo	Delete specified logo (image) in a printer	4-33
registerStyleSheet	Register style sheet to a printer	4-33
unregisterStyleSheet	Delete specified style sheet in a printer	4-34
resetPrinter	Printer hardware reset	4-34
getPrinterResponse	Obtain various responses from a printer	4-34
startDiscoveryPrinter	Start printer search (Bluetooth)	4-36
startDiscoveryPrinter	Start printer search (USB)	4-37
startDiscoveryPrinter	Start printer search (TCP/IP)	4-37
cancelDiscoveryPrinter	Cancel printer search	4-38
getFoundPrinter	Obtain searched printer information	4-38
getSendTimeout	Obtain send timeout period	4-38
setSendTimeout	Set send timeout period.....	4-38
getReceiveTimeout	Obtain receive timeout period.....	4-39
setReceiveTimeout	Set receive timeout period.....	4-39
getInternationalCharacter	Obtain international character set.....	4-39
setInternationalCharacter	Set international character set.....	4-40
getCodePage	Obtain codepage	4-40
setCodePage	Set codepage	4-40
getPrinterModel	Obtain printer model	4-41
getPortType	Obtain connection port type.....	4-41
isConnect	Verify connection state with a printer	4-41
getSocketKeepingTime	Obtain socket keeping time	4-42
setSocketKeepingTime	Set socket keeping time	4-42
controlTransaction	Start/End batch processing	4-42
(b) Dedicated method for standard mode	4-44
sendText	Send text data	4-44
sendTextEx	Send format specified text data.....	4-44
sendTextEx	Send format specified text data	4-45
printBarcode	Print barcode	4-46
printPDF417	Print PDF417	4-48
printQRcode	Print QR code	4-49
printDataMatrix	Print Data Matrix.....	4-49
printMaxiCode	Print MaxiCode	4-50
sendBinary	Send binary data.....	4-50
sendDataFile	Send specified file	4-51
printLogo	Print specified logo (image) in printer	4-52
printLogo	Print specified logo (image) in printer	4-52
(c) Dedicated method for page mode.....		4-53

enterPageMode	Start page mode	4-54	
exitPageMode	End page mode	4-54	
setPageModeArea	Specify print area of page mode.....	4-54	
setPageModeDirection	Specify print direction of page mode	4-56	
setPageModeLineSpacing	Specify line spacing of page mode.....	4-56	
printPageMode	Print page mode	4-56	
printPageModeText	Send text data of page mode	4-57	
printPageModeTextEx	Send format specified text data of page mode...	4-57	
printPageModeBarcode	Print barcode of page mode	4-58	
printPageModePDF417	Print PDF417 of page mode.....	4-60	
printPageModeQRcode	Print QR Code of page mode	4-61	
printPageModeDataMatrix	Print Data Matrix of page mode.....	4-62	
printPageModeMaxiCode	Print MaxiCode of page mode	4-62	
sendPageModeBinary	Send binary data of page mode	4-63	
printPageModelImageFile	Draw Image file of page mode	4-63	
printPageModeRectangle	Draw rectangle image of page mode	4-64	
printPageModeLine	Print ruled line of page mode.....	4-65	
printPageModeLogo	Print logo of page mode	4-67	
4.3.2 PrinterEvent Class.....		4-68	
getEventType	Obtain event type.....	4-68	
4.3.3 PrinterListener Interface		4-69	
finishEvent	Finish event of the printer search	4-69	
4.3.4 PrinterInfo Class		4-70	
getPrinterModelName	Obtain printer model name	4-70	4-70
getBluetoothAddress	Obtain Bluetooth address	4-70	
getMacAddress	Obtain MAC address	4-70	
getIpAddress	Obtain IP address.....	4-71	
getIsBonded	Obtain pairing status.....	4-71	
getDevicePath	Obtain device path.....	4-71	
4.3.5 PrinterException Class		4-72	
PrinterException	Constructor	4-74	
PrinterException	Constructor	4-74	
getErrorCode	Obtain error codes.....	4-74	
4.3.6 CallbackFunctionListener Interface		4-75	
onStatusChanged	Change event of printer status	4-75	
4.3.7 BarcodeScannerListener Interface.....		4-75	

Chapter 5	Sample Program	5-1
<hr/>		
5.1	Installation.....	5-1
5.2	Screen	5-3
5.2.1	Main screen.....	5-3
5.2.2	[SETTINGS] screen	5-4
5.3	Precaution.....	5-4
Chapter 6	Disclaimer	6-1
<hr/>		
Appendix A	Character Sets (Character Code Table)	A-1
<hr/>		
A.1	Character Code Table (Codepage)	A-1
A.2	International Character Set.....	A-10

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 Android Studio projects as a sample program.

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 program 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.

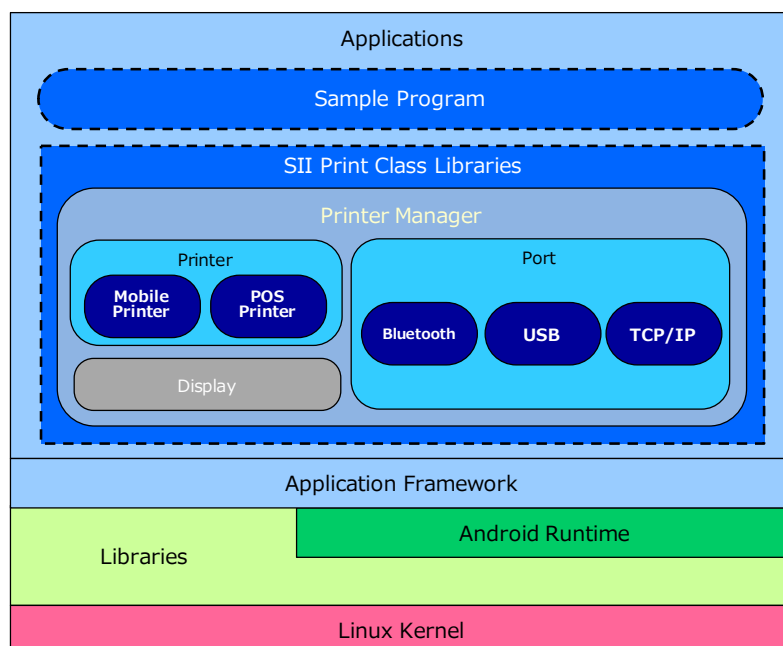


Figure 1-1

1.2.2 Function Provided by the Library

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 and/or commands*¹)
- Barcode print and 2-dimensional barcode print
- Sending a data file to a printer (print data and/or commands*¹)
- Cut paper
- Obtaining printer status
- Aborting the waiting state of a printer
- Obtaining various responses from a printer
- Bulk registration of print commands
- Registering a printer status call back function
- Printer search by Bluetooth or TCP/IP

*1: Commands that retrieve the responses 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".

(NOTE) Mobile printer and POS printer do not support the APIs of Display or the barcode scanners.

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, USB, TCP/IP : Android 5.0 (API 21) to Android 10.0 (API 29)

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 In the case of connecting with the printer in secure mode, set 1.
In the case of 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 TCP/IP connection is used in this library, the communication port cannot be shared with printer drivers or other libraries.

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

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

When TCP/IP connection is used in this library, Android device where Wireless LAN access point is connecting to and POS printer need to be connected to the same network.

Chapter 3

How to Use the Library

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

3.1 Development Environment for Android Application

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

- Android Studio

<https://developer.android.com/studio/index.html>

- USB driver for Windows (When develop on Windows environment)

<https://developer.android.com/studio/run/oem-usb.html>

In this document, in following sections are based on the premise that environments for using each tool have been set up.

3.2 Provided Files

The file configuration of the SDK is as follows.

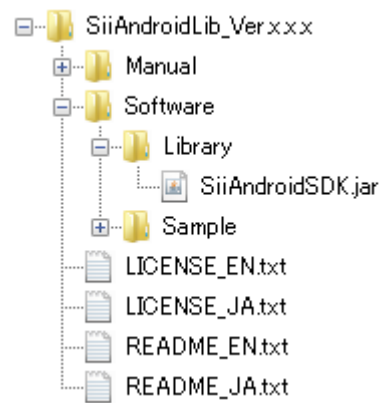


Figure 3-1

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

3.3 Add the Library to Android Studio Projects

Using the project of the sample program included in the SDK as an example, this section describes how to add the library to Android Studio projects.

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

- (1) Select and right click the module (app) displayed in the Android Project view of Android Studio, select [New] and [Directory] (Figure 3-2), and enter "libs" in the folder name to create a folder. (Figure 3-3)

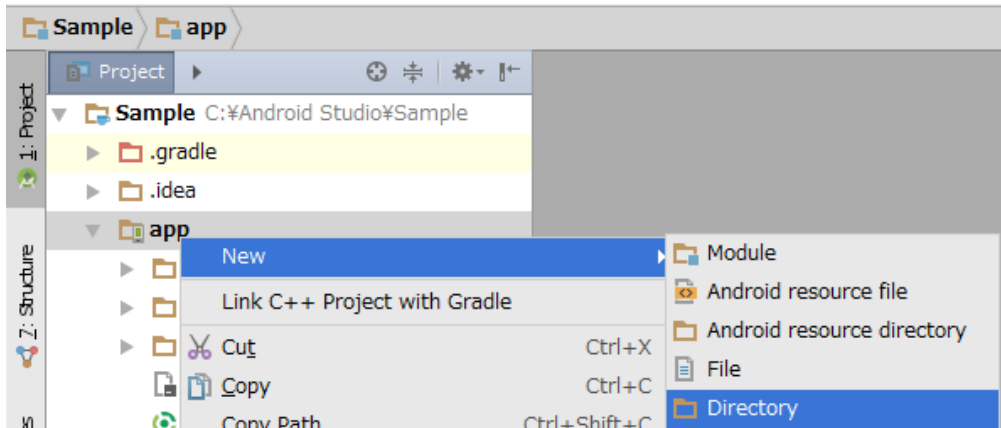


Figure 3-2

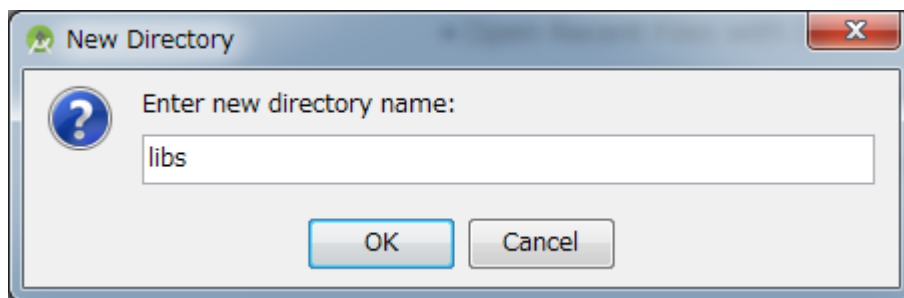


Figure 3-3

- (2) Copy the library file (SiiAndroidSDK.jar) into the folder (\Sample\app\libs) created in step (1).

- (3) After adding the library, the view looks like Figure 3-4.

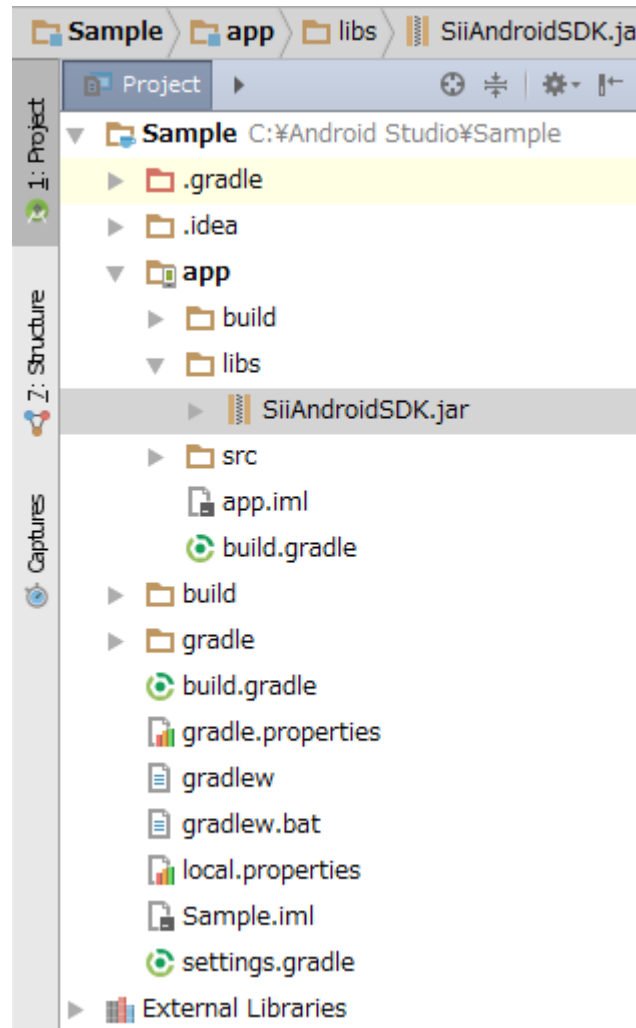


Figure 3-4

- (4) Add the following codes at the beginning of the main source file.
(Add at the beginning of MainActivity.java for the sample program.)

```
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.

3.4 Use Developed Android Application on Android Device

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

(NOTE) This procedure is based on the menu of Android 7.1. Menu contents may vary depending on the using Android device.

- (1) Select [Settings], [Security], and turn on [Unknown sources]. (Figure 3-5)

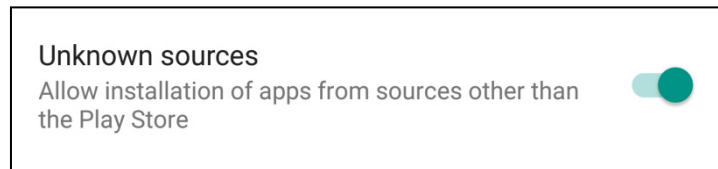


Figure 3-5

- (2) Select [Settings], [Developer options], and turn on [USB debugging]. (Figure 3-6)

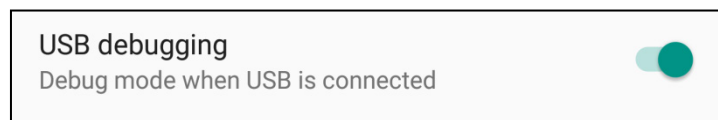


Figure 3-6

Chapter 4

Function of the Library

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

4.1 Standard Mode and Page Mode

4.1.1 Basic Operation

There are two printing modes "Standard mode" and "Page mode" in the library. The "Standard mode" and "Page mode" are described below.

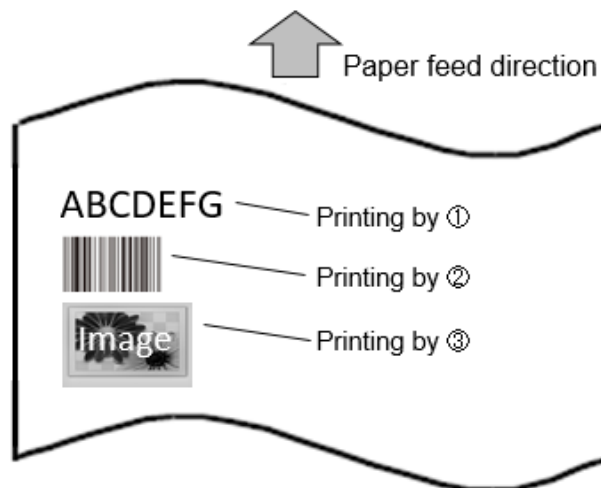
"Page mode" is supported only by POS printer.

(1) Standard mode

Standard mode is the mode to perform the printing in sequence.

Sample print command

- ① Send text data
- ② Print barcode
- ③ Send specified file (Specify an image file)



Standard mode suits the printing with an unfixed length such as a receipt.

(2) Page mode

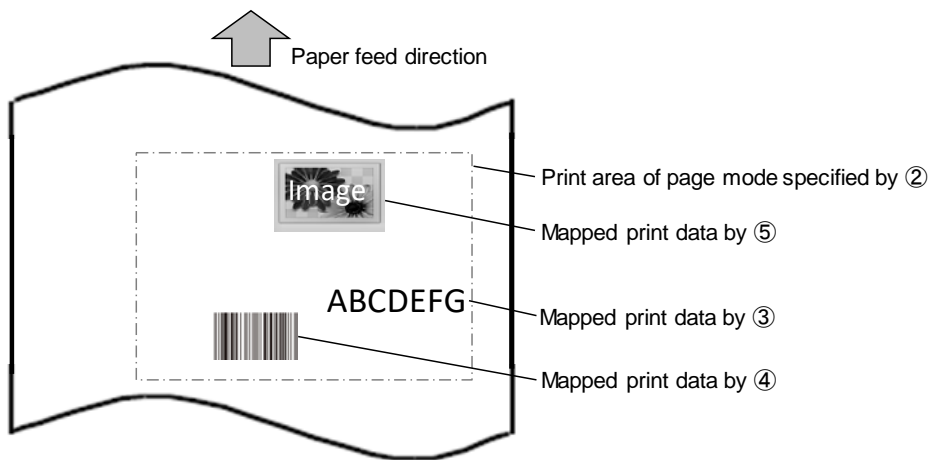
Page mode is the mode to perform the printing on a per-page basis.

In page mode, the print area of page mode is allocated at first, and then print data is mapped on an arbitrary position of the print area.

The mapped print data is printed by the print method of page mode.

Sample print command

- ① Start page mode
- ② Specify print area of page mode
- ③ Send text data of page mode
- ④ Print barcode of page mode
- ⑤ Draw image file of page mode
- ⑥ Print page mode (print the data of ③④⑤ on the print area of ②)
- ⑦ End page mode



Page mode suits the printing for the followings.

- The printing with a fixed length.
- The printing with the coordinate determination of the character starting position or the ruled line
- printing position.

4.1.2 Text Data Printing in Standard Mode

The text data in standard mode is printed each one line.

The text data is stored in the printer when the text data less than one line is specified.

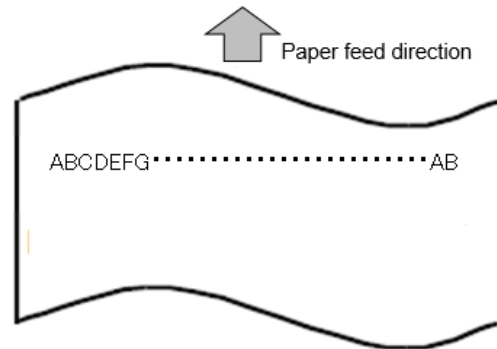
The stored text data is printed by either the following conditions.

- The text data filling for one line is specified.
- Line Feed code is specified.

- The print process when the text data filling for one line is specified.

ABCDEFGH.....ABCD
The text data filling for one line is specified.
(The continuous data is stored in the printer.)

Input data

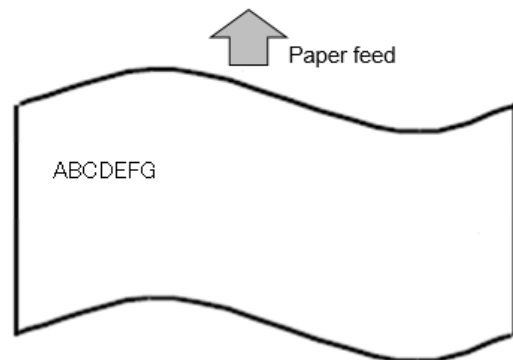


Print result

- The print process when Line Feed code is specified.

ABCDEFGH\nABCD
Line Feed code is specified.
(The continuous data is stored in the printer.)

Input data



Print result

4.1.3 Mapping Position of Print Data in Page Mode

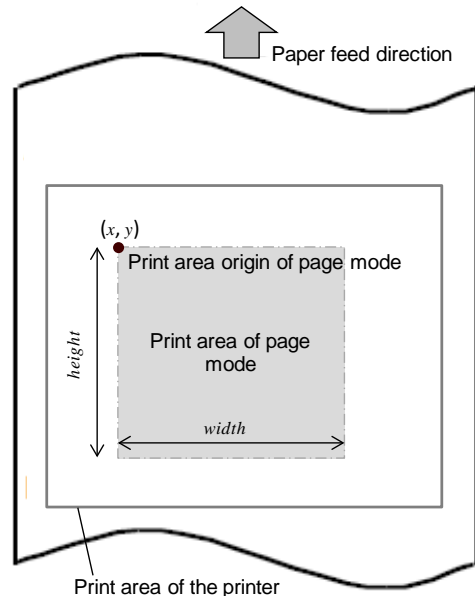
In page mode, the mapping position of print data is determined by print area, print direction, and reference point.

This section describes the print area, print direction, and reference point.

(1) Print area of page mode

The print area of page mode is specified against the print area of the printer by the print area origin, and the width and the height of page mode. The view of the print area is shown in the following figures.

The print area of page mode can be specified more than one.

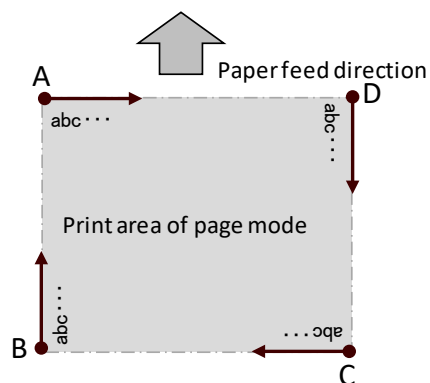


(2) Print direction

Specify the print direction at setting the print area of page mode.

The starting point is changed depending on specifying the print direction for each direction.

The relation between the print direction and the starting point is shown in the figure below.

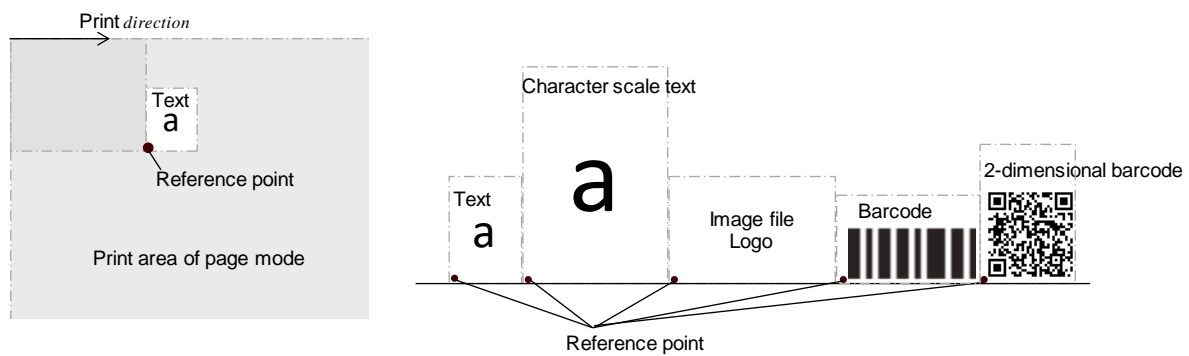


- Starting point: Upper left (A on the figure),
- Starting point: Left below (B on the figure),
- Starting point: Right below (C on the figure),
- Starting point: Upper right (D on the figure),

- Print direction: Left to Right
- Print direction: Below to Upper
- Print direction: Right to Left
- Print direction: Upper to Below

(3) Reference point

The relation between the reference point for mapping data and each print element (text, image file, logo, and barcode, etc.) is shown in the figures below.



(NOTE) The reference point cannot be specified out of the print area of page mode.

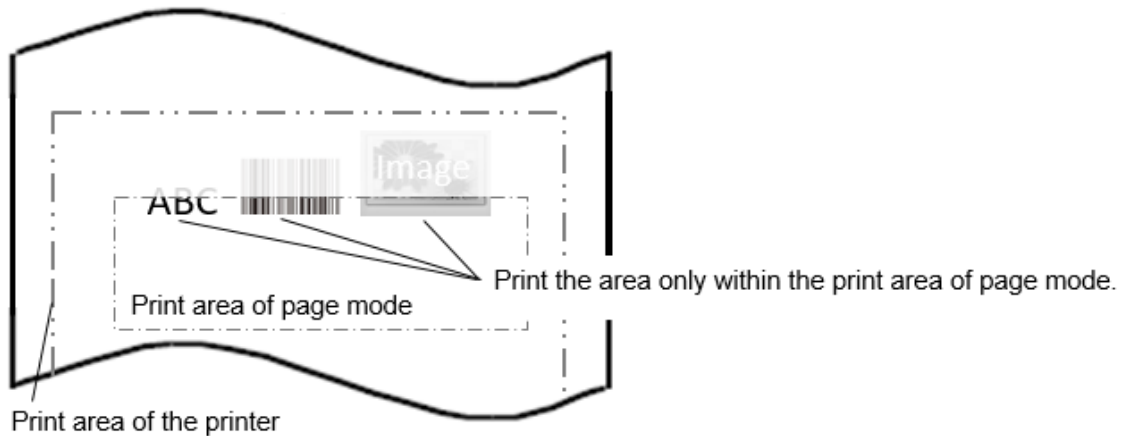
4.1.4 Print Data Process at Out of Print Area of Page Mode

This section describes the process when mapped data is to be mapped on out of the print area of page mode.

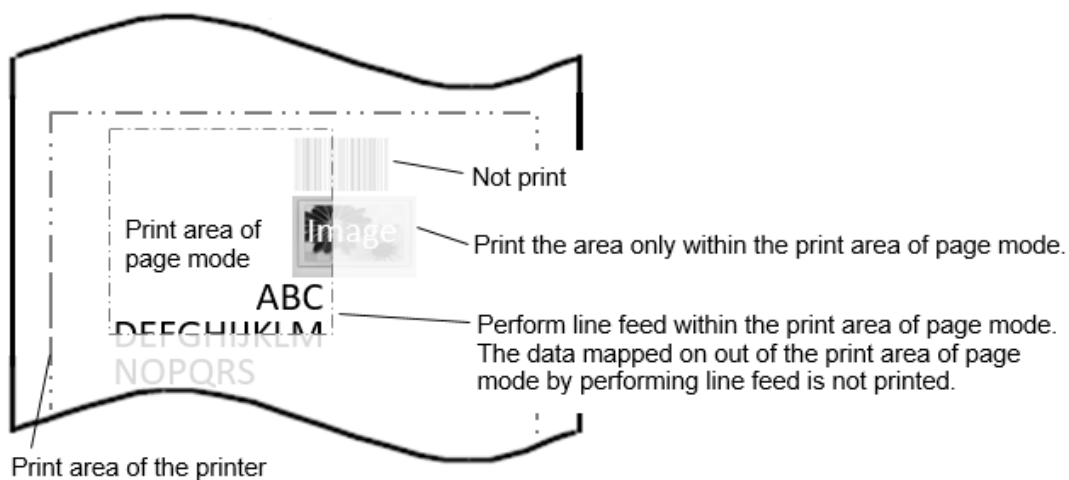
Type of Print Data

Text	Barcode, 2-dimensional Barcode	Image File, Logo, Rectangle, Ruled Line
ABC		

- (1) The print data is mapped on the upper of the print area of page mode.



- (2) The print data is mapped on the right of print area of page mode.



(NOTE) Read error or incorrect reading may occur when the part of mapped barcode data is on out of the print area of page mode.

4.2 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.
CallbackFunctionListener	Interface for getting the change event of printer status.
BarcodeScannerListener	Interface for getting barcode scanner connection or barcode scanner disconnection, or received barcode data.

See "4.3 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;
```


4.3 API Reference

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

Class Name	Description
PrinterManager	See "4.3.1 PrinterManager Class ".
PrinterEvent	See "4.3.2 PrinterEvent Class ".
PrinterListennr	See "4.3.3 PrinterListener Interface ".
PrinterInfo	See "4.3.4 PrinterInfo Class ".
PrinterException	See "4.3.5 PrinterException Class ".
CallbackFunctionListener	See "4.3.6 CallbackFunctionListener Interface ".
BarcodeScannerListener	See "4.3.5 BarcodeScannerListenerPrinterException Class ".

(NOTE) Mobile printer and POS printer do not support the APIs of Display or the barcode scanners.

4.3.1 PrinterManager Class

(1) Method List

Methods provided by the **PrinterManager** class are shown in the following table.

"Standard mode" or "Page mode" can be selected in the **PrinterManager** class.

Available method differs depending on the target printer: Mobile printer or POS printer.

Method	Description
Common method to standard mode and page mode	The valid methods in standard mode and page mode. See "4.3.1(1)(a) Common method to standard mode and page mode" for the methods.
Dedicated method for standard mode	The valid methods in standard mode. See "4.3.1(1)(b) Dedicated method for standard mode" for the methods.
Dedicated method for page mode	The valid methods in page mode. See "4.3.1(1)(c) Dedicated method for page mode" for the methods.

(a) Common method to standard mode and page mode

Methods provided by the common method to standard mode and page mode are shown in Table 4-1. See "4.3.1(4)(a) Common method to standard mode and page mode" for details of the common methods.

**Table 4-1 Common Method to Standard Mode and Page Mode
in 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
cutPaper	Cut paper	Not supported	Supported
openDrawer	Open cash drawer	Not supported	Supported
buzzer	Sound buzzer	Not supported	Supported
externalbuzzer	Sound external buzzer	Not supported	Not supported
getStatus	Obtain printer status	Supported	Supported
setCallbackFunctionListener	Start/End callback of printer status change	Supported	Supported
abort	Abort the waiting state of a printer	Supported	Supported
registerLogo	Register logo (image) to a printer	Supported ^{*1}	Supported ^{*1}
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 ^{*1}	Supported ^{*1}
startDiscoveryPrinter	Start printer search (Bluetooth)	Supported	Supported
startDiscoveryPrinter	Start printer search (USB)	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

Method	Function Summary	Target	
		Mobile	POS
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
controlTransaction	Start/Finish batch processing	Supported	Supported

*1: Provided function varies depending on the target printer.

(b) Dedicated method for standard mode

Methods provided by the dedicated method for standard mode are shown in Table 4-2.

See "4.3.1(4)(b) Dedicated method for standard mode" for details of the dedicated methods.

**Table 4-2 Dedicated Method for Standard Mode
in PrinterManager Class**

Method	Function Summary	Target	
		Mobile	POS
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
printDataMatrix	Print Data Matrix	Supported	Supported
printMaxiCode	Print MaxiCode	Supported	Supported
sendBinary	Send binary data	Supported	Supported
sendDataFile	Send specified file	Supported*1	Supported*1
printLogo	Print specified logo (image) in printer	Supported	Not supported
printLogo	Print specified logo (image) in printer	Not supported	Supported

*1: Provided function varies depending on the target printer.

(c) Dedicated method for page mode

Methods provided by the dedicated method for page mode are shown in Table 4-3.

See "4.3.1(4)(c) Dedicated method for page mode " for details of the specified methods.

**Table 4-3 Dedicated Method for Page Mode
in PrinterManager Class**

Method	Function Summary	Target	
		Mobile	POS
enterPageMode	Start page mode	Not supported	Supported
exitPageMode	End page mode	Not supported	Supported
setPageModeArea	Specify print area of page mode	Not supported	Supported
setPageModeDirection	Specify print direction of page mode	Not supported	Supported
setPageModeLineSpacing	Specify line spacing of page mode	Not supported	Supported
printPageMode	Print page mode	Not supported	Supported
printPageModeText	Send text data of page mode	Not supported	Supported
printPageModeTextEx	Send format specified text data of page mode	Not supported	Supported
printPageModeBarcode	Print barcode of page mode	Not supported	Supported
printPageModePDF417	Print PDF417 of page mode	Not supported	Supported
printPageModeQRcode	Print QR Code of page mode	Not supported	Supported
printPageModeDataMatrix	Print Data Matrix of page mode	Not supported	Supported
printPageModeMaxiCode	Print MaxiCode of page mode	Not supported	Supported
sendPageModeBinary	Send binary data of page mode	Not supported	Supported
printPageModeImageFile	Draw image file of page mode	Not supported	Supported
printPageModeRectangle	Draw rectangle image of page mode	Not supported	Supported
printPageModeLine	Print ruled line of page mode	Not supported	Supported
printPageModeLogo	Print logo of page mode	Not supported	Supported

(2) Constant List

(a) International Character Set Constant

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

Table 4-4 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 I	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 for setting/ obtaining codepage are shown in Table 4-5.

Table 4-5 Codepage Constant

Constant Name	Description	Value	Target	
			Mobile	POS
CODE_PAGE_437	USA, Standard Europe (Code Page437)	0	Not supported	Supported
CODE_PAGE_KATAKANA	Katakana	1	Supported	Supported
CODE_PAGE_850	Multilingual (Code Page850)	2	Not supported	Supported
CODE_PAGE_860	Portuguese (Code Page860)	3	Not supported	Supported
CODE_PAGE_863	Canadian-French (Code Page863)	4	Not supported	Supported
CODE_PAGE_865	Nordic (Code Page865)	5	Not supported	Supported
CODE_PAGE_1252	Latin (Code Page1252)	16	Supported	Supported
CODE_PAGE_852	Eastern Europe (Code Page852)	18	Not supported	Supported
CODE_PAGE_858	Euro (Code Page858)	19	Not supported	Supported
CODE_PAGE_864*1	Arabic (Code Page864)	37	Not supported	Supported

Constant Name	Description	Value	Target	
			Mobile	POS
CODE_PAGE_1250	Central European (Code Page1250)	45	Not supported	Supported
CODE_PAGE_1251	Cyrillic (Code Page1251)	46	Not supported	Supported
CODE_PAGE_1253	Greek (Code Page1253)	47	Not supported	Supported
CODE_PAGE_1254	Turkish (Code Page1254)	48	Not supported	Supported

Note: This library does not support other than above codepage.

*1: When **CODE_PAGE_864** is specified, 20ACh of the Unicode cannot be printed.

(c) Constant for barcode or PDF417

Constants used for printing barcode or PDF417 are shown in Table 4-6.

Table 4-6 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 for obtaining printer model are shown in Table 4-7.

Table 4-7 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 for obtaining connection port type are shown in Table 4-8.

Table 4-8 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 for obtaining various responses from a printer are shown in Table 4-9.

Table 4-9 Response Type Constant

Constant Name	Description	Value	Target	
			Mobile	POS
PRINTER_RESPONSE_REQUEST	Execution response request	0	Supported	Supported
PRINTER_RESPONSE_USER_AREA	Send remaining capacity of user area	1	Supported	Supported
PRINTER_RESPONSE_ARRANGE_USER_AREA	Send remaining capacity of user area after defragment	2	Not supported	Supported
PRINTER_RESPONSE_NV_GRAPHICS	Send NV graphics memory capacity	3	Not supported	Supported
PRINTER_RESPONSE_KEY_CODE	Send key code list of defined NV graphics	4	Not supported	Supported
PRINTER_RESPONSE_BATTERY_STATUS	Battery voltage state	5	Supported	Not supported
PRINTER_RESPONSE_EXTERNAL_RAM	Send Remaining RAM 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 4-10.

Table 4-10 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 4-11.

Table 4-11 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 4-12.

Table 4-12 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 4-13.

Table 4-13 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 4-14.

Table 4-14 Character Scale (CharacterScale)

Constant Name	Description	Target	
		Mobile	Mobile
VERTICAL_1_HORIZONTAL_1	Height × 1 and width × 1	Supported	Supported
VERTICAL_1_HORIZONTAL_2	Height × 1 and width × 2	Supported	Supported
VERTICAL_1_HORIZONTAL_3	Height × 1 and width × 3	Not supported	Supported
VERTICAL_1_HORIZONTAL_4	Height × 1 and width × 4	Not supported	Supported
VERTICAL_2_HORIZONTAL_1	Height × 2 and width × 1	Supported	Supported
VERTICAL_2_HORIZONTAL_2	Height × 2 and width × 2	Supported	Supported
VERTICAL_2_HORIZONTAL_3	Height × 2 and width × 3	Not supported	Supported
VERTICAL_2_HORIZONTAL_4	Height × 2 and width × 4	Not supported	Supported
VERTICAL_2_HORIZONTAL_6	Height × 2 and width × 6	Not supported	Supported
VERTICAL_3_HORIZONTAL_1	Height × 3 and width × 1	Not supported	Supported
VERTICAL_3_HORIZONTAL_2	Height × 3 and width × 2	Not supported	Supported
VERTICAL_3_HORIZONTAL_3	Height × 3 and width × 3	Not supported	Supported
VERTICAL_3_HORIZONTAL_4	Height × 3 and width × 4	Not supported	Supported
VERTICAL_4_HORIZONTAL_1	Height × 4 and width × 1	Not supported	Supported
VERTICAL_4_HORIZONTAL_2	Height × 4 and width × 2	Not supported	Supported
VERTICAL_4_HORIZONTAL_3	Height × 4 and width × 3	Not supported	Supported
VERTICAL_4_HORIZONTAL_4	Height × 4 and width × 4	Not supported	Supported
VERTICAL_4_HORIZONTAL_6	Height × 4 and width × 6	Not supported	Supported
VERTICAL_4_HORIZONTAL_8	Height × 4 and width × 8	Not supported	Supported
VERTICAL_6_HORIZONTAL_2	Height × 6 and width × 2	Not supported	Supported
VERTICAL_6_HORIZONTAL_4	Height × 6 and width × 4	Not supported	Supported
VERTICAL_6_HORIZONTAL_6	Height × 6 and width × 6	Not supported	Supported
VERTICAL_6_HORIZONTAL_8	Height × 6 and width × 8	Not supported	Supported
VERTICAL_8_HORIZONTAL_4	Height × 8 and width × 4	Not supported	Supported
VERTICAL_8_HORIZONTAL_6	Height × 8 and width × 6	Not supported	Supported
VERTICAL_8_HORIZONTAL_8	Height × 8 and width × 8	Not supported	Supported

(f) Alignment (**PrintAlignment**)

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

Table 4-15 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 4-16.

Table 4-16 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

See **printBarcode** or **printPageModeBarcode** for the type..

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

Enumerated type constants used for module size are shown in Table 4-17.

Table 4-17 Module Size (ModuleSize)

Constant Name	Description	Using Method	Target	
			Mobile*1	POS
BARCODE_MODULE_WIDTH_2	Fine element 2 dots Module width 0.250 mm	printBarcode printPageModeBarcode	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 printPageModePDF417	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 printPageModeQRcode	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
DATAMATRIX_MODULE_SIZE_2	2 dots	printDataMatrix printPageModeDataMatrix	Supported	Supported
DATAMATRIX_MODULE_SIZE_3	3 dots		Supported	Supported

Constant Name	Description	Using Method	Target	
			Mobile ^{*1}	POS
DATAMATRIX_MODULE_SIZE_4	4 dots	printDataMatrix printPageModeD ataMatrix	Supported	Supported
DATAMATRIX_MODULE_SIZE_5	5 dots		Supported	Supported
DATAMATRIX_MODULE_SIZE_6	6 dots		Supported	Supported
DATAMATRIX_MODULE_SIZE_7	7 dots		Supported	Supported
DATAMATRIX_MODULE_SIZE_8	8 dots		Supported	Supported
DATAMATRIX_MODULE_SIZE_9	9 dots		Supported	Supported
DATAMATRIX_MODULE_SIZE_10	10 dots		Supported	Supported
DATAMATRIX_MODULE_SIZE_11	11 dots		Supported	Supported

*1: The dedicated method for page mode is not supported by Mobile printer.

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

Enumerated type constants used for HRI character print position are shown in Table 4-18.

Table 4-18 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 4-19.

Table 4-19 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

Constant Name	Description	Target	
		Mobile	POS
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 4-20.

Table 4-20 Error Correction Level (ErrorCorrection)

Constant Name	Description	Using Method	Target	
			Mobile*1	POS
PDF417_ERROR_CORRECTION_0	Error correction level 0	printPDF417 printPageModePDF417	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 printPageModeQRcode	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

*1: The dedicated method for page mode is not supported by Mobile printer.

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

Enumerated type constants used for PDF417 symbol are shown in Table 4-21.

Table 4-21 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 4-22.

Table 4-22 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 4-23.

Table 4-23 Cutting Method (CuttingMethod)

Constant Name	Description	Target	
		Mobile	POS
CUT_FULL	Paper feed operation to the paper cut position Full cut	Not supported	Supported
CUT_PARTIAL	Paper feed operation to the paper cut position Partial cut	Not supported	Supported
CUT_NONE^{*1}	No cut	Not supported	Supported

^{*1}: Supported only by **printPageMode**.

(o) Data Matrix modules (**DataMatrixModule**)

Enumerated type constants used for Data Matrix module are shown in Table 4-24.

Table 4-24 Data Matrix modules (DataMatrixModule)

Constant Name	Description
DATA_MATRIX_AUTO	Module numbers: Automatic
DATA_MATRIX_10_10	Module numbers: 10×10
DATA_MATRIX_12_12	Module numbers: 12×12
DATA_MATRIX_14_14	Module numbers: 14×14
DATA_MATRIX_16_16	Module numbers: 16×16
DATA_MATRIX_18_18	Module numbers: 18×18
DATA_MATRIX_20_20	Module numbers: 20×20
DATA_MATRIX_22_22	Module numbers: 22×22
DATA_MATRIX_24_24	Module numbers: 24×24
DATA_MATRIX_26_26	Module numbers: 26×26
DATA_MATRIX_32_32	Module numbers: 32×32
DATA_MATRIX_36_36	Module numbers: 36×36
DATA_MATRIX_40_40	Module numbers: 40×40
DATA_MATRIX_44_44	Module numbers: 44×44
DATA_MATRIX_48_48	Module numbers: 48×48
DATA_MATRIX_52_52	Module numbers: 52×52
DATA_MATRIX_64_64	Module numbers: 64×64
DATA_MATRIX_72_72	Module numbers: 72×72
DATA_MATRIX_80_80	Module numbers: 80×80
DATA_MATRIX_88_88	Module numbers: 88×88
DATA_MATRIX_96_96	Module numbers: 96×96
DATA_MATRIX_104_104	Module numbers: 104×104
DATA_MATRIX_120_120	Module numbers: 120×120
DATA_MATRIX_132_132	Module numbers: 132×132
DATA_MATRIX_144_144	Module numbers: 144×144
DATA_MATRIX_8_18	Module numbers: 8×18
DATA_MATRIX_8_32	Module numbers: 8×32
DATA_MATRIX_12_26	Module numbers: 12×26
DATA_MATRIX_12_36	Module numbers: 12×36
DATA_MATRIX_16_36	Module numbers: 16×36
DATA_MATRIX_16_48	Module numbers: 16×48

(p) MaxiCode Mode (**MaxiCodeMode**)

Enumerated type constants used for MaxiCode mode are shown in Table 4-25.

Table 4-25 MaxiCode Mode (MaxiCodeMode)

Constant Name	Description
MAXI_CODE_2	Mode2
MAXI_CODE_3	Mode3
MAXI_CODE_4	Mode4
MAXI_CODE_5	Mode5

(q) Drawer number (**DrawerNum**)

Enumerated type constants used for drawer number are shown in Table 4-26.

Table 4-26 Drawer Number (DrawerNum)

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

(r) Activation pulse width (**PulseWidth**)

Enumerated type constants used for activation pulse width are shown in Table 4-27.

Table 4-27 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

(s) Dithering (**Dithering**)

Enumerated type constants used for dithering are shown in Table 4-28.

Table 4-28 Dithering (Dithering)

Constant Name	Description
DITHERING_DISABLE	Dithering disable
DITHERING_ERRORDIFFUSION	Dithering enable

(t) Batch processing selection (**TransactionFunction**)

Enumerated type constants used for batch processing selection are shown in Table 4-29.

Table 4-29 Batch Processing Selection (TransactionFunction)

Constant Name	Description
TRANSACTION_CLEAR	Cancel batch processing
TRANSACTION_START	Start batch processing
TRANSACTION_PRINT	Finish batch printing and batch processing

(u) Pending data output specifying (**OutputPendingData**)

Enumerated type constants used for pending data output specifying are shown in Table 4-30.

Table 4-30 Pending Data Output Specifying (OutputPendingData)

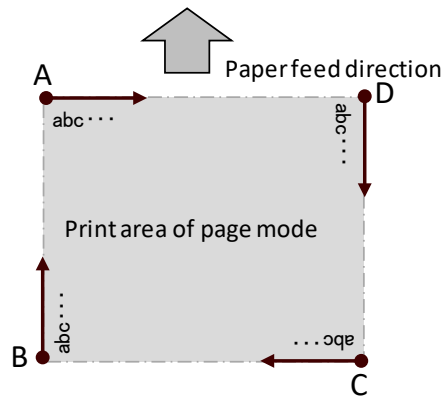
Constant Name	Description
PENDING_DATA_OUTPUT_FIRST	Output pending data at first and start the processing
PENDING_DATA_OUTPUT_TOGETHER	Output pending data at the same time as the processing

(v) Print direction (**Direction**)

Constants of enumerated type used for print direction in page mode are shown in Table 4-31.

Table 4-31 Print direction (Direction)

Constant Name	Description
DIRECTION_LEFT_TO_RIGHT	Starting point: Upper left (A on the figure), Print direction: Left to Right
DIRECTION_BOTTOM_TO_TOP	Starting point: Left below (B on the figure), Print direction: Below to Upper
DIRECTION_RIGHT_TO_LEFT	Starting point: Right below (C on the figure), Print direction: Right to Left
DIRECTION_TOP_TO_BOTTOM	Starting point: Upper right (D on the figure), Print direction: Upper to Below



(w) Line style (**LineStyle**)

Constants of enumerated type used for line style in page mode are shown in Table 4-32.

Table 4-32 Line style (LineStyle)

Constant Name	Description
LINestyle_THIN	Thin solid line (2 dots)
LINestyle_MEDIUM	Medium solid line (4 dots)
LINestyle_THICK	Thick solid line (8 dots)

(4) Method Details

(a) Common method to standard mode and page mode

The following methods are valid in standard mode and page mode. Standard mode is set immediately after **connect** method is executed.

PrinterManager		Constructor
Target	Mobile printer / POS printer	
Syntax	(a) public PrinterManager (Context <i>context</i>) (b) public PrinterManager ()	
Parameter	<i>context</i>	Specify application context to call this method. Example: MainActivity.this
Note	Use the syntax (a) when this method is used newly. Syntax (b) is a method that will be unsupported in the future.	

connect		Start communicating with a printer (Bluetooth)
Target	Mobile printer / POS printer	
Syntax	(a) public void connect (int <i>printerModel</i> , String <i>address</i>) throws PrinterException (b) public void connect (int <i>printerModel</i> , String <i>address</i> , boolean <i>secure</i>) throws PrinterException	
Parameter	<i>printerModel</i> <i>address</i> <i>secure</i>	Printer model constant for Bluetooth connection Bluetooth address Example: "00:11:22:AA:BB:CC" true Connect to a printer in secure mode false Connect to a printer in insecure mode
Description	<p>This method starts communication between Android device and a printer through Bluetooth connection. Call this method before using other methods of this class.</p> <p>This method connects to the Bluetooth address specified by <i>address</i> parameter. Also, printer default settings are executed based on <i>printerModel</i> parameter specified at the connection. See Table 4-7 Printer Model Constant for available printer model constant.</p> <p>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 <i>secure</i> parameter. Normally, it is recommended that the connection in secure mode be used.</p> <p>Monitoring of the printer status is started with this method. The latest printer status can be retrieved from getStatus method.</p> <p>Changes of the printer status can be notified as events by onStatusChanged method and setCallbackFunctionListener method.</p>	
Error	PrinterException may be thrown when the method is called.	

connect	Start communicating with a printer (USB)
---------	------------------------------------------

Target	Mobile printer / POS printer
Syntax	(a) public void connect (int <i>printerModel</i>) throws PrinterException (b) public void connect (int <i>printerModel</i> , Context <i>context</i>) throws PrinterException
Parameter	<i>printerModel</i> Printer model constant for USB connection <i>context</i> Specify application context to call this method. Example: MainActivity.this
Description	<p>This method starts communication between Android device and a printer through USB connection. Call this method before using other methods of this class.</p> <p>This method connects to a printer specified by <i>printerModel</i> parameter. Also, printer default settings are executed based on <i>printerModel</i> parameter specified at the connection. See Table 4-7 Printer Model Constant for available printer model constant.</p> <p>Monitoring of the printer status is started with this method. The latest printer status can be retrieved from getStatus method.</p> <p>Changes of the printer status can be notified as events by onStatusChanged method and setCallbackFunctionListener method.</p>
Error	PrinterException may be thrown when the method is called.
Note	Use the syntax (a) when using this method newly. Syntax (b) is a method that will be unsupported in the future.

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

Target	POS printer
Syntax	public void connect (int <i>printerModel</i> , String <i>address</i>) throws PrinterException
Parameter	<i>printerModel</i> Printer model constant for Ethernet connection <i>address</i> IP address Example: "192.168.0.190"
Description	<p>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.</p> <p>This method connects to the IP address specified by <i>address</i> parameter. TCP port number 9100 and 26100 are used for communication. Also, printer default settings are executed based on <i>printerModel</i> parameter specified at the connection. See Table 4-7 Printer Model Constant for available printer model constant.</p> <p>Monitoring of the printer status is started with this method. The latest printer status can be retrieved from getStatus method.</p> <p>Changes of the printer status can be notified as events by onStatusChanged method and setCallbackFunctionListener method.</p> <p><Introduction to creating/deleting a socket during connection with TCP/IP in the Library></p> <p>The library retains the created socket until disconnect method after connect method. Also, it is impossible to connect to the same printer from another application until disconnect method.</p>

Based on the completion of data transmission to the printer, the socket is deleted once after elapsing the socket keeping time set by **socketKeepingTime** method. Then the new socket is created immediately and used for the next connection.

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.
This method discards the print data kept by **controlTransaction** method.
The instance of **CallbackFunctionListener** interface kept by **setCallbackFunctionListener** method is discarded and the callback is stopped.

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

Note Execute this method after transmitting the data.
If executing this method before data transmission is completed, a part of the data may be deleted.
To verify data transmission completion, it is recommended to obtain execution response by **PRINTER_RESPONSE_REQUEST** of **getPrinterResponse** method before executing this method.
When this method is executed without executing **getPrinterResponse** method in your program, evaluate your program to confirm no problems arise.

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 4-23 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.

Error

buzzer

getStatus

4-29

The Mobile printer status is shown in Table 4-33.

When the connection failed, the Mobile printer status is shown in 0x00000000.

Table 4-33 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 or higher to lower than 8.0 V
1	0	7.0 or higher to lower than 7.5 V
1	1	Lower than 7.0 V

The status of POS printer is shown in Table 4-34.

When the connection failed, the POS printer status is shown in 0x80000000.

Table 4-34 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

Bit	Function	Value	
		0	1
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.

setCallbackFunctionListener Start/End callback of printer status change

Target Mobile printer / POS printer

Syntax public void **setCallbackFunctionListener**(CallbackFunctionListener *listener*)
throws **PrinterException**

Parameter *listener* Instance of **CallbackFunctionListener** interface

Description Register the process executed by callback with **onStatusChanged** method. When the instance of **CallbackFunctionListener** interface is specified in *listener* parameter and this method is executed, the callback is started. When null is specified in listener parameter and this method is executed, the callback is finished.

The instance kept by **CallbackFunctionListener** interface is discarded by any of the following:

- Execute this method specifying null in *listener* parameter
- Execute **disconnect** method

This call of the method can be used when connect is executed and **isConnect** is true.

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.

abort Abort the waiting state of a printer

Target Mobile printer / POS printer

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

Description When sending of image file by **sendDataFile** method is aborted, a printer does not accept other processes until specified image file is received completely. (Method or sent data are misinterpreted and recognized as part of the image file.) 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 file 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.

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 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. The valid range of *id* parameter is from 0 to 127.

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.
The valid range of *id* parameter is 2 characters. Subsequent characters are ignored. 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').

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 path 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.
The valid range of *num* parameter is from 1 to 4.

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.
The valid range of *num* parameter is from 1 to 4.

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.
When Bluetooth connection is used, this method resets a connected printer by using printer command.
When USB connection is used, this method resets a connected printer by using the SOFT_RESET function in USB printer class.
When TCP/IP connection is used, 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 4-9 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 at **connect** method execution is **PRINTER_MODEL_DPU_S245** or **PRINTER_MODEL_DPU_S445**, see contents of obtained response for Mobile printer. When the specified *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 4-35.

Table 4-35 Contents of Obtained Response (Mobile Printer)

Constant Name	Description
PRINTER_RESPONSE_REQUEST	Obtains "Execution response request". For <i>buf</i> parameter, specify integer array whose length is 1. Specify 0 to 15 (00h to 0Fh) for <i>buf[0]</i> . When responses are obtained successfully, 80 to 95 (50h to 5Fh) are stored to <i>buf[0]</i> .
PRINTER_RESPONSE_EXTERNAL_RAM	Obtains "Send Remaining RAM Response". For <i>buf</i> parameter, specify integer array whose length is 1. When responses are obtained successfully, remaining RAM capacity is stored to <i>buf[0]</i> in byte value.
PRINTER_RESPONSE_USER_AREA	Obtains "Send remaining capacity of user area". For <i>buf</i> parameter, specify integer array whose length is 1. When responses are obtained successfully, remaining user area is stored to <i>buf[0]</i> in byte value.
PRINTER_RESPONSE_BATTERY_STATUS	Obtains "Battery voltage state". For <i>buf</i> parameter, specify integer array whose length is 1. When responses are obtained successfully, the battery status value is stored to <i>buf[0]</i> . Meanings of battery status values are as follows.

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

Contents of obtained responses for POS printer are shown in Table 4-36.

Table 4-36 Contents of Obtained Response (POS Printer)

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

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	(a) public void startDiscoveryPrinter (PrinterListener <i>listener</i>) throws PrinterException (b) public void startDiscoveryPrinter (PrinterListener <i>listener</i> , Context <i>context</i>) throws PrinterException	
Parameter	<i>listener</i> <i>context</i>	PrinterListener Instance described in the following Context of application
Description	This method searches the printer connected to Bluetooth. The method of syntax (a) specifies instance of PrinterListener . The method of syntax (b) specifies instance of PrinterListener and specifies context to call application.	

Completion or cancellation of the printer search is notified to user application as an 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.

Do not call this method from main thread of application.

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

Note Use the syntax (a) when using this method newly.
Syntax (b) is a method that will be unsupported in the future.

startDiscoveryPrinter	Start printer search (USB)
-----------------------	----------------------------

Target Mobile printer / POS printer

Syntax public void **startDiscoveryPrinter**(PrinterListener *listener*, int *deviceType*)
throws **PrinterException**

Parameter *listener* PrinterListener Instance described in the following
deviceType Port type

Description This method searches the printer connected to USB.

Completion of the printer search or cancellation by **cancelDiscoveryDevice** method is notified to the user application as an event by **finishEvent** method through the instance specified in the *listener* parameter.
Specify **PRINTER_TYPE_USB** in the *deviceType* parameter.

This method searches for SII printer. The printer information of the found printer is stored to **PrinterInfo** class described later.

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 connecting to the same network. Completion or cancellation of the printer search is notified to user application as an event by **finishEvent** through instance specified in *listener* parameter.
The valid range of *retry* parameter is 1 to 5. When the value below 1 is set, the value is set to 1. When the value exceeds 5 is set, the value is set to 5.
The valid range of *timeout* parameter is 3000 to 60000. When the value below 3000 is set, the value is set to 3000. When the value exceeds 60000 is set, the value is set to 60000.
The searched printer information is stored in **PrinterInfo** class described in the following.

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 <i>listener</i> 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 <i>sendTimeout</i>)
Parameter	<i>sendTimeout</i> 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 values outside the scope of the valid range is configured.
Default	10000 msec (10 seconds)
Effective range	100 msec to 90000 msec (90 seconds)

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 <i>receiveTimeout</i>)
Parameter	<i>receiveTimeout</i> 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 values outside the scope of the valid range is configured. 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 the text data is sent by sendText method, sendTextEx method, sendDataFile method, printPageModeText method, or printPageModeTextEx method, the print result of the following character codes varies. See "Appendix A Character Sets (Character Code Table)" for details about characters to be printed. Character codes with the varying print result depending on the configuration of the international character set: 0x23, 0x24, 0x40, 0x5B, 0x5C, 0x5D, 0x5E, 0x60, 0x7B, 0x7C, 0x7D, 0x7E
Return value	See Table 4-4 International Character Set Constant for details.

setInternationalCharacter**Set international character set**

Target	Mobile printer / POS printer
Syntax	public void setInternationalCharacter (int <i>internationalCharacter</i>)
Parameter	<i>internationalCharacter</i> International character set constant
Description	<p>This method sets international character set. See Table 4-4 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 an invalid value to <i>internationalCharacter</i> parameter is set, the following values are used as well.</p> <p>When a language setting of an Android device is selected Japanese: COUNTRY_JAPAN</p> <p>When a language setting of an Android device is selected other languages than Japanese: COUNTRY_USA</p>

getCodePage**Obtain codepage**

Target	Mobile printer / POS printer
Syntax	public int getCodePage ()
Description	<p>This method obtains setting values of codepage. The encoder used for sending the text data by sendText method, sendTextEx method, sendDataFile method, printPageModeText method, or printPageModeTextEx is changed. See "Appendix A Character Sets (Character Code Table)" for details about characters to be printed.</p>
Return value	See Table 4-5 Codepage Constant for details.

setCodePage**Set codepage**

Target	Mobile printer / POS printer
Syntax	public void setCodePage (int <i>codePage</i>)
Parameter	<i>codePage</i> Codepage constant
Description	<p>This method sets codepage. See Table 4-5 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 an invalid value to <i>codePage</i> parameter is set, it will be ignored.</p> <p>When a language setting of an Android device is selected Japanese: CODE_PAGE_KATAKANA</p> <p>When a language setting of an Android device is selected other languages than Japanese: CODE_PAGE_1252</p>

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 the printer is not connected, when connect method has been succeeded once, the printer model value successfully connected last time is returned.
Return value	See Table 4-7 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 the printer is not connected, when connect method has been succeeded once, the port type value successfully connected last time is returned.
Return value	See Table 4-8 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 the printer, the false is returned in this method. If false is returned, 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. <table> <tr> <td>true</td> <td>Connected to a printer</td> </tr> <tr> <td>false</td> <td>Disconnected to a printer</td> </tr> </table>	true	Connected to a printer	false	Disconnected to 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 <i>socketKeepingTime</i>)
Parameter	<i>socketKeepingTime</i> Socket keeping time
Description	<p>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. The setting of Network Printer Receive Timeout can be changed in "SII RP Utility" with the Android app on Google Play.</p> <p>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.</p>
Default	300000msec (5 minutes)
Effective range	60000 to 300000msec (5 minutes)

controlTransaction	Start/End batch processing
--------------------	----------------------------

Target	Mobile printer / POS printer
Syntax	public void controlTransaction (TransactionFunction <i>transactionFunction</i>) throws PrinterException
Parameter	<i>transactionFunction</i> Batch processing selection
Description	See Table 4-29 Batch Processing Selection (TransactionFunction) for available setting in <i>transactionFunction</i> parameter.

The procedure of batch processing is as follows:

(1) Start batch processing.

Specify **TRANSACTION_START**.

(2) Execute the method.

In the case of the batch processing target method, buffering of transmission data is started.

The transmission data of the batch processing target method executed during buffering is buffered in the transmission buffer without being sent to the printer.

The maximum size of transmission data to be buffered is system dependent.

If the buffered transmission data exceeds the maximum size, the batch processing target method at the time of exceeding becomes an error. If an error occurs, the transmission data up to the error is retained.

As for the retained transmission data, finish the batch processing in step (3).

In the case of a method other than the batch processing target method, transmission data is immediately executed without being buffered.

(3) Finish batch processing.

When **TRANSACTION_PRINT** is specified, the buffered transmission data is sent to the printer. The buffered transmission data is retained even after sent to the printer.

The retained transmission data is discarded by any of the following:

- Specify **TRANSACTION_CLEAR**

- Specify **TRANSACTION_START**

- Execute **disconnect** method

The batch processing target methods are as follows:

- **sendText**
- **sendTextEx**
- **printBarcode**
- **printPDF417**
- **printQRcode**
- **printDataMatrix**
- **printMaxiCode**
- **cutPaper**^{*1}
- **openDrawer**^{*1}
- **buzzer**^{*1}
- **sendBinary**
- **sendDataFile**
- **printLogo**^{*2}
- **enterPageMode**^{*1}
- **exitPageMode**^{*1}
- **setPageModeArea**^{*1}
- **setPageModeDirection**^{*1}
- **setPageModeLineSpacing**^{*1}
- **printPageMode**^{*1}
- **printPageModeText**^{*1}
- **printPageModeTextEx**^{*1}
- **printPageModeBarcode**^{*1}
- **printPageModePDF417**^{*1}
- **printPageModeQRcode**^{*1}
- **printPageModeDataMatrix**^{*1}
- **printPageModeMaxiCode**^{*1}
- **sendPageModeBinary**^{*1}
- **printPageModeImageFile**^{*1}
- **printPageModeRectangle**^{*1}
- **printPageModeLine**^{*1}
- **printPageModeLogo**^{*1*2}

*1: This method is not supported by Mobile printer.

*2: The method under batch processing does not notify the error even when the registered logo does not exist.

(b) Dedicated method for standard mode

The following methods are valid in standard mode. **PrinterException** is thrown when the dedicated method for standard mode are executed in page mode.

sendText		Send text data
Target	Mobile printer / POS printer	
Syntax	public void sendText (String <i>text</i>) throws PrinterException	
Parameter	<i>text</i>	Text data sent to a printer
Description	<p>This method sends the text data specified by <i>text</i> parameter to a printer. Data size that can be specified at a time is 16K bytes (16384 bytes).</p> <p>This method encodes the specified text data to printable text data based on the settings of setInternationalCharacter method and setCodePage method, and then sends it to a printer.</p> <p>This method does not add a line feed code at the end of the text data. In order to print to the end, add a line feed code to the end of the text data.</p>	
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 <i>text</i> , CharacterBold <i>bold</i> , CharacterUnderline <i>underline</i> , CharacterFont <i>font</i> , CharacterScale <i>scale</i>) throws PrinterException	
Parameter	<i>text</i> <i>bold</i> <i>underline</i> <i>font</i> <i>scale</i>	Text data to send to a printer Bold print Underline Character font Character scale
Description	<p>This method is valid only for Mobile printer. This method encodes the specified text data to printable text data based on setInternationalCharacter method and setCodePage method, and then sends it to the printer. Data size that can be specified at one time is 16K bytes (16384 bytes).</p> <p>See Table 4-10 Bold Print (CharacterBold) for available setting in <i>bold</i> parameter.</p> <p>See Table 4-11 Underline (CharacterUnderline) for available setting in <i>underline</i> parameter.</p> <p>See Table 4-13 Character Font (CharacterFont) for available setting in <i>font</i> parameter.</p> <p>See Table 4-14 Character Scale (CharacterScale) for available setting in <i>scale</i> parameter.</p>	

This method does not add a line feed code at the end of the text data. In order to print to the end, add a line feed code to the end of the 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	(a) public void sendTextEx (String <i>text</i> , CharacterBold <i>bold</i> , CharacterUnderline <i>underline</i> , CharacterReverse <i>reverse</i> , CharacterFont <i>font</i> , CharacterScale <i>scale</i> , PrintAlignment <i>alignment</i>) throws PrinterException (b) public void sendTextEx (String <i>text</i> , CharacterBold <i>bold</i> , CharacterUnderline <i>underline</i> , CharacterReverse <i>reverse</i> , CharacterFont <i>font</i> , CharacterScale <i>scale</i> , PrintAlignment <i>alignment</i> , OutputPendingData <i>output</i>) throws PrinterException	
Parameter	<i>text</i> <i>bold</i> <i>underline</i> <i>reverse</i> <i>font</i> <i>scale</i> <i>alignment</i> <i>output</i>	Text data to send to a printer Bold print Underline Reverse print Character font Character scale Alignment Pending data output specifying
Description	This method is valid only for POS printer. This method encodes the specified text data to printable text data based on setInternationalCharacter method and setCodePage method, and then sends it to the printer. Data size that can be specified at one time is 16K bytes (16384 bytes). See Table 4-10 Bold Print (CharacterBold) for available setting in <i>bold</i> parameter. See Table 4-11 Underline (CharacterUnderline) for available setting in <i>underline</i> parameter. See Table 4-12 Reverse Print (CharacterReverse) for available setting in <i>reverse</i> parameter. See Table 4-13 Character Font (CharacterFont) for available setting in <i>font</i> parameter. See Table 4-14 Character Scale (CharacterScale) for available setting in <i>scale</i> parameter. See Table 4-15 Alignment (PrintAlignment) for available setting in <i>alignment</i> parameter. See Table 4-30 Pending Data Output Specifying (OutputPendingData) for available setting in <i>output</i> parameter.	

For laying out text data by sending following printer commands with **sendBinary** method or **sendDataFile** method, specify **PENDING_DATA_OUTPUT_TOGETHER** at *output* parameter in the method of syntax (b).

- "Horizontal Tab"
- "Specify Absolute Position"
- "Specify Relative Position"

When the method of syntax (a) is executed or **PENDING_DATA_OUTPUT_FIRST** is specified at *output* parameter in the method of syntax (b), the print position set in above becomes invalid.

The method of syntax (b) does not add a line feed code at the end of the text data. In order to print to the end, add a line feed code to the end of the 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	<p>(a) public void printBarcode(BarcodeSymbol <i>barcodeSymbol</i>, String <i>text</i>, ModuleSize <i>moduleSize</i>, int <i>moduleHeight</i>, HriPosition <i>hriPosition</i>, CharacterFont <i>hriFont</i>, PrintAlignment <i>alignment</i>) throws PrinterException</p> <p>(b) public void printBarcode(BarcodeSymbol <i>barcodeSymbol</i>, String <i>text</i>, ModuleSize <i>moduleSize</i>, int <i>moduleHeight</i>, HriPosition <i>hriPosition</i>, CharacterFont <i>hriFont</i>, PrintAlignment <i>alignment</i>, NwRatio <i>nwRatio</i>) throws PrinterException</p> <p>(c) public void printBarcode(BarcodeSymbol <i>barcodeSymbol</i>, byte[] <i>data</i>, ModuleSize <i>moduleSize</i>, int <i>moduleHeight</i>, HriPosition <i>hriPosition</i>, CharacterFont <i>hriFont</i>, PrintAlignment <i>alignment</i>) throws PrinterException</p>	
Parameter	<i>barcodeSymbol</i> <i>text (data)</i> <i>moduleSize</i> <i>moduleHeight</i> <i>hriPosition</i> <i>hriFont</i> <i>alignment</i> <i>nwRatio</i>	Barcode symbol Text data to send to a printer Barcode width Barcode height (dot) HRI character print position HRI character font Alignment N:W ratio
Description	<p>This method executes barcode print.</p> <p>See Table 4-16 Barcode Symbol (BarcodeSymbol) for available setting in <i>barcodeSymbol</i> parameter.</p>	

See Table 4-17 Module Size (ModuleSize) for available setting in *moduleSize* parameter.

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

See Table 4-18 HRI Character Print Position (HriPosition) for available setting in *hriPosition* parameter.

See Table 4-13 Character Font (CharacterFont) for available setting in *hriFont* parameter.

See Table 4-15 Alignment (PrintAlignment) for available setting in *alignment* parameter.

See Table 4-19 N:W Ratio (NwRatio) for available setting in *nwRatio* parameter.

Depending on the relationship between *moduleSize* parameter and *nwRatio* parameter, the wide element width is set in the following tables. See Table 4-37 for Mobile printer, and Table 4-38 for POS printer.

Table 4-37 N:W Ratio for Mobile Printer

<i>moduleSize</i>	<i>nwRatio</i>			
	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 4-38 N:W Ratio for POS Printer

<i>moduleSize</i>	<i>nwRatio</i>		
	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 *moduleSize*,
 int *moduleHeight*,
 PrintAlignment *alignment*,
 Pdf417Symbol *pdf417Symbol*) throws **PrinterException**

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

Parameter	<i>text</i>	Barcode data to send a printer
	<i>errorCorrection</i>	Error correction level
	<i>row</i>	The number of row
	<i>column</i>	The number of columns in data area
	<i>moduleSize</i>	Nominal fine element width
	<i>moduleHeight</i>	Module height (dot)
	<i>alignment</i>	Alignment
	<i>pdf417Symbol</i>	Symbol of PDF417

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

See Table 4-20 Error Correction Level (ErrorCorrection) for available setting in *errorCorrection* parameter.

The valid range of *row* parameter is from 0, 3 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 4-17 Module Size (ModuleSize) for available setting in *moduleSize* parameter.

The valid range of *moduleHeight* parameter is from 2 to 127. When the module height is set smaller, some barcode scanners may not read it. Specify 3 or more for normal use.

See Table 4-15 Alignment (PrintAlignment) for available setting in *alignment* parameter.

See Table 4-21 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.

Target	Mobile printer / POS printer	
Syntax	(a) public void printQRcode (String <i>text</i> , ErrorCorrection <i>errorCorrection</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i>) throws PrinterException (b) public void printQRcode (String <i>text</i> , ErrorCorrection <i>errorCorrection</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i> , QrModel <i>model</i>) throws PrinterException	
Parameter	<i>text</i>	Barcode data to send a printer
	<i>errorCorrection</i>	Error correction level
	<i>moduleSize</i>	Module Size
	<i>alignment</i>	Alignment
	<i>model</i>	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 <i>text</i> parameter. See Table 4-20 Error Correction Level (ErrorCorrection) for available setting in <i>errorCorrection</i> parameter. See Table 4-17 Module Size (ModuleSize) for available setting in <i>moduleSize</i> parameter. See Table 4-15 Alignment (PrintAlignment) for available setting in <i>alignment</i> parameter. See Table 4-22 QR Code Model (QrModel) for available setting in <i>model</i> 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.	

Target	Mobile printer / POS printer	
Syntax	public void printDataMatrix (String <i>text</i> , DataMatrixModule <i>dataMatrixModule</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i>) throws PrinterException	
Parameter	<i>text</i>	Barcode data to send to the printer
	<i>dataMatrixModule</i>	The number of Data Matrix modules
	<i>moduleSize</i>	Module size
	<i>alignment</i>	Alignment
Description	This method prints Data Matrix. See Table 4-24 Data Matrix modules (DataMatrixModule) for available setting in <i>dataMatrixModule</i> parameter.	

See Table 4-17 Module Size (ModuleSize) for available setting in *moduleSize* parameter.

See Table 4-15 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.

printMaxiCode		Print MaxiCode
Target	Mobile printer / POS printer	
Syntax	public void printMaxiCode (String <i>text</i> , MaxiCodeMode <i>maxiCodeMode</i> , PrintAlignment <i>alignment</i>) throws PrinterException	
Parameter	<i>text</i>	Barcode data to send to the printer <ul style="list-style-type: none"> When <i>maxiCodeMode</i> parameter is MAXI_CODE_2 Add the service class (3 digits), the country code (3 digits), and the postal code (9 digits) in the beginning of the data. When <i>maxiCodeMode</i> parameter is MAXI_CODE_3 Add the service class (3 digits), the country code (3 digits), and the postal code (6 digits) in the beginning of the data.
	<i>maxiCodeMode</i>	MaxiCode mode
	<i>alignment</i>	Alignment
Description	<p>This method prints MaxiCode.</p> <p>See Table 4-25 MaxiCode Mode (MaxiCodeMode) for available setting in <i>maxiCodeMode</i> parameter.</p> <p>See Table 4-15 Alignment (PrintAlignment) for available setting in <i>alignment</i> parameter.</p>	
Error	<p>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.</p>	

sendBinary		Send binary data
Target	Mobile printer / POS printer	
Syntax	public void sendBinary (byte [] <i>binary</i>) throws PrinterException	
Parameter	<i>binary</i>	Binary data sent to a printer
Description	<p>This method sends the binary data specified by <i>binary</i> parameter to a printer. Data size that can be specified at a time is 16K bytes (16384 bytes).</p> <p>In this method, specified binary data is sent to a printer without conversion.</p> <p>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.</p>	

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	(a) public void sendDataFile (String <i>fileName</i>) throws PrinterException (b) public void sendDataFile (String <i>fileName</i> , Dithering <i>dithering</i>) throws PrinterException (c) public void sendDataFile (String <i>fileName</i> , PrintAlignment <i>alignment</i>) throws PrinterException (d) public void sendDataFile (String <i>fileName</i> , PrintAlignment <i>alignment</i> , Dithering <i>dithering</i>) throws PrinterException	
Parameter	<i>fileName</i>	File path of the data to sent to a printer
	<i>alignment</i>	Alignment
	<i>dithering</i>	Dithering
Description	<p>This method determines data format based on the file extension specified by <i>fileName</i> 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).</p> <p>The method of syntax (a) specifies the sending file. The method of syntax (b) specifies the sending file and Dithering. The method of syntax (c) is available for only POS printer and specifies Alignment to the sending file. The method of syntax (d) is available for only POS printer and specifies Alignment to the sending file and Dithering.</p> <p>The file extensions capable of sending by <i>fileName</i> parameter and sending the file are describes below.</p> <ul style="list-style-type: none"> .bmp, .jpg, .jpeg, or .png: Send data to a printer as image file. When that image file is colored one, it is converted to monochrome image by binarization, and sent to a printer. .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 setInternationalCharacter method and setCodePage method, and then sends it to a printer. Also, in this method, the line spacing code is not added to the end of text data. .bin or .dat: Data is sent to a printer as binary data without conversion. .htm or .html: Only valid for POS printer. Data is sent to a printer as html data without conversion. <p><i>alignment</i> parameter is valid only when the file extension specified on <i>fileName</i> parameter is .bmp, .jpg, .jpeg, .png or .txt. See Table 4-15 Alignment (PrintAlignment) for available setting.</p> <p><i>dithering</i> parameter is valid only when the file extension specified on <i>fileName</i> parameter is .bmp, .jpg, .jpeg, or .png. See Table 4-28 Dithering (Dithering) for available setting.</p>	

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
-----------	-----------------------------------------

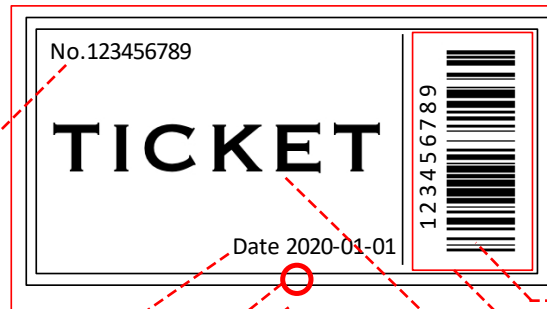
Target	Mobile printer
Syntax	public void printLogo (int <i>id</i>) throws PrinterException
Parameter	<i>id</i> Logo ID to print
Description	This method is valid only for Mobile printer. This method print the logo (image) registered by registerLogo method. Specify registered logo ID in <i>id</i> parameter. The valid range of <i>id</i> 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 <i>id</i> , PrintAlignment <i>alignment</i>) throws PrinterException
Parameter	<i>id</i> Logo ID to print <i>alignment</i> 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 <i>id</i> parameter. The valid range of <i>id</i> 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 4-15 Alignment (PrintAlignment) for available setting in <i>alignment</i> 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.

(c) Dedicated method for page mode

The following methods are dedicated methods to use page mode. An example for the print process in page mode is shown below.



① Start page mode

```
mPrinterManager.enterPageMode();
```

② Specify print area of page mode

```
mPrinterManager.setPageModeArea(0, 0, 576, 355);
```

③ Specify a rectangle and a ruled line

```
mPrinterManager.printPageModeRectangle(0, 0, 575, 344, LineStyle.LINESTYLE_THIN);  
mPrinterManager.printPageModeRectangle(7, 7, 567, 336, LineStyle.LINESTYLE_THIN);  
mPrinterManager.printPageModeLine(404, 11, 404, 334, LineStyle.LINESTYLE_THIN);
```

④ Specify a character

```
mPrinterManager.printPageModeText(21, 37, "NO.123456789");  
mPrinterManager.printPageModeText(212, 330, "Date 2020-01-01");
```

⑤ Specify an image file

```
mPrinterManager.printPageModeImageFile(  
    10,  
    212,  
    Environment.getExternalStorageDirectory().getPath() + "/TicketImage.jpg",  
    Dithering.DITHERING_DISABLE);
```

⑥ Specify print area of page mode

```
mPrinterManager.setPageModeArea(404, 0, 163, 345);
```

⑦ Specify print direction

```
mPrinterManager.setPageModeDirection(Direction.DIRECTION_BOTTOM_TO_TOP);
```

⑧ Specify a barcode

```
mPrinterManager.printPageModeBarcode(  
    20,  
    132,  
    BarcodeSymbol.BARCODE_SYMBOL_CODE128,  
    new byte[]{0x67, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19x, 0x68},  
    ModuleSize.BARCODE_MODULE_WIDTH_2,  
    80,  
    HriPosition.HRI_POSITION_ABOVE,  
    CharacterFont.FONT_A);
```

⑨ Print in page mode

```
mPrinterManager.printPageMode(CuttingMethod.CUT_PARTIAL);
```

⑩ Ends page mode

```
mPrinterManager.exitPageMode();
```

enterPageMode	Start page mode
---------------	-----------------

Target	POS printer
Syntax	public void enterPageMode() throws PrinterException
Description	<p>This method starts page mode. The dedicated method for page mode and common methods to standard mode and page mode can be used after this method execution.</p> <p>Executing exitPageMode method discards the print data kept in the page data buffer and changes the mode to standard mode.</p> <p>Executing printPageMode method prints the print data kept in the page data buffer.</p>
Error	PrinterException may be thrown when the method is called.

exitPageMode	End page mode
--------------	---------------

Target	POS printer
Syntax	public void exitPageMode() throws PrinterException
Description	Discards the print data kept in the page data buffer and changes the mode to standard mode.
Error	PrinterException may be thrown when the method is called.

setPageModeArea	Specify print area of page mode
-----------------	---------------------------------

Target	POS printer
Syntax	public void setPageModeArea (int <i>x</i> , int <i>y</i> , int <i>width</i> , int <i>height</i>) throws PrinterException
Parameter	<p><i>x</i> The horizontal origin (dot) of the print area of page mode 0 represents the left edge on the print area of the printer.</p> <p><i>y</i> The vertical origin (dot) of the print area of page mode The valid range is 0 to 2399. 0 represents the position where paper feed has not been performed.</p> <p><i>width</i> The print area width (dot) of page mode</p> <p><i>height</i> The print area height (dot) of page mode The valid range is 1 to (2400-<i>y</i>).</p>

The valid range of x and $width$ are shown in figure below.

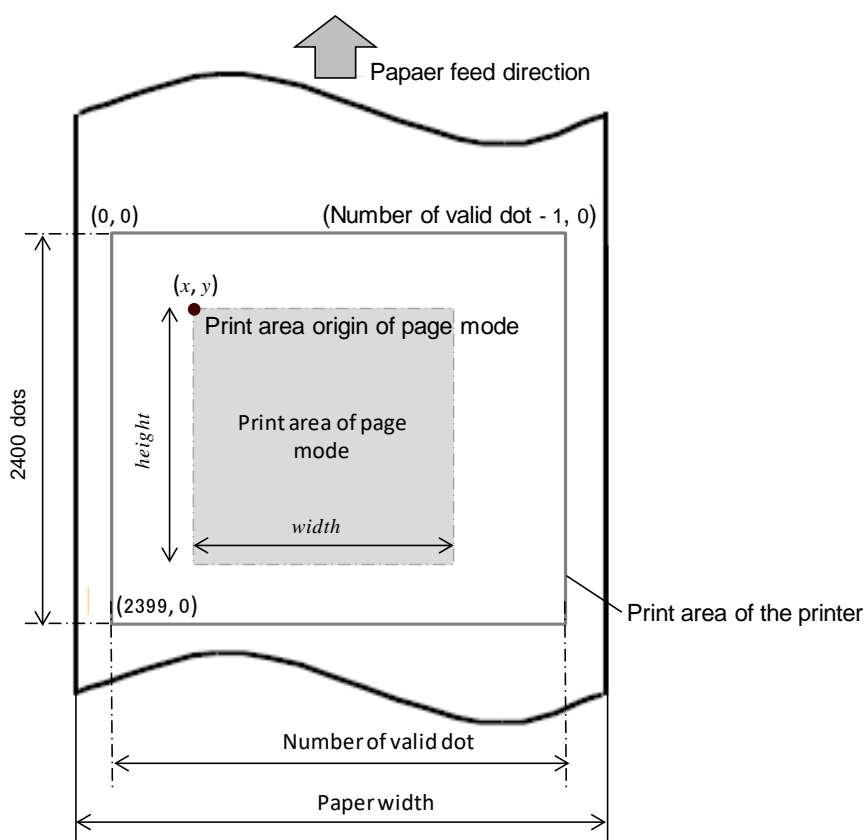


Table 4-39 Valid Range of x and $width$ (POS printer)

Memory Switch Setting of Printer		Number of Valid Dot	setPageModeArea	
MS4-4 (Paper Width)	MS4-5 (Number of Effective Dots)		x	$width$
80 mm	576	576	0 to 575	1 to (576 - x)
	512	512	0 to 511	1 to (512 - x)
58 mm	432	432	0 to 431	1 to (432 - x)
	360	360	0 to 359	1 to (360 - x)

Number of valid dot varies depending on the memory switch setting of the printer.
See the technical reference of POS printer for details about function settings.

Description This method specifies the print area of page mode.

Start page mode by **enterPageMode** method before executing this method.

The print area of page mode can be specified when page mode is started by **enterPageMode** method and then this method is executed after executing the dedicated method for page mode. The data that has been mapped is kept.

The data of the dedicated method for page mode is mapped to the print area of page mode added by this method after executing this method.

The print area of page mode is $x = 0$, $y = 0$, $width$ = number of a valid dot, $height = 2400$ after executing **enterPageMode** method.

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

setPageModeDirection	Specify print direction of page mode
----------------------	--------------------------------------

Target	POS printer
Syntax	public void setPageModeDirection (Direction <i>direction</i>) throws PrinterException
Parameter	<i>direction</i> Print direction See Table 4-31 Print direction (Direction) for available constants.
Description	This method specifies print direction of page mode. Start page mode by enterPageMode method before executing this method. The print direction is left to right after executing enterPageMode method.
Error	PrinterException may be thrown when the method is called.

setPageModeLineSpacing	Specify line spacing of page mode
------------------------	-----------------------------------

Target	POS printer
Syntax	public void setPageModeLineSpacing (int <i>lineSpacing</i>) throws PrinterException
Parameter	<i>linespacing</i> Line spacing (dot) of page mode The valid range is 0 to 255.
Description	This method specifies line spacing of page mode. Start page mode by enterPageMode method before executing this method. The line spacing is 34 dots after executing enterPageMode method.
Error	PrinterException may be thrown when the method is called.

printPageMode	Print page mode
---------------	-----------------

Target	POS printer
Syntax	public void printPageMode (CuttingMethod <i>cuttingMethod</i>) throws PrinterException
Parameter	<i>cuttingMethod</i> Cutting method See Table 4-23 Cutting Method (CuttingMethod) for available constants.
Description	This method prints the print data kept in page data buffer. The print data is kept after printing. The print data is discarded at the timing of the following: <ul style="list-style-type: none"> • Execute enterPageMode method • Execute disconnect method • Execute exitPageMode method
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.

printPageModeText

Send text data of page mode

Target	POS printer	
Syntax	public void printPageModeText (int <i>startX</i> , int <i>startY</i> , String <i>text</i>) throws PrinterException	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Text data Data size that can be specified at 1 time is 16 KB (16384 bytes).
Description	This method encodes the specified text data to printable text data based on setInternationalCharacter method and setCodePage method. Start page mode by enterPageMode method before executing this method.	
Error	PrinterException may be thrown when the method is called.	

printPageModeTextEx

Send format specified text data of page mode

Target	POS printer	
Syntax	public void printPageModeTextEx (int <i>startX</i> , int <i>startY</i> , String <i>text</i> , CharacterBold <i>bold</i> , CharacterUnderline <i>underline</i> , CharacterReverse <i>reverse</i> , CharacterFont <i>font</i> , CharacterScale <i>scale</i>) throws PrinterException	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Text data Data size that can be specified at 1 time is 16 KB (16384 bytes).
	<i>bold</i>	Bold print See Table 4-10 Bold Print (CharacterBold) for available constants.
	<i>underline</i>	Underline See Table 4-11 Underline (CharacterUnderline) for available constants.
	<i>reverse</i>	Reverse print See Table 4-12 Reverse Print (CharacterReverse) for available constants.
	<i>font</i>	Font See Table 4-13 Character Font (CharacterFont) for available constants.

	<i>scale</i>	Character scale See Table 4-14 Character Scale (CharacterScale) for available constants.
Description	This method encodes the specified text data to printable text data based on setInternationalCharacter method and setCodePage method. Start page mode by enterPageMode before executing this method.	
Error	PrinterException may be thrown when the method is called.	

printPageModeBarcode	Print barcode of page mode
-----------------------------	-----------------------------------

Target	POS printer	
Syntax	<p>(a) public void printPageModeBarcode(int <i>startX</i>, int <i>startY</i>, BarcodeSymbol <i>barcodeSymbol</i>, String <i>text</i>, ModuleSize <i>moduleSize</i>, int <i>moduleHeight</i>, HriPosition <i>hriPosition</i>, CharacterFont <i>hriFont</i>) throws PrinterException</p> <p>(b) public void printPageModeBarcode(int <i>startX</i>, int <i>startY</i>, BarcodeSymbol <i>barcodeSymbol</i>, String <i>text</i>, ModuleSize <i>moduleSize</i>, int <i>moduleHeight</i>, HriPosition <i>hriPosition</i>, CharacterFont <i>hriFont</i>, NwRatio <i>nwRatio</i>) throws PrinterException</p> <p>(c) public void printPageModeBarcodeint <i>startX</i>, int <i>startY</i>, BarcodeSymbol <i>barcodeSymbol</i>, byte[] <i>data</i>, ModuleSize <i>moduleSize</i>, int <i>moduleHeight</i>, HriPosition <i>hriPosition</i>, CharacterFont <i>hriFont</i>) throws PrinterException</p>	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>barcodeSymbol</i>	BarcodeSymbol See Table 4-16 Barcode Symbol (BarcodeSymbol) for available constants and correspondent syntax.
	<i>text(data)</i>	Barcode data
	<i>moduleSize</i>	Barcode width See Table 4-17 Module Size (ModuleSize) for available constants.
	<i>moduleHeight</i>	Barcode height (dot)

<i>hriPosition</i>	HRI character print position See Table 4-18 HRI Character Print Position (HriPosition) for available constants.
<i>hriFont</i>	HRI character font See Table 4-13 Character Font (CharacterFont) for available constants.
<i>nwRatio</i>	N:W ratio See Table 4-19 N:W Ratio (NwRatio) for available constants. Depending on specified <i>nwRatio</i> and <i>moduleSize</i> , the wide element width is set as shown in Table 4-40.

Table 4-40 NW Ratio

<i>moduleSize</i>	<i>nwRatio</i>		
	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)

Description	<p>This method maps the barcode on the print area of page mode.</p> <p>The method of syntax (a) specifies the barcode data by character string.</p> <p>The method of syntax (b) specifies the barcode data by character string and specifies N:W ratio of the barcode.</p> <p>The method of syntax (c) specifies the barcode data by the array of bytes.</p> <p>Start page mode by enterPageMode method before executing this method.</p>
Error	PrinterException may be thrown when the method is called.
Note	Map the print data of the barcode not to overlap the other print data.

Target	POS printer	
Syntax	<p>(a) public void printPageModePDF417(int <i>startX</i>, int <i>startY</i>, String <i>text</i>, ErrorCorrection <i>ErrorCorrection</i>, int <i>row</i>, int <i>column</i>, ModuleSize <i>moduleSize</i>, int <i>moduleHeight</i>, Pdf417Symbol <i>pdf417Symbol</i>) throws PrinterException</p> <p>(b) public void printPageModePDF417(int <i>startX</i>, int <i>startY</i>, String <i>text</i>, ErrorCorrection <i>ErrorCorrection</i>, int <i>row</i>, int <i>column</i>, ModuleSize <i>moduleSize</i>, int <i>moduleHeight</i>) throws PrinterException</p>	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Barcode data
	<i>errorCorrection</i>	Error correction level See Table 4-20 Error Correction Level (ErrorCorrection) for available constants.
	<i>row</i>	The number of rows (row) The valid range is 0, 3 to 90. When 0 is specified, the number of rows is automatically set.
	<i>column</i>	The number of columns in data area The valid range is 0 to 30. When 0 is specified, the number of columns in the data area is automatically set.
	<i>moduleSize</i>	Nominal fine element width See Table 4-17 Module Size (ModuleSize) for available constants.
	<i>moduleHeight</i>	Module height (dot) The valid range is 2 to 127. When the module height is set smaller, some barcode scanners may not read it. Set 3 or more for normal use.
	<i>pdf417Symbol</i>	Symbol of PDF417 See Table 4-21 PDF417 Symbol (Pdf417Symbol) for available constants.
Description	<p>This method maps PDF417 on the print area of page mode.</p> <p>The method of syntax (a) specifies PDF417 symbol.</p> <p>The method of syntax (b) is fixed to standard PDF417.</p>	

Start page mode by **enterPageMode** method before executing this method.

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

Note Map the print data of the barcode not to overlap the other print data.

printPageModeQRcode	Print QR Code of page mode
---------------------	----------------------------

Target POS printer

Syntax

```
(a) public void printPageModeQRcode(int startX,
                                     int startY,
                                     String text,
                                     ErrorCorrection errorCorrection,
                                     ModuleSize moduleSize) throws PrinterException

(b) public void printPageModeQRcode(int startX,
                                     int startY,
                                     String text,
                                     ErrorCorrection errorCorrection,
                                     ModuleSize moduleSize,
                                     QrModel model) throws PrinterException
```

Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Barcode data The version for either syntax (a) or (b) is automatically set depending on the number of data specified on <i>text</i> .
	<i>errorCorrection</i>	Error correction level See Table 4-20 Error Correction Level (ErrorCorrection) for available constants.
	<i>moduleSize</i>	Module size See Table 4-17 Module Size (ModuleSize) for available constants.
	<i>model</i>	QR Code Model See Table 4-22 QR Code Model (QrModel) for available constants.

Description This method maps QR Code on the print area of page mode.
The method of syntax (a) is fixed to QR Code Model 2.
The method of syntax (b) specifies QR Code Model.

Start page mode by **enterPageMode** method before executing this method.

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

Note Map the print data of the barcode not to overlap the other print data.

printPageModeDataMatrix**Print Data Matrix of page mode**

Target	POS printer	
Syntax	public void printPageModeDataMatrix (int <i>startX</i> , int <i>startY</i> , String <i>text</i> , DataMatrixModule <i>dataMatrixModule</i> , ModuleSize <i>moduleSize</i>) throws PrinterException	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Barcode data
	<i>dataMatrixModule</i>	The number of Data Matrix modules See Table 4-17 Module Size (ModuleSize) for available constants.
	<i>moduleSize</i>	Module size See Table 4-15 Alignment (PrintAlignment) for available constants.
Description	This method maps Data Matrix on the print area of page mode.	
	Start page mode by enterPageMode method before executing this method.	
Error	PrinterException may be thrown when the method is called.	
Note	Map the print data of the barcode not to overlap the other print data.	

printPageModeMaxiCode**Print MaxiCode of page mode**

Target	POS printer	
Syntax	public void printPageModeMaxiCode (int <i>startX</i> , int <i>startY</i> , String <i>text</i> , MaxiCodeMode <i>maxiCodeMode</i>) throws PrinterException	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.

<i>text</i>	Barcode data <ul style="list-style-type: none"> When <i>maxiCodeMode</i> is MAXI_CODE_2 Add the service class (3 digits), the country code (3 digits), and the postal code (9 digits) to the beginning of the data. When <i>maxiCodeMode</i> is MAXI_CODE_3 Add the service class (3 digits), the country code (3 digits), and the postal code (6 digits) to the beginning of the data.
<i>maxiCodeMode</i>	MaxiCode Mode See Table 4-25 MaxiCode Mode (MaxiCodeMode) for available constants.
Description	This method maps MaxiCode on the print area of page mode. Start page mode by enterPageMode method before executing this method.
Error	PrinterException may be thrown when the method is called.
Note	Map the print data of the barcode not to overlap the other print data.

sendPageModeBinary	Send binary data of page mode
--------------------	-------------------------------

Target	POS printer
Syntax	public void sendPageModeBinary (byte [] <i>binary</i>) throws PrinterException
Parameter	<i>binary</i> Binary data Data size that can be specified at 1 time is 16 KB (16384 bytes).
Description	This method maps binary data on the print area of page mode. Start page mode by enterPageMode method before executing this method. This method sends the specified binary data to the printer without conversion. By sending printer commands as binary data with this method, printer functions which are not supported in the library become available.
Error	PrinterException may be thrown when the method is called.
Note	This method may execute unexpected performance depending on the data to send. Please ensure the performance with your actual device in advance.

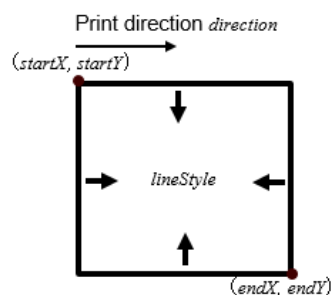
printPageModelImageFile	Draw Image file of page mode
-------------------------	------------------------------

Target	POS printer
Syntax	public void printPageModelImageFile (int <i>startX</i> , int <i>startY</i> , String <i>fileName</i> , Dithering <i>dithering</i>) throws PrinterException
Parameter	<i>startX</i> The horizontal reference point (dot) from the starting point The valid range is 0 to 2399. <i>startY</i> The vertical reference point (dot) from the starting point The valid range is 0 to 2399.

<i>fileName</i>	File path of the data The maximum file size that can be specified is 1 MB (1048576 bytes). The image files that can be sent are .bmp, .jpg, .jpeg, .png. Colored image file is converted to monochrome image by binarization and registered.
<i>dithering</i>	Dithering See Table 4-28 Dithering (Dithering) for available constants.
Description	This method maps the image file on the print area of page mode. Start page mode by enterPageMode method before executing this method.
Error	PrinterException may be thrown when the method is called.

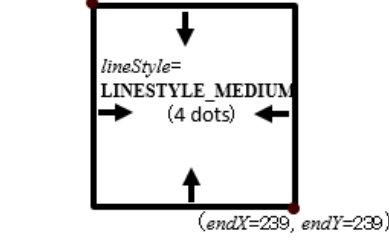
printPageModeRectangle	Draw rectangle image of page mode
-------------------------------	------------------------------------------

Target	POS printer
Syntax	public void printPageModeRectangle (int <i>startX</i> , int <i>startY</i> , int <i>endX</i> , int <i>endY</i> , LineStyle <i>LineStyle</i>) throws PrinterException
Parameter	<div> <div><i>startX</i></div> <div>The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.</div> </div> <div> <div><i>startY</i></div> <div>The vertical reference point (dot) from the starting point The valid range is 0 to 2399.</div> </div> <div> <div><i>endX</i></div> <div>The horizontal reference point (dot) from the ending point The valid range is 0 to 2399.</div> </div> <div> <div><i>endY</i></div> <div>The vertical reference point (dot) from the ending point The valid range is 0 to 2399.</div> </div> <div> <div><i>lineStyle</i></div> <div>Line style See Table 4-32 Line style (LineStyle) for available constants.</div> </div>
Description	<p>This method maps the rectangle image on the print area of page mode.</p> <p>Start page mode by enterPageMode method before executing this method.</p> <p>The rectangle is mapped to <i>direction</i> of setPageModeDirection method as shown in the figure below.</p>



The example of the parameter setting to the image is shown below.

Example: Draw a square with a medium solid line (4 dots) at 240 dots (30 mm) from the starting point.

Image	Parameter
 <p>The diagram shows a square drawn with a medium solid line (4 dots) at 240 dots (30 mm) from the starting point. The starting point is at (startX=0, startY=0) and the ending point is at (endX=239, endY=239). The line style is LINESTYLE_MEDIUM.</p>	<pre> startX 0 startY 0 endX 239 endY 239 lineStyle LINESTYLE_MEDIUM </pre>

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

printPageModeLine Print ruled line of page mode

Target POS printer

Syntax `public void printPageModeLine(int startX,
int startY,
int endX,
int endY,
LineStyle LineStyle) throws PrinterException`

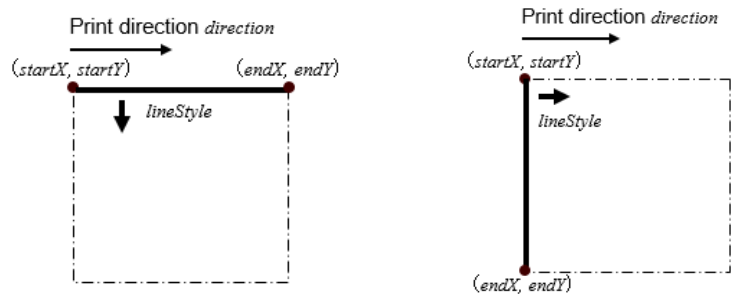
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>endX</i>	The horizontal reference point (dot) from the ending point The valid range is 0 to 2399.
	<i>endY</i>	The vertical reference point (dot) from the ending point The valid range is 0 to 2399.
	<i>lineStyle</i>	Line style See Table 4-32 Line style (LineStyle) for available constants.

Description This method maps the ruled line on the print area of page mode.

Start page mode by **enterPageMode** method before executing this method.

A diagonal stroke cannot be drawn by this method.

The ruled line is mapped to the *direction* of **setPageModeDirection** method as shown in the figure below.



Mapping direction of horizontal ruled line Mapping direction of vertical ruled line

The setting example of the parameter to the image is shown below.
Example: Draw a horizontal ruled line of a square with a medium solid line (4 dots) at 240 dots (30 mm) from the starting point.

Image	Parameter
<p>The diagram shows a square with two horizontal ruled lines. The top line is labeled ① and has coordinates $(startX=0, startY=0)$ and $(endX=239, endY=0)$. The bottom line is labeled ② and has coordinates $(startX=0, startY=236)$ and $(endX=239, endY=236)$. Both lines are labeled with $lineStyle=$ LINESTYLE_MEDIUM (4 dots). Arrows indicate the direction of the lines.</p>	<p>①</p> <p><i>startX</i> 0 <i>startY</i> 0 <i>endX</i> 239 <i>endY</i> 0 <i>lineStyle</i> LINESTYLE_MEDIUM</p> <p>②</p> <p><i>startX</i> 0 <i>startY</i> 236 <i>endX</i> 239 <i>endY</i> 236 <i>lineStyle</i> LINESTYLE_MEDIUM</p>

Example: Draw a vertical ruled line of a square with a medium solid line (4 dots) at 240 dots (30 mm) from the starting point.

Image	Parameter
<p>The diagram shows a square with two vertical ruled lines. The left line is labeled ① and has coordinates $(startX=0, startY=0)$ and $(endX=0, endY=239)$. The right line is labeled ② and has coordinates $(startX=236, startY=0)$ and $(endX=236, endY=239)$. Both lines are labeled with $lineStyle=$ LINESTYLE_MEDIUM (4 dots). Arrows indicate the direction of the lines.</p>	<p>①</p> <p><i>startX</i> 0 <i>startY</i> 0 <i>endX</i> 0 <i>endY</i> 239 <i>lineStyle</i> LINESTYLE_MEDIUM</p> <p>②</p> <p><i>startX</i> 236 <i>startY</i> 0 <i>endX</i> 236 <i>endY</i> 239 <i>lineStyle</i> LINESTYLE_MEDIUM</p>

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

Target	POS printer	
Syntax	public void printPageModeLogo (int <i>startX</i> , int <i>startY</i> , String <i>id</i>) throws PrinterException	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>id</i>	Logo ID to print (key code) Specify the ID of the registered logo as a character string
Description	This method maps the registered logo on the print area of page mode. Start page mode by enterPageMode before executing this method.	
Error	PrinterException may be thrown when the method is called.	

4.3.2 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 4-41.

Table 4-41 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 4-42.

Table 4-42 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 4-42 Event Type Constant for details.

4.3.3 PrinterListener Interface

(1) Method List

This interface obtains the complete event of printer searching.
Methods of **PrinterListener** Interface are shown in Table 4-43.

Table 4-43 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.

4.3.4 PrinterInfo Class

(1) Method List

This class stores the printer information searched by printer searching method.

Printer model name, Bluetooth address, MAC address, IP address, port name (device path) and pairing status can be obtained by searched printer information. Methods of **PrinterInfo** class are shown in Table 4-44.

The available method varies depending on the target printer either of Mobile printer or POS printer.

Table 4-44 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
getIsBonded	Obtain pairing status	Not supported	Supported
getDevicePath	Obtain device path	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.

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.
Return value	IP address

getIsBonded	Obtain pairing status
--------------------	------------------------------

Target	POS printer
Syntax	public String getIsBonded()
Description	This method is valid only for POS printer. This method obtains the status of pairing from printer information searched by the printer search.
Return value	true Paired false Not paired

getDevicePath	Obtain device path
----------------------	---------------------------

Target	Mobile printer / POS printer
Syntax	public String getDevicePath()
Description	This method obtains the character string of the USB device file path from the printer information found by the printer search.
Return value	Device path

4.3.5 PrinterException Class

(1) Method List

The list of methods provided by **PrinterException** class is shown in Table 4-45.

Table 4-45 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 4-46.

Table 4-46 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_OFFLINE	Disconnected state or the printer is offline.	-22	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

Constant Name	Description	Value	Target	
			Mobile	POS
ERROR_TIMEOUT	Send timeout happened.	-201	Supported	Supported
	Receive timeout happened.		Supported	Supported
ERROR_FILE_NOT_FOUND	Specified file not found.	-301	Supported	Supported
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 file file.	-311	Supported	Supported
ERROR_OVER_MAX_IMAGE	Either width or height of image file, or both of them exceed the number of printable maximum dots.	-312	Supported	Supported
ERROR_LOGO_NOT_DEFINED	The logo 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	The specified value for the style sheet number is invalid.	-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_PAGE_MODE_SPECIFIED	Page mode is specified.	-511	Not supported	Supported
ERROR_PAGE_MODE_NOT_SPECIFIED	Page mode is not specified.	-512	Not supported	Supported
ERROR_INVALID_PARAM	Invalid parameter specified.	-9999	Supported	Supported

*1 Abnormality processing might have happened.

(3) Method Details

PrinterException	Constructor
------------------	-------------

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

PrinterException	Constructor
------------------	-------------

Target	Mobile printer / POS printer
Syntax	public PrinterException (int <i>code</i> , String <i>message</i> , String <i>detail</i>)
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 4-46 Error Codes List for details.

4.3.6 CallbackFunctionListener Interface

CallbackFunctionListener Interface is an interface for getting the change event of printer status.

(1) Method List

Methods provided by **CallbackFunctionListener** interface are shown in Table 4-1. Available method differs depending on the target printer: Mobile printer or POS printer.

Table 4-47 Method of CallbackFunctionListener Interface

Method	Function Summary	Target	
		Mobile	POS
onStatusChanged	Change event of printer status	Supported	Supported

(2) Method Details

onStatusChanged	Change event of printer status
------------------------	--------------------------------

Target Mobile printer / POS printer

Syntax public void **onStatusChanged**(int *status*)

Parameter *status* Printer status

Description This method is called at the following timing.
 ·When **setCallbackFunctionListener** is executed.
 ·When the printer status is changed.

The change event of printer status is notified when **isConnect** is true.

This method is an interface, so it is not implemented.
Implement the optional process in the class that receives a callback of the printer status change.

Do not execute the APIs of **PrinterManager** within this method.

4.3.7 BarcodeScannerListener Interface

BarcodeScannerListener Interface is an interface for getting barcode scanner connection, barcode scanner disconnection, or received barcode data.

Mobile printer and POS printer do not support this interface.

Chapter 5

Sample Program

This chapter describes the sample programs provided by the SDK.

The SDK includes the sample program with Android Studio project format.

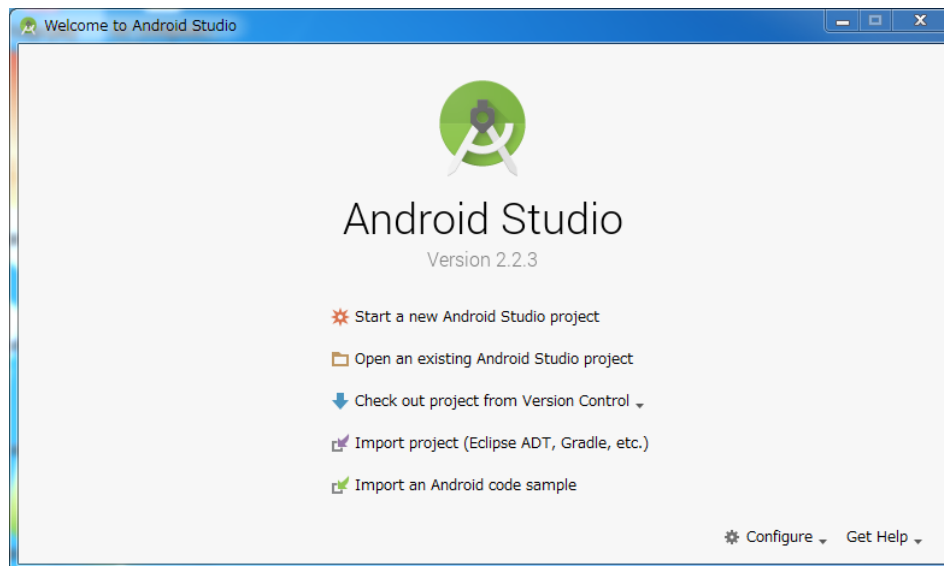
5.1 Installation

Install the sample program.

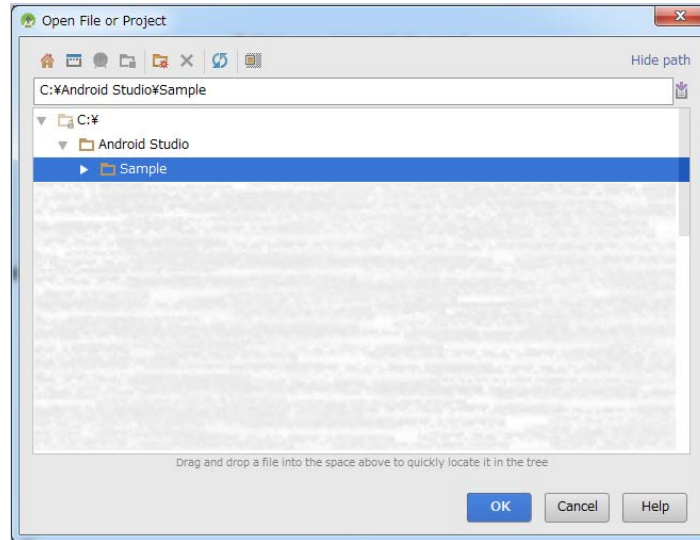
Ensure that the environment for developing Android application is prepared. See "Chapter 3 How to Use the Library" for details about required development environment.

Procedures are shown below.

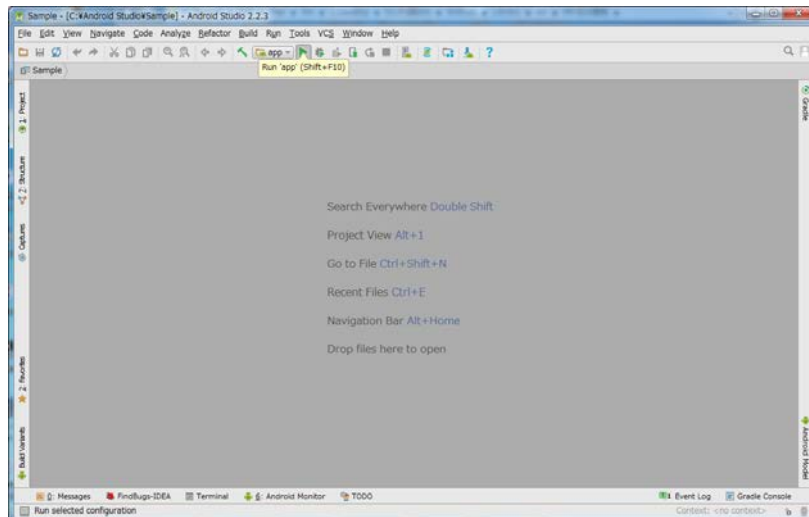
- (1) Deploy the sample folder at any location.
- (2) Start Android Studio, and click "Open an existing Android Studio Project".



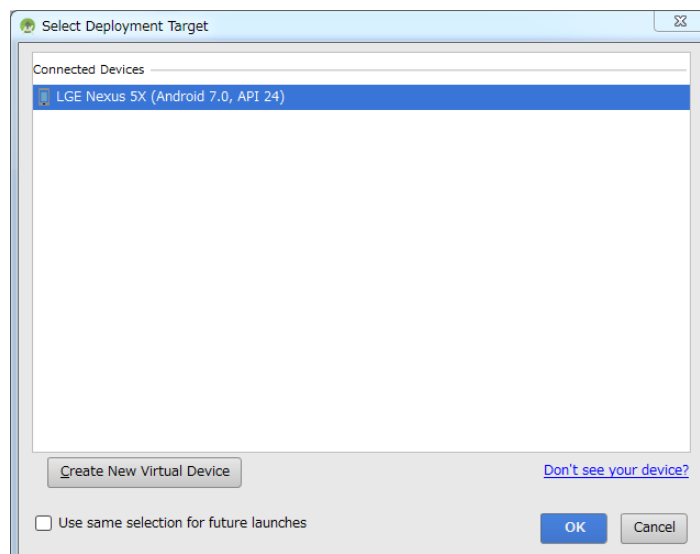
(3) Select the folder that has been deployed in procedure (1), and click [OK].



(4) Click [Run 'app'].



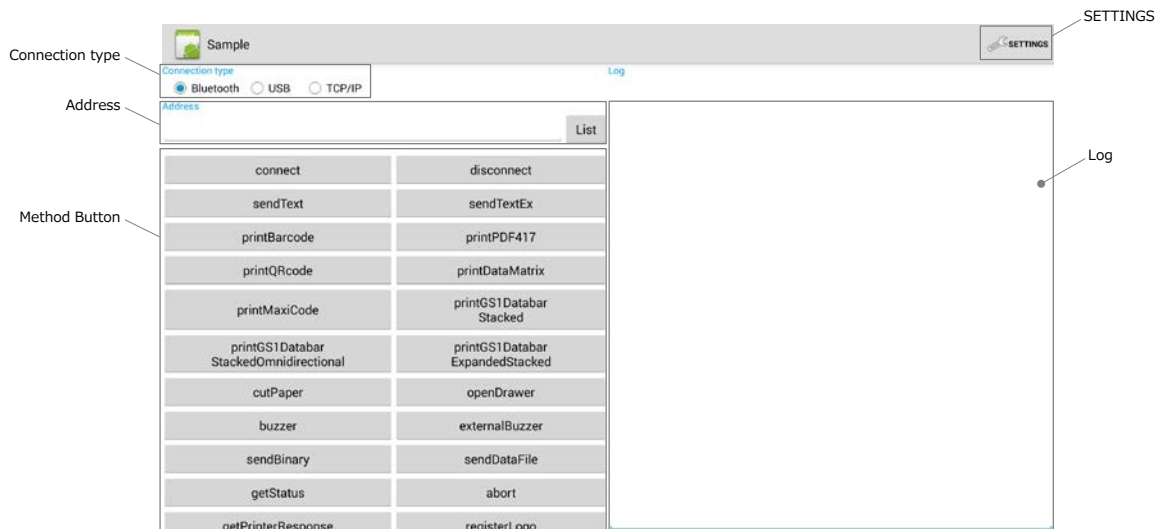
(5) Select the device, and click [OK].




5.2 Screen

This chapter describes the screen of the sample program.

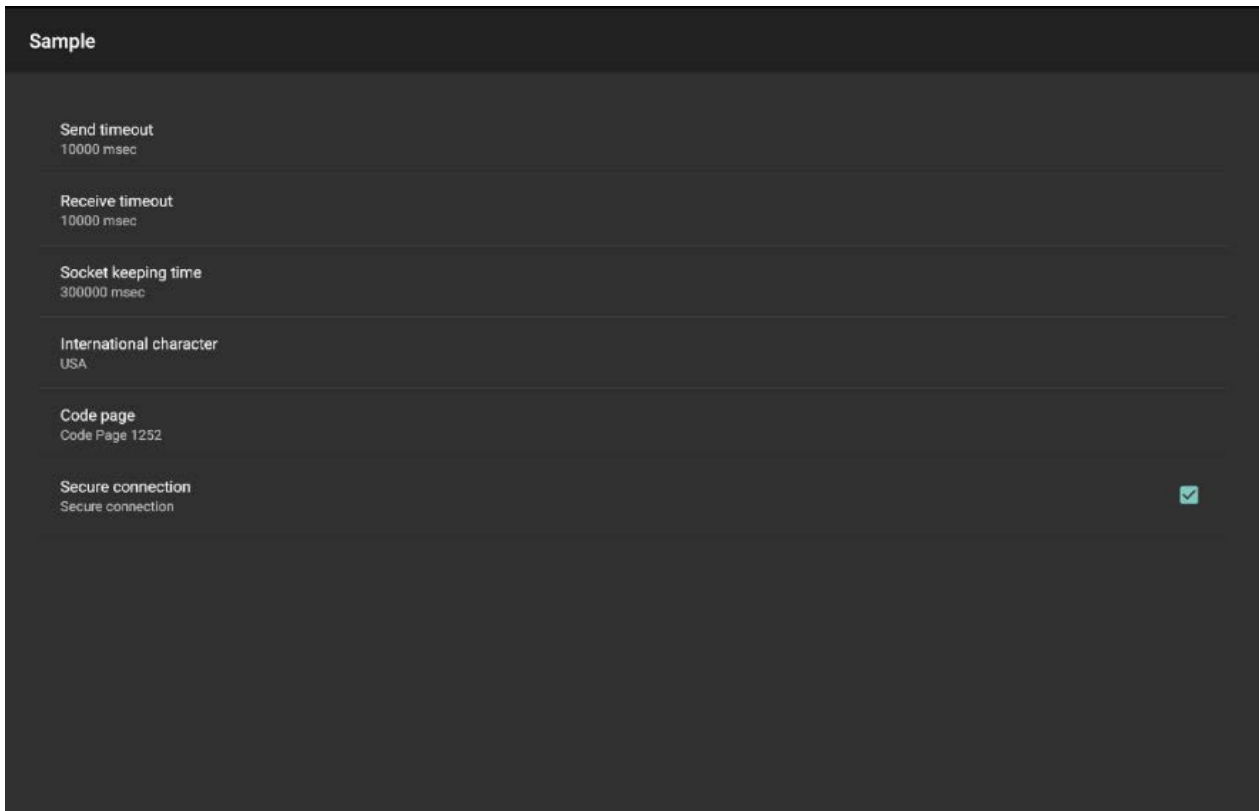
5.2.1 Main screen



Item	Description
Connection type	Selects connection form to a printer.
Address	<p>Specifies the printer address.</p> <p>For manual input: When connecting with Bluetooth, enter the Bluetooth address. When connecting with TCP/IP, enter the MAC address.</p> <p>For automatic input: By tapping the [List] button, the information of printers found by startDiscoveryPrinter(Bluetooth) is displayed in a list. When a printer from the displayed list is selected, the Bluetooth address is automatically entered. When USB is selected in Connection type, the address of the printer cannot be specified.</p>
Method Button	<p>The buttons for executing each method are located.</p> <p>By scrolling, it is possible to see the methods and properties that are not displayed.</p> <p>See "Chapter 4 Functions of the Library" for details of each method.</p>
SETTINGS	<p>Tapping the [SETTINGS] button opens the function setting screen.</p> <p>In order to go back to the main screen, tap  on the screen.</p>
Log	Executing each method of "Method Button" displays the method execution logs.

5.2.2 [SETTINGS] screen

Each setting function is displayed in [SETTINGS].



5.3 Precaution

Sample programs are subject to change without notice.

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

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)

The codepages when **COUNTRY_USA** is set for the international character set are shown below. Print results of the specific character codes vary depending on the setting of the international character set. See "A.2 International Character Set" for the specific character codes.

- Mobile printer

	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	ア	イ	ウ	エ	オ	ヤ	ユ	ヨ	ツ	
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	ゝ	。
E0																
F0																

Figure A-1 CODE_PAGE_KATAKANA

	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	€	,	;	„	”	•	-	-	~	™	š	<	œ	ž		
90																
A0	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯	
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figure A-2 CODE_PAGE_1252 (Latin)

- POS printer

	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	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ï	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ü	ÿ	Ö	Ü	φ	£	¥	℔	ƒ
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	¡	«	»	
B0	▒	▒	▒													
C0	L	┐	└	┌	┐	└	┌	┐	└	┌	┐	└	┌	┐	└	┌
D0	┐	└	┌	┐	└	┌	┐	└	┌	┐	└	┌	┐	└	┌	┐
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤	┌	┐	÷	≈	°	•	•	√	n	2	■	

Figure A-3 CODE_PAGE_437 (USA, Standard Europe)

	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	ア	イ	ウ	エ	オ	ヤ	ユ	ヨ	ッ	
B0	ー	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	ゝ	。
E0																
F0																

Figure A-4 CODE_PAGE_KATAKANA

	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	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ü	ÿ	Ö	Ü	ø	£	Ø	×	f
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©	¶		¶	¶	¶	¥	γ
C0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	ð	Ð	Ê	Ë	È	Ì	Í	Î	Ï	⌋	⌋	■	■	■	■	■
E0	Ó	β	ô	ò	õ	õ	μ	þ	þ	Ú	Û	Ü	ý	Ý	-	'
F0	-	±	=	¾	¶	§	÷	,	°	..	.	1	3	2	■	

Figure A-5 CODE_PAGE_850 (Multilingual)

	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	Ç	ü	é	â	ã	à	Á	ç	ê	Ê	è	Í	Ô	ì	Ã	Â
90	É	À	È	ô	õ	ò	Ú	ù	Ì	Õ	Ü	φ	£	Ù	Þ	Ó
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	Ò	¬	½	¼	¡	«	»
B0	☐	☐	☐													
C0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	.	.	√	n	2	■	

Figure A-6 CODE_PAGE_860 (Portuguese)

	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	Ç	ü	é	â	À	à	¶	ç	ê	ë	è	ï	î	≡	À	§
90	É	È	Ê	Ô	Ë	Ï	Ù	Ú	×	Ô	Ü	¢	£	Ü	û	f
A0		´	ó	ú	¨	³	-	î	¬	½	¼	¾	«	»		
B0	▒	▒	▒		†	‡		π	¶			¶	¶	¶	¶	¶
C0	L	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	•	•	√	n	²	■	

Figure A-7 CODE_PAGE_863 (Canadian-French)

	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	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ü	ÿ	Ö	Ü	ø	£	Ø	Pt	f
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	¾	«	»	α
B0	▒	▒	▒		†	‡		π	¶			¶	¶	¶	¶	¶
C0	L	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	•	•	√	n	²	■	

Figure A-8 CODE_PAGE_865 (Nordic)

	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	€	‘	‚	“	”	…	†	‡	^	‰	Š	<	Œ		Ž	
90		‚	‚	“	”	•	-	-	~	™	š	>	œ		ž	ÿ
A0		ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	-
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figure A-9 CODE_PAGE_1252 (Latin)

	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	Ç	ü	é	â	ä	û	ç	ı	ë	Ö	ö	ı	ž	Ä	Ć	
90	É	Í	Í	ô	ö	ı	š	š	Ö	Ü	ř	ř	ı	×	č	
A0	á	í	ó	ú	À	à	Ž	ž	È	è	¬	ž	Č	š	«	»
B0	▒	▒	▒			Á	Â	Ë	Š	š	Œ	Œ	Œ	Ž	ž	ı
C0	L	L	T			Ä	ä	ı	ı	ı	ı	ı	ı	=	ı	ı
D0	đ	Đ	Đ	È	đ	Ň	Í	Î	ě	ı	ı	ı	ı	ı	ı	ı
E0	ó	ß	ô	Ň	ň	ň	š	š	Ř	Ú	ř	Ú	ý	Ý	ı	ı
F0	-	"	ı	ı	ı	š	÷	ı	ı	ı	ı	ı	ı	ı	ı	ı

Figure A-10 CODE_PAGE_852 (Eastern Europe)

	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	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ü	ÿ	Ö	Ü	ø	£	Ø	×	f
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©	¶	¶	¶	¶	¶	¥	₱
C0	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł	Ł
D0	đ	Đ	Ê	Ë	È	€	Í	Î	Ï	Ј	Ј	■	■	■	■	■
E0	ó	ß	ô	ò	õ	õ	μ	þ	þ	ú	û	ü	ý	ý	-	'
F0	-	±	=	¾	¶	§	÷	,	°	..	.	1	3	2	■	

Figure A-11 CODE_PAGE_858 (Euro)

	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	β	∞	φ	±	½	¼	≈	«	»	ل	أ					
A0	-	ل	£	¤	ل			ل	ب	ت	ث	،	ج	ح	خ	
B0	•	١	٢	٣	٤	٥	٦	٧	٨	٩	ف	؛	س	ص	ش	؟
C0	¢	ء	آ	أ	ؤ	ع	ئ	ب	ة	ت	ث	ج	ح	خ	د	
D0	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ	ف	ق	ك	خ	
E0	-	ف	ق	ك	م	ل	هـ	و	ي	ض	ع	غ	ف	ق	ك	
F0	-	”	”	ن	هـ	ي	ي	ق	ل	ل	ل	ل	ل	ل	ل	■

Figure A-12 CODE_PAGE_864 (Arabic)

	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	€	‘	’	“	”	…	†	‡	‰	Š	Š	Š	Š	Š	Š	Š
90		‘	’	“	”	•	-	-	™	š	š	š	š	š	š	š
A0	˘	˘	Ł	ł	Ą	ą	Ś	ś	©	§	«	»	–	®	Ž	
B0	°	±	ı	ı	μ	¶	•	·	ª	§	»	Ł	”	ł	ž	
C0	Ř	Á	Â	Ã	Ä	Å	Ĺ	Ć	Ç	Č	É	Ê	Ë	Ě	Í	Î
D0	Đ	Ń	Ň	Ó	Ô	Õ	Ö	×	Ř	Ů	Ú	Ů	Ů	Ů	Ý	Ů
E0	ř	á	â	ã	ä	å	ĺ	ć	ç	č	é	ê	ë	ě	í	î
F0	đ	ń	ň	ó	ô	õ	ö	÷	ř	ů	ú	ů	ů	ů	ý	ţ

Figure A-13 CODE_PAGE_1250 (Central European)

	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	ђ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ
90	ђ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ	ѓ
A0	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ	ѐ
B0	°	±	І	і	г	г	г	г	г	г	г	г	г	г	г	г
C0	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D0	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Figure A-14 CODE_PAGE_1251 (Cyrillic)

	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	“	À	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	-	
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	í	À	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O
D0	Π	P		Σ	T	Υ	Φ	X	Ψ	Ω	İ	ÿ	á	é	ή	ί
E0	ÿ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
F0	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ï	ÿ	ó	ύ	ώ	

Figure A-15 CODE_PAGE_1253 (Greek)

	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	€	‘	‚	“	”	…	†	‡	‰	™	Š	<	Œ			
90		‚	“	”	•	-	-	~	™	š	>	œ			ÿ	
A0	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	-	
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ğ	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	İ	Ş	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ğ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ı	ş	ÿ

Figure A-16 CODE_PAGE_1254 (Turkish)

A.2 International Character Set

Print results of the specific character codes vary depending on the setting of the international character set. The following table shows the specific character codes and their print results.

	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-17 International Character Set