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

U00139060116

Seiko Instruments Inc.

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
U00139060110	March 2022
U00139060111	October 2022
U00139060112	April 2023
U00139060113	March 2024
U00139060114	January 2025
U00139060115	February 2026
U00139060116	June 2026

Copyright © 2012-2026 by Seiko Instruments Inc. All rights reserved.

Microsoft®, Windows®, and Microsoft Store are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.
Android™ is a trademark of Google LLC.
Google Play and the Google Play logo are trademarks of Google LLC.
The Bluetooth® word mark is registered trademarks owned by Bluetooth SIG, Inc.
Oracle and Java® are registered trademarks of Oracle Corporation and/or its affiliates.
The company names and product names appearing on our manual are trademarks or registered 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-S245-01C-E
		USB	DPU-S245-0xC-E
	DPU-S445	Bluetooth	DPU-S445-01C-E
		USB	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 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 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 Precautions 2-2

Chapter 3 How to Use library 3-1

3.1 Development Environment for Android Application 3-1
3.2 Provided Files 3-2
3.3 Add Library to Android Studio Projects 3-3
3.4 Use Developed Android Application on Android Device 3-5
3.5 Precautions 3-5

Chapter 4 Function of Library 4-1

4.1 Standard Mode and Page Mode 4-1
4.1.1 Basic Operation 4-1
(1) Standard mode 4-1
(2) Page mode 4-2
4.1.2 Text Data Printing in Standard Mode 4-3
4.1.3 Mapping Position of Print Data in Page Mode 4-4
(1) Print area of page mode 4-4
(2) Print direction 4-4
(3) Reference point 4-5
4.1.4 Print Data Process at Out of Print Area of Page Mode 4-6
4.2 Log File Output Function 4-7
4.2.1 How to Set Log Output 4-7
4.2.2 Log Output Settings 4-7
4.2.3 Log File 4-7
4.3 Package of Library 4-8
4.4 API Reference 4-9
4.4.1 PrinterManager Class 4-9
(1) Method List 4-9
(a) Common method to standard mode and page mode 4-10
(b) Dedicated method for standard mode 4-11

(c) Dedicated method for page mode 4-12
(2) Constant List 4-13
(a) Printer Model Constant 4-13
(b) Response Type Constant 4-13
(c) International Character Set Constant 4-14
(d) Codepage Constant 4-14
(e) Port Type Constant 4-15
(f) Constant for barcode or PDF417 4-15
(3) Constant List of Enumerated Type 4-16
(a) Drawer number (DrawerNum) 4-16
(b) Activation pulse width (PulseWidth) 4-16
(c) Dithering (Dithering) 4-16
(d) Batch processing selection (TransactionFunction) 4-17
(e) Bold print (CharacterBold) 4-17
(f) Underline (CharacterUnderline) 4-17
(g) Reverse print (CharacterReverse) 4-17
(h) Inversion print (CharacterInversion) 4-18
(i) Character font (CharacterFont) 4-18
(j) Character Scale (CharacterScale) 4-18
(k) Alignment (PrintAlignment) 4-19
(l) Pending data output specifying (OutputPendingData) 4-19
(m) Barcode symbol (BarcodeSymbol) 4-20
(n) Module size (ModuleSize) 4-20
(o) HRI character print position (HriPosition) 4-22
(p) N:W ratio (NwRatio) 4-22
(q) Error correction level (ErrorCorrection) 4-23
(r) PDF417 symbol (Pdf417Symbol) 4-23
(s) QR code model (QrModel) 4-24
(t) Data Matrix modules (DataMatrixModule) 4-24
(u) MaxiCode Mode (MaxiCodeMode) 4-25
(v) Cutting method (CuttingMethod) 4-25
(w) Image rotation direction (Rotate) 4-25
(x) Image scaling (ImageScale) 4-26
(y) Print direction (Direction) 4-26
(z) Line style (LineStyle) 4-26
(4) Method Details 4-27
(a) Common method to standard mode and page mode 4-27
PrinterManager Constructor 4-27
connect Start communicating with a printer (Bluetooth) 4-27
connect Start communicating with a printer (USB) 4-28
connect Start communicating with a printer (TCP/IP) 4-28
disconnect Disconnect a printer 4-29
setBarcodeScannerListener
Start/End callback of barcode scanner 4-29
openDrawer Open cash drawer 4-29
buzzer Sound buzzer 4-30
externalBuzzer Sound external buzzer 4-30

getStatus Obtain printer status 4-30
setCallbackFunctionListener
Start/End callback of printer status change 4-32
abort Abort the waiting state of a printer 4-32
registerLogo Register logo (image) to a printer 4-33
registerLogo Register logo (image) to a printer 4-33
unregisterLogo Delete specified logo (image) in a printer 4-34
unregisterLogo Delete specified logo (image) in a printer 4-35
registerStyleSheet Register style sheet to a printer 4-35
unregisterStyleSheet Delete specified style sheet in a printer 4-36
resetPrinter Printer hardware reset 4-36
getPrinterResponse Obtain various responses from a printer 4-36
startDiscoveryPrinter Start printer search (Bluetooth) 4-38
startDiscoveryPrinter Start printer search (USB) 4-39
startDiscoveryPrinter Start printer search (TCP/IP) 4-39
cancelDiscoveryPrinter
Cancel printer search 4-40
getFoundPrinter Obtain searched printer information 4-40
getSendTimeout Obtain send timeout period 4-40
setSendTimeout Set send timeout period 4-40
getReceiveTimeout Obtain receive timeout period 4-41
setReceiveTimeout Set receive timeout period 4-41
getInternationalCharacter
Obtain international character set 4-41
setInternationalCharacter
Set international character set 4-42
getCodePage Obtain codepage 4-42
setCodePage Set codepage 4-42
getPrinterModel Obtain printer model 4-43
getPortType Obtain connection port type 4-43
isConnect Verify connection state with a printer 4-43
getSocketKeepingTime
Obtain socket keeping time 4-44
setSocketKeepingTime
Set socket keeping time 4-44
getVersion Get SDK version 4-44
controlTransaction Start/End batch processing 4-44
(b) Dedicated method for standard mode 4-46
sendText Send text data 4-46
sendTextEx Send format specified text data 4-46
sendTextEx Send format specified text data 4-47
printBarcode Print barcode 4-48
printPDF417 Print PDF417 4-51
printQRcode Print QR code 4-52
printDataMatrix Print Data Matrix 4-53
printMaxiCode Print MaxiCode 4-53

printGS1DataBarStacked
Print GS1 Databar Stacked 4-54
printGS1DataBarStackedOmnidirectional
Print GS1 Databar Stacked Omni-directional..... 4-54
printGS1DataBarExpandedStacked
Print GS1 Databar Expanded Stacked..... 4-54
printAztecCode Print Aztec Code 4-55
cutPaper Cut paper..... 4-55
feedPosition Paper form feed..... 4-55
sendBinary Send binary data 4-55
sendDataFile Send specified file 4-56
sendImage Send image 4-57
printPDF Print PDF page..... 4-58
printLogo Print specified logo (image) in printer..... 4-59
printLogo Print specified logo (image) in printer..... 4-60
(c) Dedicated method for page mode 4-61
enterPageMode Start page mode..... 4-62
exitPageMode End page mode 4-62
setPageModeArea Specify print area of page mode 4-62
setPageModeDirection
Specify print direction of page mode..... 4-64
setPageModeLineSpacing
Specify line spacing of page mode 4-64
printPageMode Print page mode 4-64
printPageModeText Send text data of page mode 4-65
printPageModeTextEx
Send format specified text data of page mode... 4-65
printPageModeBarcode
Print barcode of page mode 4-66
printPageModePDF417
Print PDF417 of page mode..... 4-68
printPageModeQRcode
Print QR Code of page mode 4-69
printPageModeDataMatrix
Print Data Matrix of page mode..... 4-70
printPageModeMaxiCode
Print MaxiCode of page mode..... 4-71
printPageModeGS1DataBarStacked
Print GS1 Databar Stacked of page mode..... 4-71
printPageModeGS1DataBarStackedOmnidirectional
Print GS1 Databar Stacked Omni-directional
of page mode..... 4-72
printPageModeGS1DataBarExpandedStacked
Print GS1 Databar Expanded Stacked
of page mode..... 4-72
printPageModeAztecCode
Print Aztec Code of page mode 4-72

sendPageModeBinary
Send binary data of page mode 4-72
printPageModeImageFile
Draw Image file of page mode 4-73
printPageModeRectangle
Draw rectangle image of page mode 4-74
printPageModeLine Print ruled line of page mode 4-75
printPageModeLogo Print logo of page mode 4-77
4.4.2 PrinterEvent Class 4-78
(1) Method List 4-78
(2) Constant List 4-78
(3) Method Details 4-78
getEventType Obtain event type 4-78
4.4.3 PrinterListener Interface 4-79
(1) Method List 4-79
(2) Method Details 4-79
finishEvent Finish event of the printer search 4-79
4.4.4 PrinterInfo Class 4-80
(1) Method List 4-80
(2) Method Details 4-80
getPrinterModelName
Obtain printer model name 4-80
getBluetoothAddress Obtain Bluetooth address 4-80
getMacAddress Obtain MAC address 4-80
getIpAddress Obtain IP address 4-81
getIsBonded Obtain pairing status 4-81
getDevicePath Obtain device path 4-81
4.4.5 PrinterIdentifier Class 4-82
(1) Method List 4-82
(2) Method Details 4-82
getPrinterModelID Get printer model ID 4-82
getTypeID Get type ID 4-82
getRomVersionID Get ROM version ID 4-83
getMainFirmwareVersion
Get firmware version (main) 4-83
getManufacturer Get manufacturer 4-83
getModelName Get model name 4-83
getSerialNumber Get serial number 4-83
getBootFirmwareVersion.....
Get firmware version (boot) 4-84
getBootFirmwareChecksum
Get firmware checksum (boot) 4-84
getMainFirmwareChecksum
Get firmware checksum (main) 4-84
getFirmwareChecksum
Get firmware checksum (main + boot) 4-84

4.4.6 PrinterException Class 4-85
(1) Method List 4-85
(2) Constant List 4-85
(3) Method Details 4-87
PrinterException Constructor 4-87
PrinterException Constructor 4-87
getErrorCode Obtain error codes 4-87
4.4.7 CallbackFunctionListener Interface 4-88
(1) Method List 4-88
(2) Method Details 4-88
onStatusChanged Change event of printer status 4-88
4.4.8 BarcodeScannerListener Interface 4-89
4.4.9 SmartLabelManager Class 4-90

Chapter 5 Sample Program 5-1

5.1 Screen 5-1
5.1.1 Main Screen 5-1
5.1.2 [Setting] Screen 5-3
5.2 Precaution 5-3

Chapter 6 Disclaimer 6-1

Appendix A Character Sets (Character Code Table) A-1

A.1 Character Code Table (Codepage) A-1
A.2 International Character Set A-10

Appendix B Barcode Size List B-1

B.1 Barcode Size List B-1
B.1.1 printBarcode, printPageModeBarcode B-1
B.1.2 printPDF417, printPageModePDF417 ..... B-6
B.1.3 printQRCode, printPageModeQRCode B-7
B.1.4 printDataMatrix, printPageModeDataMatrix B-8
B.1.5 printMaxicode, printPageModeMaxicode B-10

Chapter 1
Product Overview

This chapter describes the product overview of the SDK.

1.1 Function Provided by 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.

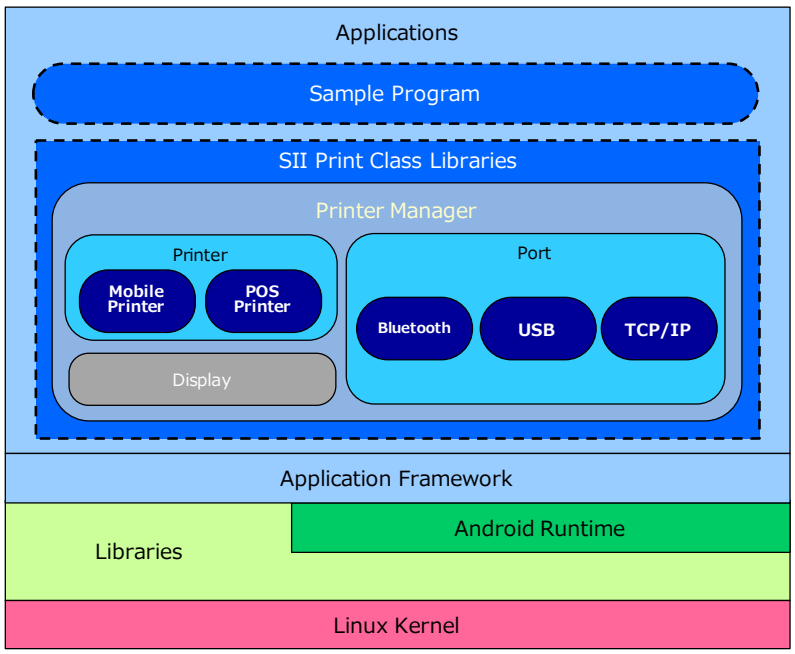


Figure 1-1

1.2.2 Function Provided by 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
- Outputting a log file

*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 relating to Display, the barcode scanners, or label printing function.

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 7.0 (API 24) to Android 16.0 (API 36)

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 Precautions

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 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.

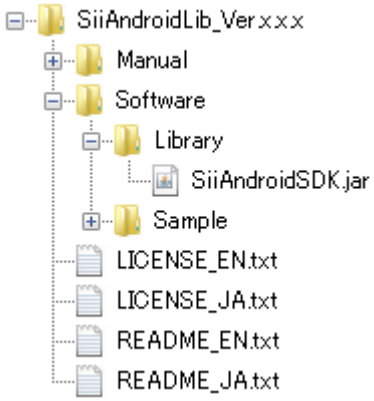


Figure 3-1

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

3.3 Add Library to Android Studio Projects

This section describes how to add the SDK to Android Studio projects.

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

(1) Create a project in Android Studio and copy the library file (SiiAndroidSDK.jar) to the [libs] folder. When the [libs] folder is not automatically created, add the folder manually.
For sample programs included in the SDK, the folder is "\Sample\app\libs".

(2) After adding the library, the view looks like Figure 3-2.

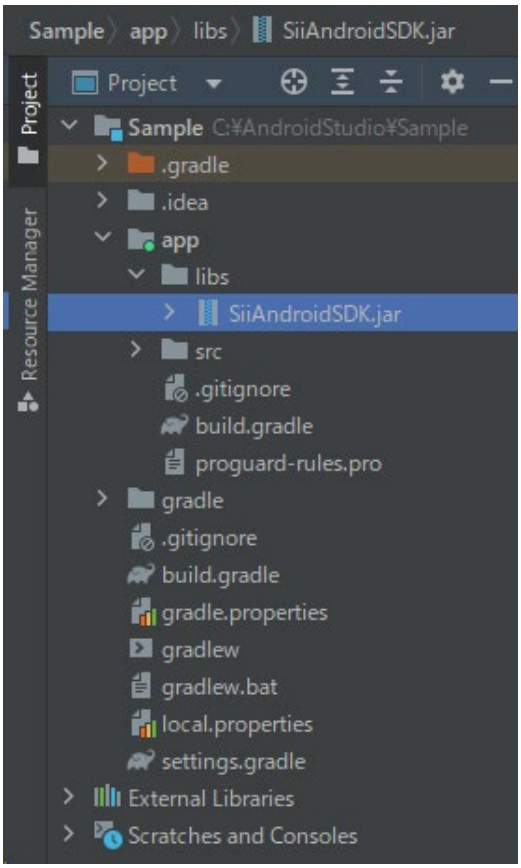


Figure 3-2

(3) Add the following to dependencies{} of build.gradle(:app) in the application.

```
implementation 'com.journeyapps:zxing-android-embedded:3.4.0@aar'
implementation 'com.google.zxing:core:3.4.1'
implementation files ('libs/SiiAndroidSDK.jar')
```

(4) Add the following codes to the beginning of the class file that uses the print class library.
(Import xxxx according to the function to use.)

```
import com.seikoinstruments.sdk.thermalprinter.PrinterManager;
import com.seikoinstruments.sdk.thermalprinter.xxxx;
```

(5) Add the following permission declaration to the application manifest (AndroidManifest.xml).
Also, implement a process (requestPermissions()) that requests the appropriate permissions for the application.

[When using Bluetooth]

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT"/>
<uses-permission android:name="android.permission.BLUETOOTH_SCAN"/>
```

[When using TCP/IP]

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
```

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, make the following settings on the Android device.

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

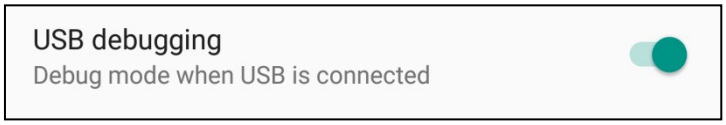


Figure 3-3

3.5 Precautions

- **About Scoped Storage**

  "Scoped Storage" that is introduced in Android 10 distinguishes between app-specific storage and external storage.
  When targeting Android 10 (API 29) or later, files that do not correspond to media files in the external storage cannot be handled directly. Files that do not correspond to media files can be handled by using the "Storage Access Framework".
  See below for details of Scoped Storage.

  - Data and file storage overview
    https://developer.android.com/training/data-storage

Chapter 4
Function of 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

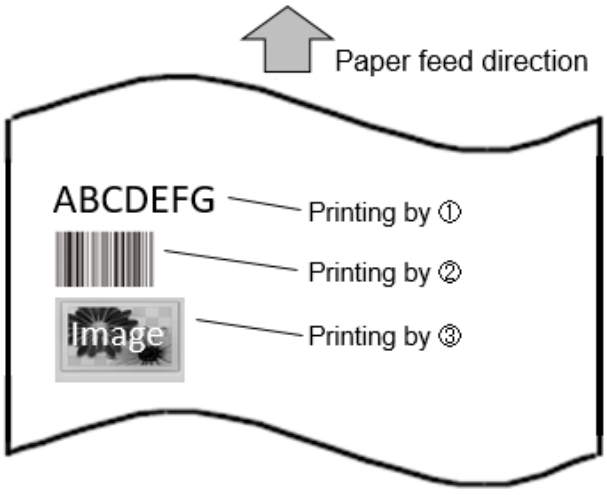
There are two printing modes "Standard mode" and "Page mode" in the library.
The "Standard mode" and "Page mode" are described below.

"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)



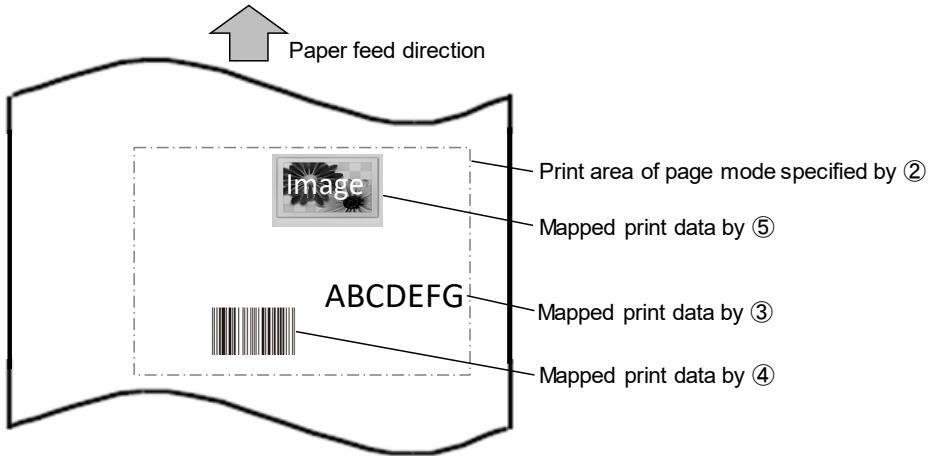
Standard mode suits the printing with an unfixed length such as a receipt.

(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.

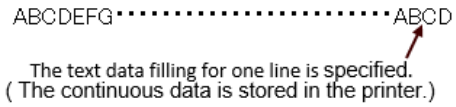
4.1.2 Text Data Printing in Standard Mode

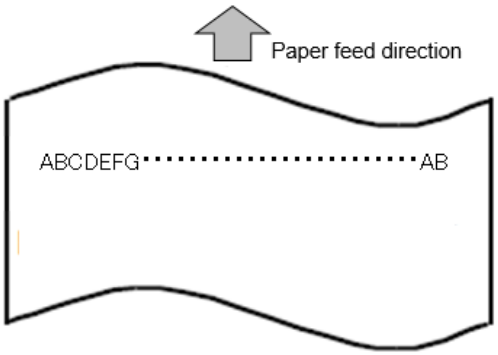
The text data in standard mode is printed each one line.

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.

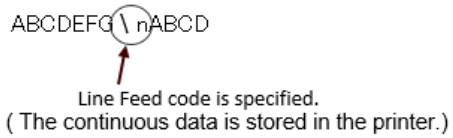
- **The print process when the text data filling for one line is specified.**

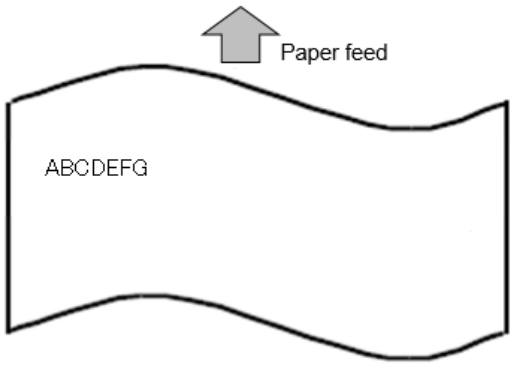




Input data **Print result**

- **The print process when Line Feed code is specified.**





Input data **Print result**

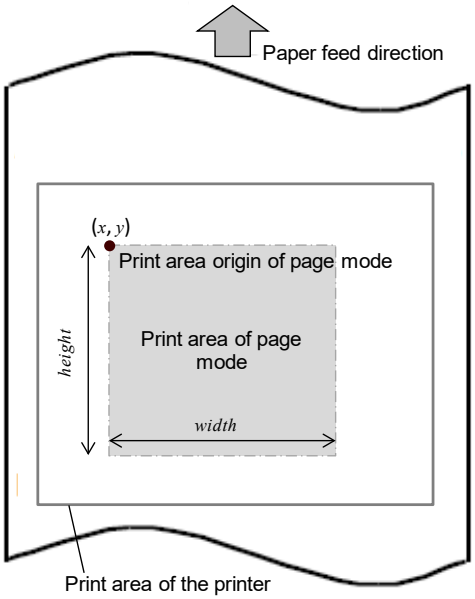
4.1.3 Mapping Position of Print Data in Page Mode

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

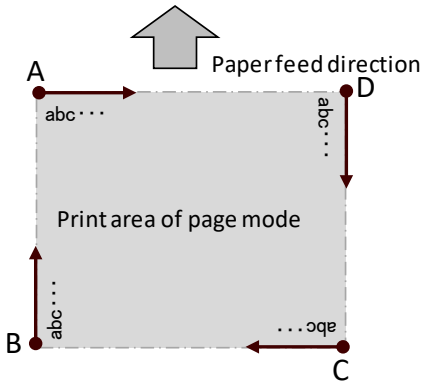
The print area of page mode is specified against the print area of the printer by the print area origin, and the width and the height of page mode. The view of the print area is shown in the following figures.

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



(2) Print direction

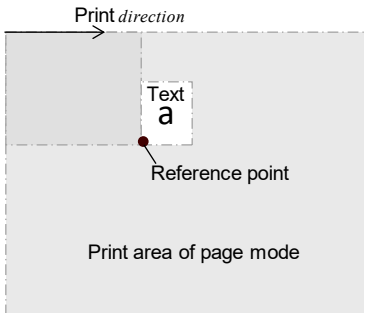
Specify the print direction at setting the print area of page mode.
The starting point is changed depending on specifying the print direction for each direction.
The relation between the print direction and the starting point is shown in the figure below.

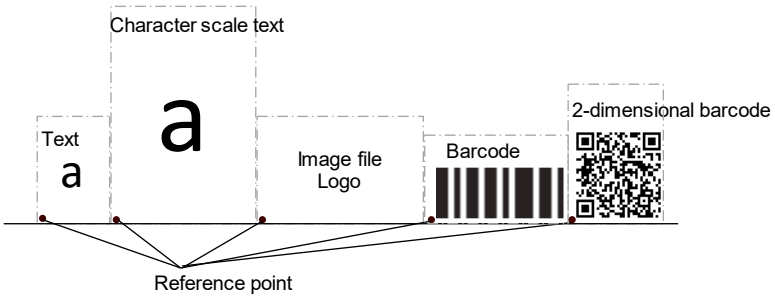


・Starting point: Upper left (A on the figure), Print direction: Left to Right
・Starting point: Left below (B on the figure), Print direction: Below to Upper
・Starting point: Right below (C on the figure), Print direction: Right to Left
・Starting point: Upper right (D on the figure), Print direction: 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.

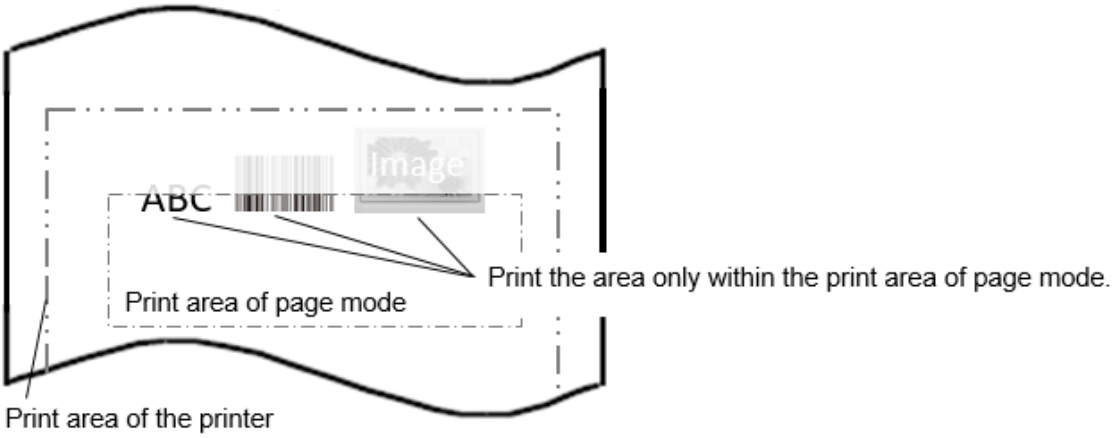
4.1.4 Print Data Process at Out of Print Area of Page Mode

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

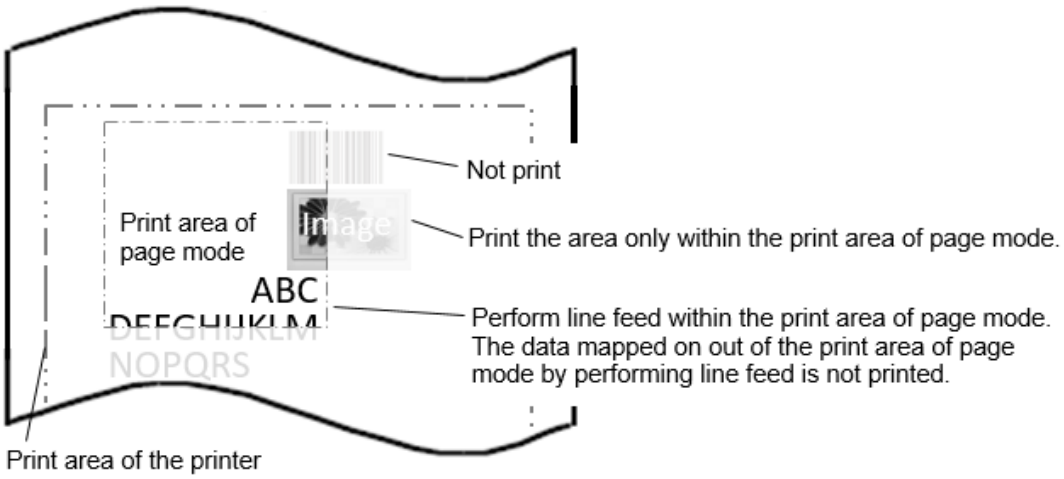
Type of Print Data

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

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



(b) 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 Log File Output Function

The logs can be retrieved and the log files can be output using the library.

4.2.1 How to Set Log Output

Log output settings can be configured by adding the config.ini file with the following content to the specific directory of the Android application that incorporates the library (e.g. internal shared storage \Android\data\<package name>\files).

config.ini

```
LOGLEVEL=x
LOGSIZEMAX=xMB
LOGOUTPUT=x
```

Reference: See "4.2.2 Log Output Settings" for details on the settings for x.

4.2.2 Log Output Settings

Item	Description	Settings
LOGLEVEL	Log level	0 : Not record the log. 1 : Records an error log when **PrinterException** is thrown. 2 : Records API execution history.
LOGSIZEMAX	Log file maximum size	1MB : Log file maximum size is 1 MB 5MB : Log file maximum size is 5 MB 10MB : Log file maximum size is 10 MB 50MB : Log file maximum size is 50 MB
LOGOUTPUT	Logcat output enabled/disabled	0 : Logcat output is disabled 1 : Logcat output is enabled

4.2.3 Log File

Log files are saved as local files in the Android application that incorporates the library.

Log file name : PrinterManagerX.log (range of X is 0 to 4)

The 1st log file is created as PrinterManager0.log. If the log file maximum size is exceeded, changes the file name to PrinterManager1.log and creates a new PrinterManager0.log.

Up to 5 log files can be created.

4.3 Package of 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.
PrinterIdentifier	Class that stores the printer ID of the printer searched by **getPrinterResponse** 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.
SmartLabelManager	Provides the API to specify label files or replace data.

See "4.4 API Reference" for more details about APIs for each class.

4.4 API Reference

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

Class Name	Description
PrinterManager	See "**4.4.1 PrinterManager Class**".
PrinterEvent	See "**4.4.2 PrinterEvent Class**".
PrinterListener	See "**4.4.3 PrinterListener Interface**".
PrinterInfo	See "**4.4.4 PrinterInfo Class**".
PrinterIdentifier	See "**4.4.5 PrinterIdentifier Class**".
PrinterException	See "**4.4.6 PrinterException Class**".
CallbackFunctionListener	See "**4.4.7 CallbackFunctionListener Interface**".
BarcodeScannerListener	See "**4.4.8 BarcodeScannerListener Interface**".
SmartLabelManager	See "**4.4.9 SmartLabelManager Class**".

(NOTE) Mobile printer and POS printer do not support the APIs relating to Display, the barcode scanners, or label printing function.

4.4.1 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 "4.4.1(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 "4.4.1(1)(b) Dedicated method for standard mode" for the methods.
Dedicated method for page mode	The valid methods in page mode. See "4.4.1(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 the following table.
See "4.4.1(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
setBarcodeScannerListener	Start/End callback of barcode scanner	Not supported	Not 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
setCallbackFunctionListerner	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

Method	Function Summary	Target	
		Mobile	POS
setCodePage	Set codepage	Supported	Supported
getPrinterModel	Obtain printer model	Supported	Supported
getPortType	Obtain connection port type	Supported	Supported
isConnect	Verify connection state with a printer	Supported	Supported
getSocketKeepingTime	Obtain socket keeping time	Not supported	Supported
setSocketKeepingTime	Set socket keeping time	Not supported	Supported
getVersion	Get SDK version	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 the following table. See "4.4.1(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
printGS1DataBarStacked	Print GS1 Databar Stacked	Not supported	Not supported
printGS1DataBarStackedOmnidirectional	Print GS1 Databar Stacked Omni-directional	Not supported	Not supported
printGS1DataBarExpandedStacked	Print GS1 Databar Expanded Stacked	Not supported	Not supported
printAztecCode	Print Aztec Code	Not supported	Not supported
cutPaper	Cut paper	Not supported	Supported
feedPosition	Paper form feed	Not supported	Not supported
sendBinary	Send binary data	Supported	Supported
sendDataFile	Send specified file	Supported*1	Supported*1
sendImage	Send image	Supported	Supported
printPDF	Print PDF page	Not supported	Supported
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 the following table.
See "4.4.1(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
printPageModeGS1DataBarStacked	Print GS1 Databar Stacked of page mode	Not supported	Not supported
printPageModeGS1DataBarStackedOmnidirectional	Print GS1 Databar Stacked Omni-directional of page mode	Not supported	Not supported
printPageModeGS1DataBarExpandedStacked	Print GS1 Databar Expanded Stacked of page mode	Not supported	Not supported
printPageModeAztecCode	Print Aztec Code of page mode	Not supported	Not 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) Printer Model Constant

Constants used for obtaining printer model are shown in the following table.

Table 4-4 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

(b) Response Type Constant

Constants used for obtaining various responses from a printer are shown in the following table.

Table 4-5 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
PRINTER_RESPONSE_ MAINTENANCE_COUNTER	Send maintenance counter	7	Not supported	Supported
PRINTER_RESPONSE_ PRINTER_IDENTIFIER	Send printer ID	8	Not supported	Supported

(c) International Character Set Constant

Constants used for setting/obtaining international character set are shown in the following table.

Table 4-6 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

(d) Codepage Constant

Constants used for setting/ obtaining codepage are shown in the following table.

Table 4-7 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
CODE_PAGE_1250	Central European (Code Page1250)	45	Not supported	Supported

Constant Name	Description	Value	Target	
			Mobile	POS
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.

(e) Port Type Constant

Constants used for obtaining connection port type are shown in the following table.

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) Constant for barcode or PDF417

Constants used for printing barcode or PDF417 are shown in the following table.

Table 4-9 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

(3) Constant List of Enumerated Type

(a) Drawer number (**DrawerNum**)

Enumerated type constants used for drawer number are shown in the following table.

Table 4-10 Drawer Number (DrawerNum)

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

(b) Activation pulse width (**PulseWidth**)

Enumerated type constants used for activation pulse width are shown in the following table.

Table 4-11 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

(c) Dithering (**Dithering**)

Enumerated type constants used for dithering are shown in the following table.

Table 4-12 Dithering (Dithering)

Constant Name	Description
DITHERING_DISABLE	Dithering is disabled
DITHERING_ERRORDIFFUSION	Dithering is enable

(d) Batch processing selection (**TransactionFunction**)

Enumerated type constants used for batch processing selection are shown in the following table.

Table 4-13 Batch Processing Selection (TransactionFunction)

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

(e) Bold print (**CharacterBold**)

Enumerated type constants used for bold print are shown in the following table.

Table 4-14 Bold Print (CharacterBold)

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

(f) Underline (**CharacterUnderline**)

Enumerated type constants used for underline are shown in the following table.

Table 4-15 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

(g) Reverse print (**CharacterReverse**)

Enumerated type constants used for reverse print are shown in the following table.

Table 4-16 Reverse Print (CharacterReverse)

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

(h) Inversion print (**CharacterInversion**)

Enumerated type constants used for inversion print are shown in the following table.
Inversion print cannot be added to the text data before inserting a new line feed.

Table 4-17 Inversion Print (CharacterInversion)

Constant Name	Description	Target	
		Mobile	POS
INVERSION_CANCEL	Release inversion print	Not supported	Supported
INVERSION	Specify inversion print	Not supported	Supported

(i) Character font (**CharacterFont**)

Enumerated type constants used for Character font are shown in the following table.

Table 4-18 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

(j) Character Scale (**CharacterScale**)

Enumerated type constants used for character scale are shown in the following table.

Table 4-19 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

Constant Name	Description	Target	
		Mobile	Mobile
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

(k) Alignment (**PrintAlignment**)

Enumerated type constants used for alignment are shown in the following table.
Alignment cannot be added to the text data before inserting a new line feed.

Table 4-20 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

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

Enumerated type constants used for pending data output specifying are shown in the following table.

Table 4-21 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

(m) Barcode symbol (**BarcodeSymbol**)

Enumerated type constants used for barcode symbol are shown in the following table.

Table 4-22 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.

(n) Module size (**ModuleSize**)

Enumerated type constants used for module size are shown in the following table.

Table 4-23 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

Constant Name	Description	Using Method	Target	
			Mobile*1	POS
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
DATAMATRIX_MODULE_SIZE_4	4 dots		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.

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

Enumerated type constants used for HRI character print position are shown in the following table.

Table 4-24 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

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

Enumerated type constants used for N:W ratio are shown in the following table.

Table 4-25 N:W Ratio (NwRatio)

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

(q) Error correction level (**ErrorCorrection**)

Enumerated type constants used for error correction level are shown in the following table.

Table 4-26 Error Correction Level (ErrorCorrection)

Constant Name	Description	Using Method	Target	
			Mobile*1	POS
PDF417_ERROR_ CORRECTION_0	Error correction level 0	**printPDF417 printPageMo dePDF417**	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 printPageMo deQRcode**	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.

(r) PDF417 symbol (**Pdf417Symbol**)

Enumerated type constants used for PDF417 symbol are shown in the following table.

Table 4-27 PDF417 Symbol (Pdf417Symbol)

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

(s) QR code model (**QrModel**)

Enumerated type constants used for QR code model symbol are shown in the following table.

Table 4-28 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

(t) Data Matrix modules (**DataMatrixModule**)

Enumerated type constants used for Data Matrix module are shown in the following table.

Table 4-29 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

Constant Name	Description
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

(u) MaxiCode Mode (**MaxiCodeMode**)

Enumerated type constants used for MaxiCode mode are shown in the following table.

Table 4-30 MaxiCode Mode (MaxiCodeMode)

Constant Name	Description
MAXI_CODE_2	Mode2
MAXI_CODE_3	Mode3
MAXI_CODE_4	Mode4
MAXI_CODE_5	Mode5

(v) Cutting method (**CuttingMethod**)

Enumerated type constants used for cutting method are shown in the following table.

Table 4-31 Cutting Method (CuttingMethod)

Constant Name	Description		Target	
	Paper Feed to Cut Position	Cutting Method	Mobile	POS
CUT_FULL	Enabled	Full cut	Not supported	Supported
CUT_FULL_NO_FEED	Disabled		Not supported	Supported
CUT_PARTIAL	Enabled	Partial cut	Not supported	Supported
CUT_PARTIAL_NO_FEED	Disabled		Not supported	Supported
CUT_NONE[*1]	Disabled	No cut	Not supported	Supported

*1: Supported only by **printPageMode**.

(w) Image rotation direction (**Rotate**)

Constants of enumerated type used for image rotation direction are shown in the following table.

Table 4-32 Image Rotation Direction (Rotate)

Constant Name	Description	Target	
		Mobile	POS
ROTATE_NONE	No rotation	Not supported	Supported
ROTATE_180	Rotate 180 degrees	Not supported	Supported

(x) Image scaling (**ImageScale**)

Constants of enumerated type used for image scaling are shown in the following table.

Table 4-33 Image Scaling (ImageScale)

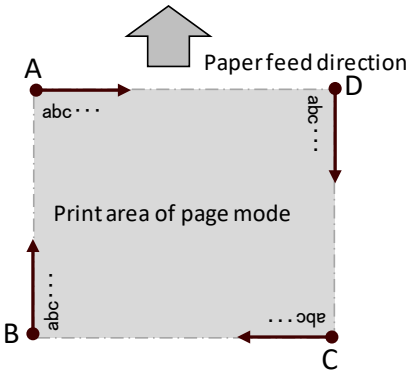
Constant Name	Description	Target	
		Mobile	POS
IMAGE_SCALE_NONE	No scaling	Not supported	Supported
IMAGE_SCALE_WIDTH_FIT	Scale to fit print width	Not supported	Supported

(y) Print direction (**Direction**)

Constants of enumerated type used for print direction in page mode are shown in the following table.

Table 4-34 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



(z) Line style (**LineStyle**)

Constants of enumerated type used for line style in page mode are shown in the following table.

Table 4-35 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 *context*)

(b) public **PrinterManager**()

Parameter:

context	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. Also, syntax (b) does not support the log function.

connect — Start communicating with a printer (Bluetooth)

Target: Mobile printer / POS printer

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

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

Parameter:

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

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

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

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

Monitoring of the printer status is started with this method. The latest printer status can be retrieved from **getStatus** method.
Changes of the printer status can be notified as events by **onStatusChanged** method and **setCallbackFunctionListener** method.

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 *printerModel*) throws **PrinterException**

(b) public void **connect**(int *printerModel*, Context *context*) throws **PrinterException**

Parameter:

printerModel	Printer model constant for USB connection
context	Specify application context to call this method. Example: `MainActivity.this`

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

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

Monitoring of the printer status is started with this method. The latest printer status can be retrieved from **getStatus** method.
Changes of the printer status can be notified as events by **onStatusChanged** method and **setCallbackFunctionListener** method.

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 *printerModel*, String *address*) throws **PrinterException**

Parameter:

printerModel	Printer model constant for Ethernet connection
address	IP address Example: "192.168.0.190"

Description: This method is valid only for POS printer. This method starts communication between Android device and a printer connected to the same network through TCP/IP connection. Call this method before using other methods of this class.

This method connects to the IP address specified by *address* parameter. TCP port number 9100 and 26100 are used for communication. Also, printer default settings are executed based on *printerModel* parameter specified at the connection. See Table 4-4 Printer Model Constant for available printer model constant.

Monitoring of the printer status is started with this method. The latest printer status can be retrieved from **getStatus** method.
Changes of the printer status can be notified as events by **onStatusChanged** method and **setCallbackFunctionListener** method.

<Creating/discarding of socket in TCP/IP connection of the library>

After **connect** method, the library retains the created socket until **disconnect** method. And connecting to the same printer from other applications is not possible until **disconnect** method.

Based on the completion of data transmission to the printer, the socket is once discarded after elapsing the socket keeping time set by **setSocketKeepingTime** 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.

setBarcodeScannerListener — Start/End callback of barcode scanner

This method is not supported. When executing this method, **PrinterException** is thrown.

Target Mobile printer / POS printer

Syntax public void **setBarcodeScannerListener**(BarcodeScannerListener *listener*) throws **PrinterException**

openDrawer — Open cash drawer

Target POS printer

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

Parameter

drawerNum	Drawer number
onOffTime	pulse width

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

See Table 4-10 Drawer Number (DrawerNum) for available setting in *drawerNum* parameter.

See Table 4-11 Activation Pulse Width (PulseWidth) for available setting in *onOffTime* parameter.

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

buzzer — Sound buzzer

Target POS printer

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

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

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

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

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

externalBuzzer — Sound external buzzer

This method is not supported. When executing this method, **PrinterException** is thrown.

Target Mobile printer / POS printer

Syntax public void **externalbuzzer**(BuzzerPattern *buzzer pattern*, int *buzzerCount*) throws **PrinterException**

getStatus — Obtain printer status

Target Mobile printer / POS printer

Syntax (a) public int **getStatus**() throws **PrinterException**

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

Parameter
buf	Status obtained from a printer

Description This method obtains printer status. The method of syntax (a) returns status retrieved from the printer with return value. The method of syntax (b) stores status retrieved from the printer in an array of int type.
Contents of status vary between Mobile printer and POS printer.
When *printerModel* parameter specified while executing **connect** method is **PRINTER_MODEL_DPU_S245** or **PRINTER_MODEL_DPU_S445**, see contents of the Mobile printer status.
When *printerModel* parameter is **PRINTER_MODEL_RP-D10** or **PRINTER_MODEL_RP-E10**, see the POS printer status.

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

Table 4-36 Printer Status (Mobile Printer)

Bit	Function	Value	
		0	1
0	Out-of-paper error	OK	Error
1	Head up error	OK	Error
2	V_P 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 the following table.
When the connection failed, the POS printer status is shown in 0x80000000.

Table 4-37 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 **CallbackFunctionLister** 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.

registerLogo — Register logo (image) to a printer

Target: Mobile printer

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

(b) public void **registerLogo**(String *fileName*, int *id,* Dithering *dithering*) throws **PrinterException**

Parameter

fileName: File path of image file to register as logo
The formats that can be entered are described below.

- Absolute path string handled by Java standard class "java.io.File"
  When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.
  See "3.5 Precautions - About Scoped Storage" for details.
- URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android
  - file://
  - content://

  It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.

id: Logo ID to register
dithering: Dithering

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

In syntax (a), dithering is fixed to be disabled.
In syntax (b), dithering can be specified.

File extension for image file supported by *fileName* parameter is .bmp, .jpg, or .jpeg.
The valid range is a character string indicating the value from 0 to 127.

See Table 4-12 Dithering (Dithering) for details about available constants for *dithering* parameter.

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

registerLogo — Register logo (image) to a printer

Target: POS printer

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

(b) public void **registerLogo**(String *fileName*,
String *id,*
Dithering *dithering*) throws **PrinterException**

Parameter *fileName* File path of image file to register as logo
The formats that can be entered are described below.

- Absolute path string handled by Java standard class "java.io.File"
  When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.
  See "3.5 Precautions - About Scoped Storage" for details.

- URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android
  - file://
  - content://

  It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.

id Logo ID to register
dithering Dithering

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

In syntax (a), dithering is fixed to be disabled.
In syntax (b), dithering can be specified.

File extension for image file supported by *fileName* parameter is .bmp, .jpg, or .jpeg.
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').

See Table 4-12 Dithering (Dithering) for details about available constants for *dithering* parameter.

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

unregisterLogo — Delete specified logo (image) in a printer

Target 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
  The formats that can be entered are described below.
  - Absolute path string handled by Java standard class "java.io.File"
    When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.
    See "3.5 Precautions - About Scoped Storage" for details.
  - URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android
    - file://
    - content://

    It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.
- *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 4-5 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 the following table.

Table 4-38 Contents of Obtained Response (Mobile Printer)

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

Battery Status Value	Battery Voltage State
0	8.0 V or higher
1	7.5 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 the following table.

Table 4-39 Contents of Obtained Response (POS Printer)

Constant Name	Description
PRINTER_RESPONSE_ REQUEST	Obtains "Execution response request". For *buf* parameter, specify integer array whose length is 1. Specify 0 to 15 (00h to 0Fh) for *buf[0]*. When responses are obtained successfully, 128 to 143 (80h to 8Fh) are stored to *buf[0]*.
PRINTER_RESPONSE_ USER_AREA	Obtains "Send remaining capacity of user area". For *buf* parameter, specify integer array whose length is 1. When responses are obtained successfully, remaining user area is stored to *buf[0]* in byte value.
PRINTER_RESPONSE_ ARRANGE_USER_AREA	Obtains "Send remaining capacity of user area after defragment". For *buf* parameter, specify integer array whose length is 1. When responses are obtained successfully, remaining memory after user area defragment is stored to *buf[0]* in byte value.
PRINTER_RESPONSE_ NV_GRAPHICS	Obtains "Send NV graphics memory capacity". For *buf* parameter, specify integer array whose length is 1. When responses are obtained successfully, NV graphics memory capacity is stored to *buf[0]* in byte value.
PRINTER_RESPONSE_ KEY_CODE	Obtains "Send key code list of defined NV graphics". For *buf* parameter, specify ArrayList<String> type array. When responses are obtained successfully, the key code of NV graphics is stored to *buf* parameter in string array. Example: *buf.size()* = 3, *buf[0]* = "22", *buf[1]* = "23", *buf[2]* = "24" etc.
PRINTER_RESPONSE_ MAINTENANCE_COUNTER	Obtains "Send maintenance counter". Specify an int array of length 8. When the response is retrieved successfully, the maintenance counter is stored as shown below. *buf[0]* : Number of paper feed lines (unit: 100 dot-line) *buf[1]* : Number of thermal head activation times (unit: 100 dot-line) *buf[2]* : Number of autocutter drive times *buf[3]* : Drive time of printer (unit: minute) *buf[4]* : Number of paper feed lines (integrated value) (unit: 100 dot-line) *buf[5]* : Number of thermal head activation times (integrated value) (unit: 100 dot-line) *buf[6]* : Number of autocutter drive times (integrated value) *buf[7]* : Drive time of printer (integrated value) (unit: minute)
PRINTER_RESPONSE_ PRINTER_IDENTIFIER	Obtains "Send printer ID". Specify an ArrayList<PrinterIdentifier> array. When the response is retrieved successfully, a **PrinterIdentifier** class object is stored in the first element of *buf*. Example: *buf.size()* = 1, *buf.get[0]* = "**PrinterIdentifier** class object" See "**4.4.5 PrinterIdentifier Class**" for the **PrinterIdentifier** class object.

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

startDiscoveryPrinter Start printer search (Bluetooth)

Target Mobile printer / POS printer

Syntax (a) public void **startDiscoveryPrinter**(PrinterListener *listener*) throws **PrinterException**

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

Parameter

listener	PrinterListener Instance described in the following
context	Context of application

Description This method searches the printer connected to Bluetooth. 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 **cancelDiscoveryPrinter** 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 *listener* parameter of **startDiscoveryPrinter** method.

getFoundPrinter — Obtain searched printer information

Target: Mobile printer / POS printer

Syntax: public ArrayList<**PrinterInfo**> **getFoundPrinter**()

Description: This method obtains the printer information searched by the printer search in ArrayList of **PrinterInfo** class.

Return value: ArrayList of **PrinterInfo** class.

getSendTimeout — Obtain send timeout period

Target: Mobile printer / POS printer

Syntax: public int **getSendTimeout**()

Description: This method obtains the timeout period when data is sent. This method is obtainable whether a printer is connected or not. Obtained timeout period is expressed in msec (millisecond) value.

Return value: Send timeout period in msec (millisecond)

setSendTimeout — Set send timeout period

Target: Mobile printer / POS printer

Syntax: public void **setSendTimeout**(int *sendTimeout*)

Parameter: *sendTimeout* Send timeout period

Description: This method sets the timeout period when data is sent in msec (millisecond). This method is configurable whether a printer is connected or not. However, it is not until sending data next time that the configured timeout period is enabled. Moreover, the default is set when 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 *receiveTimeout*)

Parameter: *receiveTimeout* Receive timeout period

Description: This method sets the timeout period when data is received in msec (millisecond). This method is configurable whether a printer is connected or not. However, it is not until receiving data next time that the configured timeout period is enabled. Moreover, the default is set when 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 4-6 International Character Set Constant for details.

setInternationalCharacter Set international character set

Target Mobile printer / POS printer

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

Parameter *internationalCharacter* International character set constant

Description This method sets international character set. See Table 4-6 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 *internationalCharacter* parameter is set, the following values are used as well.

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

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

getCodePage Obtain codepage

Target Mobile printer / POS printer

Syntax public int **getCodePage**()

Description This method obtains setting values of codepage. The encoder used 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.

Return value See Table 4-7 Codepage Constant for details.

setCodePage Set codepage

Target Mobile printer / POS printer

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

Parameter *codePage* Codepage constant

Description This method sets codepage. See Table 4-7 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 *codePage* parameter is set, it will be ignored.

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

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

getPrinterModel — Obtain printer model

Target: Mobile printer / POS printer

Syntax: public int **getPrinterModel**()

Description: This method obtains a model value for the connected printer. When a printer is not connected, the default returns. Even though 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 4-4 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.

true	Connected to a printer
false	Disconnected to a printer

getSocketKeepingTime — Obtain socket keeping time

Target: POS printer

Syntax: public int **getSocketKeepingTime**()

Description: This method is valid only for POS printer. This method obtains socket keeping time during TCP/IP connection. This method is obtainable whether a printer is connected or not. Obtained time is expressed in msec (millisecond) value.

Return value: Socket keeping time in msec (millisecond)

setSocketKeepingTime — Set socket keeping time

Target: POS printer

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

Parameter: *socketKeepingTime* — Socket keeping time

Description: This method is valid only for POS printer. This method obtains socket keeping time during TCP/IP connection. Obtained timeout period is expressed in msec (millisecond) value. For socket keeping time, set the same time as Network Printer Receive Timeout of the printer to be connected. The setting of Network Printer Receive Timeout can be changed in "SII Printer Utility" with the Android app on Google Play.
This method is configurable whether a printer is connected or not. However, it is not until executing **connect** method (TCP/IP) next time that the configured socket keeping time is enabled. Moreover, the default is set when you configure values outside the scope of the valid range.

Default: 300000msec (5 minutes)

Effective range: 60000 to 300000msec (5 minutes)

getVersion — Get SDK version

Target: Mobile printer / POS printer

Syntax: public String **getVersion**()

Return value: SDK version character string (Example: When the SDK version is Ver.1.0.0, the return value is "1.0.0")

Description: This method is configurable whether a printer is connected or not.

controlTransaction — Start/End batch processing

Target: Mobile printer / POS printer

Syntax: public void **controlTransaction**(TransactionFunction *transactionFunction*) throws **PrinterException**

Parameter: *transactionFunction* — Batch processing selection

Description: See Table 4-13 Batch Processing Selection (TransactionFunction) for available setting in *transactionFunction* 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**
- **sendImage**
- **printPDF**[*1]
- **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 *text*) throws **PrinterException**

Parameter:

text		Text data sent to a printer

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

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.

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

sendTextEx — Send format specified text data

Target: Mobile printer

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

Parameter:

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

Description: This method is valid only for Mobile printer. This method encodes 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 16 KB (16384 bytes).

See Table 4-14 Bold Print (CharacterBold) for available setting in *bold* parameter.

See Table 4-15 Underline (CharacterUnderline) for available setting in *underline* parameter.

See Table 4-18 Character Font (CharacterFont) for available setting in *font* parameter.

See Table 4-19 Character Scale (CharacterScale) for available setting in *scale* parameter.

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 *text*,
CharacterBold *bold*,
CharacterUnderline *underline*,
CharacterReverse *reverse*,
CharacterFont *font*,
CharacterScale *scale*,
PrintAlignment *alignment*) throws **PrinterException**

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

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

Parameter

text	Text data to send to a printer
bold	Bold print
underline	Underline
reverse	Reverse print
inversion	Inversion print
font	Character font
scale	Character scale
alignment	Alignment
output	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 16 KB (16384 bytes).

See Table 4-14 Bold Print (CharacterBold) for available setting in *bold* parameter.

See Table 4-15 Underline (CharacterUnderline) for available setting in *underline* parameter.

See Table 4-16 Reverse Print (CharacterReverse) for available setting in *reverse* parameter.

See Table 4-17 Inversion Print (CharacterInversion) for available setting in *inversion* parameter.

See Table 4-18 Character Font (CharacterFont) for available setting in *font* parameter.

See Table 4-19 Character Scale (CharacterScale) for available setting in *scale* parameter.

See Table 4-20 Alignment (PrintAlignment) for available setting in *alignment* parameter.

See Table 4-21 Pending Data Output Specifying (OutputPendingData) for available setting in *output* 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) or (c) 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.

When **PENDING_DATA_OUTPUT_TOGETHER** is specified at *output* in the method of syntax (b), 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.

printBarcode — Print barcode

Target Mobile printer / POS printer
The syntax (d) is not supported.

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

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

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

(d) public void **printBarcode**(BarcodeSymbol *barcodeSymbol*,
String *text*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

Parameter

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

Description

This method executes barcode print.

See Table 4-22 Barcode Symbol (BarcodeSymbol) for available setting in *barcodeSymbol* parameter.

The input conditions for barcode that can be used with the parameter *text(data)* are as follows. See Table 4-40 Barcode Input Conditions for Mobile Printer for Mobile printer, and Table 4-41 Barcode Input Conditions for POS Printer for POS printer.

Table 4-40 Barcode Input Conditions for Mobile Printer

Barcode	Number of Data	Inputtable Data Character String (Data)	Remarks
UPC-A	11 characters	'0' to '9'	
UPC-E	11 characters	'0' to '9'	
EAN13 JAN13	12 characters	'0' to '9'	
EAN8 JAN8	7 characters	'0' to '9'	
CODE39	1 to 254 characters	'0' to '9' 'A' to 'Z' ' ', '$', '%', '+', '-', '.', '/'	Start code and stop code ('*') are automatically added.
CODE128	2 to 255 bytes	(0x00 to 0x66)	Input data with 0x67 or more at the end.
ITF	2 to 254 characters (However, an even number)	'0' to '9'	
CODABAR	1 to 254 characters	'0' to '9' '$', '+', '-', '.', '/', ':'	It is needed to specify one of 'A' to 'D' at the beginning and end.

Table 4-41 Barcode Input Conditions for POS Printer

Barcode	Number of Data	Inputtable Data Character String (Data)	Remarks
UPC-A	11 to 12 characters	'0' to '9'	
UPC-E	11 to 12 characters	'0' to '9'	
EAN13 JAN13	12 to 13 characters	'0' to '9'	

Barcode	Number of Data	Inputtable Data Character String (Data)	Remarks
EAN8 JAN8	7 to 8 characters	'0' to '9'	
CODE39	1 to 150 characters	'0' to '9' 'A' to 'Z' ' ', '$', '%', '+', '-', '.', '/'	Start code and stop code ('*') are automatically added.
CODE93	1 to 150 bytes	(0x00 to 0x2E)	Input data with 0x2F or more at the end.
CODE128	2 to 150 bytes	(0x00 to 0x66)	When inputting the start code (0x67 to 0x69) of the CODE128 code set. Input data with 0x67 or more at the end.
		(0x00 to 0x7F)	When starting with a CODE128 special code start code ("{A", "{B", "{C").
ITF	2 to 150 characters (However, an even number)	'0' to '9'	
CODABAR	1 to 150 characters	'0' to '9' '$', '+', '-', '.', '/', ':'	It is needed to specify one of 'A' to 'D' at the beginning and end.
EAN13 add-on JAN13 add-on	Add-on 2: 14 to 15 characters Add-on 5: 17 to 18 characters	'0' to '9'	

See Table 4-23 Module Size (ModuleSize) for available setting in *moduleSize* parameter.

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

See Table 4-24 HRI Character Print Position (HriPosition) for available setting in *hriPosition* parameter.

See Table 4-18 Character Font (CharacterFont) for available setting in *hriFont* parameter.

See Table 4-20 Alignment (PrintAlignment) for available setting in *alignment* parameter.

See Table 4-25 N:W Ratio (NwRatio)Table 4-25 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 4-42 N:W Ratio for Mobile Printer for Mobile printer, and Table 4-43 N:W Ratio for POS Printer for POS printer.

Table 4-42 N:W Ratio for Mobile Printer

moduleSize	*nwRatio*			
	WIDE_WIDTH_1	**WIDE_WIDTH_2**	**WIDE_WIDTH_3**	**WIDE_WIDTH_4**
BARCODE_MODULE_WIDTH_2	0.625 mm (5 dots)	0.750 mm (6 dots)	0.750 mm (6 dots)	0.750 mm (6 dots)
BARCODE_MODULE_WIDTH_3	0.875 mm (7 dots)	1.000 mm (8 dots)	1.125 mm (9 dots)	1.125 mm (9 dots)

moduleSize	*nwRatio*			
	WIDE_WIDTH_1	**WIDE_WIDTH_2**	**WIDE_WIDTH_3**	**WIDE_WIDTH_4**
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 4-43 N:W Ratio for POS Printer

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

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

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printPDF417 Print PDF417

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

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

See Table 4-26 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 4-23 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.

SeeTable 4-20 Alignment (PrintAlignment)Table 4-20 Alignment (PrintAlignment) for available setting in *alignment* parameter.

See Table 4-27 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.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printQRcode — Print QR code

Target Mobile printer / POS printer

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

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

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

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

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

See Table 4-26 Error Correction Level (ErrorCorrection) for available setting in *errorCorrection* parameter.

See Table 4-23 Module Size (ModuleSize) for available setting in *moduleSize* parameter.

See Table 4-20 Alignment (PrintAlignment) for available setting in *alignment* parameter.

See Table 4-28 QR Code Model (QrModel) for available setting in *model* parameter.

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

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printDataMatrix Print Data Matrix

Target Mobile printer / POS printer

Syntax public void **printDataMatrix**(String *text*,
DataMatrixModule *dataMatrixModule*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

Parameter

text	Barcode data to send to the printer
dataMatrixModule	The number of Data Matrix modules
moduleSize	Module size
alignment	Alignment

Description This method prints Data Matrix.

See Table 4-29 Data Matrix modules (DataMatrixModule) for available setting in *dataMatrixModule* parameter.

See Table 4-23 Module Size (ModuleSize) for available setting in *moduleSize* parameter.

See Table 4-20 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.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printMaxiCode Print MaxiCode

Target Mobile printer / POS printer

Syntax public void **printMaxiCode**(String *text*,
MaxiCodeMode *maxiCodeMode*,
PrintAlignment *alignment*) throws **PrinterException**

Parameter

text	Barcode data to send to the printer

- When *maxiCodeMode* 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 *maxiCodeMode* 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.

maxiCodeMode MaxiCode mode

alignment Alignment

Description This method prints MaxiCode.

See Table 4-30 MaxiCode Mode (MaxiCodeMode) for available setting in *maxiCodeMode* parameter.

See Table 4-20 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.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printGS1DataBarStacked Print GS1 Databar Stacked

This method is not supported. When executing this method, **PrinterException** is thrown.

Target Mobile printer / POS printer

Syntax public void **printGS1DataBarStacked**(String *text*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

printGS1DataBarStackedOmnidirectional Print GS1 Databar Stacked Omni-directional

This method is not supported. When executing this method, **PrinterException** is thrown.

Target Mobile printer / POS printer

Syntax public void **printGS1DataBarStackedOmnidirectional**(String *text*,
int *moduleHeight*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

printGS1DataBarExpandedStacked Print GS1 Databar Expanded Stacked

This method is not supported. When executing this method, **PrinterException** is thrown.

Target Mobile printer / POS printer

Syntax public void **printGS1DataBarExpandedStacked**(String *text*,
int *column*,
ModuleSize *moduleSize*,
PrintAlignment *alignment*) throws **PrinterException**

printAztecCode — Print Aztec Code

This method is not supported. When executing this method, **PrinterException** is thrown.

Target Mobile printer / POS printer

Syntax public void **printAztecCode**(String *text*,
int *layer*,
int *errorCorrection*,
ModuleSize *moduleSize*,
AztecSymbol *aztecSymbol*,
PrintAlignment *alignment*) throws **PrinterException**

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. Selects enabled/disabled of the paper feed to the cut position and cuts the paper.

See Table 4-31 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.

feedPosition — Paper form feed

This method is not supported. When executing this method, **PrinterException** is thrown.

Target Mobile printer / POS printer

Syntax public void **feedPosition**(FeedPosition *feedPosition*) throws **PrinterException**

sendBinary — Send binary data

Target Mobile printer / POS printer

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

Parameter *binary* Binary data sent to a printer

Description This method sends the binary data specified by *binary* parameter to a printer. Data size that can be specified at 1 time is 3 MB (3145728 bytes).

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

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

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

sendDataFile — Send specified file

Target Mobile printer / POS printer

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

(b) public void **sendDataFile**(String *fileName*, Dithering *dithering*) throws **PrinterException**

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

(d) public void **sendDataFile**(String *fileName*,
PrintAlignment *alignment,*
Dithering *dithering*) throws **PrinterException**

Parameter

fileName File path of the data to sent to a printer
The formats that can be entered are described below.

- Absolute path string handled by Java standard class "java.io.File"
  When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.
  See "3.5 Precautions - About Scoped Storage" for details.
- URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android
  - file://
  - content://

  It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.

alignment Alignment

dithering Dithering

Description This method determines data format based on the file extension specified by *fileName* parameter, convert it to printer-enabled data format, and send it to a printer. The maximum file size that can be specified is 16 MB (16777216 bytes).

In syntax (a), the sending file can be specified.
In syntax (b), the sending file and dithering can be specified.
The syntax (c) is only available on the POS printer, alignment of the sending file can be specified.
The syntax (d) is only available on the POS printer, alignment and dithering of the sending file can be specified.

The file extensions capable of sending by *fileName* parameter and sending the file are describes below.

- .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.

alignment parameter is valid only when the file extension specified on *fileName* parameter is .bmp, .jpg, .jpeg, .png or .txt. See Table 4-20 Alignment (PrintAlignment)Table 4-20 Alignment (PrintAlignment) for available setting.

dithering parameter is valid only when the file extension specified on *fileName* parameter is .bmp, .jpg, .jpeg, or .png. See Table 4-12 Dithering (Dithering) for available setting.

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.

sendImage Send image

Target Mobile printer / POS printer

Syntax (a) public void **sendImage**(Bitmap *bitmap*,
PrintAlignment *alignment,*
Dithering *dithering*) throws **PrinterException**

(b) public void **sendImage**(Bitmap *bitmap*,
PrintAlignment *alignment*) throws **PrinterException**

(c) public void **sendImage**(Bitmap *bitmap*) throws **PrinterException**

Parameter

bitmap	"android.graphics.Bitmap" type object to send to the printer
alignment	Alignment
dithering	Dithering

Description This method prints an image of the specified “android.graphics.Bitmap” type.

In syntax (a), for mobile printer, image alignment is fixed to left alignment, and dithering can be specified. For POS printer, image alignment and dithering can be specified.
In syntax (b), for mobile printer, image alignment is fixed to left alignment, and dithering is fixed to disable. For POS printer, image alignment can be specified, and dithering is fixed to disabled.
In syntax (c), image alignment is fixed to left alignment, and dithering is fixed to disabled.

Send the image to the printer as an image file in *bitmap* parameter.
When that image file is colored one, it is converted to monochrome image by binarization, and sent to a printer. The image file is expanded onto the printer’s memory and then printed in a batch.

See Table 4-20 Alignment (PrintAlignment) for available setting in *alignment* parameter.

See Table 4-12 Dithering (Dithering) for available setting in *dithering* parameter.

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

printPDF Print PDF page

Target POS printer

Syntax public void **printPDF**(String *fileName*,
int *startIndex*,
int *endIndex*,
Rotate *rotate*,
ImageScale *imageScale*,
int *bottomMargin*,
Dithering *dithering*,
PrintAlignment *alignment*) throws **PrinterException**

Parameter

fileName File path of the PDF file
The formats that can be entered are described below.

- Absolute path string handled by Java standard class "java.io.File"
  When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.
  See "3.5 Precautions - About Scoped Storage" for details.

- URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android
  - file://
  - content://

  It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.

startIndex Start number of the printed page

endIndex End number of the printed page

rotate Rotation direction of the image

imageScale Image scaling

bottomMargin Bottom margin (dot)

alignment Alignment

dithering Dithering

Description Prints the specified page of the selected PDF file.
PDF is converted to printable format for the printer and is sent to the printer.

The file extension for supporting PDF is .pdf.
The maximum file size that can be specified is 16 MB (16777216 bytes).
The color image data is converted to monochrome image by binarization.

The valid range of *startIndex* parameter is -1, 1 to the number of pages in the PDF file.
When -1 is specified for *startIndex* parameter, all pages are printed.
When -1 is specified for *startIndex* parameter, the value of *endIndex* parameter is ignored.
When 1 or more is specified for *startIndex* parameter, pages from the page number specified in *startIndex* parameter to the page number specified in *endIndex* parameter are printed.
When a value more than the value specified for *endIndex* parameter is specified for *startIndex* parameter, an error occurs.
When an out-of-range value is specified for *startIndex* parameter or *endIndex* parameter, an error occurs.

The valid range of *endIndex* parameter is 1 to 2147483647.
When a value more than the number of pages in the PDF file is specified for *endIndex* parameter, pages from the page number specified in *startIndex* parameter to the last page of the PDF file are printed.

See Table 4-32 Image Rotation Direction (Rotate) for available setting in *rotate* parameter.

See Table 4-33 Image Scaling (ImageScale) for available setting in *imageScale* parameter.
When **IMAGE_SCALE_WIDTH_FIT** is specified for imageScale, the image width is converted to the print width of the printer while maintaining the aspect ratio.

The valid range of *bottomMargin* parameter is -1, 0 to 2400.
When -1 is specified for *bottomMargin* parameter, the image is created and printed while maintaining the bottom margin.
When a value between 0 and 2400 is specified for *bottomMargin* parameter, the bottom margin is changed to the specified size.
When a value between 0 and 2400 is specified for *bottomMargin* parameter, blank pages are not printed.

See Table 4-12 Dithering (Dithering) for available setting in *dithering* parameter.

See Table 4-20 Alignment (PrintAlignment) for available setting in *alignment* parameter.

When the memory switches MS 1-3 (Mark Mode Selection) of RP-E10 is set to "Enable", marked paper form feed is performed after printing the PDF.

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.

Note No guarantee of printing operation when more than 100 pages are printed at a time.

printLogo — Print specified logo (image) in printer

Target Mobile printer

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

Parameter *id* Logo ID to print

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

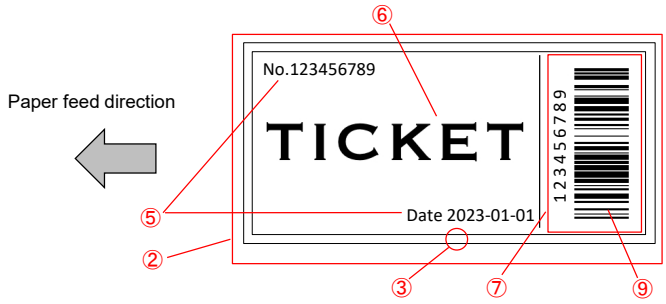
Error **PrinterException** may be thrown when this method is called. The printer may be disconnected when **PrinterException** is thrown during data sending.
See **isConnect** method for verifying connection state.

printLogo — Print specified logo (image) in printer

Target	POS printer
Syntax	public void **printLogo**(String *id*, PrintAlignment *alignment*) throws **PrinterException**
Parameter	*id* Logo ID to print *alignment* Alignment
Description	This method is valid only for POS printer. This method print the logo (image) registered by **registerLogo** method. Specify registered logo ID in *id* parameter. The valid range of *id* parameter is 2 characters. The valid characters are ASCII character code from 20h (space) to 7Eh (tilde) such as alphanumeric ('0' to '9', 'A' to 'Z', 'a' to 'z'). See Table 4-20 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.

(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, 355, 576);
```

③ Specify a rectangle and a ruled line

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

④ Specify print direction of page mode

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

⑤ Specify a character

```
mPrinterManager.printPageModeText(21, 47, "NO.123456789");
mPrinterManager.printPageModeText(212, 340, "Date 2023-01-01");
```

⑥ Specify an image file

```
mPrinterManager.printPageModeImageFile(
            10,
            222,
            Environment.getExternalStorageDirectory().getPath() + "/TicketImage.jpg",
            Dithering.DITHERING_DISABLE);
```

⑦ Specify print area of page mode

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

⑧ Specify print direction

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

⑨ 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);
```

⑩ Print in page mode

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

⑪ Ends page mode

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

enterPageMode — Start page mode

Target POS printer

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

Description 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.

Executing **exitPageMode** method discards the print data kept in the page data buffer and changes the mode to standard mode.
Executing **printPageMode** method prints the print data kept in the page data buffer.

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 *x,*
int *y,*
int *width,*
int *height*) throws **PrinterException**

Parameter

x The horizontal origin (dot) of the print area of page mode
0 represents the left edge on the print area of the printer.

y 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.

width The print area width (dot) of page mode

height The print area height (dot) of page mode
The valid range is 1 to (2400-*y*).

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

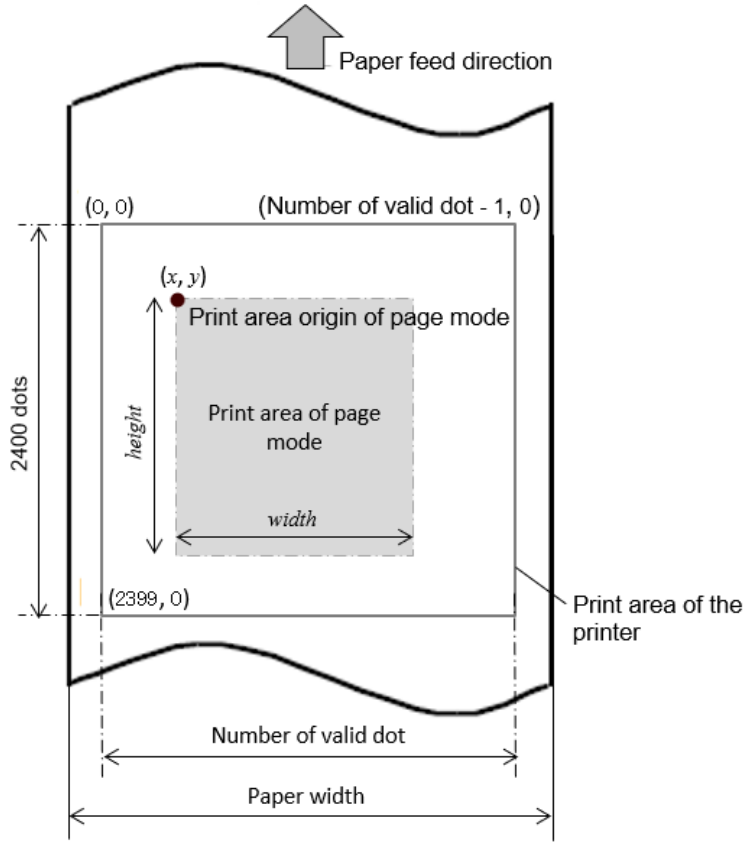


Table 4-44 Valid Range of *x* and *width* (POS printer)

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 *direction*) throws **PrinterException**

Parameter:

direction	Print direction See Table 4-34 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 *lineSpacing*) throws **PrinterException**

Parameter:

linespacing	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 *cuttingMethod*) throws **PrinterException**

Parameter:

cuttingMethod	Cutting method See Table 4-31 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:

- 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 *startX,* int *startY,* String *text*) throws **PrinterException**

Parameter

startX	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
startY	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
text	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 *startX,*
int *startY,*
String *text,*
CharacterBold *bold,*
CharacterUnderline *underline,*
CharacterReverse *reverse,*
CharacterFont *font,*
CharacterScale *scale*) throws **PrinterException**

Parameter

startX	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
startY	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
text	Text data Data size that can be specified at 1 time is 16 KB (16384 bytes).
bold	Bold print See Table 4-14 Bold Print (CharacterBold) for available constants.
underline	Underline See Table 4-15 Underline (CharacterUnderline) for available constants.
reverse	Reverse print See Table 4-16 Reverse Print (CharacterReverse) for available constants.
font	Font See Table 4-18 Character Font (CharacterFont) for available constants.

scale Character scale
See Table 4-19 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 (a) public void **printPageModeBarcode**(int *startX,*
int *startY,*
BarcodeSymbol *barcodeSymbol,*
String *text,*
ModuleSize *moduleSize,*
int *moduleHeight,*
HriPosition *hriPosition,*
CharacterFont *hriFont*) throws **PrinterException**

(b) public void **printPageModeBarcode**(int *startX,*
int *startY,*
BarcodeSymbol *barcodeSymbol,*
String *text,*
ModuleSize *moduleSize,*
int *moduleHeight,*
HriPosition *hriPosition,*
CharacterFont *hriFont,*
NwRatio *nwRatio*) throws **PrinterException**

(c) public void **printPageModeBarcodeint** *startX,*
int *startY,*
BarcodeSymbol *barcodeSymbol,*
byte[] *data,*
ModuleSize *moduleSize,*
int *moduleHeight,*
HriPosition *hriPosition,*
CharacterFont *hriFont*) throws **PrinterException**

(d) public void **printPageModeBarcode**(int *startX,*
int *startY,*
BarcodeSymbol *barcodeSymbol,*
String *text,*
ModuleSize *moduleSize*) throws **PrinterException**

Parameter *startX* The horizontal reference point (dot) from the starting point
The valid range is 0 to 2399.

startY The vertical reference point (dot) from the starting point
The valid range is 0 to 2399.

barcodeSymbol BarcodeSymbol
See Table 4-22 Barcode Symbol (BarcodeSymbol) for available constants and correspondent syntax.

text(data) Barcode data to send to the printer
See Table 4-45 Barcode Input Conditions.

Table 4-45 Barcode Input Conditions

Barcode	Number of Data	Inputtable Data Character String (Data)	Remarks
UPC-A	11 to 12 characters	'0' to '9'	
UPC-E	11 to 12 characters	'0' to '9'	
EAN13 JAN13	12 to 13 characters	'0' to '9'	
EAN8 JAN8	7 to 8 characters	'0' to '9'	
CODE39	1 to 150 characters	'0' to '9' 'A' to 'Z' ' ', '$', '%', '+', '-', '.', '/'	Start code and stop code ('*') are automatically added.
CODE93	1 to 150 bytes	(0x00 to 0x2E)	Input data with 0x2F or more at the end.
CODE128	2 to 150 bytes	(0x00 to 0x66)	When inputting the start code (0x67 to 0x69) of the CODE128 code set. Input data with 0x67 or more at the end.
		(0x00 to 0x7F)	When starting with a CODE128 special code start code ("{A", "{B", "{C").
ITF	2 to 150 characters (However, an even number)	'0' to '9'	
CODABAR	1 to 150 characters	'0' to '9' '$', '+', '-', '.', '/', ':'	It is needed to specify one of 'A' to 'D' at the beginning and end.
EAN13 add-on JAN13 add-on	Add-on 2: 14 to 15 characters Add-on 5: 17 to 18 characters	'0' to '9'	

moduleSize Barcode width
See Table 4-23 Module Size (ModuleSize) for available constants.

moduleHeight Barcode height (dot)

hriPosition HRI character print position
See Table 4-24 HRI Character Print Position (HriPosition) for available constants.

hriFont HRI character font
See Table 4-18 Character Font (CharacterFont) for available constants.

nwRatio N:W ratio
See Table 4-25 N:W Ratio (NwRatio) for available constants.
Depending on specified *nwRatio* and *moduleSize*, the wide element width is set as shown in the following table.

Table 4-46 NW Ratio

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

Description This method maps the barcode on the print area of page mode.
In syntax (a), barcode data can be specified as a character string.
In syntax (b), barcode data can be specified as a character string, and the N:W ratio can also be specified.
In syntax (c), barcode data can be specified as a byte array.
The syntax (d) is not supported.

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.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printPageModePDF417 Print PDF417 of page mode

Target POS printer

Syntax (a) public void **printPageModePDF417**(int *startX,*
int *startY,*
String *text,*
ErrorCorrection *ErrorCorrection,*
int *row*,
int *column*,
ModuleSize *moduleSize*,
int *moduleHeight*,
Pdf417Symbol *pdf417Symbol*) throws **PrinterException**

(b) public void **printPageModePDF417**(int *startX,*
int *startY,*
String *text,*
ErrorCorrection *ErrorCorrection,*
int *row*,
int *column*,
ModuleSize *moduleSize*,
int *moduleHeight*) throws **PrinterException**

Parameter *startX* The horizontal reference point (dot) from the starting point
The valid range is 0 to 2399.

	startY	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	text	Barcode data
	errorCorrection	Error correction level See Table 4-26 Error Correction Level (ErrorCorrection) for available constants.
	row	The number of rows (row) The valid range is 0, 3 to 90. When 0 is specified, the number of rows is automatically set.
	column	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.
	moduleSize	Nominal fine element width See Table 4-23 Module Size (ModuleSize) for available constants.
	moduleHeight	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.
	pdf417Symbol	Symbol of PDF417 See Table 4-27 PDF417 Symbol (Pdf417Symbol) for available constants.
Description	This method maps PDF417 on the print area of page mode. In syntax (a), PDF417 symbol can be specified. In syntax (b), PDF417 symbol is fixed to standard PDF417. 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.	
Reference	See "Appendix B Barcode Size List" for details of the barcode size.	

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	*startX*	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.

	startY	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	text	Barcode data The version for either syntax (a) or (b) is automatically set depending on the number of data specified on *text*.
	errorCorrection	Error correction level See Table 4-26 Error Correction Level (ErrorCorrection) for available constants.
	moduleSize	Module size See Table 4-23 Module Size (ModuleSize) for available constants.
	model	QR Code Model See Table 4-28 QR Code Model (QrModel) for available constants.

Description This method maps QR Code on the print area of page mode.
In syntax (a), QR Code Model is fixed to QR Code Model 2.
In syntax (b), QR Code Model can be specified.

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.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printPageModeDataMatrix — Print Data Matrix of page mode

Target POS printer

Syntax public void **printPageModeDataMatrix**(int *startX,*
int *startY,*
String *text,*
DataMatrixModule *dataMatrixModule*,
ModuleSize *moduleSize*) throws **PrinterException**

Parameter	*startX*	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	startY	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	text	Barcode data
	dataMatrixModule	The number of Data Matrix modules See Table 4-23 Module Size (ModuleSize) for available constants.
	moduleSize	Module size See Table 4-20 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.

Reference See "Appendix B Barcode Size List" for details of the barcode size.

printPageModeMaxiCode — Print MaxiCode of page mode

Target: POS printer

Syntax:
public void **printPageModeMaxiCode**(int *startX,*
int *startY,*
String *text,*
MaxiCodeMode *maxiCodeMode*) throws **PrinterException**

Parameter:

startX	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
startY	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
text	Barcode data • When *maxiCodeMode* 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 *maxiCodeMode* 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.
maxiCodeMode	MaxiCode Mode See Table 4-30 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.

Reference: See "Appendix B Barcode Size List" for details of the barcode size.

printPageModeGS1DataBarStacked — Print GS1 Databar Stacked of page mode

This method is not supported. When executing this method, **PrinterException** is thrown.

Target: Mobile printer / POS printer

Syntax:
public void **printPageModeGS1DataBarStacked**(int *startX*,
int *startY*,
String *text*,
ModuleSize *moduleSize*) throws **PrinterException**

printPageModeGS1DataBarStackedOmnidirectional
Print GS1 Databar Stacked Omni-directional of page mode

This method is not supported. When executing this method, **PrinterException** is thrown.

Target Mobile printer / POS printer

Syntax public void **printPageModeGS1DataBarStackedOmnidirectional**(int *startX*,
int *startY*,
String *text*,
int *moduleHeight*,
ModuleSize *moduleSize*) throws **PrinterException**

printPageModeGS1DataBarExpandedStacked
Print GS1 Databar Expanded Stacked of page mode

This method is not supported. When executing this method, **PrinterException** is thrown.

Target Mobile printer / POS printer

Syntax public void **printPageModeGS1DataBarExpandedStacked**(int *startX*,
int *startY*,
String *text*,
int *column*,
ModuleSize *moduleSize*) throws **PrinterException**

printPageModeAztecCode — Print Aztec Code of page mode

This method is not supported. When executing this method, **PrinterException** is thrown.

Target Mobile printer / POS printer

Syntax public void **printPageModeAztecCode**(int *startX*,
int *startY*,
String *text*,
int *layer*,
int *errorCorrection*,
ModuleSize *moduleSize*,
AztecSymbol *aztecSymbol*) throws **PrinterException**

sendPageModeBinary — Send binary data of page mode

Target POS printer

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

Parameter *binary* Binary data
Data size that can be specified at 1 time is 3 MB (3145728 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.

printPageModeImageFile Draw Image file of page mode

Target POS printer

Syntax public void **printPageModeImageFile**(int *startX,*
int *startY,*
String *fileName,*
Dithering *dithering*) throws **PrinterException**

Parameter

startX The horizontal reference point (dot) from the starting point
The valid range is 0 to 2399.

startY The vertical reference point (dot) from the starting point
The valid range is 0 to 2399.

fileName File path of the data
The formats that can be entered are described below.

- Absolute path string handled by Java standard class "java.io.File"
  When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.
  See "3.5 Precautions - About Scoped Storage" for details.

- URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android
  - file://
  - content://

  It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.

The maximum file size that can be specified is 16 MB (16777216 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.

dithering Dithering
See Table 4-12 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 *startX,*
int *startY,*
int *endX,*
int *endY,*
LineStyle *LineStyle*) throws **PrinterException**

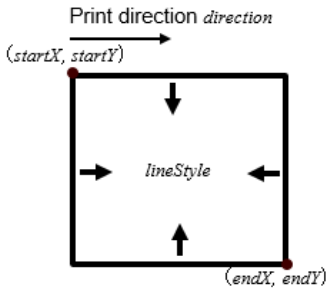
Parameter:

startX	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
startY	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
endX	The horizontal reference point (dot) from the ending point The valid range is 0 to 2399.
endY	The vertical reference point (dot) from the ending point The valid range is 0 to 2399.
lineStyle	Line style See Table 4-35 Line style (LineStyle) for available constants.

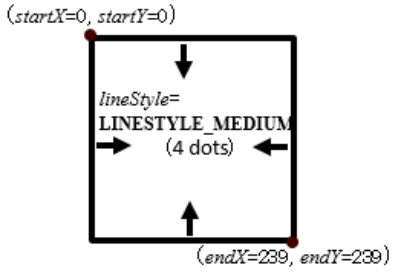
Description: This method maps the rectangle image on the print area of page mode.

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

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



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
	*startX* 0 *startY* 0 *endX* 239 *endY* 239 *lineStyle* **LINESTYLE_MEDIUM**

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

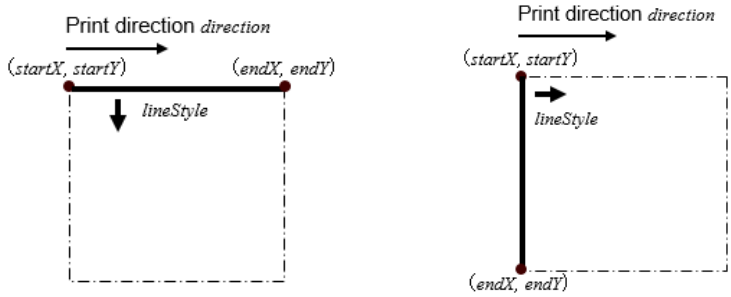
startX	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
startY	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
endX	The horizontal reference point (dot) from the ending point The valid range is 0 to 2399.
endY	The vertical reference point (dot) from the ending point The valid range is 0 to 2399.
lineStyle	Line style See Table 4-35 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.

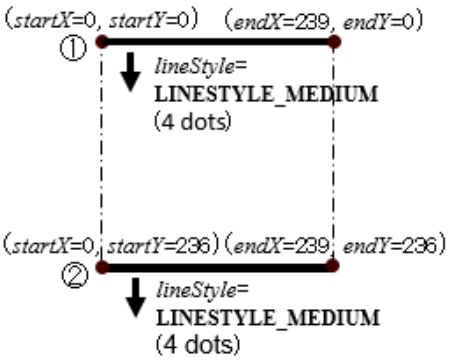
A diagonal stroke cannot be drawn by this method.

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

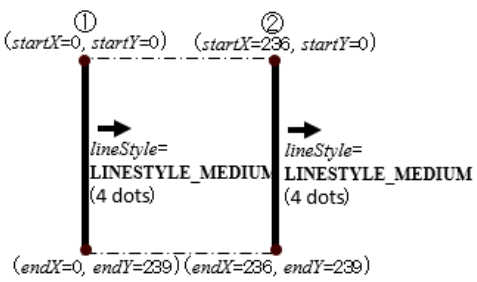


Mapping direction of horizontal ruled line **Mapping direction of vertical ruled line**

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
	①
	startX 0
	startY 0
	endX 239
	endY 0
	lineStyle **LINESTYLE_MEDIUM**
	②
	startX 0
	startY 236
	endX 239
	endY 236
	lineStyle **LINESTYLE_MEDIUM**

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
	①
	startX 0
	startY 0
	endX 0
	endY 239
	lineStyle **LINESTYLE_MEDIUM**
	②
	startX 236
	startY 0
	endX 236
	endY 239
	lineStyle **LINESTYLE_MEDIUM**

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

printPageModeLogo — Print logo of page mode

Target	POS printer	
Syntax	public void **printPageModeLogo**(int *startX,* int *startY,* String *id*) throws **PrinterException**	
Parameter	*startX*	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	startY	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	id	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.	

4.4.2 PrinterEvent Class

(1) Method List

This class obtains the event type proceeded when printer searching is completed.
Methods provided by **PrintEvent** class are shown in the following table.

Table 4-47 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 the following table.

Table 4-48 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 4-48 Event Type Constant for details.

4.4.3 PrinterListener Interface

(1) Method List

This interface obtains the complete event of printer searching.
Methods of **PrinterListener** interface are shown in the following table.

Table 4-49 Method of PrinterListener Interface

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

(2) Method Details

finishEvent — Finish event of the printer search

Target: Mobile printer / POS printer

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

Parameter: *event* Printer event

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

4.4.4 PrinterInfo Class

(1) Method List

This class stores the printer information searched by printer searching method.
Printer model name, Bluetooth address, MAC address, IP address, port name (device path) and pairing status can be obtained by searched printer information. Methods of **PrinterInfo** class are shown in the following table.
The available method varies depending on the target printer either of Mobile printer or POS printer.

Table 4-50 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

4.4.5 PrinterIdentifier Class

PrinterIdentifier class stores the printer ID of the printer retrieved by **getPrinterResponse** method.
This class is valid only for POS printer.

(1) Method List

Printer model ID, type ID, ROM version ID, firmware version (main), manufacturer, model name, serial number, firmware version (boot), firmware checksum (boot), firmware checksum (main) and firmware checksum (main + boot) can be retrieved. Methods of **PrinterIdentifier** class are shown in the following table.

Table 4-51 Method of PrinterIdentifier Class

Name	Description	Target	
		Mobile	Mobile
getPrinterModelID	Get printer model ID	Not supported	Supported
getTypeID	Get type ID	Not supported	Supported
getRomVersionID	Get ROM version ID	Not supported	Supported
getMainFirmwareVersion	Get firmware version (main)	Not supported	Supported
getManufacturer	Get manufacturer	Not supported	Supported
getModelName	Get model name	Not supported	Supported
getSerialNumber	Get serial number	Not supported	Supported
getBootFirmwareVersion	Get firmware version (boot)	Not supported	Supported
getBootFirmwareChecksum	Get firmware checksum (boot)	Not supported	Supported
getMainFirmwareChecksum	Get firmware checksum (main)	Not supported	Supported
getFirmwareChecksum	Get firmware checksum (main + boot)	Not supported	Supported

(2) Method Details

getPrinterModelID — Get printer model ID

Target	POS printer
Syntax	public byte **getPrinterModelID**()
Description	Gets the printer model ID from the printer ID of the printer retrieved by **getPrinterResponse** method.
Return value	Printer model ID

getTypeID — Get type ID

Target	POS printer
Syntax	public byte **getTypeID**()
Description	Gets the type ID from the printer ID of the printer retrieved by **getPrinterResponse** method.
Return value	Type ID

getRomVersionID — Get ROM version ID

Target	POS printer
Syntax	public byte **getRomVersionID**()
Description	Gets the ROM version ID from the printer ID of the printer retrieved by **getPrinterResponse** method.
Return value	ROM version ID

getMainFirmwareVersion — Get firmware version (main)

Target	POS printer
Syntax	public String **getMainFirmwareVersion**()
Description	Gets the firmware version (main) as a string from the printer ID of the printer retrieved by **getPrinterResponse** method.
Return value	Firmware version (main)

getManufacturer — Get manufacturer

Target	POS printer
Syntax	public String **getManufacturer**()
Description	Gets the manufacturer name as a string from the printer ID of the printer retrieved by **getPrinterResponse** method.
Return value	Manufacturer name

getModelName — Get model name

Target	POS printer
Syntax	public String **getModelName**()
Description	Gets the model name as a string from the printer ID of the printer retrieved by **getPrinterResponse** method.
Return value	Model name

getSerialNumber — Get serial number

Target	POS printer
Syntax	public String **getSerialNumber**()
Description	Gets the serial number as a string from the printer ID of the printer retrieved by **getPrinterResponse** method.
Return value	Serial number

getBootFirmwareVersion — Get firmware version (boot)

Target	POS printer
Syntax	public String **getBootFirmwareVersion**()
Description	Gets the firmware version (boot) as a string from the printer ID of the printer retrieved by **getPrinterResponse** method.
Return value	Firmware version (boot)

getBootFirmwareChecksum — Get firmware checksum (boot)

Target	POS printer
Syntax	public byte[] **getBootFirmwareChecksum**()
Description	Gets the firmware checksum (boot) from the printer ID of the printer retrieved by **getPrinterResponse** method.
Return value	Firmware checksum (boot)

getMainFirmwareChecksum — Get firmware checksum (main)

Target	POS printer
Syntax	public byte[] **getMainFirmwareChecksum**()
Description	Gets the firmware checksum (main) from the printer ID of the printer retrieved by **getPrinterResponse** method.
Return value	Firmware checksum (main)

getFirmwareChecksum — Get firmware checksum (main + boot)

Target	POS printer
Syntax	public byte[] **getFirmwareChecksum**()
Description	Gets the firmware checksum (main + boot) from the printer ID of the printer retrieved by **getPrinterResponse** method.
Return value	Firmware checksum (main + boot)

4.4.6 PrinterException Class

(1) Method List

The list of methods provided by **PrinterException** class is shown in the following table.

Table 4-52 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 the following table.

Table 4-53 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_SPE CIFIED	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 *code*, String *message*)
Description	Constructor for **com.seikoinstruments.sdk.thermalprinter.PrinterException** class.

PrinterException — Constructor

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

getErrorCode — Obtain error codes

Target	Mobile printer / POS printer
Syntax	public int **getErrorCode**()
Description	This method obtains error code for thrown exception.
Return value	See Table 4-53 Error Codes List for details.

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

Table 4-54 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.

4.4.8 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.

4.4.9 SmartLabelManager Class

SmartLabelManager class provides the function to convert the label file (*.sl, *.slex) created using SII Layout Editor into the printable data from the printer.

Mobile printer and POS printer do not support this class.

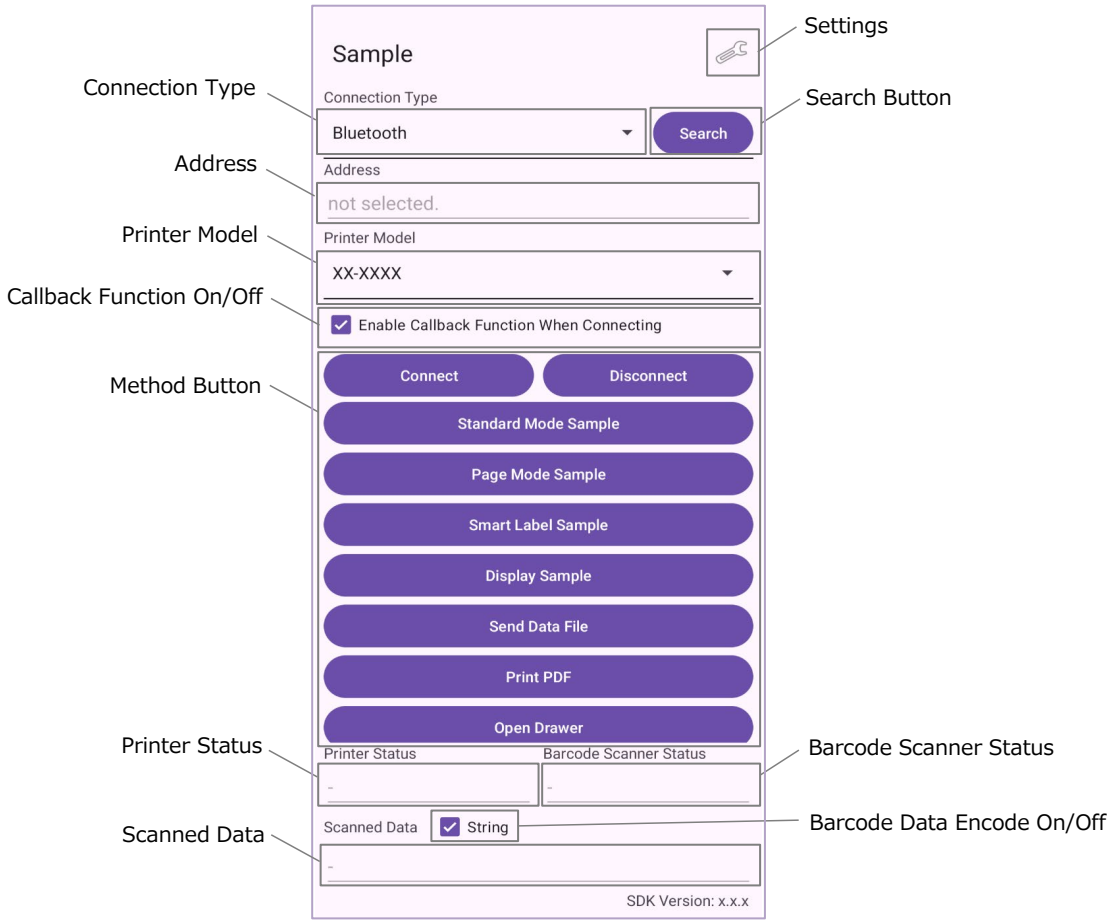
Chapter 5
Sample Program

This chapter describes the sample programs provided by the SDK.

5.1 Screen

The SDK includes the sample program with Android Studio project format.
This chapter describes the screen of the sample program.

5.1.1 Main Screen

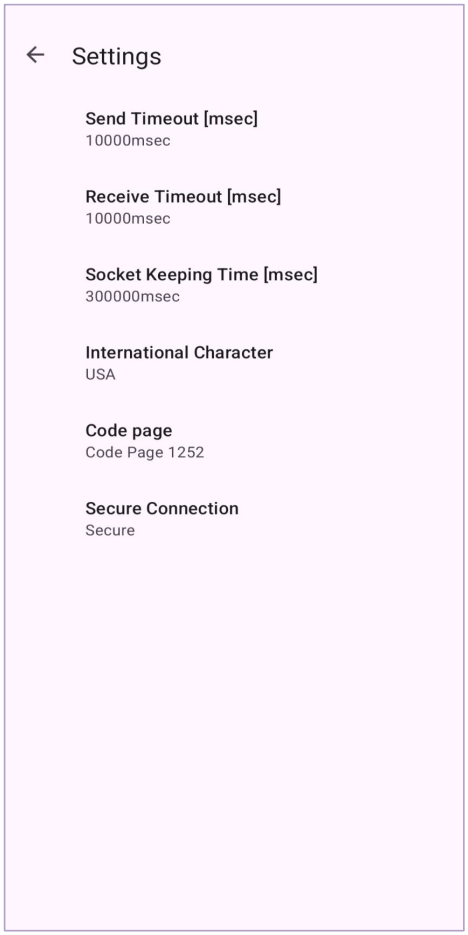


Item	Description
Settings	Tapping the [Settings] button opens the function setting screen. In order to go back to the main screen, tap ← button on the top left of the screen.
Connection type	Selects the connection type to the printer.
Search Button	Starts searching for the type of printer specified in [Connection Type]. Transits to the printer search view. A list of the searched printers is displayed. The printer is selected by tapping the searched printer and returns to the main view.
Address	Displays the information about the selected printer.
Printer Model	Specifies the printer model.
Callback Function On/Off	Select whether to enable the callback function when connecting to the printer. On : Starts the callback function when connecting. Off : The callback function does not respond.
Method Button*1	In addition to the method buttons for executing **connect** and **disconnect**, the sample by the combination of some methods can be printed and checked for the operation of peripheral devices.
Printer Status	Displays the printer status. When [Callback Function On/Off] is On, the latest printer status is displayed.
Barcode Scanner Status	Displays the connection status of the barcode scanner. Mobile printer and POS printer do not support the barcode scanner.
Barcode Data Encode On/Off	Selects the conversion of barcode data read by the barcode scanner. Mobile printer and POS printer do not support the barcode scanner.
Scanned Data	Displays the barcode data read by the barcode scanner. Mobile printer and POS printer do not support the barcode scanner.

*1: Supported functions vary by model. Only supported functions can be operated.

5.1.2 [Setting] Screen

Various setting functions are displayed in [Settings].



5.2 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**

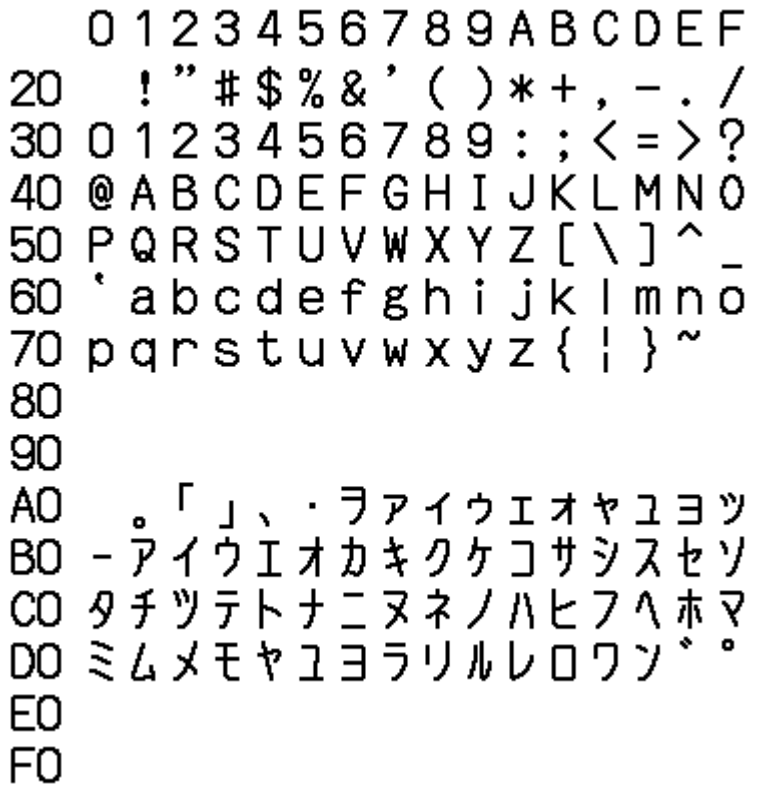


Figure A-1 CODE_PAGE_KATAKANA



Figure A-2 CODE_PAGE_1252 (Latin)

- **POS printer**

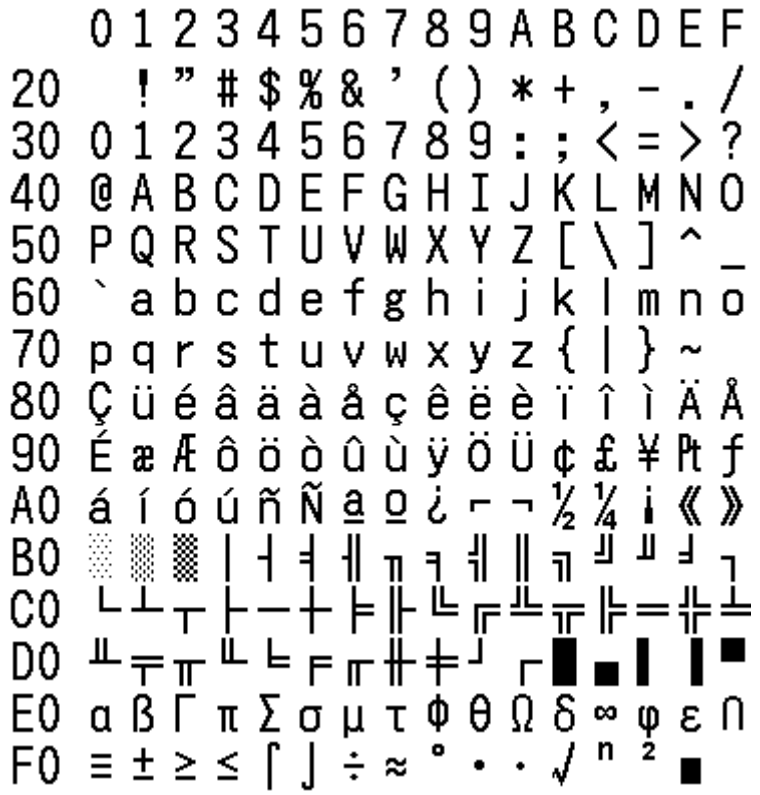


Figure A-3　CODE_PAGE_437 (USA, Standard Europe)

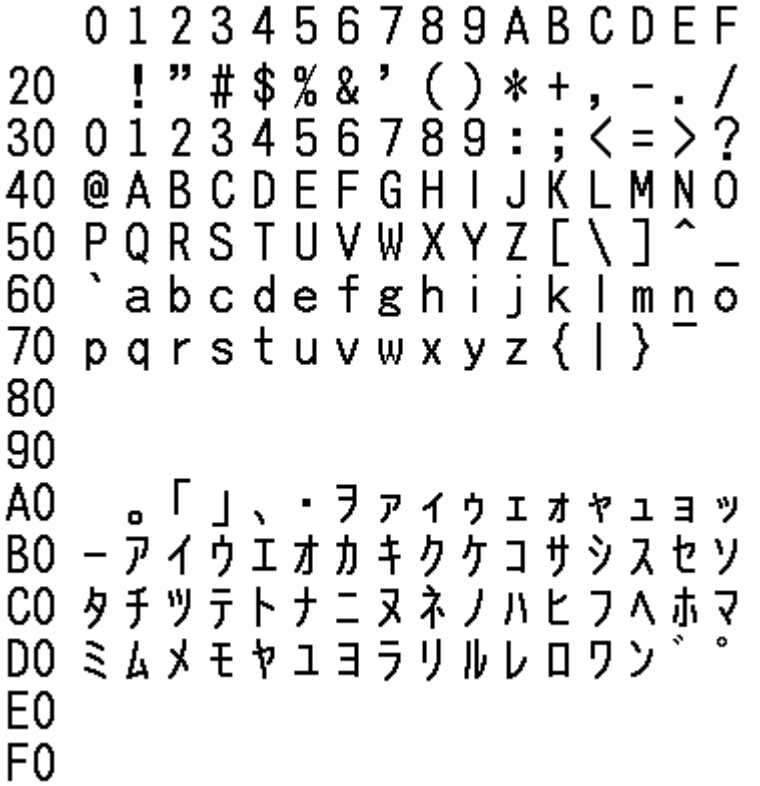


Figure A-4　CODE_PAGE_KATAKANA

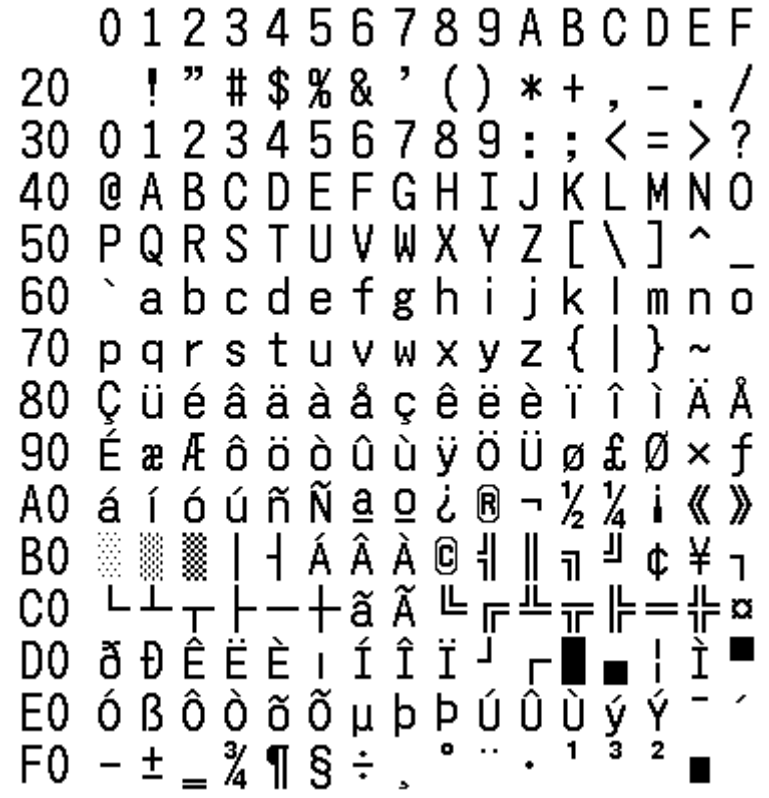


Figure A-5 CODE_PAGE_850 (Multilingual)

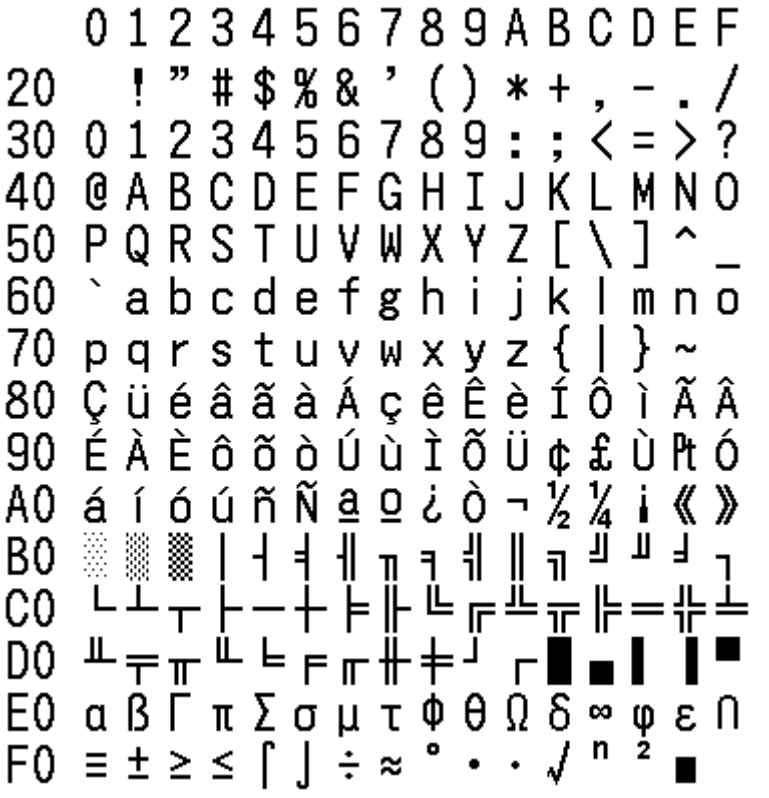


Figure A-6 CODE_PAGE_860 (Portuguese)

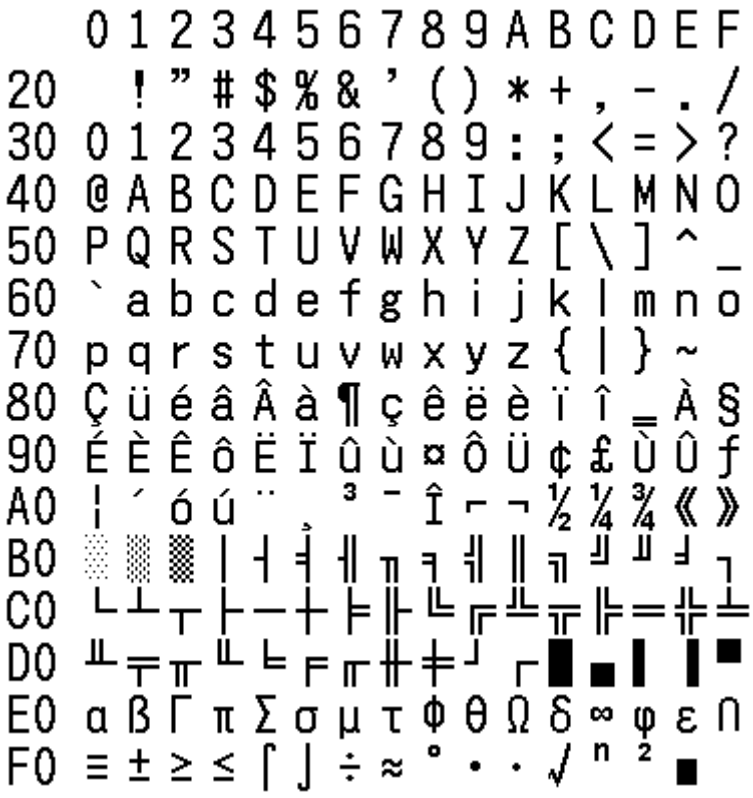


Figure A-7 CODE_PAGE_863 (Canadian-French)

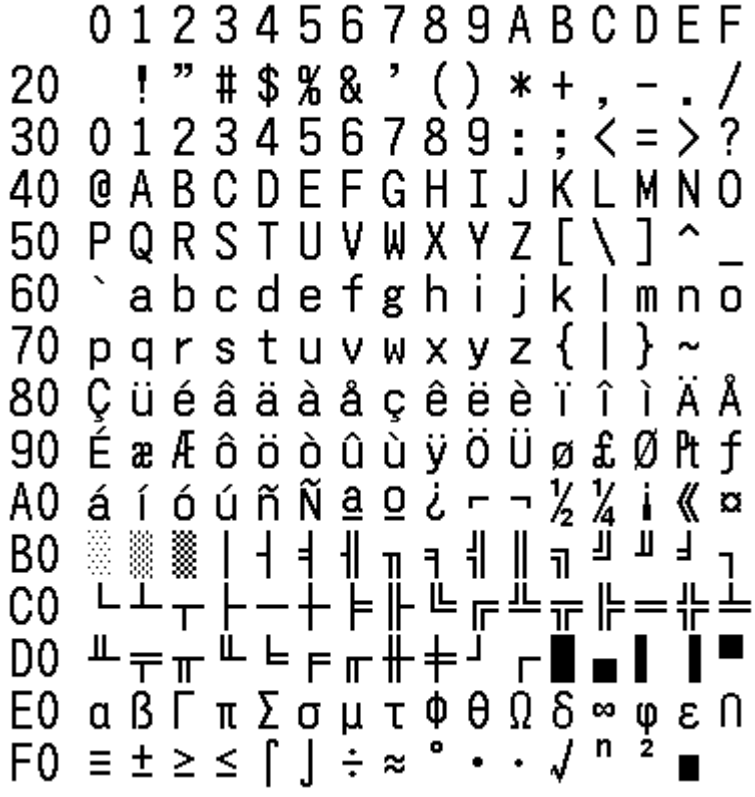


Figure A-8 CODE_PAGE_865 (Nordic)

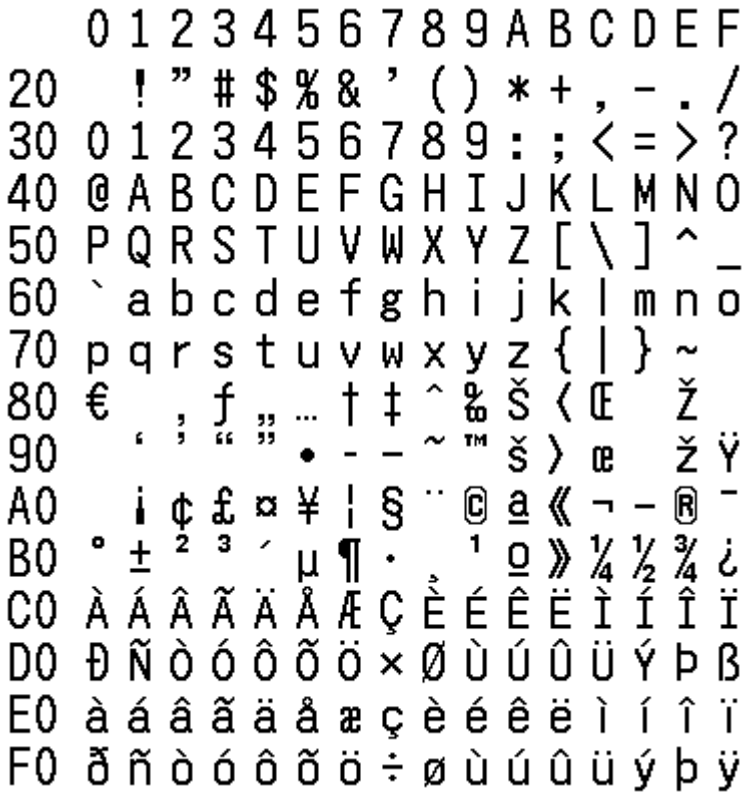


Figure A-9 CODE_PAGE_1252 (Latin)

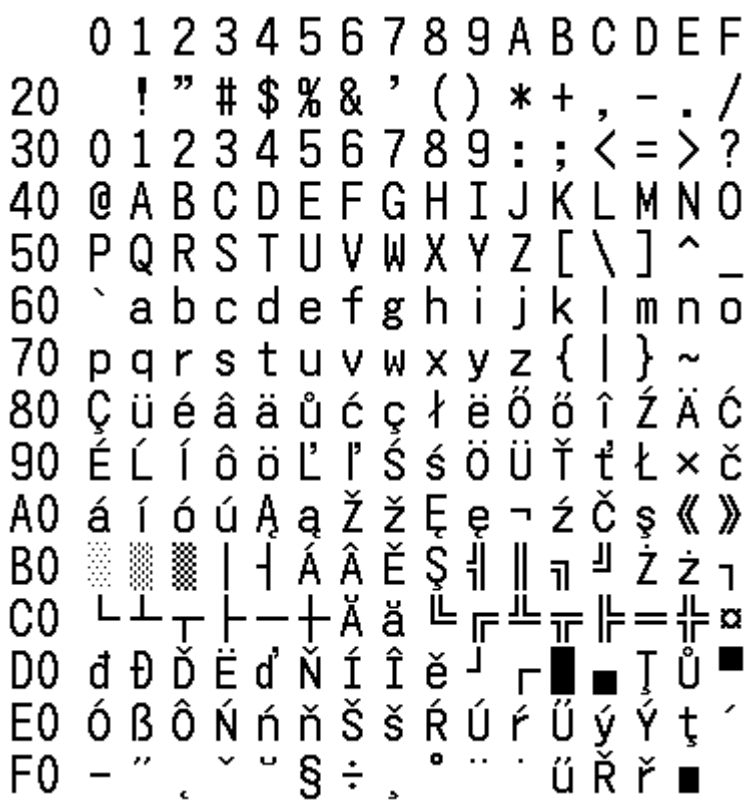


Figure A-10 CODE_PAGE_852 (Eastern Europe)

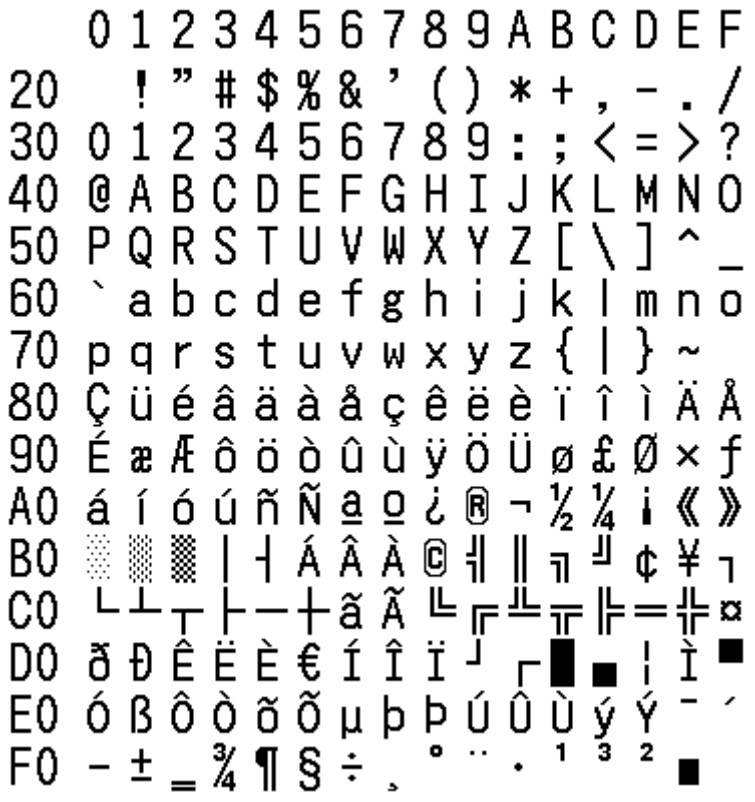


Figure A-11 CODE_PAGE_858 (Euro)

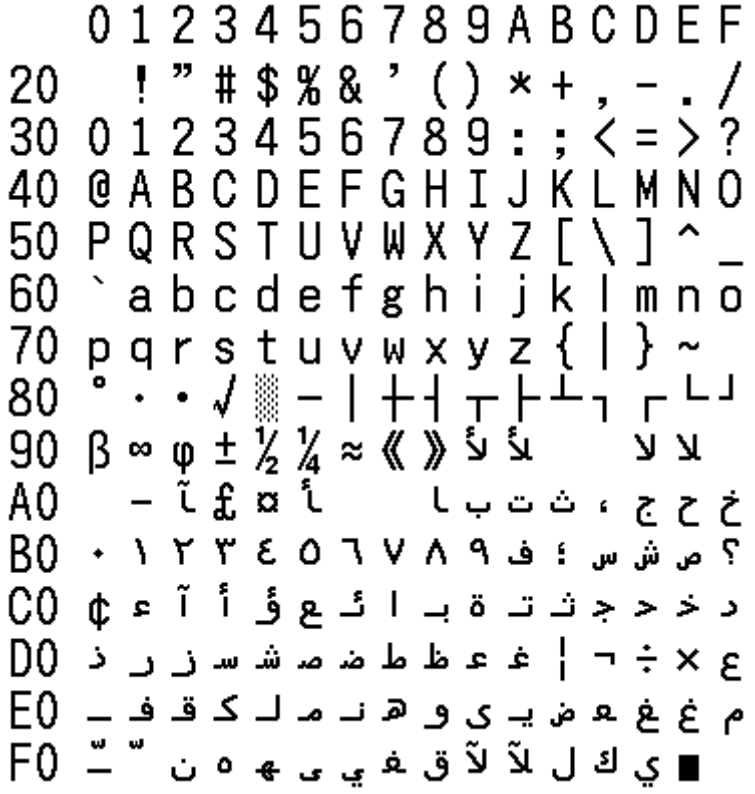


Figure A-12 CODE_PAGE_864 (Arabic)



Figure A-13 CODE_PAGE_1250 (Central European)



Figure A-14 CODE_PAGE_1251 (Cyrillic)

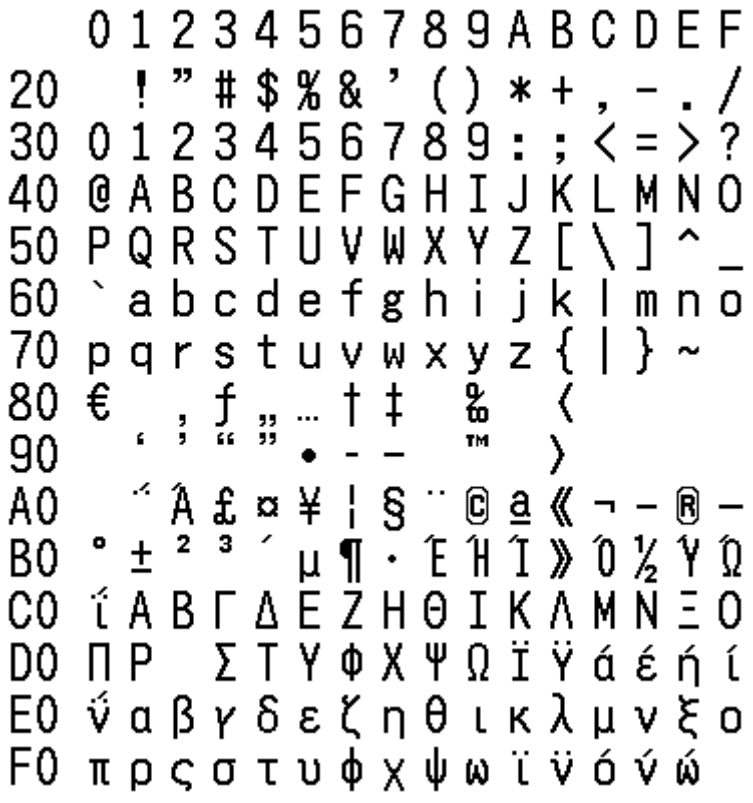


Figure A-15 CODE_PAGE_1253 (Greek)



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

Appendix B
Barcode Size List

B.1 Barcode Size List

B.1.1 printBarcode，printPageModeBarcode



(1) Height of the barcode image

hriFont	*hriPosition*	Length from Top of Barcode to Reference Point	Height of Barcode Image
FONT_A	**HRI_NONE**	*moduleHeight*	*moduleHeight*
	HRI_POSITION_ABOVE	*moduleHeight* + 32	*moduleHeight* + 32
	HRI_POSITION_BELOW	*moduleHeight*	*moduleHeight* + 32
	HRI_POSITION_ABOVE_BELOW	*moduleHeight* + 64	*moduleHeight* + 64
FONT_B	**HRI_NONE**	*moduleHeight*	*moduleHeight*
	HRI_POSITION_ABOVE	*moduleHeight* + 24	*moduleHeight* + 24
	HRI_POSITION_BELOW	*moduleHeight*	*moduleHeight* + 24
	HRI_POSITION_ABOVE_BELOW	*moduleHeight* + 48	*moduleHeight* + 48

(2) Width of the barcode image

barcodeSymbol	*moduleSize*	**Width of Barcode Image**
BARCODE_UPC_A	**BARCODE_MODULE_WIDTH_2**	226
	BARCODE_MODULE_WIDTH_3	339
	BARCODE_MODULE_WIDTH_4	452
	BARCODE_MODULE_WIDTH_5	565
	BARCODE_MODULE_WIDTH_6	678
BARCODE_UPC_E	**BARCODE_MODULE_WIDTH_2**	130
	BARCODE_MODULE_WIDTH_3	195
	BARCODE_MODULE_WIDTH_4	260
	BARCODE_MODULE_WIDTH_5	325
	BARCODE_MODULE_WIDTH_6	390
BARCODE_EAN13	**BARCODE_MODULE_WIDTH_2**	226
	BARCODE_MODULE_WIDTH_3	339
	BARCODE_MODULE_WIDTH_4	452
	BARCODE_MODULE_WIDTH_5	565
	BARCODE_MODULE_WIDTH_6	678
BARCODE_JAN13	**BARCODE_MODULE_WIDTH_2**	226
	BARCODE_MODULE_WIDTH_3	339
	BARCODE_MODULE_WIDTH_4	452
	BARCODE_MODULE_WIDTH_5	565
	BARCODE_MODULE_WIDTH_6	678
BARCODE_EAN8	**BARCODE_MODULE_WIDTH_2**	162
	BARCODE_MODULE_WIDTH_3	243
	BARCODE_MODULE_WIDTH_4	324
	BARCODE_MODULE_WIDTH_5	405
	BARCODE_MODULE_WIDTH_6	486
BARCODE_JAN8	**BARCODE_MODULE_WIDTH_2**	162
	BARCODE_MODULE_WIDTH_3	243
	BARCODE_MODULE_WIDTH_4	324
	BARCODE_MODULE_WIDTH_5	405
	BARCODE_MODULE_WIDTH_6	486
BARCODE_CODE93	**BARCODE_MODULE_WIDTH_2**	18 × number of barcode data + 96
	BARCODE_MODULE_WIDTH_3	27 × number of barcode data + 144
	BARCODE_MODULE_WIDTH_4	36 × number of barcode data + 192
	BARCODE_MODULE_WIDTH_5	45 × number of barcode data + 240
	BARCODE_MODULE_WIDTH_6	54 × number of barcode data + 288

barcodeSymbol	*moduleSize*	**Width of Barcode Image**
BARCODE_CODE128	**BARCODE_MODULE_WIDTH_2**	22 × number of barcode data + 66
	BARCODE_MODULE_WIDTH_3	33 × number of barcode data + 99
	BARCODE_MODULE_WIDTH_4	44 × number of barcode data + 132
	BARCODE_MODULE_WIDTH_5	55 × number of barcode data + 165
	BARCODE_MODULE_WIDTH_6	66 × number of barcode data + 198

barcodeSymbol	*nwRatio*	*moduleSize*	**Width of Barcode Image**
BARCODE_CODE39	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	26 × number of barcode data + 90
		BARCODE_MODULE_WIDTH_3	39 × number of barcode data + 135
		BARCODE_MODULE_WIDTH_4	52 × number of barcode data + 180
		BARCODE_MODULE_WIDTH_5	65 × number of barcode data + 225
		BARCODE_MODULE_WIDTH_6	78 × number of barcode data + 270
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	29 × number of barcode data + 96
		BARCODE_MODULE_WIDTH_3	45 × number of barcode data + 147
		BARCODE_MODULE_WIDTH_4	58 × number of barcode data + 192
		BARCODE_MODULE_WIDTH_5	74 × number of barcode data + 243
		BARCODE_MODULE_WIDTH_6	87 × number of barcode data + 288
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	32 × number of barcode data + 102
		BARCODE_MODULE_WIDTH_3	48 × number of barcode data + 153
		BARCODE_MODULE_WIDTH_4	64 × number of barcode data + 204
		BARCODE_MODULE_WIDTH_5	80 × number of barcode data + 255
		BARCODE_MODULE_WIDTH_6	96 × number of barcode data + 306

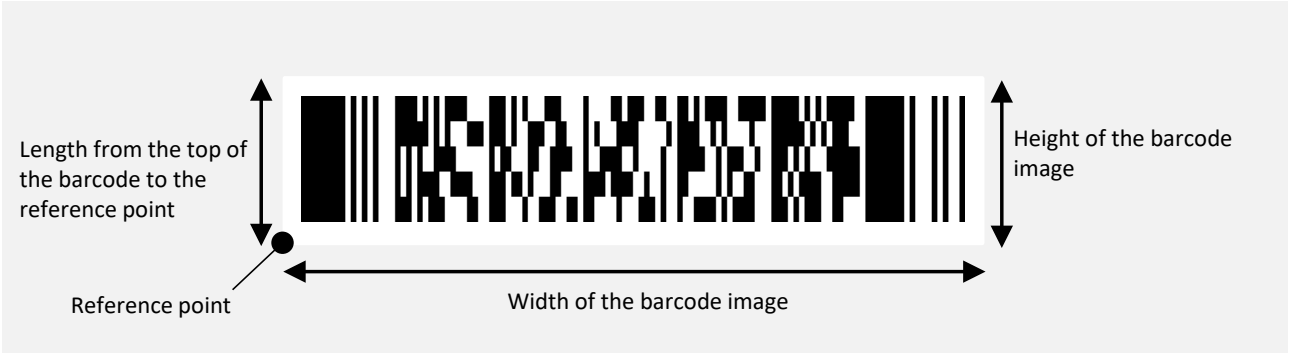
barcodeSymbol	*nwRatio*	*moduleSize*	**Width of Barcode Image**
BARCODE_ITF	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	14 × number of barcode data + 56
		BARCODE_MODULE_WIDTH_3	21 × number of barcode data + 84
		BARCODE_MODULE_WIDTH_4	28 × number of barcode data + 112
		BARCODE_MODULE_WIDTH_5	35 × number of barcode data + 140
		BARCODE_MODULE_WIDTH_6	42 × number of barcode data + 168
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	16 × number of barcode data + 57
		BARCODE_MODULE_WIDTH_3	25 × number of barcode data + 86
		BARCODE_MODULE_WIDTH_4	32 × number of barcode data + 114
		BARCODE_MODULE_WIDTH_5	41 × number of barcode data + 143
		BARCODE_MODULE_WIDTH_6	48 × number of barcode data + 171
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	18 × number of barcode data + 58
		BARCODE_MODULE_WIDTH_3	27 × number of barcode data + 87
		BARCODE_MODULE_WIDTH_4	36 × number of barcode data + 116
		BARCODE_MODULE_WIDTH_5	45 × number of barcode data + 145
		BARCODE_MODULE_WIDTH_6	54 × number of barcode data + 174
BARCODE_CODABAR*1	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	20 × number of data + 2 × (2 + number of wide data) + 38
		BARCODE_MODULE_WIDTH_3	30 × number of data + 3 × (2 + number of wide data) + 57
		BARCODE_MODULE_WIDTH_4	40 × number of data + 4 × (2 + number of wide data) + 76
		BARCODE_MODULE_WIDTH_5	50 × number of data + 5 × (2 + number of wide data) + 95
		BARCODE_MODULE_WIDTH_6	60 × number of data + 6 × (2 + number of wide data) + 114
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	22 × number of data + 3 × (2 + number of wide data) + 38
		BARCODE_MODULE_WIDTH_3	34 × number of data + 5 × (2 + number of wide data) + 57
		BARCODE_MODULE_WIDTH_4	44 × number of data + 6 × (2 + number of wide data) + 76

barcodeSymbol	*nwRatio*	*moduleSize*	**Width of Barcode Image**
BARCODE_CODABAR[*1]	**NWRATIO_1TO2_5**	**BARCODE_MODULE_WIDTH_5**	56 × number of data + 8 × (2 + number of wide data) + 95
		BARCODE_MODULE_WIDTH_6	66 × number of data + 9 × (2 + number of wide data) + 114
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	24 × number of data + 4 × (2 + number of wide data) + 38
		BARCODE_MODULE_WIDTH_3	36 × number of data + 6 × (2 + number of wide data) + 57
		BARCODE_MODULE_WIDTH_4	48 × number of data + 8 × (2 + number of wide data) + 76
		BARCODE_MODULE_WIDTH_5	60 × number of data + 10 × (2 + number of wide data) + 95
		BARCODE_MODULE_WIDTH_6	72 × number of data + 12 × (2 + number of wide data) + 114

*1: The number of data is the number of all characters except for the start and stop characters.
The wide data is the number of ": / . +".

barcodeSymbol	**Number of Data**	*moduleSize*	**Width of Barcode Image**
BARCODE_EAN13_ADDON	14 or 15	**BARCODE_MODULE_WIDTH_2**	276
		BARCODE_MODULE_WIDTH_3	414
		BARCODE_MODULE_WIDTH_4	552
		BARCODE_MODULE_WIDTH_5	690
		BARCODE_MODULE_WIDTH_6	828
	17 or 18	**BARCODE_MODULE_WIDTH_2**	330
		BARCODE_MODULE_WIDTH_3	495
		BARCODE_MODULE_WIDTH_4	660
		BARCODE_MODULE_WIDTH_5	825
		BARCODE_MODULE_WIDTH_6	990
BARCODE_JAN13_ADDON	14 or 15	**BARCODE_MODULE_WIDTH_2**	276
		BARCODE_MODULE_WIDTH_3	414
		BARCODE_MODULE_WIDTH_4	552
		BARCODE_MODULE_WIDTH_5	690
		BARCODE_MODULE_WIDTH_6	828
	17 or 18	**BARCODE_MODULE_WIDTH_2**	330
		BARCODE_MODULE_WIDTH_3	495
		BARCODE_MODULE_WIDTH_4	660
		BARCODE_MODULE_WIDTH_5	825
		BARCODE_MODULE_WIDTH_6	990

B.1.2 printPDF417, printPageModePDF417



(1) Height of the barcode image

moduleSize	**Height of Barcode Image**[*1]
PDF417_MODULE_WIDTH_2	*moduleHeight* × *row*[*2] + 8
PDF417_MODULE_WIDTH_3	*moduleHeight* × *row*[*2] + 12
PDF417_MODULE_WIDTH_4	*moduleHeight* × *row*[*2] + 16
PDF417_MODULE_WIDTH_5	*moduleHeight* × *row*[*2] + 20
PDF417_MODULE_WIDTH_6	*moduleHeight* × *row*[*2] + 24
PDF417_MODULE_WIDTH_7	*moduleHeight* × *row*[*2] + 28
PDF417_MODULE_WIDTH_8	*moduleHeight* × *row*[*2] + 32

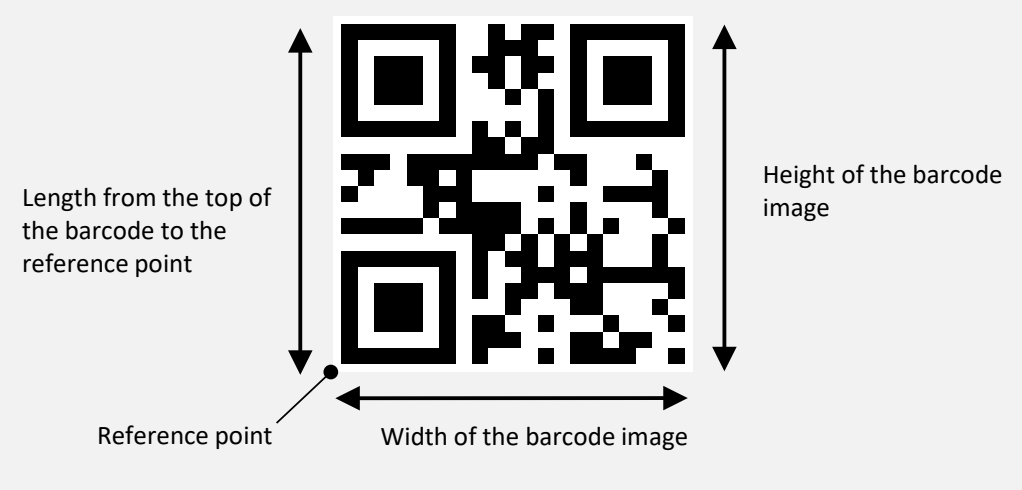
*1: Height of the barcode image = Length from the top of the barcode to the reference point
*2: Parameter $row \neq 0$

(2) Width of the barcode image

pdf417Symol	*moduleSize*	**Width of Barcode Image**
PDF417_STANDARD	**PDF417_MODULE_WIDTH_2**	(17 × *column*[*1] + 69) × 2 + 8
	PDF417_MODULE_WIDTH_3	(17 × *column*[*1] + 69) × 2 + 12
	PDF417_MODULE_WIDTH_4	(17 × *column*[*1] + 69) × 2 + 16
	PDF417_MODULE_WIDTH_5	(17 × *column*[*1] + 69) × 2 + 20
	PDF417_MODULE_WIDTH_6	(17 × *column*[*1] + 69) × 2 + 24
	PDF417_MODULE_WIDTH_7	(17 × *column*[*1] + 69) × 2 + 28
	PDF417_MODULE_WIDTH_8	(17 × *column*[*1] + 69) × 2 + 32
PDF417_COMPACT	**PDF417_MODULE_WIDTH_2**	(17 × *column*[*1] + 35) × 2 + 8
	PDF417_MODULE_WIDTH_3	(17 × *column*[*1] + 35) × 2 + 12
	PDF417_MODULE_WIDTH_4	(17 × *column*[*1] + 35) × 2 + 16
	PDF417_MODULE_WIDTH_5	(17 × *column*[*1] + 35) × 2 + 20
	PDF417_MODULE_WIDTH_6	(17 × *column*[*1] + 35) × 2 + 24
	PDF417_MODULE_WIDTH_7	(17 × *column*[*1] + 35) × 2 + 28
	PDF417_MODULE_WIDTH_8	(17 × *column*[*1] + 35) × 2 + 32

*1: Parameter $column \neq 0$

B.1.3 printQRCode， printPageModeQRCode



(1) Height and width of the barcode image

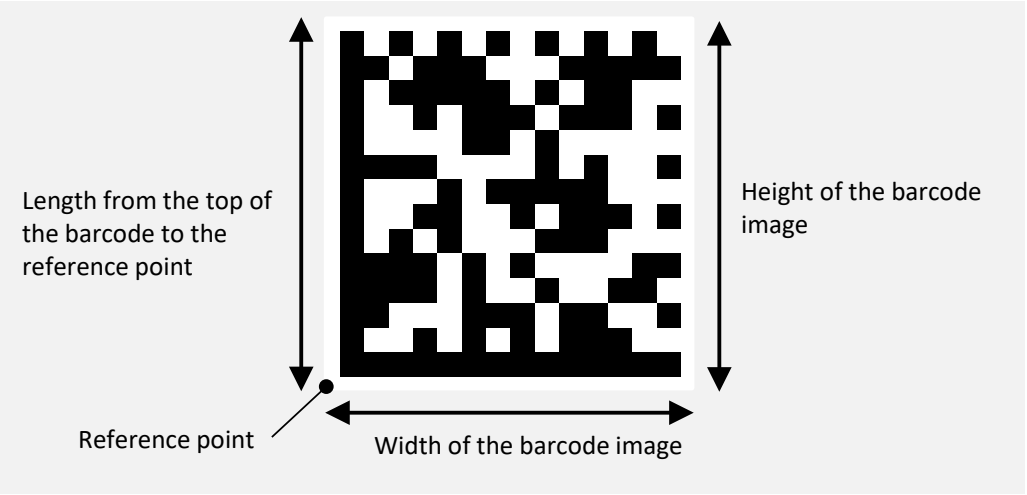
Height[*1] and width of the barcode image = (4 × version[*2] + 17 + 8) × module size value

*1: Height of the barcode image = Length from the top of the barcode to the reference point
*2: The version is determined by the content of the barcode data and the error correction level.

Module Size Value

moduleSize	**Module Size Value**
QR_MODULE_SIZE_2	2
QR_MODULE_SIZE_3	3
QR_MODULE_SIZE_4	4
QR_MODULE_SIZE_5	5
QR_MODULE_SIZE_6	6
QR_MODULE_SIZE_7	7
QR_MODULE_SIZE_8	8
QR_MODULE_SIZE_9	9
QR_MODULE_SIZE_10	10
QR_MODULE_SIZE_11	11
QR_MODULE_SIZE_12	12
QR_MODULE_SIZE_13	13
QR_MODULE_SIZE_14	14
QR_MODULE_SIZE_15	15
QR_MODULE_SIZE_16	16

B.1.4 printDataMatrix， printPageModeDataMatrix



(1) Height and width of the barcode image

Height of the barcode image = (number of vertical module + 2) × module size value

Width of the barcode image = (number of horizontal module + 2) × module size value

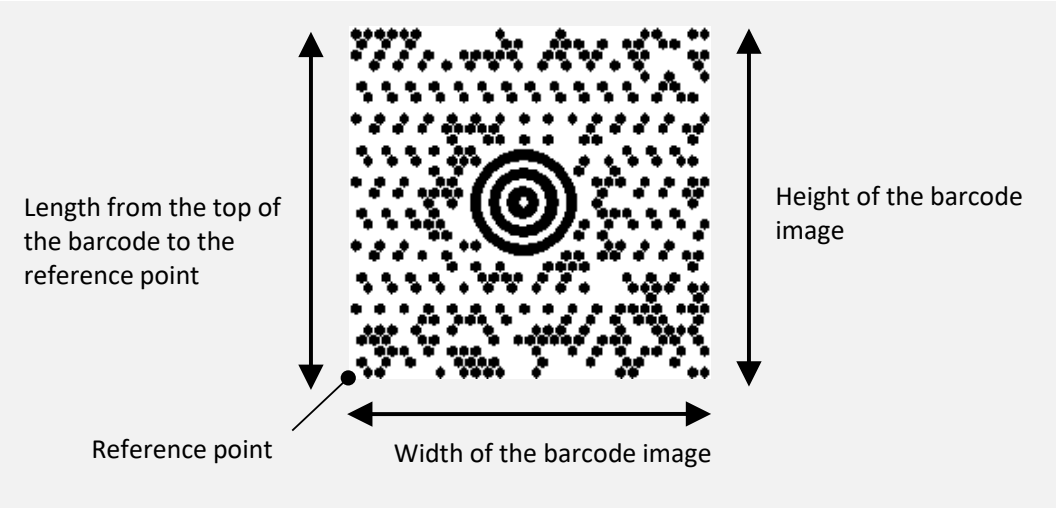
dataMatrixModule	**Number of Vertical Module**	**Number of Horizontal Module**
DATA_MATRIX_10_10	10	10
DATA_MATRIX_12_12	12	12
DATA_MATRIX_14_14	14	14
DATA_MATRIX_16_16	16	16
DATA_MATRIX_18_18	18	18
DATA_MATRIX_20_20	20	20
DATA_MATRIX_22_22	22	22
DATA_MATRIX_24_24	23	23
DATA_MATRIX_26_26	26	26
DATA_MATRIX_32_32	32	32
DATA_MATRIX_36_36	36	36
DATA_MATRIX_40_40	40	40
DATA_MATRIX_44_44	44	44
DATA_MATRIX_48_48	48	48
DATA_MATRIX_52_52	52	52
DATA_MATRIX_64_64	64	64
DATA_MATRIX_72_72	72	72
DATA_MATRIX_80_80	80	80
DATA_MATRIX_88_88	88	88
DATA_MATRIX_96_96	96	96
DATA_MATRIX_104_104	104	104
DATA_MATRIX_120_120	120	120

dataMatrixModule	Number of Vertical Module	Number of Horizontal Module
DATA_MATRIX_132_132	132	132
DATA_MATRIX_144_144	144	144
DATA_MATRIX_8_18	8	18
DATA_MATRIX_8_32	8	32
DATA_MATRIX_12_26	12	26
DATA_MATRIX_12_36	12	36
DATA_MATRIX_16_36	16	36
DATA_MATRIX_16_48	16	48

Module Size Value

moduleSize	Module Size Value
DATAMATRIX_MODULE_SIZE_2	2
DATAMATRIX_MODULE_SIZE_3	3
DATAMATRIX_MODULE_SIZE_4	4
DATAMATRIX_MODULE_SIZE_5	5
DATAMATRIX_MODULE_SIZE_6	6
DATAMATRIX_MODULE_SIZE_7	7
DATAMATRIX_MODULE_SIZE_8	8
DATAMATRIX_MODULE_SIZE_9	9
DATAMATRIX_MODULE_SIZE_10	10
DATAMATRIX_MODULE_SIZE_11	11
DATAMATRIX_MODULE_SIZE_12	12
DATAMATRIX_MODULE_SIZE_13	13
DATAMATRIX_MODULE_SIZE_14	14
DATAMATRIX_MODULE_SIZE_15	15
DATAMATRIX_MODULE_SIZE_16	16

B.1.5 printMaxicode， printPageModeMaxicode



(1) Height of the barcode image

Height of the barcode image[*1] = 215

*1: Height of the barcode image = Length from the top of the barcode to the reference point

(2) Width of the barcode image

Width of the barcode image = 225