



# SII Print Class Library for Android™ Application Programmer's Guide

Rev.04

[Products]

MP-B21L Series

Seiko Instruments Inc.

Rev.01	March 2024
Rev.02	November 2024
Rev.03	January 2025
Rev.04	February 2026

Copyright © 2024-2026 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.

App Store™ is a service mark of Apple Inc.

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.

SII reserves the right to make changes without notice to the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

## Introduction

This manual describes "SII Print Class Library for Android™" (hereinafter referred to as "SII print class library") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

### Target printers

The printers supported by SII print class library are listed below.

Printers	Interface
MP-B21L Series	Bluetooth
	USB
	TCP/IP

### Terms

The terms used in this manual are defined as below.

Terms	Description
Printer command	Command for controlling the printer described in "MP-B21L SERIES THERMAL PRINTER TECHNICAL REFERENCE"

# Table of Contents

<b>Chapter 1</b>	<b>Product Overview</b>	<b>1-1</b>
1.1	Functions Provided by SII Print Class Library .....	1-1
1.2	SII Print Class Library Overview .....	1-1
1.2.1	SII Print Class Library Configuration .....	1-1
1.2.2	Functions Provided by Library .....	1-2
<b>Chapter 2</b>	<b>Product Specifications</b>	<b>2-1</b>
2.1	Operating Environment.....	2-1
2.2	Printer Settings .....	2-2
2.3	Precautions .....	2-3
<b>Chapter 3</b>	<b>How to Use Library</b>	<b>3-1</b>
3.1	Android Application Development Environment .....	3-1
3.2	Provided Files .....	3-2
3.3	Build Library into Android Studio Project .....	3-3
3.4	Use Developed Android Application on Android Device .....	3-5
3.5	Precautions .....	3-5
<b>Chapter 4</b>	<b>Functions of Library</b>	<b>4-1</b>
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
(1)	The print data is mapped on the upper of the print area of page mode.....	4-6
(2)	The print data is mapped on the right of print area of page mode.....	4-6
4.2	Printing Label Function .....	4-7
4.2.1	Structure of Label File .....	4-7
(1)	Types of objects and support in the library .....	4-8
(2)	Precautions for printing the label file using the library .....	4-8
①	All object.....	4-8
②	Text object.....	4-8
③	Image object.....	4-8
④	Barcode object.....	4-8
⑤	Contact object.....	4-9

⑥ DateTime object .....	4-9
⑦ Drawing object.....	4-9
4.2.2 Method for Using Label File .....	4-10
(1) Print the label file as it is from the library .....	4-10
(2) Replace the object data in the label file and print .....	4-10
4.3 Log File Output Function .....	4-11
4.3.1 How to Set Log Output.....	4-11
4.3.2 Log Output Settings .....	4-11
4.3.3 Log File.....	4-11
4.4 API Reference .....	4-12
4.4.1 PrinterManager Class .....	4-13
(1) Method List.....	4-13
① Common method to standard mode and page mode.....	4-13
② Dedicated method for standard mode .....	4-14
③ Dedicated method for page mode .....	4-15
(2) Constant List .....	4-16
① Printer model .....	4-16
② Response type.....	4-16
③ Battery remaining capacity level.....	4-16
④ International character set.....	4-17
⑤ Codepage .....	4-17
⑥ Port type .....	4-18
⑦ Paper selection with or without mark when printing label file.....	4-18
⑧ Barcode or PDF417 .....	4-18
(3) Enumerated Constant List.....	4-18
① Dithering (Dithering) .....	4-18
② Batch processing selection (TransactionFunction) .....	4-19
③ Bold print (CharacterBold).....	4-19
④ Underline (CharacterUnderline) .....	4-19
⑤ Reverse print (CharacterReverse) .....	4-19
⑥ Inversion print (CharacterInversion).....	4-19
⑦ Character font (CharacterFont) .....	4-20
⑧ Character scale (CharacterScale) .....	4-20
⑨ Alignment (PrintAlignment).....	4-21
⑩ Pending data output specifying (OutputPendingData).....	4-21
⑪ Barcode symbol (BarcodeSymbol).....	4-21
⑫ Module size (ModuleSize) .....	4-22
⑬ HRI character print position (HriPosition).....	4-24
⑭ N:W ratio (NwRatio) .....	4-25
⑮ Error correction level (ErrorCorrection) .....	4-25
⑯ PDF417 symbol (Pdf417Symbol).....	4-25
⑰ QR Code Model (QrModel) .....	4-25
⑱ Data Matrix module (DataMatrixModule).....	4-26
⑲ MaxiCode Mode (MaxiCodeMode).....	4-27
⑳ Aztec Symbol (AztecSymbol) .....	4-27
㉑ Cutting method (CuttingMethod) .....	4-27
㉒ Form feed position (FeedPosition) .....	4-27
㉓ Image rotation direction (Rotate).....	4-29

②4	Image scaling (ImageScale).....	4-29
②5	Print direction (Direction).....	4-29
②6	Line style (LineStyle).....	4-30
(4)	Method Details .....	4-31
①	Common method to standard mode and page mode.....	4-31
	PrinterManager      Constructor.....	4-31
	connect              Start communicating with printer (Bluetooth) .....	4-31
	connect              Start communicating with printer (USB) .....	4-32
	connect              Start communicating with printer (TCP/IP).....	4-32
	disconnect          Stop communicating with printer .....	4-33
	setBarcodeScannerListener	
	Start/End callback of barcode scanner .....	4-33
	openDrawer          Open cash drawer .....	4-33
	buzzer                Sound buzzer .....	4-33
	externalBuzzer      Sound external buzzer .....	4-33
	getStatus            Get printer status .....	4-34
	setCallbackFunctionListener	
	Start/End callback of printer status change.....	4-35
	abort                 Abort waiting state of printer.....	4-35
	registerLogo          Register logo .....	4-36
	unregisterLogo        Delete registered logo .....	4-36
	registerStyleSheet    Register style sheet.....	4-37
	unregisterStyleSheet   Delete registered style sheet.....	4-37
	resetPrinter          Reset printer .....	4-37
	getPrinterResponse    Get various responses from printer .....	4-37
	startDiscoveryPrinter   Start printer search (Bluetooth) .....	4-39
	startDiscoveryPrinter   Start printer search (USB).....	4-40
	startDiscoveryPrinter   Start printer search (TCP/IP).....	4-40
	cancelDiscoveryPrinter   Cancel printer search .....	4-41
	getFoundPrinter        Get found printer information.....	4-41
	getSendTimeout        Get send timeout period .....	4-41
	setSendTimeout        Set send timeout period .....	4-41
	getReceiveTimeout     Get receive timeout period .....	4-41
	setReceiveTimeout     Set receive timeout period.....	4-42
	getInternationalCharacter	
	Get international character set .....	4-42
	setInternationalCharacter	
	Set international character set.....	4-42
	getCodePage            Get codepage .....	4-43
	setCodePage            Set codepage .....	4-43
	getPrinterModel        Get printer model.....	4-43
	getPortType            Get connecting port type .....	4-43
	isConnect              Verify connection state with printer .....	4-44
	getSocketKeepingTime   Get socket keeping time.....	4-44
	setSocketKeepingTime   Set socket keeping time .....	4-44
	getPrintSmartLabelMode	
	Get paper when printing label file.....	4-44

setPrintSmartLabelMode		
	Set paper when printing label file .....	4-45
getVersion	Get SDK version.....	4-45
controlTransaction	Start/End batch processing .....	4-45
② Dedicated method for standard mode .....		4-47
sendText	Send text data .....	4-47
sendTextEx	Send format specified text data.....	4-47
printBarcode	Print barcode .....	4-49
printPDF417	Print PDF417 .....	4-52
printQRcode	Print QR Code .....	4-53
printDataMatrix	Print Data Matrix.....	4-54
printMaxiCode	Print MaxiCode .....	4-55
printGS1DataBarStacked		
	Print GS1 Databar Stacked .....	4-55
printGS1DataBarStackedOmnidirectional		
	Print GS1 Databar Stacked Omni-directional.....	4-56
printGS1DataBarExpandedStacked		
	Print GS1 Databar Expanded Stacked.....	4-56
printAztecCode	Print Aztec Code .....	4-57
cutPaper	Cut paper.....	4-58
feedPosition	Paper form feed.....	4-58
sendBinary	Send binary data .....	4-58
sendDataFile	Send specified file .....	4-59
sendImage	Send image .....	4-60
printPDF	Print PDF page.....	4-61
printLogo	Print logo .....	4-62
printSmartLabelImageData		
	Print label.....	4-62
③ Dedicated method for page mode .....		4-64
enterPageMode	Start page mode .....	4-65
exitPageMode	End page mode .....	4-65
setPageModeArea	Specify print area of page mode .....	4-65
setPageModeDirection	Specify print direction of page mode .....	4-67
setPageModeLineSpacing		
	Specify line spacing of page mode .....	4-67
printPageMode	Print page mode .....	4-67
printPageModeText	Send text data of page mode .....	4-68
printPageModeTextEx	Send format specified text data of page mode...	4-68
printPageModeBarcode	Print barcode of page mode .....	4-69
printPageModePDF417	Print PDF417 of page mode.....	4-73
printPageModeQRcode	Print QR Code of page mode .....	4-74
printPageModeDataMatrix		
	Print Data Matrix of page mode.....	4-75
printPageModeMaxiCode		
	Print MaxiCode of page mode.....	4-75
printPageModeGS1DataBarStacked		
	Print GS1 Databar Stacked of page mode .....	4-76

printPageModeGS1DataBarStackedOmnidirectional	Print GS1 Databar Stacked Omni-directional of .....	
	page mode.....	4-76
printPageModeGS1DataBarExpandedStacked	Print GS1 Databar Expanded Stacked of	
	page mode.....	4-77
printPageModeAztecCode	Print Aztec Code of page mode .....	4-78
sendPageModeBinary	Send binary data of page mode .....	4-79
printPageModelImageFile	Draw Image file of page mode .....	4-79
printPageModeRectangle	Draw rectangle image of page mode .....	4-80
printPageModeLine	Print ruled line of page mode .....	4-81
printPageModeLogo	Print logo of page mode .....	4-83
4.4.2 PrinterEvent Class.....		4-84
(1) Method List.....		4-84
(2) End event constant .....		4-84
(3) Method Details .....		4-84
getEventType	Get end event.....	4-84
4.4.3 PrinterListener Interface.....		4-85
(1) Method List.....		4-85
(2) Method Details .....		4-85
finishEvent	End event of printer search .....	4-85
4.4.4 PrinterInfo Class.....		4-86
(1) Method List.....		4-86
(2) Method Details .....		4-86
getPrinterModelName	Get printer model name.....	4-86
getBluetoothAddress	Get Bluetooth address.....	4-86
getMacAddress	Get MAC address .....	4-86
getIPAddress	Get IP address .....	4-87
getIsBonded	Get pairing status .....	4-87
getDevicePath	Get device path .....	4-87
4.4.5 PrinterIdentifier Class.....		4-88
(1) Method List.....		4-88
(2) Method Details .....		4-88
getPrinterModelID	Get printer model ID .....	4-88
getTypeID	Get type ID .....	4-88
getRomVersionID	Get ROM version ID .....	4-88
getMainFirmwareVersion	Get firmware version (main).....	4-89
getManufacturer	Get manufacturer .....	4-89
getModelName	Get model name .....	4-89
getSerialNumber	Get serial number .....	4-89
getBootFirmwareVersion	Get firmware version (boot).....	4-89
getBootFirmwareChecksum	Get firmware checksum (boot) .....	4-90



getMainFirmwareChecksum	Get firmware checksum (main) .....	4-90
getFirmwareChecksum	Get firmware checksum (main + boot) .....	4-90
4.4.6 PrinterException Class .....		4-91
(1) Method List .....		4-91
(2) Constant List .....		4-91
① Error code .....		4-91
(3) Method Details .....		4-92
PrinterException	Constructor .....	4-92
getErrorCode	Get error codes .....	4-92
4.4.7 CallbackFunctionListener Interface .....		4-93
(1) Method List .....		4-93
(2) Method Details .....		4-93
onStatusChanged	Change event of printer status .....	4-93
4.4.8 BarcodeScannerListener Interface .....		4-94
4.4.9 SmartLabelManager Class .....		4-95
(1) Method List .....		4-95
(2) Method Details .....		4-96
SmartLabelManager	Constructor .....	4-96
selectSmartLabelFile	Specify label file .....	4-96
replaceSmartLabelTextData	Replace text data of label .....	4-96
replaceSmartLabelImageData	Replace image data of label .....	4-97
replaceSmartLabelBarcodeData	Replace barcode data of label .....	4-98

---

## Chapter 5    Sample Program 5-1

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

---

## Appendix A    Character Set A-1

A.1 Codepage Table (Character Code Table) .....	A-1
A.2 International Character Set .....	A-11

---

## Appendix B    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-8
B.1.3 printQRCode, printPageModeQRCode .....	B-9
B.1.4 printDataMatrix, printPageModeDataMatrix .....	B-10
B.1.5 printMaxicode, printPageModeMaxicode .....	B-12

B.1.6	printGS1DataBarStacked, printPageModeGS1DataBarStacked.....	B-13
B.1.7	printGS1DataBarStackedOmnidirectional, printPageModeGS1DataBarStackedOmnidirectional .....	B-14
B.1.8	printGS1DataBarExpandedStacked, printPageModeGS1DataBarExpandedStacked .....	B-15
B.1.9	printAztecCode, printPageModeAztecCode .....	B-16

# Chapter 1

## Product Overview

This chapter describes the product overview of SII print class library.

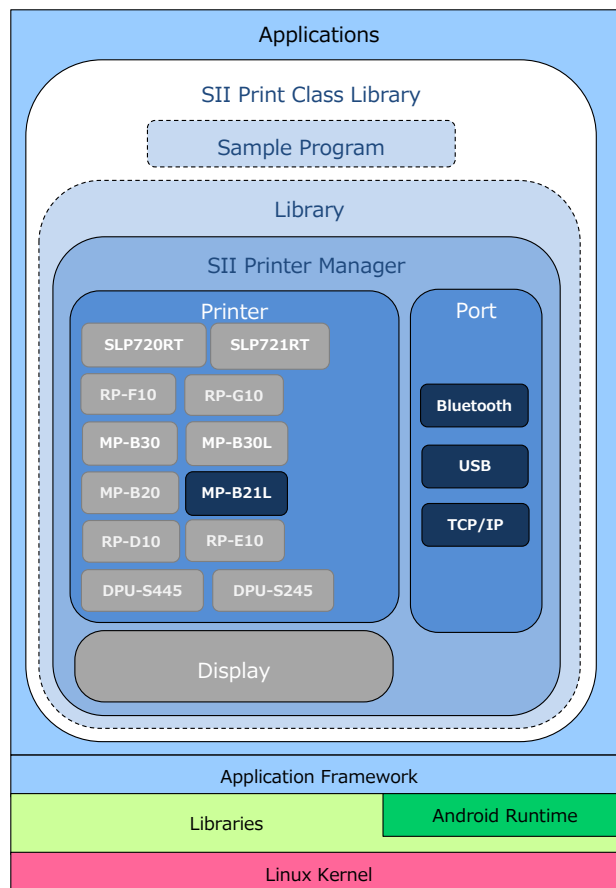
### 1.1 Functions Provided by SII Print Class Library

The SII print class library including the library and the sample program provides the functions to use SII printer MP-B21L series (hereinafter referred to as "printer") in Android applications. Moreover, the SII print class library provides the library sample program in Android Studio project.

### 1.2 SII Print Class Library Overview

#### 1.2.1 SII Print Class Library Configuration

The library and sample program in the SII print class library are indicated with dashed lines in the figure below.



### 1.2.2 Functions Provided by Library

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

The library provides the following functions:

- Connecting to / disconnecting from a printer
- Sending data to a printer (print data and/or printer commands<sup>\*1</sup>)
- Printing barcode and 2-dimensional barcode
- Sending a data file to a printer (print data and/or printer commands<sup>\*1</sup>)
- Getting the printer status
- Aborting the waiting state of a printer
- Getting various responses from a printer
- Bulk registration of print commands
- Registering a printer status call back function
- Searching the printer by Bluetooth or TCP/IP
- Printing a label file
- Replacing object data in a label file
- Outputting a log file

<sup>\*1</sup>: Commands that read responses from the printer are not supported. In order to read responses from the printer, use `getStatus` or `getPrinterResponse`.

<b>(NOTE) MP-B21L does not support the APIs of Display or the barcode scanner.</b>
--

## Chapter 2

### Product Specifications

This chapter describes the product specifications of the library.

#### 2.1 Operating Environment

Operating environment for the library is shown in the following table.

Printer	Model		MP-B21L		
	Communication Interface		Bluetooth	USB	TCP/IP
Android Device	Communication Port		Bluetooth* <sup>1</sup>	USB* <sup>2</sup>	TCP/IP* <sup>3</sup>
	OS	Android 7.0 (API 24)	Supported	Supported	Supported
		Android 7.1 (API 25)			
		Android 8.0 (API 26)			
		Android 8.1 (API 27)			
		Android 9.0 (API 28)			
		Android 10.0 (API 29)			
		Android 11.0 (API 30)			
		Android 12.0 (API 31)			
		Android 12.1 (API 32)			
		Android 13.0 (API 33)			
		Android 14.0 (API 34)			
		Android 15.0 (API 35)			
		Android 16.0 (API 36)			
Supported Language			Japanese, English		

<sup>\*1</sup>: Bluetooth connection needs to be established by SPP (Serial Port Profile).

<sup>\*2</sup>: Android device needs to support USB host function.

<sup>\*3</sup>: The wireless LAN access point that the Android device is connected and the printer need to be connected to the same network.

## 2.2 Printer Settings

Set the memory switches of the printer to [Value] in the following table when using the library.  
The memory switch of the printer can be set in the Android app "SII Printer Utility" on the Google Play.  
See "User's Guide" for details about the memory switches and the factory default settings.

MS	Function	Value
1-1	Interface Selection (Interface)	1 : USB/Wireless* <sup>1</sup> 0 : USB* <sup>2</sup>
1-2	Mark Gap Mode Selection (Mark/Gap Mode)	1 : Disable* <sup>3</sup> 0 : Enable* <sup>4</sup>
1-3 to 1-5	Command System Selection (Command System)	000B : ESC/POS
1-6	Data Discard Selection When Error Occurs (Error Through)	0 : Enable
1-8	Data Discard Selection When Output Buffer Full Occurs (Response Data Discarding)	1 : Disable
2-2	Realtime Command Selection (Realtime Command)	0 : Enable
9-1	Automatic Status Response Selection (Auto Status Back)	0 : Enable
9-2	Initialized Response Selection (Init. Response)	0 : Enable

\*1: Select "USB/Wireless" for Bluetooth connection or TCP/IP Connection.

\*2: Select "USB" for USB connection.

\*3: Select "Disable" when using **cutPaper**.

\*4: Select "Enable" when using **feedPosition**.

## 2.3 Precautions

Communication ports cannot be shared with the printer driver and other libraries when using TCP/IP.

## Chapter 3

### How to Use Library

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

#### 3.1 Android Application Development Environment

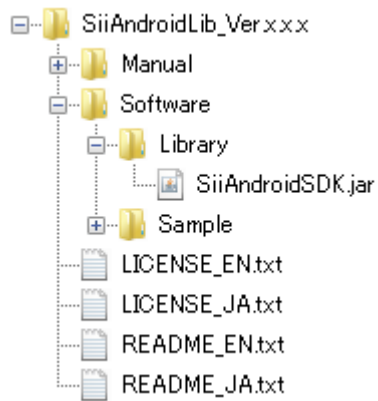
In order to develop Android applications, the following tools are required.  
See each of the following URLs for more details.

- Android Studio  
<https://developer.android.com/studio/index.html>
- USB driver for Windows (When developing in Windows environment)  
<https://developer.android.com/studio/run/oem-usb.html>

The description in and after this chapter is on the premise that the environment where each tool is available is prepared.

## 3.2 Provided Files

The file configuration of the SII print class library 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 Build Library into Android Studio Project

This section describes how to build the SII print class library into Android Studio project.

See "Chapter 5 Sample Program" for the sample program included in the SII print class library.

- (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 SII print class library, the folder is "\Sample\app\libs".
- (2) After adding the library, the view looks like Figure 3-2.

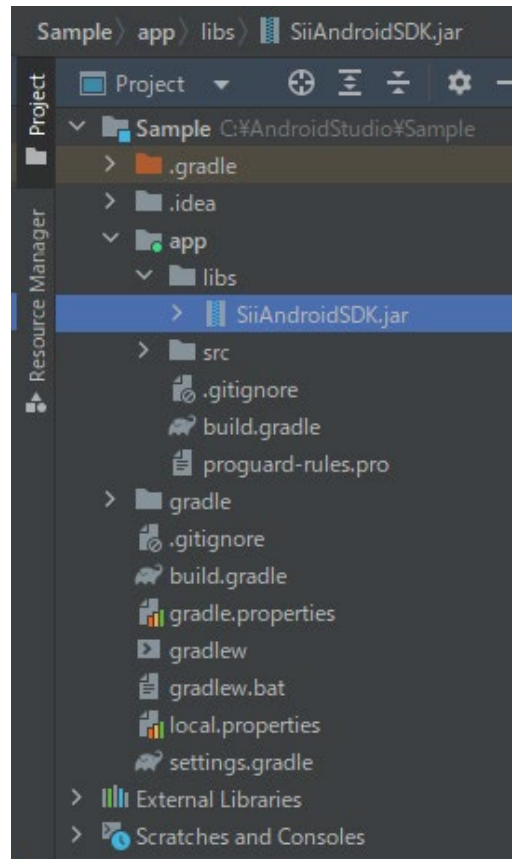


Figure 3-2

- (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 to the beginning of the class file that uses the print class library.  
(Import xxxx according to the function to use.)

```
import com.seiko instruments.sdk.thermalprinter.PrinterManager;
import com.seiko instruments.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, the library function becomes 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)

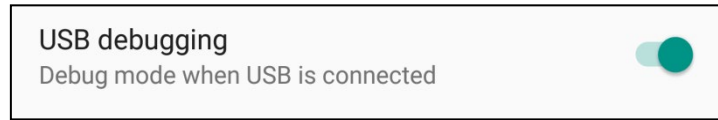


Figure 3-3

### 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

### Functions 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

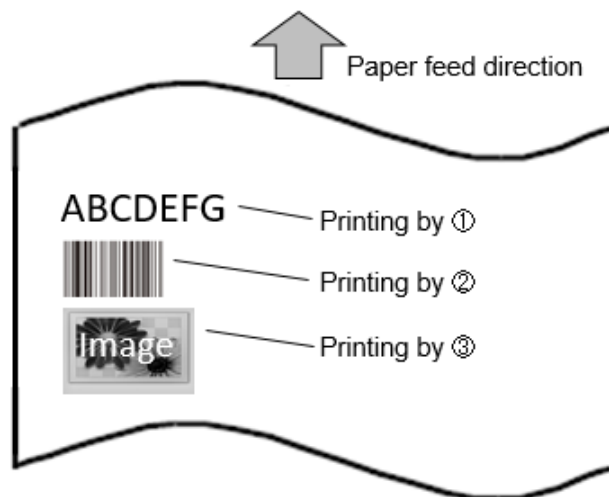
There are two printing modes "Standard mode" and "Page mode" in the library. The "Standard mode" and "Page mode" are described below.

###### (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

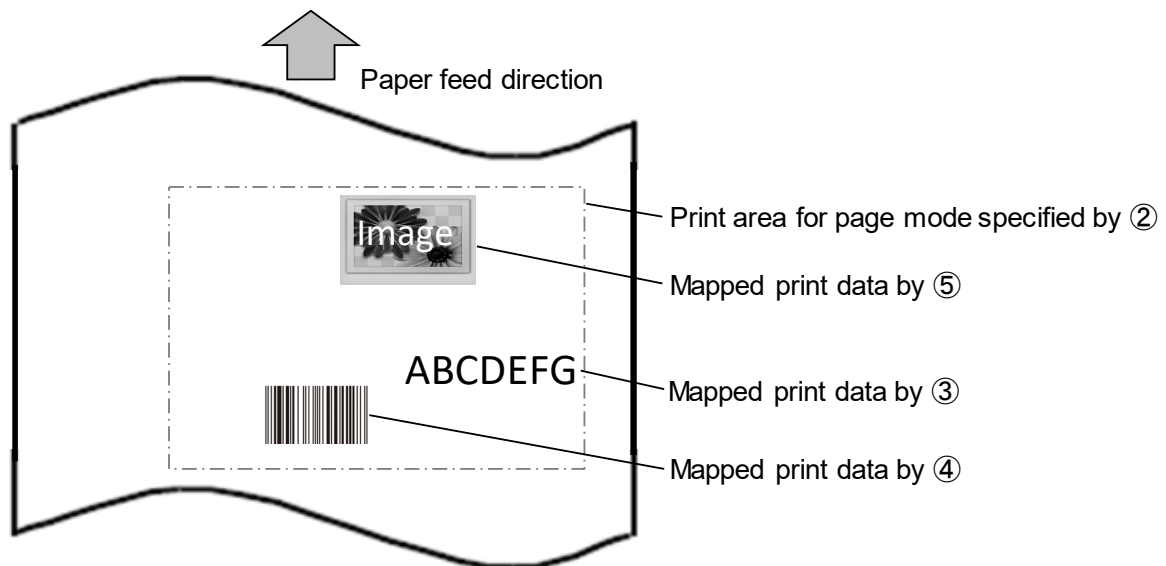
Page mode is the mode to perform the printing on a per-page basis.

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 following.

- 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

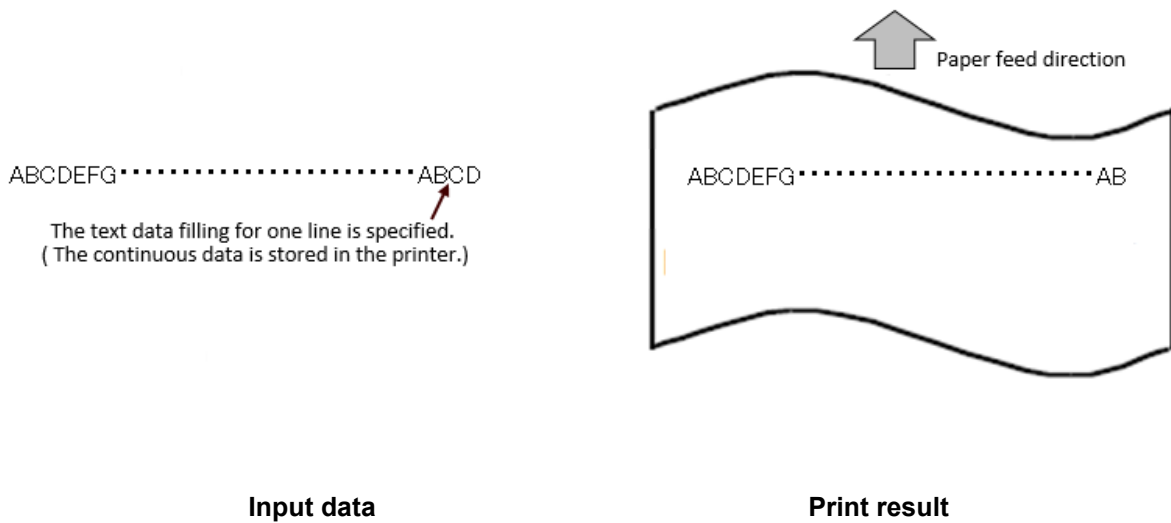
The text data in standard mode is printed each one line.

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

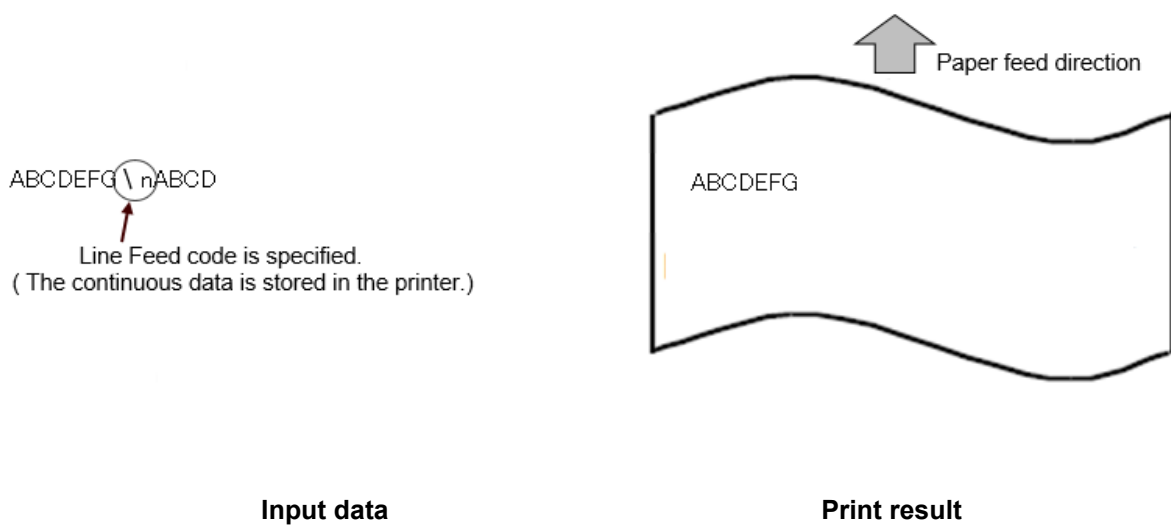
The stored text data is printed by either the following conditions.

- 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.**



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



### 4.1.3 Mapping Position of Print Data in Page Mode

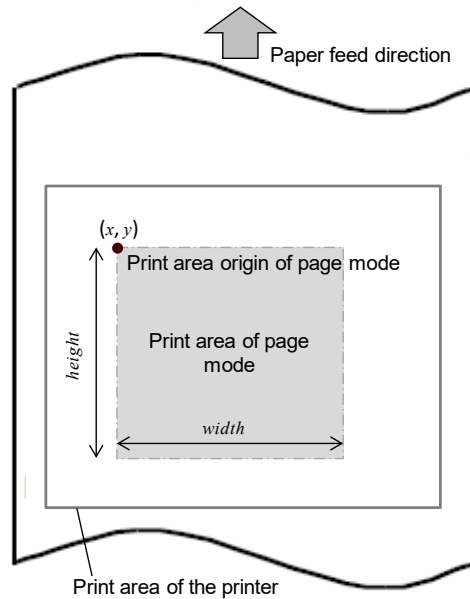
In page mode, the mapping position of print data is determined by print area, print direction, and reference point.

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.

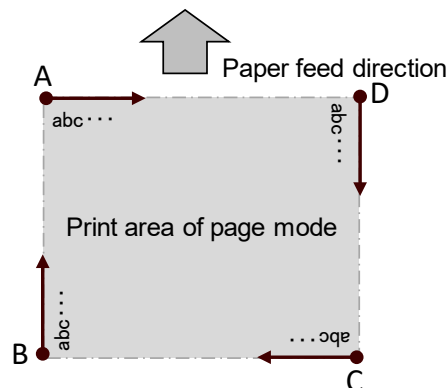


#### (2) Print direction

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.

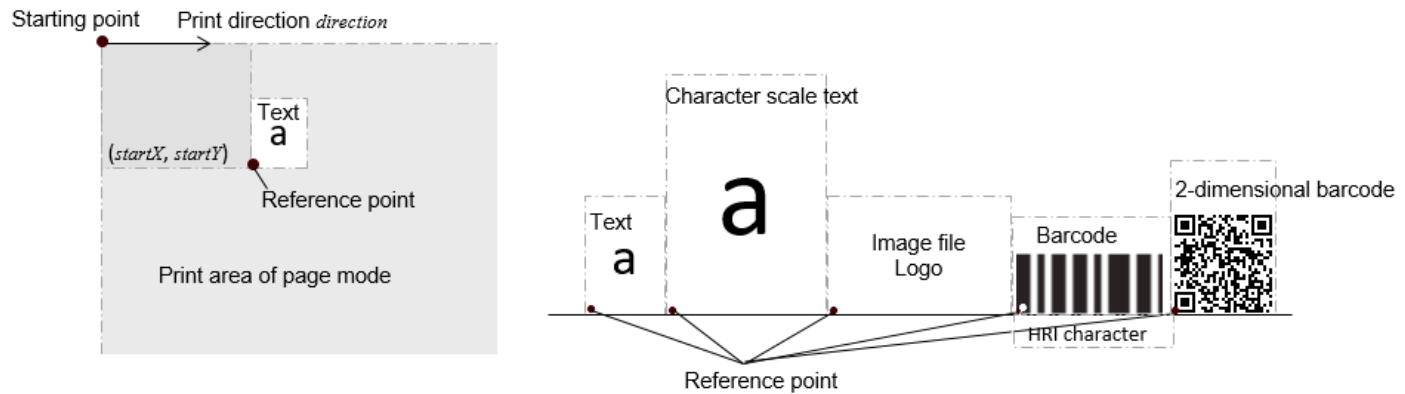
The relation between the print direction and the starting point is shown in the figure below.



- 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.**



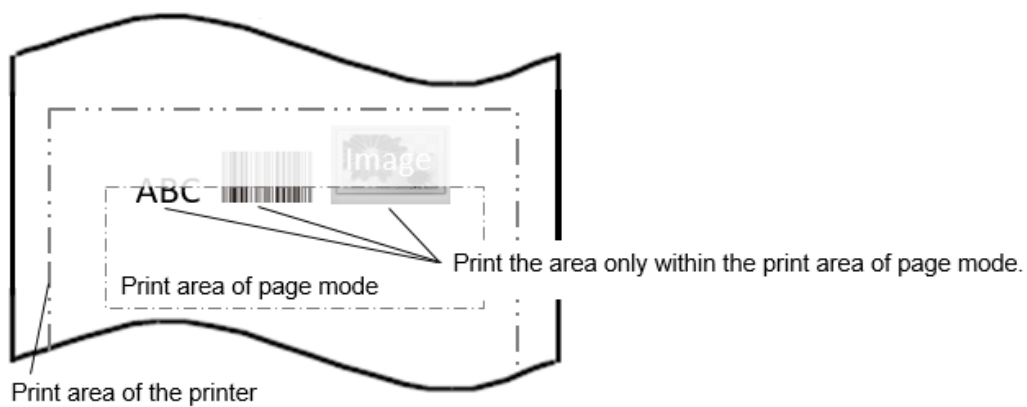
#### 4.1.4 Print Data Process at Out of Print Area of Page Mode

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**

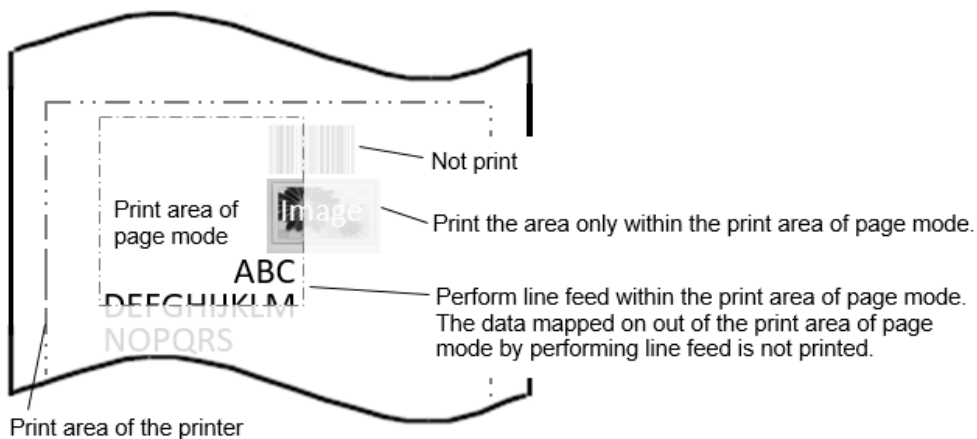
Text	Barcode, 2-dimensional Barcode	Image File, Logo, Rectangle, Ruled Line
ABC		

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



The barcode specified by **BARCODE\_CUSTOMER\_BAR\_CODE\_JP** is not printed when the part of the barcode is out of the print area of page mode.

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



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

## 4.2 Printing Label Function

The label files (\*.sl, \*.slex) created using SII Layout Editor can be printed using the library. It also provides the function to replace text data, image data, or barcode data using the label file and print.

SII Layout Editor is software that can create labels.

In this manual, when describing SII Layout Editor, it is referred to as the "app".

### iOS or Android

By scanning a QR code below with the smartphone, redirected to the store and the app can be installed.

- iOS



- Android



### Windows

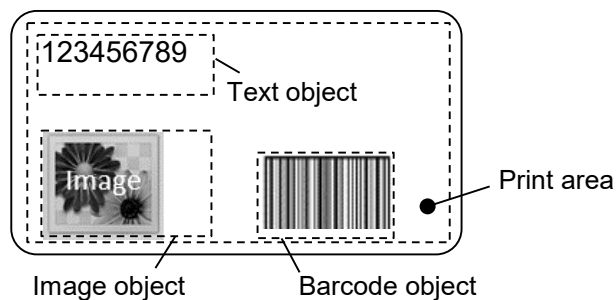
By clicking on the URL below or inputting it into a web browser to go to the store, the app can be installed.

<https://www.microsoft.com/store/apps/9P5G2R1PS76W>

Reference "SII Layout Editor" can also be searched and installed in App Store, Google Play, or Microsoft Store.

### 4.2.1 Structure of Label File

The label file is the file where objects are mapped within the print area for the label.



Example of label file (\*.sl, \*.slex)

(1) Types of objects and support in the library

Objects in the app are shown in the following table.

Object	Description	Supported in Library
Text object	Handle text data	✓
Image object	Handle image data	✓
Barcode object	Handle barcode data	✓
Contact object	Retrieve data from the device contact book	✓
DateTime object	Handle the data of the date and time	✓
Drawing (rectangle) object	Handle the drawn data of a figures (rectangle)	✓
Drawing (circle/oval) object	Handle the drawn data of a figures (circle/oval)	✓
Drawing (line) object	Handle the drawn data of a figures (line)	✓

(2) Precautions for printing the label file using the library

Printing the label file using the app may differ from printing the label file using the library. Verify the performance with your actual device in advance.

Note the following when printing label files using the library.

① All object

- The drawing object mapped outside the print area is not supported.

② Text object

- The "Serialization" is not supported.
- If the font set using the app is not in the library, the text data is printed in the system standard font.

③ Image object

- When the setting of dithering is set to "Burkes" or "Bayer", the "Floyd–Steinberg" is used in the library.

④ Barcode object

- Among the barcodes supported by the app, the following barcodes are supported by the library.
  - CODE39
  - ITF
  - CODE128
  - UPC-A
  - EAN13
  - CODABAR
  - UPC-E
  - EAN8
  - PDF417
  - Data Matrix
  - QR Code
- The "Serialization" is not supported.
- The barcode setting shown in following is not reflected.
  - Ratio of bar width

- The barcode image created using the app and the barcode image created by the library may not become the same barcode image.
  - If the height of the barcode object is specified to be lower than the bar height using the app, the barcode will be reduced to fit within the object in the library and printed.
  - When the security of the PDF417 is set to "-1" using the app, it is fixed to "0" in the library and the object is drawn.
  - If the font set using the app is not in the library, the text data is printed in the system standard font.
- ⑤ Contact object
- If the font set using the app is not in the library, the text data is printed in the system standard font.
- ⑥ DateTime object
- If the font set using the app is not in the library, the text data is printed in the system standard font.
- ⑦ Drawing object
- When "Line Width" using the app is too thin, dashed, long dashed, or double lines may be squished.
  - The drawn position of the drawing object may differ between the app and the library.

#### 4.2.2 Method for Using Label File

The printing method using the label file is described below.

- (1) Print the label file as it is from the library

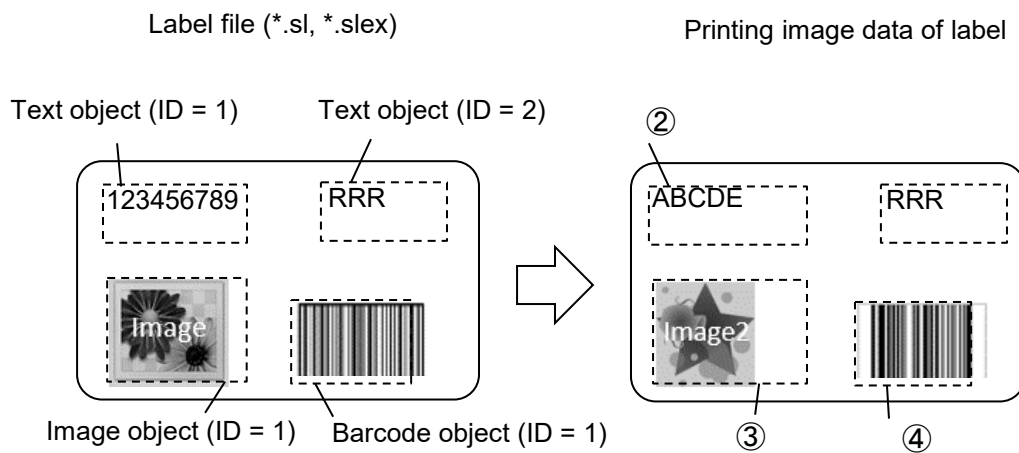
Print command example

- ① Specify label file
- ② Print label

- (2) Replace the object data in the label file and print

Print command example

- ① Specify label file
- ② Replace text data of label (text object ID = 1)
- ③ Replace image data of label (image object ID = 1)
- ④ Replace barcode data of label (barcode object ID = 1)
- ⑤ Print label



## 4.3 Log File Output Function

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

### 4.3.1 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 "4.3.2 Log Output Settings" for details on the settings for x.

### 4.3.2 Log Output Settings

Item	Description	Settings
LOGLEVEL	Log level	0 : Not record the log. 1 : Records an error log when <b>PrinterException</b> 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

### 4.3.3 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.

## 4.4 API Reference

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

Class Name	Description	Supported <sup>*1</sup>
<b>PrinterManager</b>	Provides the APIs used for communication with the printer and for printing. See " <b>4.4.1 PrinterManager Class</b> " for more details.	✓
<b>PrinterEvent</b>	Provides the API that gets the end event when <b>startDiscoveryPrinter</b> is terminated. See " <b>4.4.2 PrinterEvent Class</b> " for more details.	✓
<b>PrinterListener</b>	Interface for getting the end event when <b>startDiscoveryPrinter</b> is terminated. See " <b>4.4.3 PrinterListener Interface</b> " for more details.	✓
<b>PrinterInfo</b>	Stores the printer information found by <b>startDiscoveryPrinter</b> . See " <b>4.4.4 PrinterInfo Class</b> ".	✓
<b>PrinterIdentifier</b>	Stores the printer ID of the printer retrieved by <b>getPrinterResponse</b> . See " <b>4.4.5 PrinterIdentifier Class</b> ".	✓
<b>PrinterException</b>	Exception class that is thrown at API call. See " <b>4.4.6 PrinterException Class</b> " for more details.	✓
<b>CallbackFunctionListener</b>	Interface for getting the change event of printer status. See " <b>4.4.7 CallbackFunctionListener Interface</b> " for more details.	✓
<b>BarcodeScannerListener</b>	Interface for getting barcode scanner connection or barcode scanner disconnection, or received barcode data.	-
<b>SmartLabelManager</b>	Provides the API to specify label files or replace data. See " <b>4.4.9 SmartLabelManager Class</b> " for more details.	✓

\*1: ✓: Supported, -: Not supported

<b>(NOTE) MP-B21L does not support the APIs of Display or the barcode scanner.</b>
--

#### 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.

Method	Description
Common method to standard mode and page mode	The valid methods in standard mode and page mode. See "4.4.1(1) ① 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)② Dedicated method for standard mode" for the methods.
Dedicated method for page mode	The valid methods in page mode. See "4.4.1(1) ③ Dedicated method for page mode" for the methods.

##### ① 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)① Common method to standard mode and page mode" for details of the common methods.

Name	Description	Supported *1
<b>PrinterManager</b>	Constructor	✓
<b>connect</b>	Start communicating with printer (Bluetooth)	✓
<b>connect</b>	Start communicating with printer (USB)	✓
<b>connect</b>	Start communicating with printer (TCP/IP)	✓
<b>disconnect</b>	Stop communicating with printer	✓
<b>setBarcodeScannerListener</b>	Start/End callback of barcode scanner	-
<b>openDrawer</b>	Open cash drawer	-
<b>buzzer</b>	Sound buzzer	-
<b>externalBuzzer</b>	Sound external buzzer	-
<b>getStatus</b>	Get printer status	✓
<b>setCallbackFunctionListener</b>	Start/End callback of printer status change	✓
<b>abort</b>	Abort waiting state of printer	✓
<b>registerLogo</b>	Register logo	✓
<b>unregisterLogo</b>	Delete registered logo	✓
<b>registerStyleSheet</b>	Register style sheet	-
<b>unregisterStyleSheet</b>	Delete registered style sheet	-
<b>resetPrinter</b>	Reset printer	✓
<b>getPrinterResponse</b>	Get various responses from printer	✓
<b>startDiscoveryPrinter</b>	Start printer search (Bluetooth)	✓
<b>startDiscoveryPrinter</b>	Start printer search (USB)	✓



Name	Description	Supported <sup>*1</sup>
<b>startDiscoveryPrinter</b>	Start printer search (TCP/IP)	✓
<b>cancelDiscoveryPrinter</b>	Cancel printer search	✓
<b>getFoundPrinter</b>	Get found printer information	✓
<b>getSendTimeout</b>	Get send timeout period	✓
<b>setSendTimeout</b>	Set send timeout period	✓
<b>getReceiveTimeout</b>	Get receive timeout period	✓
<b>setReceiveTimeout</b>	Set receive timeout period	✓
<b>getInternationalCharacter</b>	Get international character set	✓
<b>setInternationalCharacter</b>	Set international character set	✓
<b>getCodePage</b>	Get codepage	✓
<b>setCodePage</b>	Set codepage	✓
<b>getPrinterModel</b>	Get printer model	✓
<b>getPortType</b>	Get connecting port type	✓
<b>isConnect</b>	Verify connection state with printer	✓
<b>getSocketKeepingTime</b>	Get socket keeping time	✓
<b>setSocketKeepingTime</b>	Set socket keeping time	✓
<b>getPrintSmartLabelMode</b>	Get paper when printing label file	✓
<b>setPrintSmartLabelMode</b>	Set paper when printing label file	✓
<b>getVersion</b>	Get SDK version	✓
<b>controlTransaction</b>	Start/End batch processing	✓

\*1: ✓: Supported, -: Not supported

## ② 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)② Dedicated method for standard mode" for details of the dedicated methods.

Name	Description	Supported <sup>*1</sup>
<b>sendText</b>	Send text data	✓
<b>sendTextEx</b>	Send format specified text data	✓
<b>printBarcode</b>	Print barcode	✓
<b>printPDF417</b>	Print PDF417	✓
<b>printQRcode</b>	Print QR Code	✓
<b>printDataMatrix</b>	Print Data Matrix	✓
<b>printMaxiCode</b>	Print MaxiCode	✓
<b>printGS1DataBarStacked</b>	Print GS1 Databar Stacked	✓
<b>printGS1DataBarStackedOmniDirectional</b>	Print GS1 Databar Stacked Omni-directional	✓
<b>printGS1DataBarExpandedStacked</b>	Print GS1 Databar Expanded Stacked	✓
<b>printAztecCode</b>	Print Aztec Code	✓
<b>cutPaper</b>	Cut paper <sup>*2</sup>	✓
<b>feedPosition</b>	Paper form feed	✓

Name	Description	Supported <sup>*1</sup>
<b>sendBinary</b>	Send binary data	✓
<b>sendDataFile</b>	Send specified file	✓
<b>sendImage</b>	Send image	✓
<b>printPDF</b>	Print PDF page	✓
<b>printLogo</b>	Print logo	✓
<b>printSmartLabelImageData</b>	Print label	✓

\*1: ✓: Supported, -: Not supported

\*2: Only the paper feed operation to the paper cut position is performed.

### ③ 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)③ Dedicated method for page mode" for details of the dedicated methods.

Name	Description	Supported <sup>*1</sup>
<b>enterPageMode</b>	Start page mode	✓
<b>exitPageMode</b>	End page mode	✓
<b>setPageModeArea</b>	Specify print area of page mode	✓
<b>setPageModeDirection</b>	Specify print direction of page mode	✓
<b>setPageModeLineSpacing</b>	Specify line spacing of page mode	✓
<b>printPageMode</b>	Print page mode	✓
<b>printPageModeText</b>	Send text data of page mode	✓
<b>printPageModeTextEx</b>	Send format specified text data of page mode	✓
<b>printPageModeBarcode</b>	Print barcode of page mode	✓
<b>printPageModePDF417</b>	Print PDF417 of page mode	✓
<b>printPageModeQRcode</b>	Print QR Code of page mode	✓
<b>printPageModeDataMatrix</b>	Print Data Matrix of page mode	✓
<b>printPageModeMaxiCode</b>	Print MaxiCode of page mode	✓
<b>printPageModeGS1DataBarStacked</b>	Print GS1 Databar Stacked of page mode	✓
<b>printPageModeGS1DataBarStackedOmnidirectional</b>	Print GS1 Databar Stacked Omnidirectional of page mode	✓
<b>printPageModeGS1DataBarExpandedStacked</b>	Print GS1 Databar Expanded Stacked of page mode	✓
<b>printPageModeAztecCode</b>	Print Aztec Code of page mode	✓
<b>sendPageModeBinary</b>	Send binary data of page mode	✓
<b>printPageModeImageFile</b>	Draw image file of page mode	✓
<b>printPageModeRectangle</b>	Draw rectangle image of page mode	✓
<b>printPageModeLine</b>	Print ruled line of page mode	✓
<b>printPageModeLogo</b>	Print logo of page mode	✓

\*1: ✓: Supported, -: Not supported

(2) Constant List

① Printer model

Constants used for starting communicating with a printer and getting the printer model are shown in the following table.

Constant Name	Description	Value
PRINTER_MODEL_MP_B21L	MP-B21L	307
PRINTER_MODEL_DEFAULT	Default of printer model	284

② Response type

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

Constant Name	Description	Value
PRINTER_RESPONSE_REQUEST	Request of execution response	0
PRINTER_RESPONSE_USER_AREA	Send remaining capacity of user area	1
PRINTER_RESPONSE_ARRANGE_USER_AREA	Send remaining capacity of user area after defragment	2
PRINTER_RESPONSE_NV_GRAPHICS	Send NV graphics memory capacity	3
PRINTER_RESPONSE_KEY_CODE	Send key code list of defined NV graphics	4
PRINTER_RESPONSE_BATTERY_STATUS	Battery remaining capacity level	5
PRINTER_RESPONSE_MAINTENANCE_COUNTER	Send maintenance counter	7
PRINTER_RESPONSE_PRINTER_IDENTIFIER	Send printer ID	8

③ Battery remaining capacity level

Constants of the battery remaining capacity level retrieved from a printer are shown in the following table.

Constant Name	Description	Value
BATTERY_STATUS_FULL	Battery remaining capacity: approx. 80%	0
BATTERY_STATUS_MIDDLE	Battery remaining capacity: approx. 40%	1
BATTERY_STATUS_LOW	Battery remaining capacity: approx. 10%	2
BATTERY_STATUS_EMPTY	No battery	3

④ International character set

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

Constant Name	Description	Value
COUNTRY_USA	USA	0
COUNTRY_FRANCE	France	1
COUNTRY_GERMANY	Germany	2
COUNTRY_ENGLAND	United Kingdom	3
COUNTRY_DENMARK_1	Denmark I	4
COUNTRY_SWEDEN	Sweden	5
COUNTRY_ITALY	Italy	6
COUNTRY_SPAIN	Spain I	7
COUNTRY_JAPAN	Japan	8
COUNTRY_NORWAY	Norway	9
COUNTRY_DENMARK_2	Denmark II	10
COUNTRY_SPAIN_2	Spain II	11
COUNTRY_LATIN_AMERICA	Latin America	12
COUNTRY_ARABIA	Arabia	17

⑤ Codepage

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

Constant Name	Description	Value
CODE_PAGE_437	USA, Standard Europe (Code Page 437)	0
CODE_PAGE_KATAKANA	Katakana	1
CODE_PAGE_850	Multilingual (Code Page 850)	2
CODE_PAGE_860	Portuguese (Code Page 860)	3
CODE_PAGE_863	Canadian-French (Code Page 863)	4
CODE_PAGE_865	Nordic (Code Page 865)	5
CODE_PAGE_857	Turkish (Code Page 857)	13
CODE_PAGE_737	Greek (Code Page 737)	14
CODE_PAGE_1252	Latin (Code Page 1252)	16
CODE_PAGE_866	Russian (Code Page 866)	17
CODE_PAGE_852	Eastern Europe (Code Page 852)	18
CODE_PAGE_858	Euro (Code Page 858)	19
CODE_PAGE_855	Cyrillic (Code Page 855)	34
CODE_PAGE_864 <sup>*1*2</sup>	Arabic (Code Page 864)	37
CODE_PAGE_1250	Central European (Code Page 1250)	45
CODE_PAGE_1251	Cyrillic (Code Page 1251)	46
CODE_PAGE_1253	Greek (Code Page 1253)	47
CODE_PAGE_1254	Turkish (Code Page 1254)	48

\*1: 20ACh of the Unicode cannot be printed.

\*2: Font B cannot be printed.

## ⑥ Port type

Constants used for starting communicating with a printer and getting the connecting port type are shown in the following table.

Constant Name	Description	Value
PRINTER_TYPE_BLUETOOTH	Bluetooth	0
PRINTER_TYPE_USB	USB	1
PRINTER_TYPE_TCP	TCP/IP	2

## ⑦ Paper selection with or without mark when printing label file

Constants used for selecting the paper when printing label file.

Constant Name	Description	Value
PRINTSMARTLABEL_MODE_MARK	Marked paper	0
PRINTSMARTLABEL_MODE_NONEMARK	Paper without mark	1

## ⑧ Barcode or PDF417

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

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

## (3) Enumerated Constant List

### ① Dithering (Dithering)

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

Constant Name	Description
DITHERING_DISABLE	Dithering is disabled
DITHERING_ERRORDIFFUSION	Dithering is enabled

② Batch processing selection (TransactionFunction)

Constants of enumerated type used for batch processing selection are shown in the following table.

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

③ Bold print (CharacterBold)

Constants of enumerated type used for bold print are shown in the following table.

Constant Name	Description
BOLD_CANCEL	Cancel bold print
BOLD	Specify bold print

④ Underline (CharacterUnderline)

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

Constant Name	Description
UNDERLINE_CANCEL	Cancel underline print
UNDERLINE_1	Specify 1-dot width underline print
UNDERLINE_2	Specify 2-dot width underline print

⑤ Reverse print (CharacterReverse)

Constants of enumerated type used for reverse print are shown in the following table.

Constant Name	Description
REVERSE_CANCEL	Cancel reverse print
REVERSE	Specify reverse print

⑥ Inversion print (CharacterInversion)

Constants of enumerated type 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.

Constant Name	Description
INVERSION_CANCEL	Cancel inversion print
INVERSION	Specify inversion print

⑦ Character font (CharacterFont)

Constants of enumerated type used for character font are shown in the following table.

Constant Name	Description
FONT_A	Font A (24 × 12)
FONT_B	Font B (16 × 8)

⑧ Character scale (CharacterScale)

Constants of enumerated type used for character scale are shown in the following table.

Constant Name	Description
VERTICAL_1_HORIZONTAL_1	Height × 1 and width × 1
VERTICAL_1_HORIZONTAL_2	Height × 1 and width × 2
VERTICAL_1_HORIZONTAL_3	Height × 1 and width × 3
VERTICAL_1_HORIZONTAL_4	Height × 1 and width × 4
VERTICAL_2_HORIZONTAL_1	Height × 2 and width × 1
VERTICAL_2_HORIZONTAL_2	Height × 2 and width × 2
VERTICAL_2_HORIZONTAL_3	Height × 2 and width × 3
VERTICAL_2_HORIZONTAL_4	Height × 2 and width × 4
VERTICAL_2_HORIZONTAL_6	Height × 2 and width × 6
VERTICAL_3_HORIZONTAL_1	Height × 3 and width × 1
VERTICAL_3_HORIZONTAL_2	Height × 3 and width × 2
VERTICAL_3_HORIZONTAL_3	Height × 3 and width × 3
VERTICAL_3_HORIZONTAL_4	Height × 3 and width × 4
VERTICAL_4_HORIZONTAL_1	Height × 4 and width × 1
VERTICAL_4_HORIZONTAL_2	Height × 4 and width × 2
VERTICAL_4_HORIZONTAL_3	Height × 4 and width × 3
VERTICAL_4_HORIZONTAL_4	Height × 4 and width × 4
VERTICAL_4_HORIZONTAL_6	Height × 4 and width × 6
VERTICAL_4_HORIZONTAL_8	Height × 4 and width × 8
VERTICAL_6_HORIZONTAL_2	Height × 6 and width × 2
VERTICAL_6_HORIZONTAL_4	Height × 6 and width × 4
VERTICAL_6_HORIZONTAL_6	Height × 6 and width × 6
VERTICAL_6_HORIZONTAL_8	Height × 6 and width × 8
VERTICAL_8_HORIZONTAL_4	Height × 8 and width × 4
VERTICAL_8_HORIZONTAL_6	Height × 8 and width × 6
VERTICAL_8_HORIZONTAL_8	Height × 8 and width × 8

⑨ Alignment (PrintAlignment)

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

Constant Name	Description
ALIGNMENT_LEFT	Aligned left
ALIGNMENT_CENTER	Centered
ALIGNMENT_RIGHT	Aligned right

⑩ Pending data output specifying (OutputPendingData)

Constants of enumerated type used for pending data output specifying are shown in the following table.

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

⑪ Barcode symbol (BarcodeSymbol)

Constants of enumerated type used for barcode symbol are shown in the following table.

Constant Name	Description	Syntax <sup>*1</sup>
BARCODE_SYMBOL_UPC_A	UPC-A	(a)
BARCODE_SYMBOL_UPC_E	UPC-E	(a)
BARCODE_SYMBOL_EAN13	EAN13	(a)
BARCODE_SYMBOL_JAN13	JAN13	(a)
BARCODE_SYMBOL_EAN8	EAN8	(a)
BARCODE_SYMBOL_JAN8	JAN8	(a)
BARCODE_SYMBOL_CODE39	CODE39	(a), (b)
BARCODE_SYMBOL_CODE93	CODE93	(c)
BARCODE_SYMBOL_CODE128	CODE128	(c)
BARCODE_SYMBOL_ITF	ITF	(a), (b)
BARCODE_SYMBOL_CODABAR	CODABAR	(a), (b)
BARCODE_SYMBOL_EAN13_ADDON	EAN13 add-on	(a)
BARCODE_SYMBOL_JAN13_ADDON	JAN13 add-on	(a)
BARCODE_SYMBOL_CUSTOMER_BAR_CODE_JP	Customer Bar Code_JP	(d)
BARCODE_SYMBOL_GS1_OMNI_DIRECTIONAL	GS1 Databar Omni-directional	(a)
BARCODE_SYMBOL_GS1_TRUNCATED	GS1 Databar Truncated	(a)
BARCODE_SYMBOL_GS1_LIMITED	GS1 Databar Limited	(a)
BARCODE_SYMBOL_GS1_EXPANDED	GS1 Databar Expanded	(a)

\*1: See printBarcode or printPageModeBarcode for details of syntax.



⑫ Module size (ModuleSize)

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

Constant Name	Description	Method to Use
<b>BARCODE_MODULE_WIDTH_2</b>	Fine element 2 dots Module width 0.250 mm	<ul style="list-style-type: none"> <li>● <b>printBarcode</b></li> <li>● <b>printPageModeBarcode</b></li> </ul>
<b>BARCODE_MODULE_WIDTH_3</b>	Fine element 3 dots Module width 0.375 mm	
<b>BARCODE_MODULE_WIDTH_4</b>	Fine element 4 dots Module width 0.500 mm	
<b>BARCODE_MODULE_WIDTH_5</b>	Fine element 5 dots Module width 0.625 mm	
<b>BARCODE_MODULE_WIDTH_6</b>	Fine element 6 dots Module width 0.750 mm	
<b>CUSTOMERBARCODEJP_MODULE_4</b>	Customer Bar Code_JP size 4 points*1	
<b>CUSTOMERBARCODEJP_MODULE_5</b>	Customer Bar Code_JP size 5 points*1	
<b>CUSTOMERBARCODEJP_MODULE_6</b>	Customer Bar Code_JP size 6 points*1	
<b>CUSTOMERBARCODEJP_MODULE_7</b>	Customer Bar Code_JP size 7 points*1	
<b>CUSTOMERBARCODEJP_MODULE_8</b>	Customer Bar Code_JP size 8 points*1	
<b>CUSTOMERBARCODEJP_MODULE_9</b>	Customer Bar Code_JP size 9 points*1	
<b>CUSTOMERBARCODEJP_MODULE_10</b>	Customer Bar Code_JP size 10 points*1	

\*1: The structure of the bar according to the points of the Customer Bar Code\_JP is as follows.

Customer Bar Code_JP Size	Structure (mm)				
	Long Bar	Timing Bar	Bar Pitch	Bar Width	Bar Space
<b>CUSTOMERBARCODEJP_MODULE_4</b>	1.500 (12 dots)	0.500 (4 dots)	0.500 (4 dots)	0.250 (2 dots)	0.250 (2 dots)
<b>CUSTOMERBARCODEJP_MODULE_5</b>	1.875 (15 dots)	0.625 (5 dots)	0.625 (5 dots)	0.375 (3 dots)	0.250 (2 dots)
<b>CUSTOMERBARCODEJP_MODULE_6</b>	2.250 (18 dots)	0.750 (6 dots)	0.750 (6 dots)	0.375 (3 dots)	0.375 (3 dots)
<b>CUSTOMERBARCODEJP_MODULE_7</b>	2.625 (21 dots)	0.875 (7 dots)	0.875 (7 dots)	0.500 (4 dots)	0.375 (3 dots)
<b>CUSTOMERBARCODEJP_MODULE_8</b>	3.000 (24 dots)	1.000 (8 dots)	1.000 (8 dots)	0.500 (4 dots)	0.500 (4 dots)
<b>CUSTOMERBARCODEJP_MODULE_9</b>	3.375 (27 dots)	1.125 (9 dots)	1.125 (9 dots)	0.625 (5 dots)	0.500 (4 dots)
<b>CUSTOMERBARCODEJP_MODULE_10</b>	3.750 (30 dots)	1.250 (10 dots)	1.250 (10 dots)	0.625 (5 dots)	0.625 (5 dots)

Constant Name	Description	Method to Use
PDF417_MODULE_WIDTH_2	Nominal fine element width 2 dots	<ul style="list-style-type: none"> <li>● <b>printPDF417</b></li> <li>● <b>printPageModePDF417</b></li> </ul>
PDF417_MODULE_WIDTH_3	Nominal fine element width 3 dots	
PDF417_MODULE_WIDTH_4	Nominal fine element width 4 dots	
PDF417_MODULE_WIDTH_5	Nominal fine element width 5 dots	
PDF417_MODULE_WIDTH_6	Nominal fine element width 6 dots	
PDF417_MODULE_WIDTH_7	Nominal fine element width 7 dots	
PDF417_MODULE_WIDTH_8	Nominal fine element width 8 dots	
QR_MODULE_SIZE_2	2 dots	<ul style="list-style-type: none"> <li>● <b>printQRcode</b></li> <li>● <b>printPageModeQRcode</b></li> </ul>
QR_MODULE_SIZE_3	3 dots	
QR_MODULE_SIZE_4	4 dots	
QR_MODULE_SIZE_5	5 dots	
QR_MODULE_SIZE_6	6 dots	
QR_MODULE_SIZE_7	7 dots	
QR_MODULE_SIZE_8	8 dots	
QR_MODULE_SIZE_9	9 dots	
QR_MODULE_SIZE_10	10 dots	
QR_MODULE_SIZE_11	11 dots	
QR_MODULE_SIZE_12	12 dots	
QR_MODULE_SIZE_13	13 dots	
QR_MODULE_SIZE_14	14 dots	
QR_MODULE_SIZE_15	15 dots	
QR_MODULE_SIZE_16	16 dots	
DATAMATRIX_MODULE_SIZE_2	2 dots	<ul style="list-style-type: none"> <li>● <b>printDataMatrix</b></li> <li>● <b>printPageModeDataMatrix</b></li> </ul>
DATAMATRIX_MODULE_SIZE_3	3 dots	
DATAMATRIX_MODULE_SIZE_4	4 dots	
DATAMATRIX_MODULE_SIZE_5	5 dots	
DATAMATRIX_MODULE_SIZE_6	6 dots	
DATAMATRIX_MODULE_SIZE_7	7 dots	
DATAMATRIX_MODULE_SIZE_8	8 dots	
DATAMATRIX_MODULE_SIZE_9	9 dots	
DATAMATRIX_MODULE_SIZE_10	10 dots	
DATAMATRIX_MODULE_SIZE_11	11 dots	
DATAMATRIX_MODULE_SIZE_12	12 dots	
DATAMATRIX_MODULE_SIZE_13	13 dots	
DATAMATRIX_MODULE_SIZE_14	14 dots	
DATAMATRIX_MODULE_SIZE_15	15 dots	
DATAMATRIX_MODULE_SIZE_16	16 dots	

Constant Name	Description	Method to Use
GS1DATABAR_MODULE_SIZE_2	2 dots	<ul style="list-style-type: none"> <li>● <b>printGS1DataBarStacked</b></li> <li>● <b>printGS1DataBarStackedOmni</b> <b>directional</b></li> <li>● <b>printGS1DataBarExpandedSt</b> <b>acked</b></li> <li>● <b>printPageModeGS1DataBarS</b> <b>tacked</b></li> <li>● <b>printPageModeGS1DataBarS</b> <b>tackedOmni</b> <b>directional</b></li> <li>● <b>printPageModeGS1DataBarE</b> <b>xpanded</b> <b>Stacked</b></li> </ul>
GS1DATABAR_MODULE_SIZE_3	3 dots	
GS1DATABAR_MODULE_SIZE_4	4 dots	
GS1DATABAR_MODULE_SIZE_5	5 dots	
GS1DATABAR_MODULE_SIZE_6	6 dots	
GS1DATABAR_MODULE_SIZE_7	7 dots	
GS1DATABAR_MODULE_SIZE_8	8 dots	
GS1DATABAR_MODULE_SIZE_9	9 dots	
GS1DATABAR_MODULE_SIZE_10	10 dots	
GS1DATABAR_MODULE_SIZE_11	11 dots	
GS1DATABAR_MODULE_SIZE_12	12 dots	
GS1DATABAR_MODULE_SIZE_13	13 dots	
GS1DATABAR_MODULE_SIZE_14	14 dots	
GS1DATABAR_MODULE_SIZE_15	15 dots	
GS1DATABAR_MODULE_SIZE_16	16 dots	
AZTECCODE_MODULE_SIZE_2	2 dots	<ul style="list-style-type: none"> <li>● <b>printAztecCode</b></li> <li>● <b>printPageModeAztecCode</b></li> </ul>
AZTECCODE_MODULE_SIZE_3	3 dots	
AZTECCODE_MODULE_SIZE_4	4 dots	
AZTECCODE_MODULE_SIZE_5	5 dots	
AZTECCODE_MODULE_SIZE_6	6 dots	
AZTECCODE_MODULE_SIZE_7	7 dots	
AZTECCODE_MODULE_SIZE_8	8 dots	
AZTECCODE_MODULE_SIZE_9	9 dots	
AZTECCODE_MODULE_SIZE_10	10 dots	
AZTECCODE_MODULE_SIZE_11	11 dots	
AZTECCODE_MODULE_SIZE_12	12 dots	
AZTECCODE_MODULE_SIZE_13	13 dots	
AZTECCODE_MODULE_SIZE_14	14 dots	
AZTECCODE_MODULE_SIZE_15	15 dots	
AZTECCODE_MODULE_SIZE_16	16 dots	

⑬ HRI character print position (HriPosition)

Constants of enumerated type used for HRI character print position are shown in the following table.

Constant Name	Description
HRI_NONE	Not printed
HRI_POSITION_ABOVE	Above barcode
HRI_POSITION_BELOW	Below barcode
HRI_POSITION_ABOVE_BELOW	Above and below barcode (both)

⑭ N:W ratio (NwRatio)

Constants of enumerated type used for N:W ratio are shown in the following table.

Constant Name	Description
NWRATIO_1TO2	1:2
NWRATIO_1TO2_5	1:2.5
NWRATIO_1TO3	1:3

⑮ Error correction level (ErrorCorrection)

Constants of enumerated type used for error correction level are shown in the following table.

Constant Name	Description	Method to Use
PDF417_ERROR_CORRECTION_0	Error correction level 0	<ul style="list-style-type: none"> <li>● printPDF417</li> <li>● printPageModePDF417</li> </ul>
PDF417_ERROR_CORRECTION_1	Error correction level 1	
PDF417_ERROR_CORRECTION_2	Error correction level 2	
PDF417_ERROR_CORRECTION_3	Error correction level 3	
PDF417_ERROR_CORRECTION_4	Error correction level 4	
PDF417_ERROR_CORRECTION_5	Error correction level 5	
PDF417_ERROR_CORRECTION_6	Error correction level 6	
PDF417_ERROR_CORRECTION_7	Error correction level 7	
PDF417_ERROR_CORRECTION_8	Error correction level 8	
QR_ERROR_CORRECTION_L	Error correction level L	<ul style="list-style-type: none"> <li>● printQRcode</li> <li>● printPageModeQRcode</li> </ul>
QR_ERROR_CORRECTION_M	Error correction level M	
QR_ERROR_CORRECTION_H	Error correction level H	
QR_ERROR_CORRECTION_Q	Error correction level Q	

⑯ PDF417 symbol (Pdf417Symbol)

Constants of enumerated type used for PDF417 symbol are shown in the following table.

Constant Name	Description
PDF417_STANDARD	PDF417
PDF417_COMPACT	Compact PDF417

⑰ QR Code Model (QrModel)

Constants of enumerated type used for QR Code Model are shown in the following table.

Constant Name	Description
QR_MODEL_1	QR Code Model 1
QR_MODEL_2	QR Code Model 2

⑱ Data Matrix module (DataMatrixModule)

Constants of enumerated type used for Data Matrix module are shown in the following table.

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

⑲ MaxiCode Mode (MaxiCodeMode)

Constants of enumerated type used for MaxiCode Mode are shown in the following table.

Constant Name	Description
<b>MAXI_CODE_2</b>	Mode2
<b>MAXI_CODE_3</b>	Mode3
<b>MAXI_CODE_4</b>	Mode4
<b>MAXI_CODE_5</b>	Mode5

⑳ Aztec Symbol (AztecSymbol)

Constants of enumerated type used for Aztec Symbol are shown in the following table.

Constant Name	Description
<b>AZTECCODE_FULLRANGE</b>	Full-range
<b>AZTECCODE_COMPACT</b>	Compact

㉑ Cutting method (CuttingMethod)

Constants of enumerated type used for cutting method are shown in the following table.

Constant Name	Description
<b>CUT_FULL</b>	No cut
<b>CUT_PARTIAL</b>	Paper feed operation to the paper cut position
<b>CUT_NONE</b> *1	No cut

\*1: Supported only by `printPageMode`

㉒ Form feed position (FeedPosition)

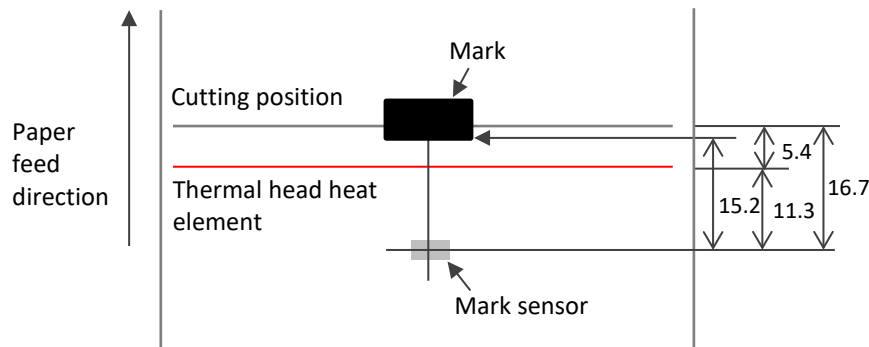
Constants of enumerated type used for form feed position of marked paper or label are shown in the following table.

Constant Name	Description
<b>FEED_CUTTER</b>	After detecting the mark or the gap, feeds the paper to the cutting position. The paper feed length is the length of the memory switches MS 21 to 22 (Mark Detection Cut Position Correction) of the printer. The default of the paper feed is 122 dots (15.2 mm).
<b>FEED_NEXT_TOF</b>	After detecting the next mark or the gap, feeds the paper to the printing position. The paper feed length is the length of the memory switches MS 26 to 27 (Mark Detection Print Position Correction) of the printer. The default of the paper feed is 122 dots (15.2 mm).

Reference See "MP-B21L SERIES Thermal Printer USER'S GUIDE" for details of the memory switch of the printer.  
The memory switch of the printer can be changed in the Android app "SII Printer Utility" on the Google Play.

The relation between the sensor position and the defaults of the memory switches MS 21 to 22 (Mark Detection Cut Position Correction) of the printer and the memory switches MS 26 to 27 (Mark Detection Print Position Correction) of the printer are shown in the following figure.

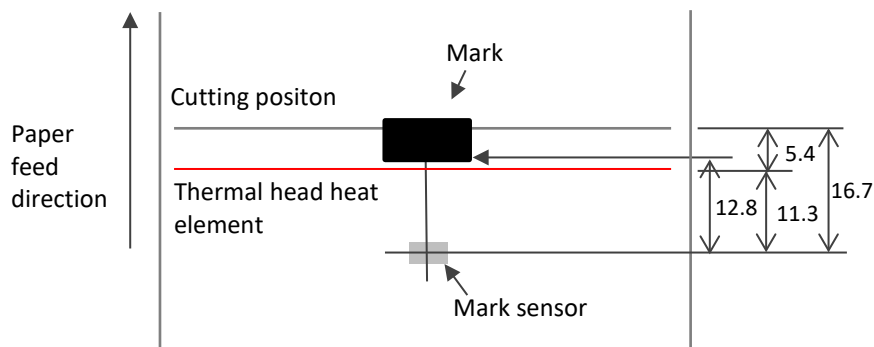
When the memory switch of the printer is set to the default, the cutting position of **FEED\_CUTTER** and the next printing position of **FEED\_NEXT\_TOF** is the same paper feed length.



Unit : mm

To set to shorter the next print position for the mark to save paper, set the values of the memory switches MS 26 to 27 (Mark Detection Print Position Correction) of the printer shorter.

As an example, the relation of the sensor position when the values of the memory switches MS 26 to 27 of the printer are set to 103 dots (12.8 mm) and paper form feed is performed with specifying **FEED\_NEXT\_TOF** is shown in the following figure.



Unit : mm

Note When using label, set the values of the memory switches MS 26 to 27 (Mark Detection Print Position Correction) of the printer so that the print position can be inside the label.

②③ Image rotation direction (Rotate)

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

Constant Name	Description
ROTATE_NONE	No rotation
ROTATE_180	Rotate 180 degrees

②④ Image scaling (ImageScale)

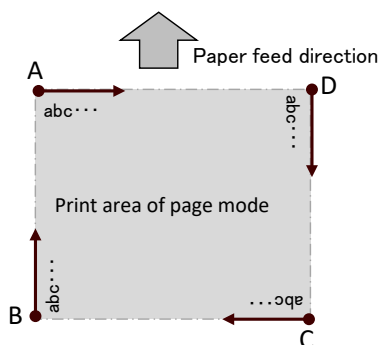
Constants of enumerated type used for image scaling are shown in the following table.

Constant Name	Description
IMAGE_SCALE_NONE	No scaling
IMAGE_SCALE_WIDTH_FIT	Scale to fit print width

②⑤ Print direction (Direction)

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

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





②⑥ Line style (LineStyle)

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

Constant Name	Description
<b>LINESTYLE_THIN</b>	Thin solid line (2 dots)
<b>LINESTYLE_MEDIUM</b>	Medium solid line (4 dots)
<b>LINESTYLE_THICK</b>	Thick solid line (8 dots)

#### (4) Method Details

##### ① 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** is executed.

PrinterManager	Constructor
----------------	-------------

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

Syntax      `public PrinterManager(Context context)`

Parameter    *context*                      Specify application context to call this method.  
Example: **MainActivity.this**

connect	Start communicating with printer (Bluetooth)
---------	--

Starts communication with a printer by Bluetooth connection.

The method of syntax (a) always communicates with a printer in secure mode.

The method of syntax (b) communicates with a printer by specifying secure mode or insecure mode.

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  
See "4.4.1(2)① Printer model" for available constants.

*address*                              Bluetooth address  
Example: "00:11:22:AA:BB:CC"

*secure*                                true      Communicates with a printer in secure mode  
false     Communicates with a printer in insecure mode  
Normally, communication in secure mode is recommended.

Exception    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.

Description    Call this method before using other **Printer Manager** class methods.

The printer specified by *printerModel* is connected to the Bluetooth address specified by *address*.

Also, printer initial setting is performed at the connection based on *printerModel* specified.

Monitoring of the printer status is started with this method. The latest printer status can be retrieved from **getStatus**.

Changes of the printer status can be notified as events by **onStatusChanged** and **setCallbackFunctionListener**.

Starts communication with a printer by USB connection.

Syntax      `public void connect(int printerModel)` throws **PrinterException**

Parameter   *printerModel*      Printer model constant for USB connection  
See "4.4.1(2) ① Printer model" for available constants.

Exception    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.

Description Call this method before using other **Printer Manager** class methods.

The printer specified by *printerModel* is connected.  
Also, printer initial setting is performed at the connection based on *printerModel* specified.

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

Starts communication with a printer by TCP/IP connection.

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

Parameter   *printerModel*      Printer model constant for Ethernet connection  
See "4.4.1(2) ① Printer model" for available constants.

*address*      IP address  
Example: "192.168.0.190"

Exception    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.

Description Call this method before using other **PrinterManager** class methods.

Starts communication with a printer connected to the same network as the Android device by TCP/IP connection.

Connects to the IP address specified by *address*. TCP ports 9100 and 26100 are used for communication. Also, printer initial setting is performed at the connection based on *printerModel* specified.

#### • Creating/discarding of socket in TCP/IP connection of the library

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

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

Changes of the printer status can be notified as events by `onStatusChanged` and `setCallbackFunctionListener`.

Stop communicating with printer

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

Description	This method discards the print data kept by <b>controlTransaction</b> . The instance of <b>CallbackFunctionListener</b> interface kept by <b>setCallbackFunctionListener</b> is discarded and the callback is stopped.
-------------	---

## Start/End callback of barcode scanner

**Syntax**

```
public void setBarcodeScannerListener(BarcodeScannerListener listener) throws PrinterException
```

Open cash drawer

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

## Sound buzzer

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

Sound external buzzer

**Syntax**

```
public void externalBuzzer(BuzzerPattern buzzer pattern, int buzzerCount)  
    throws PrinterException
```

Gets the latest printer status.

The method of syntax (a) returns the printer status with return value.

The method of syntax (b) stores the printer status in an array of int type.

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

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

Return value Status retrieved from the printer

Parameter *buf* Status retrieved from the printer

Exception **PrinterException**

**PrinterException** is thrown when an error occurs while this method is being called.

See "4.4.6 **PrinterException Class**" for details of the error.

When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

Description The printer status is shown below.

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

Bit	Function	Value	
		0	1
0	Voltage error	No error	Error
1	Hardware error	No error	Error
2	Head temperature error	No error	Error
3	Reserved	Fixed	-
4	Out-of-paper error	No error	Error
5	Reserved	Fixed	-
6	Paper jam error while detecting mark or gap	No error	Error
7	Cover open error	No error	Error
8	FEED Switch status	OFF	ON
9	Reserved	Fixed	-
10	Paper feed status	Stop	Operating
11	Return-waiting status	Not waiting	Waiting
12	Reserved	Fixed	-
13	Reserved	-	Fixed
14	Reserved	-	Fixed
15	Reserved	-	Fixed
16	FLASH memory rewriting	Not rewriting	Rewriting
17	Reserved	-	Fixed
18	Reserved	-	Fixed
19	Reserved	-	Fixed



Registers image file to NV graphics memory in the printer as a logo.

The method of syntax (a), dithering is fixed to be disabled.

The method of syntax (b), dithering can be specified.

Syntax	(b) public void <b>registerLogo</b> (String <i>fileName</i> , String <i>id</i> ) throws <b>PrinterException</b>
	(a) public void <b>registerLogo</b> (String <i>fileName</i> , String <i>id</i> , Dithering <i>dithering</i> ) throws <b>PrinterException</b>
Parameter	<p><i>fileName</i> File path of image file to register as a logo The formats that can be entered are described below.</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android <ul style="list-style-type: none"> <li>• file://</li> <li>• content://</li> </ul> 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.</li> </ul> <p>The file extensions supporting image file are .bmp, .jpg, .jpeg, and .png. When the image file is colored, it is converted to monochrome image by binarization and registered.</p> <p><i>id</i> Logo ID to register (key code) Specify the logo ID to register by character string of 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').</p> <p><i>dithering</i> Dithering See "4.4.1(3)① Dithering (Dithering)" for available constants.</p>
Exception	<p><b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See "<b>4.4.6 PrinterException Class</b>" for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>

Deletes the registered logo.

Syntax	public void <b>unregisterLogo</b> (String <i>id</i> ) throws <b>PrinterException</b>
Parameter	<p><i>id</i> Logo ID to delete (key code) Specify the ID of the registered logo as a character string.</p>

Exception    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.  
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

<b>registerStyleSheet</b>	Register style sheet
---------------------------	----------------------

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

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

<b>unregisterStyleSheet</b>	Delete registered style sheet
-----------------------------	-------------------------------

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

Syntax        public void **unregisterStyleSheet**(int *num*) throws **PrinterException**

<b>resetPrinter</b>	Reset printer
---------------------	---------------

Resets the printer hardware.

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

Exception    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.  
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

Description For Bluetooth connection:  
The printer hardware reset is performed by the printer command "Printer Reset".  
  
For USB connection:  
The printer reset is performed by using the SOFT\_RESET function in USB printer class.  
  
For TCP/IP connection:  
The reset is performed to the connected printer by our proprietary command (reset request) to TCP port 26100.  
  
The connection with the printer is retained even after this method is executed.

<b>getPrinterResponse</b>	Get various responses from printer
---------------------------	------------------------------------

Gets response data from the printer.

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

Parameter    *id*                                  Response type constant  
See "4.4.1(2) ② Response type" for available constants.



*buf*

Buffer that stores the retrieved response data  
This method stores the response data specified by *id* to the object specified by *buf*.  
The buffer type varies depending on the response type constant.  
See the following table for buffer types.

Response Type Constant	
Parameter	Description
<b>PRINTER_RESPONSE_REQUEST</b> (Execution response request)	
<i>buf</i>	Specify an int type array of length 1. Specify 0 to 15 (00h to 0Fh) for <i>buf[0]</i> . When the response is retrieved successfully, the response code of the execution response request is stored to <i>buf[0]</i> with 128 to 143 (80h to 8Fh).
<b>PRINTER_RESPONSE_USER_AREA</b> (Send remaining capacity of user area)	
<i>buf</i>	Specify an int type array of length 1. When the response is retrieved successfully, the remaining capacity of the user area is stored as a numerical value in bytes.
<b>PRINTER_RESPONSE_ARRANGE_USER_AREA</b> (Send remaining capacity of user area after defragment)	
<i>buf</i>	Specify an int type array of length 1. When the response is retrieved successfully, the remaining capacity of the user area after defragment is stored as a numerical value in bytes.
<b>PRINTER_RESPONSE_NV_GRAPHICS</b> (Send NV graphics memory capacity)	
<i>buf</i>	Specify an int type array of length 1. When the response is retrieved successfully, the NV graphics memory capacity is stored as a numerical value in bytes.
<b>PRINTER_RESPONSE_KEY_CODE</b> (Send key code list of defined NV graphics)	
<i>buf</i>	Specify an ArrayList<String> array. When the response is retrieved successfully, the key code of NV graphics is stored as a string array. Example: <i>buf.size()</i> = 3, <i>buf[0]</i> = "22", <i>buf[1]</i> = "23", <i>buf[2]</i> = "24", etc.
<b>PRINTER_RESPONSE_BATTERY_STATUS</b> (Battery remaining capacity level)	
<i>buf</i>	Specify an int type array of length 1. When the response is retrieved successfully, the battery remaining capacity level is stored as a value. See "4.4.1(2) ③ Battery remaining capacity level" for details of the value. Battery remaining capacity level <b>BATTERY_STATUS_FULL</b> : Full (Battery remaining capacity: approx. 80%) <b>BATTERY_STATUS_MIDDLE</b> : Middle (Battery remaining capacity: approx. 40%) <b>BATTERY_STATUS_LOW</b> : Low (Battery remaining capacity: approx. 10%) <b>BATTERY_STATUS_EMPTY</b> : No battery

Response Type Constant	
Parameter	Description
<b>PRINTER_RESPONSE_MAINTENANCE_COUNTER</b> (Send maintenance counter)	
<i>buf</i>	<p>Specify an int array of length 8.  When the response is retrieved successfully, the maintenance counter is stored as shown below.</p> <p><i>buf</i>[0] : Number of paper feed lines (unit: 100 dot-line)  <i>buf</i>[1] : Number of thermal head activation times (unit: 100 dot-line)  <i>buf</i>[2] : Number of autocutter drive times<sup>*1</sup>  <i>buf</i>[3] : Drive time of printer (unit: minute)  <i>buf</i>[4] : Number of paper feed lines (integrated value) (unit: 100 dot-line)  <i>buf</i>[5] : Number of thermal head activation times (integrated value) (unit: 100 dot-line)  <i>buf</i>[6] : Number of autocutter drive times (integrated value)<sup>*1</sup>  <i>buf</i>[7] : Drive time of printer (integrated value) (unit: minute)</p>
<b>PRINTER_RESPONSE_PRINTER_IDENTIFIER</b> (Send printer ID)	
<i>buf</i>	<p>Specify an ArrayList&lt;PrinterIdentifier&gt; array.  When the response is retrieved successfully, a <b>PrinterIdentifier</b> object is stored in the first element of <i>buf</i>.  Example: <i>buf.size()</i> = 1, <i>buf.get</i>[0] = "<b>PrinterIdentifier</b> object"  See "<b>4.4.5 PrinterIdentifier Class</b>" for the <b>PrinterIdentifier</b> class object.</p>

\*1: Not supported. Always returns 0.

Exception **PrinterException**

**PrinterException** is thrown when an error occurs while this method is being called.

See "**4.4.6 PrinterException Class**" for details of the error.

When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

## startDiscoveryPrinter Start printer search (Bluetooth)

Searches for the printer using the Bluetooth connection. The found printer information is stored in **PrinterInfo** class.

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

Parameter *listener* Instance of **PrinterListener**  
Completion of this method or cancellation by **cancelDiscoveryPrinter** is notified to the user application as an end event by **finishEvent** through the instance set in *listener*.

Exception **PrinterException**

**PrinterException** is thrown when an error occurs while this method is being called.

See "**4.4.6 PrinterException Class**" for details of the error.

Description This method may discover other printers besides SII printer.  
In addition, the printers in which the Bluetooth connection is already established by the library or other applications are not found.

Do not call this method from the main thread of the application.

startDiscoveryPrinter

Start printer search (USB)

startDiscoveryPrinter

Start printer search (USB)

Searches for the printer using the USB connection. The found printer information is stored in **PrinterInfo** class.

**Syntax**

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

Parameter	<i>listener</i>	Instance of <b>PrinterListener</b>
		Completion of this method or cancellation by <b>cancelDiscoveryPrinter</b> is notified to the user application as an end event by <b>finishEvent</b> through the instance set in <i>listener</i> .

*deviceType* Port type  
Specify **PRINTER** **TYPE** **USB**.

Exception **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
 See "**4.4.6 PrinterException Class**" for details of the error.

Description	This method searches for SII printer. The printer information of the found printer is stored to <b>PrinterInfo</b> class described later.
-------------	---

**startDiscoveryPrinter** Start printer search (TCP/IP)

**startDiscoveryPrinter** Start printer search (TCP/IP)

Searches for SII printer connecting to the same network.

[illegible]

Parameter	<i>listener</i>	Instance of <b>PrinterListener</b> Completion of this method or cancellation by <b>cancelDiscoveryPrinter</b> is notified to the user application as an end event by <b>finishEvent</b> through the instance set in <i>listener</i> .
-----------	-----------------	--

*retry*      Retry count (times)  
Sends the local broadcast packet the number of times set by *retry*.  
The valid range is 1 to 5.  
When the value is specified less than 1, the number is set to 1.  
When the value is specified more than 5, the number is set to 5.

*timeout*

Search timeout period (millisecond: ms)

Sets the timeout period per search. Each time the local broadcast packet is sent, this method waits for a response from the printer until the period specified by *timeout* elapses.

The valid range is 3000 to 60000.

When the value is specified less than 3000, the period is set to 3000 ms.

When the value is specified more than 60000, the period is set to 60000 ms.

**PrinterException** is thrown when an error occurs while this method is being called.

Description	This method searches for SII printer. The printer information of the found printer is stored to <b>PrinterInfo</b> class described later.
-------------	---

## cancelDiscoveryPrinter

## Cancel printer search

Cancels **startDiscoveryPrinter** under execution.

Syntax      `public void cancelDiscoveryPrinter()`

Description    Cancellation by this method is notified as an end event to the user application by **finishEvent** through the instance set in *listener* of **startDiscoveryPrinter**.

## getFoundPrinter

## Get found printer information

Gets the information of the printer found by **startDiscoveryPrinter** in ArrayList from the **PrinterInfo** class, which is the storage destination.

Syntax      `public ArrayList<PrinterInfo> getFoundPrinter()`

Return value    ArrayList of **PrinterInfo** class

## getSendTimeout

## Get send timeout period

Gets the send timeout period.

Syntax      `public int getSendTimeout()`

Return value    Send timeout period (millisecond: ms)

Description    This method can get the send timeout period regardless of whether **isConnect** is true or false.

## setSendTimeout

## Set send timeout period

Sets the send timeout period.

Syntax      `public void setSendTimeout(int sendTimeout)`

Parameter    *sendTimeout*      Send timeout period (millisecond: ms)  
The valid range is 100 to 90000.  
When the value out of the valid range is specified, the value is set to 10000 ms.

Description    When the send timeout period is not set by this method, the value is set to 10000.

This method can set the send timeout period regardless of whether **isConnect** is true or false.

The set timeout period becomes effective at the next data sending.

## getReceiveTimeout

## Get receive timeout period

Gets the receive timeout period.

Syntax      `public int getReceiveTimeout()`

Return value    Receive timeout period (millisecond: ms)

Description This method can get the receive timeout period regardless of whether **isConnect** is true or false.

## **setReceiveTimeout**

## Set receive timeout period

Sets the receive timeout period.

Syntax `public void setReceiveTimeout(int receiveTimeout)`

Parameter *receiveTimeout* Receive timeout period (millisecond: ms)  
The valid range is 100 to 90000.  
When the value out of the valid range is specified, the value is set to 10000 ms.

Description When the receive timeout period is not set by this method, the value is set to 10000.

This method can set the receive timeout period regardless of whether **isConnect** is true or false.

The set timeout period becomes effective at the next data receiving.

## **getInternationalCharacter**

## Get international character set

Gets the value of international character set.

Syntax `public int getInternationalCharacter()`

Return value See "4.4.1(2) ④ International character set" for details of the value.

Description When the text data is sent by **sendText**, **sendTextEx**, **sendDataFile**, **printPageModeText** or **printPageModeTextEx**, the print result of the following character codes varies. See "Appendix A Character Set" for details about characters to be printed.

Character codes with the varying print result depending on the configuration of the international character:

0x23, 0x24, 0x40, 0x5B, 0x5C, 0x5D, 0x5E, 0x60, 0x7B, 0x7C, 0x7D, 0x7E

## **setInternationalCharacter**

## Set international character set

Sets the value of international character set.

Syntax `public void setInternationalCharacter(int internationalCharacter)`

Parameter *internationalCharacter* International character set constant  
See "4.4.1(2) ④ International character set" for the configurable values.  
When an invalid value is specified, it is ignored.

Description When the international character set is not set by this method, it is as follows depending on the language setting of an Android device.

When the language setting of the Android device is Japanese:

**COUNTRY\_JAPAN**

When the language setting of the Android device is other than Japanese:

**COUNTRY\_USA**

## getCodePage

## Get codepage

Gets the value of codepage.

Syntax      `public int getCodePage()`

Return value   See "4.4.1(2)⑤ Codepage" for details of the value.

Description   The encoder used for sending the text data by **sendText**, **sendTextEx**, **sendDataFile**, **printPageModeText**, or **printPageModeTextEx** is changed. See "Appendix A Character Set" for details about characters to be printed.

## setCodePage

## Set codepage

Sets the value of codepage.

Syntax      `public void setCodePage(int codePage)`

Parameter   *codePage*                      Codepage constant  
See "4.4.1(2) ⑤ Codepage" for the configurable values.  
When an invalid value is specified, it is ignored.

Description   When the codepage is not set by this method, it is as follows depending on the language setting of an Android device.

When the language setting of the Android device is Japanese:

**CODE\_PAGE\_KATAKANA**

When the language setting of the Android device is other than Japanese:

**CODE\_PAGE\_1252**

## getPrinterModel

## Get printer model

Gets the value of the connecting printer model.

Syntax      `public int getPrinterModel()`

Return value   See "4.4.1(2) ① Printer model" for details of the value.  
**PRINTER\_MODEL\_DEFAULT** is returned when **isConnect** is false.

Description   Even when the printer is not connected, when **connect** has succeeded once, the printer model value successfully connected last time is returned.

## getPortType

## Get connecting port type

Gets the port type used for connecting with the printer.

Syntax      `public int getPortType()`

Return value   See "4.4.1(2) ⑥ Port type" for details of the value.  
**PRINTER\_TYPE\_BLUETOOTH** is returned when **isConnect** is false.

Description   Even when the printer is not connected, when **connect** has been succeeded once, the port type value successfully connected last time is returned.

## isConnect

## Verify connection state with printer

Verifies connection state with the printer.

Syntax      public boolean **isConnect**()

Return value    true          Connected to a printer  
                 false        Not connected to a printer

Description    When the data transmission is failed, the communication with the printer is ended, and this method returns false. When false is returned, reconnect with the printer by **connect**.

## getSocketKeepingTime

## Get socket keeping time

Gets the socket keeping time.

Syntax      public int **getSocketKeepingTime**()

Return value    Socket keeping time (millisecond: ms)

Description    This method can get the socket keeping time regardless of whether **isConnect** is true or false.

## setSocketKeepingTime

## Set socket keeping time

Sets the socket keeping time.

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

Valid range    60000 to 300000 (millisecond: ms)  
                 When the value is specified less than 60000, the time is set to 60000 ms.  
                 When the value is specified more than 300000, the time is set to 300000 ms.

Default        300000

Description    This method can set the socket keeping time regardless of whether **isConnect** is true or false.

For the socket keeping time, specify a time equal to Receive Timeout of the printer to be connected. The setting of Receive Timeout can be changed in the Android app "SII Printer Utility" on the Google Play.

The set socket keeping time becomes effective at the next **connect** execution.

## getPrintSmartLabelMode

## Get paper when printing label file

Gets the paper when printing label file.

Syntax      public int **getPrintSmartLabelMode**()

Return value    See "4.4.1(2)⑦ Paper selection with or without mark when printing label file" for details of the value.

Description    This method can get the paper regardless of whether **isConnect** is true or false.

## setPrintSmartLabelMode

## Set paper when printing label file

Sets the paper when printing label file.

Syntax	public void <b>setPrintSmartLabelMode</b> (int <i>paperMode</i> )	
Parameter	<i>paperMode</i>	Paper selection with or without mark when printing label file See "4.4.1(2) ⑦ Paper selection with or without mark when printing label file" for the configurable values. When an invalid value is specified, it is ignored.
Description	When specifying the marked paper, feeds the paper to the print start position at <b>printSmartLabelImageData</b> execution.  When specifying the paper without the mark, the paper is not fed to the print start position at <b>printSmartLabelImageData</b> execution.	

## getVersion

## Get SDK version

Gets the SDK version as a character string.

Syntax	public String <b>getVersion</b> ()	
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 can get the SDK version regardless of whether <b>isConnect</b> is true or false.	

## controlTransaction

## Start/End batch processing

Starts or ends batch processing.

Syntax	public void <b>controlTransaction</b> (TransactionFunction <i>transactionFunction</i> ) throws <b>PrinterException</b>	
Parameter	<i>transactionFunction</i>	Batch processing selection See "4.4.1(3)② Batch processing selection (TransactionFunction)" for available constants.
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See " <b>4.4.6 PrinterException Class</b> " for details of the error. When data transmission fails, communication with the printer may be terminated and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.	
Description	The procedure of batch processing is as follows: (1) Start batch processing. Specify <b>TRANSACTION_START</b> . (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**

The batch processing target methods are as follows:

- **sendText**
- **sendTextEx**
- **printBarcode**
- **printPDF417**
- **printQRcode**
- **printDataMatrix**
- **printMaxiCode**
- **printGS1DataBarStacked**
- **printGS1DataBarStackedOmnidirectional**
- **printGS1DataBarExpandedStacked**
- **printAztecCode**
- **cutPaper**
- **feedPosition**
- **sendBinary**
- **sendDataFile**
- **sendImage**
- **printPDF**
- **printLogo**<sup>\*1</sup>
- **printSmartLabelImageData**
- **enterPageMode**
- **exitPageMode**
- **setPageModeArea**
- **setPageModeDirection**
- **setPageModeLineSpacing**
- **printPageMode**
- **printPageModeText**
- **printPageModeTextEx**
- **printPageModeBarcode**
- **printPageModePDF417**
- **printPageModeQRcode**
- **printPageModeDataMatrix**
- **printPageModeMaxiCode**
- **printPageModeGS1DataBarStacked**
- **printPageModeGS1DataBarStackedOmnidirectional**
- **printPageModeGS1DataBarExpandedStacked**
- **printPageModeAztecCode**
- **sendPageModeBinary**
- **printPageModeImageFile**
- **printPageModeRectangle**
- **printPageModeLine**
- **printPageModeLogo**<sup>\*1</sup>

\*1: The method under batch processing does not notify the error even when the registered logo does not exist.

## ② 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

Sends text data.

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

Parameter    *text*                              Text data to send to the printer  
Data size that can be specified at 1 time is 16 KB (16384 bytes).

Exception    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.  
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

Description   This method encodes the specified text data to printable text data based on **setInternationalCharacter** and **setCodePage**, and then sends it to the 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.

### sendTextEx Send format specified text data

Sends format specified text data to the printer.

The method of syntax (a) or (c) outputs the pending data at first and starts processing.

The method of syntax (b) starts processing according to the constants of the pending data output specifying.

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	<i>text</i>	Text data to send to the printer Data size that can be specified at 1 time is 16 KB (16384 bytes).
	<i>bold</i>	Bold print See "4.4.1(3) ③ Bold print (CharacterBold)" for available constants.
	<i>underline</i>	Underline See "4.4.1(3) ④ Underline (CharacterUnderline)" for available constants.
	<i>reverse</i>	Reverse print See "4.4.1(3) ⑤ Reverse print (CharacterReverse)" for available constants.
	<i>inversion</i>	Inversion print See "4.4.1(3) ⑥ Inversion print (CharacterInversion)" for available constants.
	<i>font</i>	Font See "4.4.1(3) ⑦ Character font (CharacterFont)" for available constants.
	<i>scale</i>	Character scale See "4.4.1(3) ⑧ Character scale (CharacterScale)" for available constants.
	<i>alignment</i>	Alignment See "4.4.1(3) ⑨ Alignment (PrintAlignment)" for available constants.
	<i>output</i>	Pending data output specifying See "4.4.1(3) ⑩ Pending data output specifying (OutputPendingData)" for available constants.

**Exception**    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "4.4.6 PrinterException Class" for details of the error.  
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

**Description**    This method encodes the specified text data to printable text data based on **setInternationalCharacter** and **setCodePage**, and then sends it to the printer.

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

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

When the method of syntax (a) is executed or **PENDING\_DATA\_OUTPUT\_FIRST** is specified at *output* 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.

The method of syntax (a) specifies the barcode data by character string.

The method of syntax (c) specifies the barcode data by the array of bytes and specifies the alignment of the barcode.

The method of syntax (d) specifies the Customer Bar Code\_JP by character string and specifies the alignment of the Customer Bar Code\_JP.

Syntax	<pre>(a) public void <b>printBarcode</b>(BarcodeSymbol <i>barcodeSymbol</i>,                                String <i>text</i>,                                ModuleSize <i>moduleSize</i>,                                int <i>moduleHeight</i>,                                HriPosition <i>hriPosition</i>,                                CharacterFont <i>hriFont</i>,                                PrintAlignment <i>alignment</i>) throws <b>PrinterException</b></pre>
--------	---

(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	<i>barcodeSymbol</i>	Barcode symbol See "4.4.1(3) ⑪ Barcode symbol (BarcodeSymbol)" for available constants and corresponding syntax.
-----------	----------------------	---

*text (data)* Barcode data to send to the printer  
The input conditions for barcode data are as follows.

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'	

Barcode	Number of Data	Inputtable Data Character String (Data)	Remarks
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'	
Customer Bar Code_JP	7 to 20 characters	'0' to '9' 'A' to 'Z' ' '	Start code, check digit, and stop code are automatically added. It is needed to specify one of '0' to '9' at the first 7 characters. A' to 'Z' is calculated as 2 characters.
GS1 Databar Omni-directional	13 characters	'0' to '9'	Check digit is automatically added.
GS1 Databar Truncated	13 characters	'0' to '9'	Check digit is automatically added.
GS1 Databar Limited	13 characters	'0' to '9'	Check digit is automatically added.
GS1 Databar Expanded	2 to 255 characters	' ' to '"' '%' to '?' 'A' to 'Z' ' ' 'a' to 'z' '{'	

*moduleSize*

Barcode width

See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.

*moduleHeight*

Barcode height (dot)

- When *barcodeSymbol* is set to the following, the valid range is 1 to 255.

**BARCODE\_SYMBOL\_UPC\_A**

**BARCODE\_SYMBOL\_UPC\_E**

BARCODE\_SYMBOL\_EAN13  
 BARCODE\_SYMBOL\_JAN13  
 BARCODE\_SYMBOL\_EAN8  
 BARCODE\_SYMBOL\_JAN8  
 BARCODE\_SYMBOL\_CODE39  
 BARCODE\_SYMBOL\_CODE93  
 BARCODE\_SYMBOL\_CODE128  
 BARCODE\_SYMBOL\_ITF  
 BARCODE\_SYMBOL\_CODABAR  
 BARCODE\_SYMBOL\_EAN13\_ADDON  
 BARCODE\_SYMBOL\_JAN13\_ADDON

- When *barcodeSymbol* is set to the following, the valid range is different by *barcodeSymbol* and *moduleSize*.

<i>barcodeSymbol</i>		
	<i>moduleSize</i>	Valid Range
<b>BARCODE_SYMBOL_GS1_OMNI_DIRECTIONAL</b>		
	BARCODE_MODULE_WIDTH_2	66 to 255
	BARCODE_MODULE_WIDTH_3	99 to 255
	BARCODE_MODULE_WIDTH_4	132 to 255
	BARCODE_MODULE_WIDTH_5	165 to 255
	BARCODE_MODULE_WIDTH_6	198 to 255
<b>BARCODE_SYMBOL_GS1_TRUNCATED</b>		
	BARCODE_MODULE_WIDTH_2	26 to 255
	BARCODE_MODULE_WIDTH_3	39 to 255
	BARCODE_MODULE_WIDTH_4	52 to 255
	BARCODE_MODULE_WIDTH_5	65 to 255
	BARCODE_MODULE_WIDTH_6	78 to 255
<b>BARCODE_SYMBOL_GS1_LIMITED</b>		
	BARCODE_MODULE_WIDTH_2	20 to 255
	BARCODE_MODULE_WIDTH_3	30 to 255
	BARCODE_MODULE_WIDTH_4	40 to 255
	BARCODE_MODULE_WIDTH_5	50 to 255
	BARCODE_MODULE_WIDTH_6	60 to 255
<b>BARCODE_SYMBOL_GS1_EXPANDED</b>		
	BARCODE_MODULE_WIDTH_2	68 to 255
	BARCODE_MODULE_WIDTH_3	102 to 255
	BARCODE_MODULE_WIDTH_4	136 to 255
	BARCODE_MODULE_WIDTH_5	170 to 255
	BARCODE_MODULE_WIDTH_6	204 to 255

*hriPosition*      HRI character print position  
 See "4.4.1(3) ⑬ HRI character print position (HriPosition)" for available constants.

*hriFont*          HRI character font  
 See "4.4.1(3) ⑦ Character font (CharacterFont)" for available constants.

*alignment* Alignment  
See "4.4.1(3)⑨ Alignment (PrintAlignment)" for available constants.

*nwRatio* N:W ratio  
See "4.4.1(3)⑭ 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.

<i>moduleSize</i>	<i>nwRatio</i>		
	NWRATIO_ 1TO2	NWRATIO_ 1TO2_5	NWRATIO_ 1TO3
BARCODE_MODULE_WIDTH_2	0.500 mm (4 dots)	0.625 mm (5 dots)	0.750 mm (6 dots)
BARCODE_MODULE_WIDTH_3	0.750 mm (6 dots)	1.000 mm (8 dots)	1.125 mm (9 dots)
BARCODE_MODULE_WIDTH_4	1.000 mm (8 dots)	1.250 mm (10 dots)	1.500 mm (12 dots)
BARCODE_MODULE_WIDTH_5	1.250 mm (10 dots)	1.625 mm (13 dots)	1.875 mm (15 dots)
BARCODE_MODULE_WIDTH_6	1.500 mm (12 dots)	1.875 mm (15 dots)	2.250 mm (18 dots)

- Exception **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "4.4.6 PrinterException Class" for details of the error.  
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.
- Note The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.
- Reference See "Appendix B Barcode Size List" for details of the barcode size.

## printPDF417

## Print PDF417

Prints PDF417.  
The method of syntax (a) specifies PDF417 symbol.  
The method of syntax (b) is fixed to standard PDF417.

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

Parameter	<i>text</i>	Barcode data to send to the printer
	<i>errorCorrection</i>	Error correction level See "4.4.1(3) ⑮ Error correction level (ErrorCorrection)" for available constants.
	<i>row</i>	The number of rows (row) The valid range is 0, 3 to 90. When 0 is specified, the number of rows is automatically set.
	<i>column</i>	The number of columns in data area The valid range is 0 to 30. When 0 is specified, the number of columns in the data area is automatically set.
	<i>moduleSize</i>	Nominal fine element width See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.
	<i>moduleHeight</i>	Module height (dot) The valid range is 2 to 127. When the module height is set smaller, some barcode scanners may not read it. Set 3 or more for normal use.
	<i>alignment</i>	Alignment See "4.4.1(3) ⑨ Alignment (PrintAlignment)" for available constants.
	<i>pdf417Symbol</i>	Symbol of PDF417 See "4.4.1(3) ⑩ PDF417 symbol (Pdf417Symbol)" for available constants.
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.	
Note	The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.	
Reference	See "Appendix B Barcode Size List" for details of the barcode size.	

## printQRcode

## Print QR Code

Prints QR Code.

The method of syntax (a) is fixed to QR Code Model 2.

The method of syntax (b) specifies QR Code Model.

Syntax	(a) public void <b>printQRcode</b> (String <i>text</i> , ErrorCorrection <i>errorCorrection</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i> ) throws <b>PrinterException</b>
	(b) public void <b>printQRcode</b> (String <i>text</i> , ErrorCorrection <i>errorCorrection</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i> , QrModel <i>model</i> ) throws <b>PrinterException</b>



Parameter	<i>text</i>	Barcode data to send to the printer The version for either syntax (a) or (b) is automatically set depending on the number of data specified on <i>text</i> .
	<i>errorCorrection</i>	Error correction level See "4.4.1(3) ⑮ Error correction level (ErrorCorrection)" for available constants.
	<i>moduleSize</i>	Module size See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.
	<i>alignment</i>	Alignment See "4.4.1(3) ⑨ Alignment (PrintAlignment)" for available constants.
	<i>model</i>	QR Code Model See "4.4.1(3) ⑰ QR Code Model (QrModel)" for available constants.
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.	
Note	The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.	
Reference	See "Appendix B Barcode Size List" for details of the barcode size.	

## printDataMatrix

## Print Data Matrix

Prints Data Matrix.

Syntax	public void <b>printDataMatrix</b> (String <i>text</i> , DataMatrixModule <i>dataMatrixModule</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i> ) throws <b>PrinterException</b>	
Parameter	<i>text</i>	Barcode data to send to the printer
	<i>dataMatrixModule</i>	The number of Data Matrix modules See "4.4.1(3) ⑱ Data Matrix module (DataMatrixModule)" for available constants.
	<i>moduleSize</i>	Module size See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.
	<i>alignment</i>	Alignment See "4.4.1(3) ⑨ Alignment (PrintAlignment)" for available constants.
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.	
Note	The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.	
Reference	See "Appendix B Barcode Size List" for details of the barcode size.	

Prints MaxiCode.

Syntax	public void <b>printMaxiCode</b> (String <i>text</i> , MaxiCodeMode <i>maxiCodeMode</i> , PrintAlignment <i>alignment</i> ) throws <b>PrinterException</b>	
Parameter	<i>text</i>	Barcode data to send to the printer <ul style="list-style-type: none"> <li>When <i>maxiCodeMode</i> is <b>MAXI_CODE_2</b> Add the service class (3 digits), the country code (3 digits), and the postal code (9 digits) to the beginning of the data.</li> <li>When <i>maxiCodeMode</i> is <b>MAXI_CODE_3</b> Add the service class (3 digits), the country code (3 digits), and the postal code (6 digits) to the beginning of the data.</li> </ul>
	<i>maxiCodeMode</i>	MaxiCode Mode See "4.4.1(3)⑱ MaxiCode Mode (MaxiCodeMode)" for available constants.
	<i>alignment</i>	Alignment See "4.4.1(3)⑨ Alignment (PrintAlignment)" for available constants.
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See " <b>4.4.6 PrinterException Class</b> " for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.	
Note	The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.	
Reference	See "Appendix B Barcode Size List" for details of the barcode size.	

Prints GS1 Databar Stacked.

Syntax	public void <b>printGS1DataBarStacked</b> (String <i>text</i> , ModuleSize <i>moduleSize</i> , PrintAlignment <i>alignment</i> ) throws <b>PrinterException</b>	
Parameter	<i>text</i>	Barcode data to send to the printer Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.
	<i>moduleSize</i>	Module size See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.
	<i>alignment</i>	Alignment See "4.4.1(3)⑨ Alignment (PrintAlignment)" for available constants.

Exception	<p><b>PrinterException</b></p> <p><b>PrinterException</b> is thrown when an error occurs while this method is being called. See "<b>4.4.6 PrinterException Class</b>" for details of the error.</p> <p>When the data transmission is failed, the communication with the printer is ended and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>
Reference	See "Appendix B Barcode Size List" for details of the barcode size.

## printGS1DataBarStackedOmniDirectional      Print GS1 Databar Stacked Omni-directional

Prints GS1 Databar Stacked Omni-directional.

Syntax	<pre>public void <b>printGS1DataBarStackedOmniDirectional</b>(String <i>text</i>,   int <i>moduleHeight</i>,   ModuleSize <i>moduleSize</i>,   PrintAlignment <i>alignment</i>) throws <b>PrinterException</b></pre>								
Parameter	<table> <tr> <td><i>text</i></td><td>Barcode data to send to the printer Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.</td></tr> <tr> <td><i>moduleHeight</i></td><td>Barcode module height (the number of the modules) The valid range is 33 to 255.</td></tr> <tr> <td><i>moduleSize</i></td><td>Module size See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.</td></tr> <tr> <td><i>alignment</i></td><td>Alignment See "4.4.1(3)⑨ Alignment (PrintAlignment)" for available constants.</td></tr> </table>	<i>text</i>	Barcode data to send to the printer Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.	<i>moduleHeight</i>	Barcode module height (the number of the modules) The valid range is 33 to 255.	<i>moduleSize</i>	Module size See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.	<i>alignment</i>	Alignment See "4.4.1(3)⑨ Alignment (PrintAlignment)" for available constants.
<i>text</i>	Barcode data to send to the printer Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.								
<i>moduleHeight</i>	Barcode module height (the number of the modules) The valid range is 33 to 255.								
<i>moduleSize</i>	Module size See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.								
<i>alignment</i>	Alignment See "4.4.1(3)⑨ Alignment (PrintAlignment)" for available constants.								
Exception	<p><b>PrinterException</b></p> <p><b>PrinterException</b> is thrown when an error occurs while this method is being called. See "<b>4.4.6 PrinterException Class</b>" for details of the error.</p> <p>When the data transmission is failed, the communication with the printer is ended and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>								
Reference	See "Appendix B Barcode Size List" for details of the barcode size.								

## printGS1DataBarExpandedStacked      Print GS1 Databar Expanded Stacked

Prints GS1 Databar Expanded Stacked.

Syntax	<pre>public void <b>printGS1DataBarExpandedStacked</b>(String <i>text</i>,   int <i>column</i>,   ModuleSize <i>moduleSize</i>,   PrintAlignment <i>alignment</i>) throws <b>PrinterException</b></pre>		
Parameter	<table> <tr> <td><i>text</i></td><td>Barcode data to send to the printer Enter any number of characters using the following: ' ', '!', '"', '%', '&amp;', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '&lt;', '=', '&gt;', '?', '_', '0' to '9', 'A' to 'Z', 'a' to 'z'. Enter '{1' to FNC1. Be sure to input the check digit because it is not automatically calculated by the printer.</td></tr> </table>	<i>text</i>	Barcode data to send to the printer Enter any number of characters using the following: ' ', '!', '"', '%', '&', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '_', '0' to '9', 'A' to 'Z', 'a' to 'z'. Enter '{1' to FNC1. Be sure to input the check digit because it is not automatically calculated by the printer.
<i>text</i>	Barcode data to send to the printer Enter any number of characters using the following: ' ', '!', '"', '%', '&', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '_', '0' to '9', 'A' to 'Z', 'a' to 'z'. Enter '{1' to FNC1. Be sure to input the check digit because it is not automatically calculated by the printer.		



*alignment*

Alignment

See "4.4.1(3)⑨ Alignment (PrintAlignment)" for available constants.

- Exception **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.  
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.
- Reference See "Appendix B Barcode Size List" for details of the barcode size.

## cutPaper

Cut paper

Feeds the paper to the paper cut position. The paper is not cut.

- Syntax `public void cutPaper(CuttingMethod cuttingMethod)` throws **PrinterException**
- Parameter *cuttingMethod* Cutting method  
See "4.4.1(3)⑪ Cutting method (CuttingMethod)" for available constants.
- Exception **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.  
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

## feedPosition

Paper form feed

Performs the paper form feed of marked paper or label.

- Syntax `public void feedPosition(FeedPosition feedPosition)` throws **PrinterException**
- Parameter *feedPosition* Form feed position  
See "4.4.1(3)⑫ Form feed position (FeedPosition)" for available constants.
- Exception **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.  
When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.
- Note The paper form feed is not performed when this method is executed at the form feed position of the marked paper or the label.

## sendBinary

Send binary data

Sends binary data to the printer.

- Syntax `public void sendBinary(byte [] binary)` throws **PrinterException**
- Parameter *binary* Binary data to send to the printer  
Data size that can be specified at 1 time is 16 KB (16384 bytes).

Exception	<p><b>PrinterException</b></p> <p><b>PrinterException</b> is thrown when an error occurs while this method is being called. See "<b>4.4.6 PrinterException Class</b>" for details of the error.</p> <p>When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>
Description	<p>This method sends the specified binary data to the printer without conversion.</p> <p>By sending printer commands 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 get responses from the printer.</p>

<b>sendDataFile</b>	<b>Send specified file</b>
---------------------	----------------------------

<p>Sends file data.</p> <p>The method of syntax (a), dithering is fixed to be disabled.</p> <p>The method of syntax (b), dithering can be specified.</p>	
Syntax	<p>(a) public void <b>sendDataFile</b>(String <i>fileName</i>, PrintAlignment <i>alignment</i>) throws <b>PrinterException</b></p> <p>(b) public void <b>sendDataFile</b>(String <i>fileName</i>, PrintAlignment <i>alignment</i>, Dithering <i>dithering</i>) throws <b>PrinterException</b></p>
Parameter	<p><i>fileName</i> File path of the data file to send to the printer The formats that can be entered are described below.</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android <ul style="list-style-type: none"> <li>• file://</li> <li>• content://</li> </ul> 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.</li> </ul> <p>The maximum file size that can be specified is 1 MB (1048576 bytes). The file extensions that can be sent and the file transmission are described below.</p> <ul style="list-style-type: none"> <li>• .bmp, .jpg, .jpeg, .png Data is sent to the printer as image file. Colored image file is converted to monochrome image by binarization and registered. Printing is performed at one time after mapping the image file in memory of the printer.</li> </ul>

- Data is sent to the printer as text data. Text data format supports UTF-8. This method encodes the text data to printable text data based on the settings of **setInternationalCharacter** and **setCodePage**, and then sends it to the 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.

- Data is sent to the printer as the binary data without conversion.

The alignment is valid only when the file extension specified on *fileName* is .bmp, .jpg, .jpeg, .png, or .txt.  
See "4.4.1(3)⑨ Alignment (PrintAlignment)" for available constants.

The dithering is valid only when the file extension specified on *fileName* is .bmp, .jpg, .jpeg, or .png.  
See "4.4.1(3)① Dithering (Dithering)" for available constants.

**PrinterException** is thrown when an error occurs while this method is being called. See **"4.4.6 PrinterException Class"** for details of the error.

When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

Send image

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

*alignment* Alignment  
See "4.4.1(3)⑨ Alignment (PrintAlignment)" for available constants.

**PrinterException** is thrown when an error occurs while this method is being called. See **"4.4.6 PrinterException Class"** for details of the error. When the data transmission is failed, the communication with the printer is ended, and **PrinterException** may be thrown. See **isConnect** for verifying the connection state with the printer.

Prints the specified page of the selected PDF file.

Syntax	<pre>public void <b>printPDF</b>(String <i>fileName</i>,                       int <i>startIndex</i>,                       int <i>endIndex</i>,                       Rotate <i>rotate</i>,                       ImageScale <i>imageScale</i>,                       int <i>bottomMargin</i>,                       Dithering <i>dithering</i>,                       PrintAlignment <i>alignment</i>) throws <b>PrinterException</b></pre>
Parameter	<p><i>fileName</i></p> <p>File path of the PDF file The file extension for supporting PDF is .pdf. A maximum of 1 MB (1048576 bytes) of file size can be specified. The formats that can be entered are described below.</p> <ul style="list-style-type: none"> <li>• 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.</li> <li>• URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android <ul style="list-style-type: none"> <li>• file://</li> <li>• content://</li> </ul> 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.</li> </ul> <p><i>startIndex</i></p> <p>Start number of the printed page The range is -1, 1 to the number of pages in the PDF file. When -1 is specified for <i>startIndex</i>, all pages are printed. When -1 is specified for <i>startIndex</i>, the value of <i>endIndex</i> is ignored. When 1 or more is specified for <i>startIndex</i>, pages from the page number specified in <i>startIndex</i> to the page number specified in <i>endIndex</i> are printed. When a value more than the value specified for <i>endIndex</i> is specified for <i>startIndex</i>, an error occurs. When an out-of-range value is specified for <i>startIndex</i> or <i>endIndex</i>, an error occurs.</p> <p><i>endIndex</i></p> <p>End number of the printed page The range is 1 to 2147483647. When a value more than the number of pages in the PDF file is specified for <i>endIndex</i>, pages from the page number specified in <i>startIndex</i> to the last page of the PDF file are printed.</p> <p><i>rotate</i></p> <p>Rotation direction of the image See "4.4.1(3)㉓ Image rotation direction (Rotate)" for available constants.</p> <p><i>imageScale</i></p> <p>Image scaling See "4.4.1(3)㉔ Image scaling (ImageScale)" for available constants. When <b>IMAGE_SCALE_WIDTH_FIT</b> is specified for <i>imageScale</i>, the image width is converted to the print width of the printer while maintaining the aspect ratio.</p>



<i>bottomMargin</i>	<p>Bottom margin (dot)  The range is -1, 0 to 2400.  When -1 is specified for <i>bottomMargin</i>, the image is created and printed while maintaining the bottom margin.  When a value between 0 and 2400 is specified for <i>bottomMargin</i>, the bottom margin is changed to the specified size.  When a value between 0 and 2400 is specified for <i>bottomMargin</i>, blank pages are not printed.</p>
<i>dithering</i>	<p>Dithering  See "4.4.1(3)① Dithering (Dithering)" for available constants.</p>
<i>alignment</i>	<p>Alignment  See "4.4.1(3)⑨ Alignment (PrintAlignment)" for available constants.</p>
Exception	<p><b>PrinterException</b>  <b>PrinterException</b> is thrown when an error occurs while this method is being called.  See "<b>4.4.6 PrinterException Class</b>" for details of the error.  When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>
Description	<p>PDF is converted to printable format for the printer and is sent to the printer.</p> <p>The color image data is converted to monochrome image by binarization.</p> <p>When the memory switches MS 1-2 (Mark/Gap Mode Selection) of the printer is set to "Enable", form feed of marked paper or label is performed after printing the PDF page.</p>
Note	<p>No guarantee of printing operation when more than 100 pages are printed at a time.</p>

## printLogo

## Print logo

Prints the registered logo.

Syntax	public void <b>printLogo</b> (String <i>id</i> , PrintAlignment <i>alignment</i> ) throws <b>PrinterException</b>	
Parameter	<i>id</i>	Logo ID to print (key code) Specify the ID of the registered logo as a character string.
	<i>alignment</i>	Alignment See "4.4.1(3)⑨ Alignment (PrintAlignment)" for available constants.
Exception	<p><b>PrinterException</b>  <b>PrinterException</b> is thrown when an error occurs while this method is being called.  See "<b>4.4.6 PrinterException Class</b>" for details of the error.  When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>	

## printSmartLabelImageData

## Print label

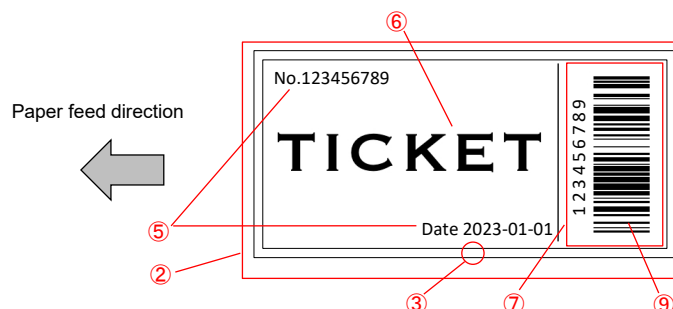
Prints labels.

Syntax	public void <b>printSmartLabelImageData</b> (SmartLabelManager <i>labelManager</i> ) throws <b>PrinterException</b>	
Parameter	<i>labelManager</i>	Instance of <b>SmartLabelManager</b> class

Exception	<p><b>PrinterException</b></p> <p><b>PrinterException</b> is thrown when an error occurs while this method is being called. See "<b>4.4.6 PrinterException Class</b>" for details of the error.</p> <p>When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.</p>
Description	See " <b>4.4.9 SmartLabelManager Class</b> " for the print example using this method.

### ③ 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

Starts page mode.

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

Exception    **PrinterException**

**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.

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** discards the print data kept in the page data buffer and changes the mode to standard mode.

Executing **printPageMode** prints the print data kept in the page data buffer.

## exitPageMode

## End page mode

Ends page mode and changes the mode to standard mode.

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

Exception    **PrinterException**

**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.

Description Discards the print data kept in the page data buffer and changes the mode to standard mode.

## setPageModeArea

## Specify print area of page mode

Specifies print area of page mode.

Syntax      public void **setPageModeArea**(int *x*,  
   int *y*,  
   int *width*,  
   int *height*) throws **PrinterException**

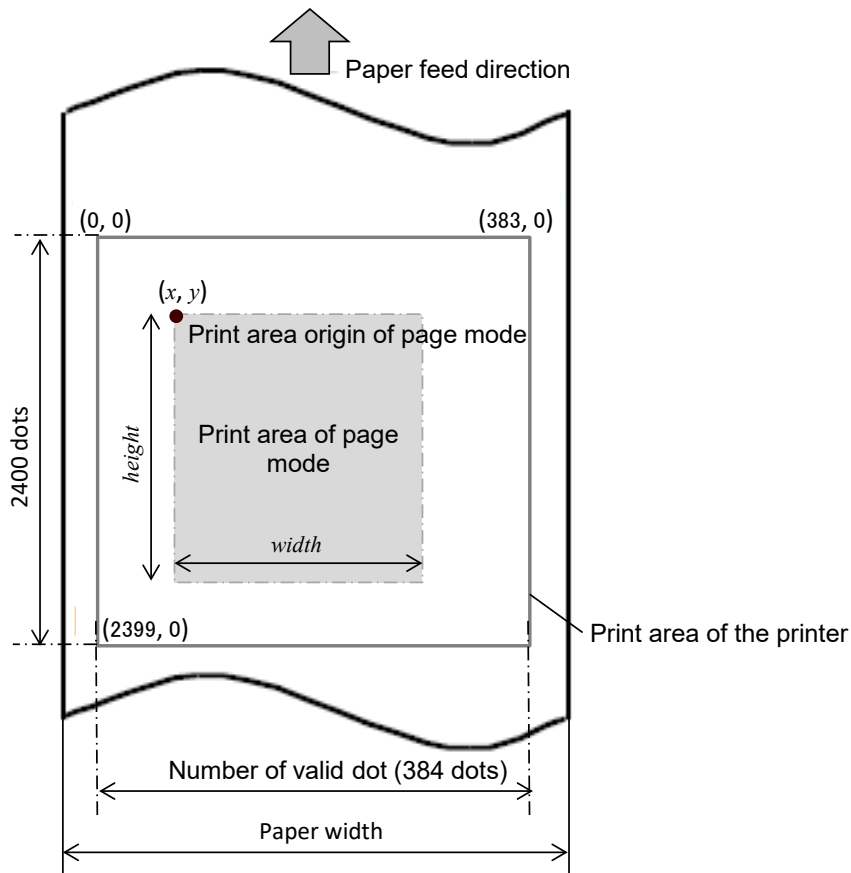
Parameter    *x*                    The horizontal origin (dot) of the print area of page mode  
                                 The valid range is 0 to (print width - 1).  
                                 0 represents the left edge on the print area of the printer.

*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  
                                 The valid range is 1 to (print width - *x*).

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

The relation between the print area of page mode and the print area of the printer is shown in figure below using an example of a print width of 384 dots.



**Exception `PrinterException`**

**`PrinterException`** is thrown when an error occurs while this method is being called. See "**4.4.6 `PrinterException` Class**" for details of the error.

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

Starting page mode by **`enterPageMode`** and executing this method after the dedicated method for page mode is executed, the print area of page mode can be additionally specified. 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  $x = 0$ ,  $y = 0$ ,  $width = \text{print width}$ ,  $height = 2400$  after executing **`enterPageMode`**.

**Reference** The print width is the width of the memory switch MS 3 (Print Width) of the printer.

## setPageModeDirection

## Specify print direction of page mode

Specifies print direction of page mode.

Syntax	public void <b>setPageModeDirection</b> (Direction <i>direction</i> ) throws <b>PrinterException</b>		
Parameter	<i>direction</i>	Print direction See "4.4.1(3)㉔" Print direction (Direction)" for available constants.	
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error.		
Description	Start page mode by <b>enterPageMode</b> before executing this method.  The print direction is left to right after executing <b>enterPageMode</b> .		

## setPageModeLineSpacing

## Specify line spacing of page mode

Specifies line spacing of page mode.

Syntax	public void <b>setPageModeLineSpacing</b> (int <i>lineSpacing</i> ) throws <b>PrinterException</b>		
Parameter	<i>linespacing</i>	Line spacing (dot) of page mode The valid range is 0 to 255.	
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error.		
Description	Start page mode by <b>enterPageMode</b> before executing this method.  The line spacing is 34 dots after executing <b>enterPageMode</b> .		

## printPageMode

## Print page mode

Prints the print data kept in page data buffer.

Syntax	public void <b>printPageMode</b> (CuttingMethod <i>cuttingMethod</i> ) throws <b>PrinterException</b>		
Parameter	<i>cuttingMethod</i>	Cutting method See "4.4.1(3)㉔" Cutting method (CuttingMethod)" for available constants.	
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error. When the data transmission is failed, the communication with the printer is ended, and <b>PrinterException</b> may be thrown. See <b>isConnect</b> for verifying the connection state with the printer.		
Description	The print data is kept after printing. The print data is discarded at the timing of the following: <ul style="list-style-type: none"><li>•Execute <b>enterPageMode</b></li><li>•Execute <b>disconnect</b></li><li>•Execute <b>exitPageMode</b></li></ul>		

Maps the text data on the print area of page mode.

Syntax	public void <b>printPageModeText</b> (int <i>startX</i> , int <i>startY</i> , String <i>text</i> ) throws <b>PrinterException</b>	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Text data Data size that can be specified at 1 time is 16 KB (16384 bytes).
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 <b>PrinterException Class</b> " for details of the error.	
Description	This method encodes the specified text data to printable text data based on <b>setInternationalCharacter</b> and <b>setCodePage</b> .	
	Start page mode by <b>enterPageMode</b> before executing this method.	

Maps the format specified text data on the print area of page mode.

Syntax	public void <b>printPageModeTextEx</b> (int <i>startX</i> , int <i>startY</i> , String <i>text</i> , CharacterBold <i>bold</i> , CharacterUnderline <i>underline</i> , CharacterReverse <i>reverse</i> , CharacterFont <i>font</i> , CharacterScale <i>scale</i> ) throws <b>PrinterException</b>	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Text data Data size that can be specified at 1 time is 16 KB (16384 bytes).
	<i>bold</i>	Bold print See "4.4.1(3) ③ Bold print (CharacterBold)" for available constants.
	<i>underline</i>	Underline See "4.4.1(3) ④ Underline (CharacterUnderline)" for available constants.
	<i>reverse</i>	Reverse print See "4.4.1(3) ⑤ Reverse print (CharacterReverse)" for available constants.
	<i>font</i>	Font See "4.4.1(3) ⑦ Character font (CharacterFont)" for available constants.

*scale* Character scale  
See "4.4.1(3) ⑧ Character scale (CharacterScale)" for available constants.

Exception **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.

Description This method encodes the specified text data to printable text data based on **setInternationalCharacter** and **setCodePage**.

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

## printPageModeBarcode

Print barcode of page mode

Maps the barcode on the print area of page mode.

The method of syntax (a) specifies the barcode data by character string.

The method of syntax (b) specifies the barcode data by character string and specifies N:W ratio of the barcode.

The method of syntax (c) specifies the barcode data by the array of bytes.

The method of syntax (d) specifies the Customer Bar Code\_JP data by character string.

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 **printPageModeBarcode**(int *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.



<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
<i>barcodeSymbol</i>	BarcodeSymbol See "4.4.1(3) ① Barcode symbol (BarcodeSymbol)" for available constants and correspondent syntax.
<i>text(data)</i>	Barcode data The input conditions for barcode data are as follows.

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'	
Customer Bar Code_JP	7 to 20 characters	'0' to '9' 'A' to 'Z' ' '	Start code, check digit, and stop code are automatically added. It is needed to specify one of '0' to '9' at the first 7 characters. A' to 'Z' is calculated as 2 characters.
GS1 Databar Omni-directional	13 characters	'0' to '9'	Check digit is automatically added.
GS1 Databar Truncated	13 characters	'0' to '9'	Check digit is automatically added.
GS1 Databar Limited	13 characters	'0' to '9'	Check digit is automatically added.

Barcode	Number of Data	Inputtable Data Character String (Data)	Remarks
GS1 Databar Expanded	2 to 255 characters	' ' to '' '%' to '?' 'A' to 'Z' '_' 'a' to 'z' '{'	

*moduleSize* Barcode width  
See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.

*moduleHeight* Barcode height (dot)

- When *barcodeSymbol* is set to the following, the valid range is 1 to 255.

**BARCODE\_SYMBOL\_UPC\_A**  
**BARCODE\_SYMBOL\_UPC\_E**  
**BARCODE\_SYMBOL\_EAN13**  
**BARCODE\_SYMBOL\_JAN13**  
**BARCODE\_SYMBOL\_EAN8**  
**BARCODE\_SYMBOL\_JAN8**  
**BARCODE\_SYMBOL\_CODE39**  
**BARCODE\_SYMBOL\_CODE93**  
**BARCODE\_SYMBOL\_CODE128**  
**BARCODE\_SYMBOL\_ITF**  
**BARCODE\_SYMBOL\_CODABAR**  
**BARCODE\_SYMBOL\_EAN13\_ADDON**  
**BARCODE\_SYMBOL\_JAN13\_ADDON**

- When *barcodeSymbol* is set to the following, the valid range is different by *barcodeSymbol* and *moduleSize*.

<i>barcodeSymbol</i>		
	<i>moduleSize</i>	Valid Range
<b>BARCODE_SYMBOL_GS1_OMNI_DIRECTIONAL</b>		
	<b>BARCODE_MODULE_WIDTH_2</b>	66 to 255
	<b>BARCODE_MODULE_WIDTH_3</b>	99 to 255
	<b>BARCODE_MODULE_WIDTH_4</b>	132 to 255
	<b>BARCODE_MODULE_WIDTH_5</b>	165 to 255
	<b>BARCODE_MODULE_WIDTH_6</b>	198 to 255
<b>BARCODE_SYMBOL_GS1_TRUNCATED</b>		
	<b>BARCODE_MODULE_WIDTH_2</b>	26 to 255
	<b>BARCODE_MODULE_WIDTH_3</b>	39 to 255
	<b>BARCODE_MODULE_WIDTH_4</b>	52 to 255
	<b>BARCODE_MODULE_WIDTH_5</b>	65 to 255
	<b>BARCODE_MODULE_WIDTH_6</b>	78 to 255

<i>barcodeSymbol</i>		
	<i>moduleSize</i>	<b>Valid Range</b>
<b>BARCODE_SYMBOL_GS1_LIMITED</b>		
	<b>BARCODE_MODULE_WIDTH_2</b>	20 to 255
	<b>BARCODE_MODULE_WIDTH_3</b>	30 to 255
	<b>BARCODE_MODULE_WIDTH_4</b>	40 to 255
	<b>BARCODE_MODULE_WIDTH_5</b>	50 to 255
	<b>BARCODE_MODULE_WIDTH_6</b>	60 to 255
<b>BARCODE_SYMBOL_GS1_EXPANDED</b>		
	<b>BARCODE_MODULE_WIDTH_2</b>	68 to 255
	<b>BARCODE_MODULE_WIDTH_3</b>	102 to 255
	<b>BARCODE_MODULE_WIDTH_4</b>	136 to 255
	<b>BARCODE_MODULE_WIDTH_5</b>	170 to 255
	<b>BARCODE_MODULE_WIDTH_6</b>	204 to 255

*hriPosition* HRI character print position  
See "4.4.1(3) ⑬ HRI character print position (HriPosition)" for available constants.

*hriFont* HRI character font  
See "4.4.1(3) ⑦ Character font (CharacterFont)" for available constants.

*nwRatio* N:W ratio  
See "4.4.1(3) ⑭ 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.

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

Exception **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.

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

Note Map the print data of the barcode not to overlap the other print data.  
The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.

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

Maps PDF417 on the print area of page mode.

The method of syntax (a) specifies PDF417 symbol.

The method of syntax (b) is fixed to standard PDF417.

Syntax	(a) public void <b>printPageModePDF417</b> (int <i>startX</i> , int <i>startY</i> , String <i>text</i> , ErrorCorrection <i>ErrorCorrection</i> , int <i>row</i> , int <i>column</i> , ModuleSize <i>moduleSize</i> , int <i>moduleHeight</i> , Pdf417Symbol <i>pdf417Symbol</i> ) throws <b>PrinterException</b>	
	(b) public void <b>printPageModePDF417</b> (int <i>startX</i> , int <i>startY</i> , String <i>text</i> , ErrorCorrection <i>ErrorCorrection</i> , int <i>row</i> , int <i>column</i> , ModuleSize <i>moduleSize</i> , int <i>moduleHeight</i> ) throws <b>PrinterException</b>	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Barcode data
	<i>errorCorrection</i>	Error correction level See "4.4.1(3) ⑮ Error correction level (ErrorCorrection)" for available constants.
	<i>row</i>	The number of rows (row) The valid range is 0, 3 to 90. When 0 is specified, the number of rows is automatically set.
	<i>column</i>	The number of columns in data area The valid range is 0 to 30. When 0 is specified, the number of columns in the data area is automatically set.
	<i>moduleSize</i>	Nominal fine element width See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.
	<i>moduleHeight</i>	Module height (dot) The valid range is 2 to 127. When the module height is set smaller, some barcode scanners may not read it. Set 3 or more for normal use.
Exception	<i>pdf417Symbol</i>	Symbol of PDF417 See "4.4.1(3) ⑩ PDF417 symbol (Pdf417Symbol)" for available constants.
	<b>PrinterException</b>	<b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error.

Description	Start page mode by <b>enterPageMode</b> before executing this method.
Note	Map the print data of the barcode not to overlap the other print data. The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.
Reference	See "Appendix B Barcode Size List" for details of the barcode size.

## printPageModeQRcode

## Print QR Code of page mode

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

Syntax	(a) public void <b>printPageModeQRcode</b> (int <i>startX</i> , int <i>startY</i> , String <i>text</i> , ErrorCorrection <i>ErrorCorrection</i> , ModuleSize <i>moduleSize</i> ) throws <b>PrinterException</b>  (b) public void <b>printPageModeQRcode</b> (int <i>startX</i> , int <i>startY</i> , String <i>text</i> , ErrorCorrection <i>ErrorCorrection</i> , ModuleSize <i>moduleSize</i> , QrModel <i>model</i> ) throws <b>PrinterException</b>	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Barcode data The version for either syntax (a) or (b) is automatically set depending on the number of data specified on <i>text</i> .
	<i>errorCorrection</i>	Error correction level See "4.4.1(3) ⑮ Error correction level (ErrorCorrection)" for available constants.
	<i>moduleSize</i>	Module size See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.
	<i>model</i>	QR Code Model See "4.4.1(3) ⑰ QR Code Model (QrModel)" for available constants.
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See " <b>4.4.6 PrinterException Class</b> " for details of the error." for details of the error.	
Description	Start page mode by <b>enterPageMode</b> before executing this method.	
Note	Map the print data of the barcode not to overlap the other print data. The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.	
Reference	See "Appendix B Barcode Size List" for details of the barcode size.	

Maps Data Matrix on the print area of page mode.

Syntax	public void <b>printPageModeDataMatrix</b> (int <i>startX</i> , int <i>startY</i> , String <i>text</i> , DataMatrixModule <i>dataMatrixModule</i> , ModuleSize <i>moduleSize</i> ) throws <b>PrinterException</b>	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Barcode data
	<i>dataMatrixModule</i>	The number of Data Matrix modules See "4.4.1(3) ⑱ Data Matrix module (DataMatrixModule)" for available constants.
	<i>moduleSize</i>	Module size See "4.4.1(3) ⑳ Module size (ModuleSize)" for available constants.
Exception	<b>PrinterException</b> <b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error.	
Description	Start page mode by <b>enterPageMode</b> before executing this method.	
Note	Map the print data of the barcode not to overlap the other print data. The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.	
Reference	See "Appendix B Barcode Size List" for details of the barcode size.	

Maps MaxiCode on the print area of page mode.

Syntax	public void <b>printPageModeMaxiCode</b> (int <i>startX</i> , int <i>startY</i> , String <i>text</i> , MaxiCodeMode <i>maxiCodeMode</i> ) throws <b>PrinterException</b>	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Barcode data
	<ul style="list-style-type: none"> <li>When <i>maxiCodeMode</i> is <b>MAXI_CODE_2</b> Add the service class (3 digits), the country code (3 digits), and the postal code (9 digits) to the beginning of the data.</li> <li>When <i>maxiCodeMode</i> is <b>MAXI_CODE_3</b> Add the service class (3 digits), the country code (3 digits), and the postal code (6 digits) to the beginning of the data.</li> </ul>	

Exception	<p><b>PrinterException</b></p> <p><b>PrinterException</b> is thrown when an error occurs while this method is being called. See "<b>4.4.6 PrinterException Class</b>" for details of the error.</p>
Description	Start page mode by <b>enterPageMode</b> before executing this method.
Note	<p>Map the print data of the barcode not to overlap the other print data.</p> <p>The quiet zone is not secured. Set the quiet zone in accordance with the standard of the barcode symbol.</p>
Reference	See "Appendix B Barcode Size List" for details of the barcode size.

Maps GS1 Databar Stacked on the print area of page mode.

Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Barcode data Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.
	<i>moduleSize</i>	Module size See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.

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

Maps GS1 Databar Stacked Omni-directional on the print area of page mode.

4-76

Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Barcode data Enter 13 characters from '0' to '9'. The leading '01' is automatically added by the printer. The check digit is automatically calculated by the printer.
	<i>moduleHeight</i>	Barcode module height (the number of the modules) The valid range is 33 to 255.
	<i>moduleSize</i>	Module size See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.
Exception	<b>PrinterException</b>	<b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error.
Description	Start page mode by <b>enterPageMode</b> before executing this method.	
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.	

## printPageModeGS1DataBarExpandedStacked

Print GS1 Databar Expanded Stacked of page mode

Maps GS1 Databar Expanded Stacked on the print area of page mode.

Syntax	<pre>public void <b>printPageModeGS1DataBarExpandedStacked</b>(int <i>startX</i>,   int <i>startY</i>,   String <i>text</i>,   int <i>column</i>,   ModuleSize <i>moduleSize</i>) throws <b>PrinterException</b></pre>	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>text</i>	Barcode data Enter any number of characters using the following: ' ', '!', '"', '%', '&', ' ', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '_ ', '0' to '9', 'A' to 'Z', 'a' to 'z'. Enter '{1' to FNC1. Be sure to input the check digit because it is not automatically calculated by the printer.
	<i>column</i>	The number of columns Specify the number of the columns in 1 line. The valid range is the even number from 2 to 20.
	<i>moduleSize</i>	Module size See "4.4.1(3) ⑫ Module size (ModuleSize)" for available constants.
Exception	<b>PrinterException</b>	<b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error.



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

## Print Aztec Code of page mode

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.

## **sendPageModeBinary**

## **Send binary data of page mode**

Maps binary data on the print area of page mode.

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

Parameter *binary* Binary data  
Data size that can be specified at 1 time is 16 KB (16384 bytes).

Exception **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.

Description Start page mode by **enterPageMode** 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.

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**

Maps the image file on the print area of page mode.

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 image file  
The formats that can be entered are described below.

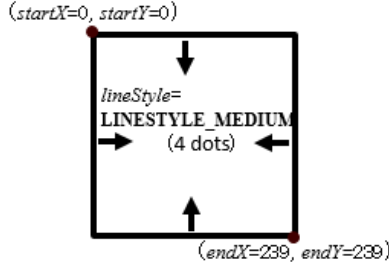
- Absolute path string handled by Java standard class "java.io.File"  
When targeting Android 10 (API 29) or later, please note that some files cannot be handled directly.  
See "3.5 Precautions - About Scoped Storage" for details.

- 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.



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

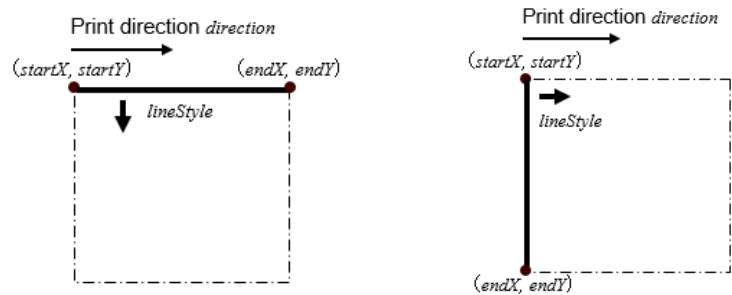
Image	Parameter
 <p>The diagram shows a square with a thick, dotted border. The top-left corner is labeled <math>(startX=0, startY=0)</math> and the bottom-right corner is labeled <math>(endX=239, endY=239)</math>. Inside the square, the text <code>lineStyle=</code> is followed by <code>LINESTYLE_MEDIUM</code> and <code>(4 dots)</code>. Arrows point from the text to the corresponding parts of the square's border.</p>	<pre>startX  0 startY  0 endX    239 endY    239 lineStyle LINESTYLE_MEDIUM</pre>

## printPageModeLine Print ruled line of page mode

Maps the ruled line on the print area of page mode.

Syntax	<pre>public void <b>printPageModeLine</b>(int <i>startX</i>,                                 int <i>startY</i>,                                 int <i>endX</i>,                                 int <i>endY</i>,                                 LineStyle <i>LineStyle</i>) throws <b>PrinterException</b></pre>	
Parameter	<i>startX</i>	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>startY</i>	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	<i>endX</i>	The horizontal reference point (dot) from the ending point The valid range is 0 to 2399.
	<i>endY</i>	The vertical reference point (dot) from the ending point The valid range is 0 to 2399.
	<i>LineStyle</i>	Line style See "4.4.1(3)②⑥ Line style (LineStyle)" for available constants.
Exception	<b>PrinterException</b>	<b>PrinterException</b> is thrown when an error occurs while this method is being called. See "4.4.6 <b>PrinterException Class</b> " for details of the error.
Description	Start page mode by <b>enterPageMode</b> before executing this method.  A diagonal stroke cannot be drawn by this method.	

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



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

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

Image	Parameter
<p>The diagram shows a square with two horizontal ruled lines. The top line is labeled ① and has coordinates (startX=0, startY=0) and (endX=239, endY=0). The bottom line is labeled ② and has coordinates (startX=0, startY=236) and (endX=239, endY=236). Both lines are labeled 'lineStyle= LINESYLE_MEDIUM (4 dots)' with a downward arrow.</p>	<p>①</p> <p>startX 0 startY 0 endX 239 endY 0 lineStyle LINESYLE_MEDIUM</p> <p>②</p> <p>startX 0 startY 236 endX 239 endY 236 lineStyle LINESYLE_MEDIUM</p>

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

Image	Parameter
<p>The diagram shows a square with two vertical ruled lines. The left line is labeled ① and has coordinates (startX=0, startY=0) and (endX=0, endY=239). The right line is labeled ② and has coordinates (startX=236, startY=0) and (endX=236, endY=239). Both lines are labeled 'lineStyle= LINESYLE_MEDIUM (4 dots)' with a rightward arrow.</p>	<p>①</p> <p>startX 0 startY 0 endX 0 endY 239 lineStyle LINESYLE_MEDIUM</p> <p>②</p> <p>startX 236 startY 0 endX 236 endY 239 lineStyle LINESYLE_MEDIUM</p>

Maps the registered logo on the print area of page mode.

**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

**Exception**    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.

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

#### 4.4.2 PrinterEvent Class

**PrinterEvent** class gets the end event that occurs when **startDiscoveryPrinter** is terminated.

##### (1) Method List

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

Name	Description
<b>getEventType</b>	Get end event

##### (2) End event constant

Constants used for getting the end event are shown in the following table.

Constant Name	Description	Value
<b>EVENT_FINISHED_DISCOVERY</b>	Completion of <b>startDiscoveryPrinter</b>	1
<b>EVENT_CANCELED_DISCOVERY</b>	Cancellation by <b>cancelDiscoveryPrinter</b>	2

##### (3) Method Details

<b>getEventType</b>	Get end event
---------------------	---------------

Gets the end event when **startDiscoveryPrinter** is terminated.

Syntax      `public int getEventType()`

Return value    See "4.4.2(2) End event constant" for details of the value.

Description    Whether **startDiscoveryPrinter** has been completed or the search has been canceled by **cancelDiscoveryPrinter** can be determined by the end event.  
Even when the printer was not discovered, **EVENT\_FINISHED\_DISCOVERY** is returned.

### 4.4.3 PrinterListener Interface

**PrinterListener** interface is for getting the end event when **startDiscoveryPrinter** is terminated.

#### (1) Method List

Methods of the **PrinterListener** interface are shown in the following table.

Name	Description
<b>finishEvent</b>	End event of printer search

#### (2) Method Details

<b>finishEvent</b>	End event of printer search
--------------------	-----------------------------

End event that is called when **startDiscoveryPrinter** is completed, or when **cancelDiscoveryPrinter** is executed.

Syntax      `public void finishEvent(PrinterEvent event)`

Parameter    *event*                      End event  
It is specified by **PrinterEvent** class.

Description    This method is an interface, so it is not implemented.  
Implement this method in the user application that receives the notification of the end event by completion of **startDiscoveryPrinter** or cancellation by **cancelDiscoveryPrinter**. Determine the type of the end event by **getEventType** in **PrinterEvent** class.



#### 4.4.4 PrinterInfo Class

**PrinterInfo** class stores the information of the printer found by **startDiscoveryPrinter**.

##### (1) Method List

Printer model name (Bluetooth device name), Bluetooth address, MAC address, IP address, port name (device path) and pairing status can be retrieved. Methods of **PrinterInfo** class are shown in the following table.

Name	Description
<b>getPrinterModelName</b>	Get printer model name
<b>getBluetoothAddress</b>	Get Bluetooth address
<b>getMacAddress</b>	Get MAC address
<b>getIpAddress</b>	Get IP address
<b>getIsBonded</b>	Get pairing status
<b>getDevicePath</b>	Get device path

##### (2) Method Details

#### **getPrinterModelName** Get printer model name

Gets the character string of the printer model name (Bluetooth device name) from the printer information found by **startDiscoveryPrinter**.

Syntax      `public String getPrinterModelName()`

Return value   Printer model name (Bluetooth device name)

#### **getBluetoothAddress** Get Bluetooth address

Gets the character string of the Bluetooth address from the printer information found by **startDiscoveryPrinter**.

Syntax      `public String getBluetoothAddress()`

Return value   Bluetooth address

#### **getMacAddress** Get MAC address

Gets the character string of the MAC address from the printer information found by **startDiscoveryPrinter**.

Syntax      `public String getMacAddress()`

Return value   MAC address

## **getIPAddress**

Get IP address

Gets the character string of the IP address from the printer information found by **startDiscoveryPrinter**.

Syntax        public String **getIpAddress()**

Return value   IP address

## **getIsBonded**

Get pairing status

Gets the status of pairing from the printer information found by **startDiscoveryPrinter**.

Syntax        public boolean **getIsBonded()**

Return value	true	Paired
	false	Not paired

## **getDevicePath**

Get device path

Gets the character string of the USB device file path from the printer information found by **startDiscoveryPrinter**.

Syntax        public String **getDevicePath()**

Return value   Device path

#### 4.4.5 PrinterIdentifier Class

**PrinterIdentifier** class stores the printer ID of the printer retrieved by **getPrinterResponse**.

##### (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.

Name	Description
<b>getPrinterModelID</b>	Get printer model ID
<b>getTypeID</b>	Get type ID
<b>getRomVersionID</b>	Get ROM version ID
<b>getMainFirmwareVersion</b>	Get firmware version (main)
<b>getManufacturer</b>	Get manufacturer
<b>getModelName</b>	Get model name
<b>getSerialNumber</b>	Get serial number
<b>getBootFirmwareVersion</b>	Get firmware version (boot)
<b>getBootFirmwareChecksum</b>	Get firmware checksum (boot)
<b>getMainFirmwareChecksum</b>	Get firmware checksum (main)
<b>getFirmwareChecksum</b>	Get firmware checksum (main + boot)

##### (2) Method Details

#### **getPrinterModelID** Get printer model ID

Gets the printer model ID from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax      public byte **getPrinterModelID()**

Return value   Printer model ID

#### **getTypeID** Get type ID

Gets the type ID from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax      public byte **getTypeID()**

Return value   Type ID

#### **getRomVersionID** Get ROM version ID

Gets the ROM version ID from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax      public byte **getRomVersionID()**

Return value   ROM version ID

**getMainFirmwareVersion****Get firmware version (main)**

Gets the firmware version (main) as a string from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax      `public String getMainFirmwareVersion()`

Return value   Firmware version (main)

**getManufacturer****Get manufacturer**

Gets the manufacturer name as a string from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax      `public String getManufacturer()`

Return value   Manufacturer name

**getModelName****Get model name**

Gets the model name as a string from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax      `public String getModelName()`

Return value   Model name

**getSerialNumber****Get serial number**

Gets the serial number as a string from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax      `public String getSerialNumber()`

Return value   Serial number

**getBootFirmwareVersion****Get firmware version (boot)**

Gets the firmware version (boot) as a string from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax      `public String getBootFirmwareVersion()`

Return value   Firmware version (boot)

**getBootFirmwareChecksum**

Get firmware checksum (boot)

Gets the firmware checksum (boot) from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax        public byte[] **getBootFirmwareChecksum()**

Return value   Firmware checksum (boot)

**getMainFirmwareChecksum**

Get firmware checksum (main)

Gets the firmware checksum (main) from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax        public byte[] **getMainFirmwareChecksum()**

Return value   Firmware checksum (main)

**getFirmwareChecksum**

Get firmware checksum (main + boot)

Gets the firmware checksum (main + boot) from the printer ID of the printer retrieved by **getPrinterResponse**.

Syntax        public byte[] **getFirmwareChecksum()**

Return value   Firmware checksum (main + boot)

#### 4.4.6 PrinterException Class

##### (1) Method List

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

Name	Description
<b>PrinterException</b>	Constructor
<b>getErrorCode</b>	Get error code

##### (2) Constant List

###### ① Error code

Constants used for getting error codes are shown in following table.

Constant Name	Description	Value
<b>ERROR_ACCESS_DENIED</b>	Failed to get the handle.* <sup>1</sup>	-1
	An unavailable port was specified.	
	An unsupported method was specified.	
<b>ERROR_SHARING_VIOLATION</b>	An already opened port was specified.	-11
<b>ERROR_PORT_NOT_OPENED</b>	The port is not opened.	-12
<b>ERROR_DEVICE_NOT_CONNECTED</b>	There is a problem with connection between the Android device and printer.	-21
<b>ERROR_DEVICE_INITIALIZE_FAILED</b>	Failed to change the printer setting. Data sending to the printer is not completed within the send timeout period, or data receiving from the printer is not completed within the receive timeout period.	-31
<b>ERROR_DATA_SIZE_ZERO</b>	0-byte data was specified.	-101
<b>ERROR_OVER_MAX_DATA_SIZE</b>	Maximum data size is exceeded.	-102
<b>ERROR_ENCODE_FAILED</b>	An error occurred in encoding text data.* <sup>1</sup>	-111
<b>ERROR_TIMEOUT</b>	Send timeout occurred.	-201
	Receive timeout occurred.	
<b>ERROR_FILE_NOT_FOUND</b>	The specified file is not found.	-301
<b>ERROR_FILE_USED</b>	The specified file is being used by another process.	-302
<b>ERROR_FILE_INVALID</b>	The specified file is invalid.	-303
<b>ERROR_LOW_MEMORY</b>	Memory shortage occurred when loading image file.	-311
<b>ERROR_OVER_MAX_IMAGE</b>	Either or both of width and height of image file exceeds the number of printable maximum dots.	-312
<b>ERROR_LOGO_NOT_DEFINED</b>	The logo is not registered.	-313
<b>ERROR_LOW_USER_AREA</b>	Remaining user area is insufficient.	-401
<b>ERROR_LOW_EXTERNAL_RAM</b>	Remaining RAM capacity is insufficient.	-402
<b>ERROR_PAGE_MODE_SPECIFIED</b>	Page mode is specified.	-511
<b>ERROR_PAGE_MODE_NOT_SPECIFIED</b>	Page mode is not specified.	-512
<b>ERROR_LABEL_FILE_NOT_SELECTED</b>	The label file is not selected.	-521

Constant Name	Description	Value
<b>ERROR_GET_LABEL_IMAGE</b>	Failed to create the label image.	-522
<b>ERROR_INVALID_PARAM</b>	Specified parameter is invalid.	-9999

\*1: Abnormal processing might have occurred.

### (3) Method Details

#### PrinterException

#### Constructor

Constructor for the **com.seikoinstruments.sdk.thermalprinter.PrinterException** class.

Syntax      `public PrinterException(int code, String message)`

#### getErrorCode

#### Get error codes

Gets the error code for thrown exception.

Syntax      `public int getErrorCode()`

Return value    See "4.4.6(2) Constant List" for details of the error.

#### 4.4.7 CallbackFunctionListener Interface

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

##### (1) Method List

Method of **CallbackFunctionListener** Interface is shown below.

Name	Description
<b>onStatusChanged</b>	Change event of printer status

##### (2) Method Details

<b>onStatusChanged</b>	Change event of printer status
------------------------	--------------------------------

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 the barcode scanner connection, barcode scanner disconnection, or received barcode data obtaining.

MP-B21L does not support this interface.

#### 4.4.9 SmartLabelManager Class

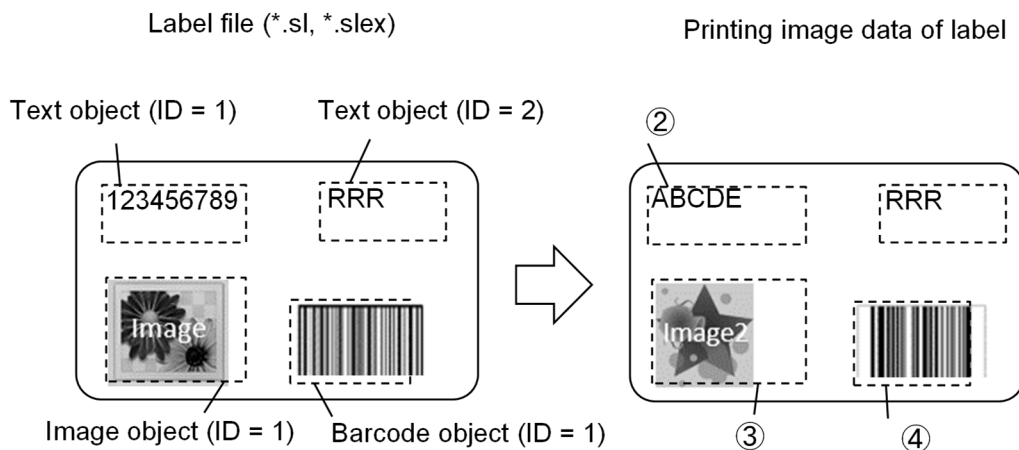
**SmartLabelManager** class provides the function to convert the label file (\*.sl, \*.slex) created using the app into the printable data from the printer.

##### (1) Method List

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

Name	Description
<b>SmartLabelManager</b>	Constructor
<b>selectSmartLabelFile</b>	Specify label file
<b>replaceSmartLabelTextData</b>	Replace text data of label
<b>replaceSmartLabelImageData</b>	Replace image data of label
<b>replaceSmartLabelBarcodeData</b>	Replace barcode data of label

The example of the procedure for replacing and printing data using the label file is described below.



① Specify a label file to print or replace data.

```
smartLabelManager.selectSmartLabelFile(labelFilePath);
```

② Replace text data.

```
smartLabelManager.replaceSmartLabelTextData(1, "ABCDE");
```

③ Replace image data.

```
smartLabelManager.replaceSmartLabelImageData(1, bitmap1);
```

④ Replace barcode data.

```
smartLabelManager.replaceSmartLabelBarcodeData(1, "123456789");
```

⑤ Print labels.

```
printerManager.printSmartLabelImageData(smartLabelManager);
```

**SmartLabelManager****Constructor**

Constructor for **com.seikoinstruments.sdk.thermalprinter.SmartLabelManager** class.

Syntax      `public SmartLabelManager(Context context)`

Parameter    *context*                      Specify application context to call this method.  
Example: **MainActivity.this**

**selectSmartLabelFile****Specify label file**

Specifies a label file (\*.sl, \*.slex).

Syntax      `public void selectSmartLabelFile(String filePath)` throws **PrinterException**

Parameter    *filePath*                      File path of label file (\*.sl, \*.slex) to use (\*.sl is deprecated.)  
The formats that can be entered are described below.

- Absolute path string handled by Java standard class "java.io.File"  
When the application targets 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 for this parameter to specify the URI string obtained from "Storage Access Framework". Please note that URI created without obtaining the URI string from "Storage Access Framework" may not be able to open the file.

Exception    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
See "**4.4.6 PrinterException Class**" for details of the error.

Description    The specified label file (\*.sl, \*.slex) is retained internally.  
After specifying the label file, the data of each object can be replaced.

The label files that can be used are restricted. See "4.2.1 Structure of Label File" for restrictions.

Note            The label files (\*.sl) are label file that will not be supported in the future.  
Use label files (\*.slex) that can be created using SII Layout Editor (app version 1.5.0 or later).

**replaceSmartLabelTextData****Replace text data of label**

Replaces the value of the text object of the label file (\*.sl, \*.slex).

Syntax      `public void replaceSmartLabelTextData(int mapID, String text)` throws **PrinterException**

Parameter    *mapID*                      ID of the text object  
Specify the ID of the text object mapped on the label file (\*.sl, \*.slex) of the app. The ID of the text object can be confirmed on the UI display of the app.  
When the specified *mapID* is not defined in the selected label, it is ignored.



UI display of the app

*text*

Text data to replace

Exception **PrinterException**

**PrinterException** is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error.

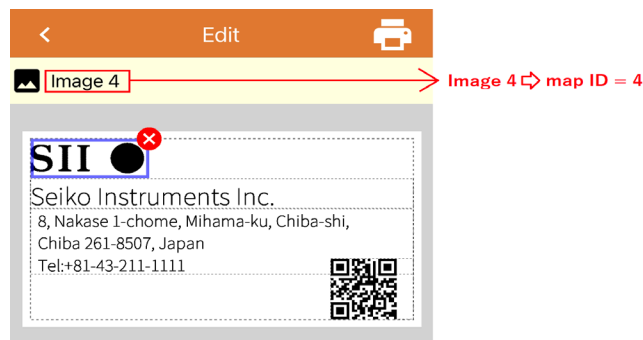
## replaceSmartLabelImageData

## Replace image data of label

Replaces the value of the image object of the label file (\*.sl, \*.slex).

Syntax `public void replaceSmartLabelImageData(int mapID, Bitmap bitmap) throws PrinterException`

Parameter *mapID* ID of the image object  
Specify the ID of the image object mapped on the label file (\*.sl, \*.slex) of the app. The ID of the image object can be confirmed on the UI display of the app.  
When the specified *mapID* is not defined in the selected label, it is ignored.



UI display of the app

*bitmap*

Image data to replace  
Specify image data conforming to the Android class "android.graphics.Bitmap".

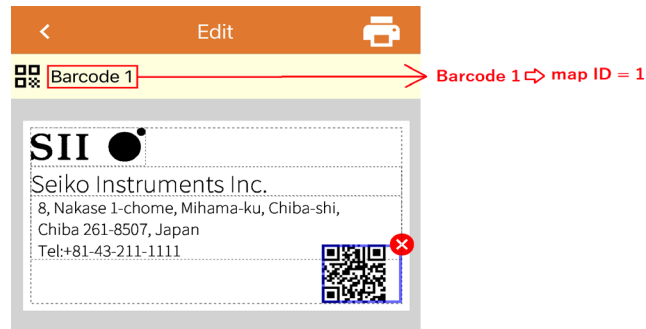
Exception **PrinterException**

**PrinterException** is thrown when an error occurs while this method is being called. See "4.4.6 PrinterException Class" for details of the error.

Replaces the value of the barcode object of the label file (\*.sl, \*.slex).

**Syntax**      `public void replaceSmartLabelBarcodeData(int mapID, String text) throws PrinterException`

**Parameter**    *mapID*                      ID of the barcode object  
    Specify the ID of the barcode object mapped on the label file (\*.sl, \*.slex) of the app. The ID of the barcode object can be confirmed on the UI display of the app.  
    When the specified *mapID* is not defined in the selected label, it is ignored.



UI display of the app

*text*                                      Text data to replace  
    Even if the text data to be replaced is invalid barcode data, an error is not caused. Make sure that the barcode data is valid before specifying it.

**Exception**    **PrinterException**  
**PrinterException** is thrown when an error occurs while this method is being called.  
 See "**4.4.6 PrinterException Class**" for details of the error.

## Chapter 5

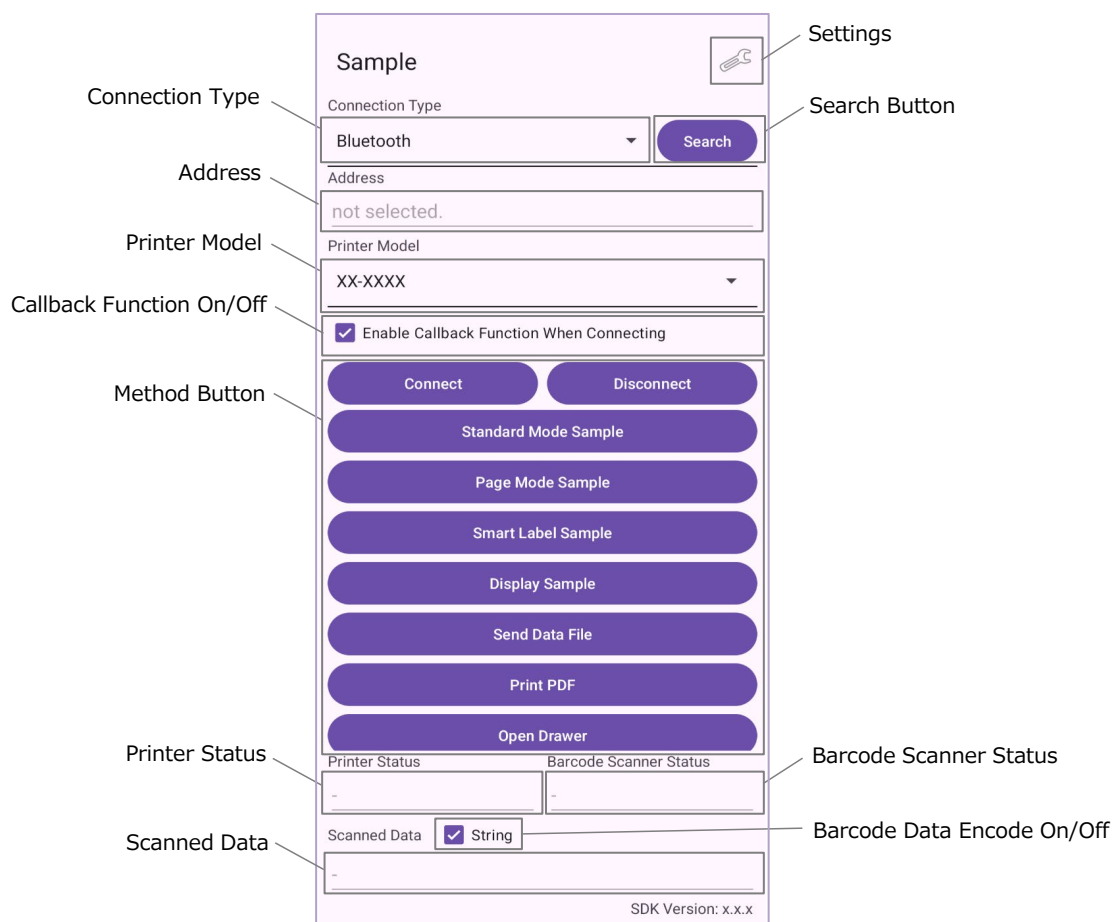
### Sample Program


This chapter describes the sample program provided by SII print class library.

#### 5.1 Screen Layout

SII print class library includes the sample program in Android Studio project format.  
This section describes the screen of the sample program.

##### 5.1.1 Main Screen

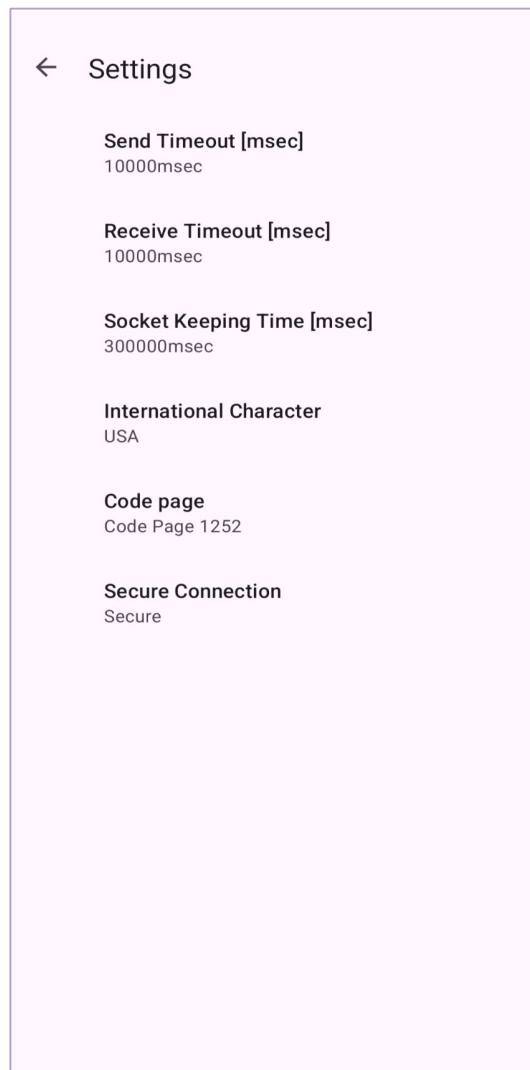


Item	Description
Settings	Tapping the [Settings] button opens the function setting screen. In order to go back to the main screen, tap  button on the top left of the screen.
Connection type	Selects the connection type to the printer.
Search Button	Starts searching for the type of printer specified in [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 <sup>*1</sup>	In addition to the method buttons for executing <b>connect</b> and <b>disconnect</b> , 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. MP-B21L does not support the barcode scanner.
Barcode Data Encode On/Off	Selects the conversion of barcode data read by the barcode scanner. MP-B21L does not support the barcode scanner.
Scanned Data	Displays the barcode data read by the barcode scanner. MP-B21L does not support the barcode scanner.

<sup>\*1</sup>: Supported functions vary by model. Only supported functions can be operated.

### 5.1.2 [Settings] Screen

Various setting functions are displayed in [Settings].



## 5.2 Precaution

The sample program is subject to change without notice.

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



## Appendix A

### Character Set

#### A.1 Codepage Table (Character Code Table)

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.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	φ	£	¥	℔	ƒ
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	¿	»		
B0	☐	☐	☐		†	‡	§	¶	§	§	§	§	§	§	§	§
C0	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞
D0	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞	⌞
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤	∫	∫	÷	≈	°	•	•	√	n	2	■	

Figure A-1 CODE\_PAGE\_437 (USA, Standard Europe)

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

Figure A-2 CODE\_PAGE\_KATAKANA

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	f
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©	¶	¶	¶	¶	¢	¥	₱
C0	⊥	⊥	⊥	⊥	⊥	ã	Ã	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	=	ℓ	α
D0	ð	Đ	Ê	Ë	È	Í	Î	Ï	⌋	⌋	■	■	■	■	■	■
E0	ó	β	ô	ò	õ	õ	μ	þ	þ	ú	û	ù	ý	ý	-	'
F0	-	±	=	¾	¶	§	÷	,	°	…	.	¹	³	²	■	

Figure A-3 CODE\_PAGE\_850 (Multilingual)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ã	à	Á	ç	ê	Ê	è	Í	Ô	ì	Ã	Â
90	É	À	È	ô	õ	ò	Ú	ù	Ì	Õ	Ü	¢	£	Ù	Þ	Ó
A0	á	í	ó	ú	ñ	Ñ	ä	ö	ï	ò	¬	½	¼	¡	«	»
B0	☐	☐	☐													
C0	L	L	T		-	+	+	+	+	+	+	+	+	+	+	+
D0	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	.	.	√	n	2	■	

Figure A-4 CODE\_PAGE\_860 (Portuguese)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	À	à	¶	ç	ê	ë	è	ï	î	≡	À	§
90	É	È	Ê	ô	Ë	Ï	Ô	Ù	⌘	Ô	Ü	¢	£	Ù	Ù	f
A0	!	'	ó	ú	..	³	-	î	¬	¬	½	¼	¾	«	»	
B0	☐	☐	☐													
C0	L	L	T		-	+	+	+	+	+	+	+	+	+	+	+
D0	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	.	.	√	n	2	■	

Figure A-5 CODE\_PAGE\_863 (Canadian-French)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	ƒ	
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	¡	«	»	
B0	☐	☐	☐													
C0	L	L	T	T	T	T	T	T	T	T	T	T	T	T	T	T
D0	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌
E0	α	β	Γ	π	Σ	σ	μ	τ	φ	θ	Ω	δ	∞	φ	ε	Π
F0	≡	±	≥	≤		J	÷	≈	°	.	.	√	n	2	■	

Figure A-6 CODE\_PAGE\_865 (Nordic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	Ş	ş
A0	á	í	ó	ú	ñ	Ñ	Ğ	ğ	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐													
C0	L	L	T	T	T	T	T	T	T	T	T	T	T	T	T	T
D0	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌	⌌
E0	ó	β	ô	ò	õ	õ	μ	×	ú	û	ü	ì	ÿ	-	'	
F0	-	±	¾	¶	§	÷	,	°	..	.	1	3	2	■		

Figure A-7 CODE\_PAGE\_857 (Turkish)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O	Π
90	P	Σ	T	Υ	Φ	X	Ψ	Ω	α	β	γ	δ	ε	ζ	η	θ
A0	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	ς	τ	υ	φ	χ	ψ
B0	⌘	⌘	⌘		†	‡	§	¶	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
C0	L	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
D0	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
E0	ω	ά	έ	ή	ï	í	ó	ύ	ÿ	ώ	À	É	Η	Ι	Ο	Υ
F0	Ω	±	≥	≤	İ	ÿ	÷	≈	°	•	•	√	n	2	■	

Figure A-8 CODE\_PAGE\_737 (Greek)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	‘	’	“	”	•	-	-	~	™	Š	<	Œ		Ž	
90											š	>	œ		ž	ÿ
A0	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯		
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figure A-9 CODE\_PAGE\_1252 (Latin)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
90	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
A0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
B0	␣	␣	␣		†	‡	§	¶	‡	§	¶	‡	§	¶	‡	§
C0	␣	␣	␣		†	‡	§	¶	‡	§	¶	␣	␣	␣	␣	␣
D0	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣	␣
E0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F0	Ё	ё	Є	є	İ	ı	Ÿ	ÿ	°	•	•	√	№	α	■	

Figure A-10 CODE\_PAGE\_866 (Russian)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	û	ç	ł	ë	ő	ö	î	ž	Ä	Ć	
90	É	Í	í	ô	ö	Ł	ł	Ś	ś	Ö	Ü	Ť	ť	Ł	×	č
A0	á	í	ó	ú	À	à	Ž	ž	Ę	ę	¬	ž	Č	š	«	»
B0	␣	␣	␣		†	‡	§	¶	‡	§	¶	␣	␣	␣	␣	␣
C0	␣	␣	␣		†	‡	§	¶	‡	§	¶	␣	␣	␣	␣	␣
D0	đ	Đ	Ď	Ě	ď	Ň	í	î	ě	Ĵ	Ĵ	■	■	Ť	Ů	■
E0	ó	ß	ô	ń	ň	š	š	ř	ú	ř	Ů	ý	Ý	ť	´	
F0	-	"	˘	˘	˘	§	÷	˘	˘	˘	˘	Ů	Ř	ř	■	

Figure A-11 CODE\_PAGE\_852 (Eastern Europe)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	â	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
A0	á	í	ó	ú	ñ	Ñ	ä	ö	¿	®	¬	½	¼	¡	«	»
B0	☐	☐	☐			Á	Â	À	©			¶	¶	¢	¥	₱
C0	L	└	└	└	└	└	ã	Ã	ℓ	ℓ	ℓ	ℓ	ℓ	=	ℓ	α
D0	ð	Ð	Ê	Ë	È	€	Í	Î	Ï	Ј	Г	■	■	І	Ì	■
E0	ó	β	ô	ò	õ	õ	μ	þ	þ	ú	û	ù	ý	Ý	-	'
F0	-	±	=	¾	¶	§	÷	,	°	..	.	1	3	2		■

Figure A-12 CODE\_PAGE\_858 (Euro)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	ђ	Ђ	ѓ	Ѓ	ё	Ё	є	Є	ѕ	Ѕ	і	І	ї	Ї	ј	Ј
90	љ	Љ	њ	Њ	ћ	Ћ	ќ	Ќ	џ	Џ	џ	џ	џ	џ	џ	џ
A0	а	А	б	Б	в	В	г	Г	д	Д	е	Е	ф	Ф	г	Г
B0	☐	☐	☐			х	Х	и	И			¶	¶	й	Й	₱
C0	L	└	└	└	└	└	к	К	ℓ	ℓ	ℓ	ℓ	ℓ	=	ℓ	α
D0	л	Л	м	М	н	Н	о	О	п	П	Г	■	■	П	я	■
E0	Я	р	Р	с	С	т	Т	у	У	ж	Ж	в	В	ь	Ь	№
F0	-	ы	Ы	э	Э	ш	Ш	э	Э	щ	Щ	ч	Ч	§		■

Figure A-13 CODE\_PAGE\_855 (Cyrillic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	”	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	°	•	•	√	■	-		+	+	+	+	+	+	+	+	+
90	β	∞	φ	±	½	¼	≈	«	»	لأ	لأ					لا
A0	-	ĩ	£	¤	£			ل	ب	ث	ج	ح	خ	ح	خ	خ
B0	•	١	٢	٣	٤	٥	٦	٧	٨	٩	ف	س	ش	ص	ش	؟
C0	φ	ء	آ	أ	ؤ	ع	ئ	ب	ة	ث	ت	ج	د	خ	د	د
D0	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ		÷	×	ع	ع
E0	-	ف	ق	ك	ل	م	ن	ه	و	ي	ض	ع	غ	غ	غ	م
F0	-	”	ن	ه	ه	ي	ي	غ	ي	غ	ي	غ	ي	غ	ي	■

Figure A-14 CODE\_PAGE\_864 (Arabic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	”	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	‘	’	“	”	…	†	‡	‰	Š	<	Š	Ť	Ž	Ž	
90		‘	’	“	”	•	-	-	™	š	>	š	ť	ž	ž	
A0	˘	˘	Ł	¤	Ą		Ś	..	©	§	«	¬	-	®	Ž	
B0	°	±	ł	´	μ	¶	•	ą	§	»	Ł	”	ł	ž	ž	
C0	Ř	Á	Ā	Ǽ	Ä	Ć	Č	Č	É	Ě	Ě	Ě	Í	Î	Ď	
D0	Đ	Ň	Ň	Ó	Ô	Õ	Ö	×	Ř	Ů	Ú	Ú	Ú	Ý	Ť	ß
E0	ř	á	â	ǻ	ä	í	ć	č	é	ě	ě	ě	í	î	ď	
F0	đ	ň	ň	ó	ô	õ	ö	÷	ř	ů	ú	ú	ú	ý	ť	·

Figure A-15 CODE\_PAGE\_1250 (Central European)



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	ђ	ѓ	;	ѓ	„	...	†	‡	€	‰	Љ	<	Њ	ќ	ћ	џ
90	ђ	‘	;	“	”	•	-	-	™	Љ	>	њ	ќ	ћ	џ	
A0	ÿ	ÿ	Ј	Ѡ	Г	І	Š	Ě	©	€	«	¬	-	®	İ	
B0	°	±	İ	ı	г	μ	¶	•	ё	№	»	j	S	s	ı	
C0	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
D0	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
E0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
F0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я

Figure A-16 CODE\_PAGE\_1251 (Cyrillic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	‘	;	ƒ	„	...	†	‡	‰		<					
90		‘	;	“	”	•	-	-	™		>					
A0	ˆ	Â	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	-	
B0	°	±	²	³	´	μ	¶	·	¸	¹	º	»	¼	½	¾	¸
C0	ı	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
D0	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	İ	ÿ	ά	έ	ή	ί
E0	ύ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
F0	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ï	ÿ	ό	ύ	ώ	

Figure A-17 CODE\_PAGE\_1253 (Greek)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	€	‘	’	“	”	…	†	‡	^	‰	Š	<	Œ			
90		‚	‚	„	„	•	-	-	~	™	š	>	œ			ÿ
A0	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯		
B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ğ	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	İ	Ş	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ğ	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ı	ş	ÿ

Figure A-18 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	#	α	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
COUNTRY_ITALY	#	\$	@	°	\	é	^	ù	à	ò	è	ì
COUNTRY_SPAIN	£	\$	@	ı	Ñ	ı	^	`	..	ñ	}	~
COUNTRY_JAPAN	#	\$	@	[	¥	]	^	`	{		}	~
COUNTRY_NORWAY	#	α	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	ı	Ñ	ı	é	`	í	ñ	ó	ú
COUNTRY_LATIN_AMERICA	#	\$	á	ı	Ñ	ı	é	ü	í	ñ	ó	ú
COUNTRY_ARABIA	#	\$	@	[	\	]	^	`	{		}	~

Figure A-19 International Character Set

## Appendix B

### Barcode Size List

#### B.1 Barcode Size List

##### B.1.1 printBarcode, printPageModeBarcode



##### (1) Height of the barcode image

<i>barcodeSymbol</i>	<i>hriFont</i>	<i>hriPosition</i>	Length from Top of Barcode to Reference Point	Height of Barcode Image
BARCODE_UPC_A BARCODE_UPC_E BARCODE_EAN13 BARCODE_JAN13 BARCODE_EAN8 BARCODE_JAN8 BARCODE_CODE39 BARCODE_CODE93 BARCODE_CODE128 BARCODE_ITF BARCODE_CODABAR BARCODE_EAN13_ADDON BARCODE_JAN13_ADDON BARCODE_GS1_OMNI_DIRECTIONAL BARCODE_GS1_TRUNCATED BARCODE_GS1_LIMITED BARCODE_GS1_EXPANDED	FONT_A	HRI_NONE	<i>moduleHeight</i>	<i>moduleHeight</i>
		HRI_POSITION_ABOVE	<i>moduleHeight</i> + 32	<i>moduleHeight</i> + 32
		HRI_POSITION_BELOW	<i>moduleHeight</i>	<i>moduleHeight</i> + 32
		HRI_POSITION_ABOVE_BELOW	<i>moduleHeight</i> + 64	<i>moduleHeight</i> + 64
	FONT_B	HRI_NONE	<i>moduleHeight</i>	<i>moduleHeight</i>
		HRI_POSITION_ABOVE	<i>moduleHeight</i> + 24	<i>moduleHeight</i> + 24
		HRI_POSITION_BELOW	<i>moduleHeight</i>	<i>moduleHeight</i> + 24
		HRI_POSITION_ABOVE_BELOW	<i>moduleHeight</i> + 48	<i>moduleHeight</i> + 48

<i>barcodeSymbol</i>	<i>moduleSize</i>	Width of Barcode Image
BARCODE_CUSTOMER_BAR_CODE_JP	CUSTOMERBARCODEJP_MODULE_4	12
	CUSTOMERBARCODEJP_MODULE_5	15
	CUSTOMERBARCODEJP_MODULE_6	18
	CUSTOMERBARCODEJP_MODULE_7	21
	CUSTOMERBARCODEJP_MODULE_8	24
	CUSTOMERBARCODEJP_MODULE_9	27
	CUSTOMERBARCODEJP_MODULE_10	30

(2) Width of the barcode image

<i>barcodeSymbol</i>	<i>moduleSize</i>	Width of Barcode Image
BARCODE_UPC_A	BARCODE_MODULE_WIDTH_2	190
	BARCODE_MODULE_WIDTH_3	285
	BARCODE_MODULE_WIDTH_4	380
	BARCODE_MODULE_WIDTH_5	475
	BARCODE_MODULE_WIDTH_6	570
BARCODE_UPC_E	BARCODE_MODULE_WIDTH_2	102
	BARCODE_MODULE_WIDTH_3	153
	BARCODE_MODULE_WIDTH_4	204
	BARCODE_MODULE_WIDTH_5	255
	BARCODE_MODULE_WIDTH_6	306
BARCODE_EAN13	BARCODE_MODULE_WIDTH_2	190
	BARCODE_MODULE_WIDTH_3	285
	BARCODE_MODULE_WIDTH_4	380
	BARCODE_MODULE_WIDTH_5	475
	BARCODE_MODULE_WIDTH_6	570
BARCODE_JAN13	BARCODE_MODULE_WIDTH_2	190
	BARCODE_MODULE_WIDTH_3	285
	BARCODE_MODULE_WIDTH_4	380
	BARCODE_MODULE_WIDTH_5	475
	BARCODE_MODULE_WIDTH_6	570
BARCODE_EAN8	BARCODE_MODULE_WIDTH_2	134
	BARCODE_MODULE_WIDTH_3	201
	BARCODE_MODULE_WIDTH_4	268
	BARCODE_MODULE_WIDTH_5	335
	BARCODE_MODULE_WIDTH_6	402
BARCODE_JAN8	BARCODE_MODULE_WIDTH_2	134
	BARCODE_MODULE_WIDTH_3	201
	BARCODE_MODULE_WIDTH_4	268
	BARCODE_MODULE_WIDTH_5	335
	BARCODE_MODULE_WIDTH_6	402

<i>barcodeSymbol</i>	<i>moduleSize</i>	<b>Width of Barcode Image</b>
<b>BARCODE_CODE93</b>	<b>BARCODE_MODULE_WIDTH_2</b>	18 × number of barcode data + 56
	<b>BARCODE_MODULE_WIDTH_3</b>	27 × number of barcode data + 84
	<b>BARCODE_MODULE_WIDTH_4</b>	36 × number of barcode data + 112
	<b>BARCODE_MODULE_WIDTH_5</b>	45 × number of barcode data + 140
	<b>BARCODE_MODULE_WIDTH_6</b>	54 × number of barcode data + 168
<b>BARCODE_CODE128</b>	<b>BARCODE_MODULE_WIDTH_2</b>	22 × number of barcode data + 26
	<b>BARCODE_MODULE_WIDTH_3</b>	33 × number of barcode data + 39
	<b>BARCODE_MODULE_WIDTH_4</b>	44 × number of barcode data + 52
	<b>BARCODE_MODULE_WIDTH_5</b>	55 × number of barcode data + 65
	<b>BARCODE_MODULE_WIDTH_6</b>	66 × number of barcode data + 78
<b>BARCODE_CUSTOMER_BAR_CODE_JP</b>	<b>CUSTOMERBARCODEJP_MODULE_4</b>	266
	<b>CUSTOMERBARCODEJP_MODULE_5</b>	333
	<b>CUSTOMERBARCODEJP_MODULE_6</b>	399
	<b>CUSTOMERBARCODEJP_MODULE_7</b>	466
	<b>CUSTOMERBARCODEJP_MODULE_8</b>	532
	<b>CUSTOMERBARCODEJP_MODULE_9</b>	599
<b>BARCODE_GS1_OMNI_DIRECTIONAL</b>	<b>BARCODE_MODULE_WIDTH_2</b>	192
	<b>BARCODE_MODULE_WIDTH_3</b>	288
	<b>BARCODE_MODULE_WIDTH_4</b>	384
	<b>BARCODE_MODULE_WIDTH_5</b>	480
	<b>BARCODE_MODULE_WIDTH_6</b>	576
<b>BARCODE_GS1_TRUNCATED</b>	<b>BARCODE_MODULE_WIDTH_2</b>	192
	<b>BARCODE_MODULE_WIDTH_3</b>	288
	<b>BARCODE_MODULE_WIDTH_4</b>	384
	<b>BARCODE_MODULE_WIDTH_5</b>	480
	<b>BARCODE_MODULE_WIDTH_6</b>	576
<b>BARCODE_GS1_LIMITED</b>	<b>BARCODE_MODULE_WIDTH_2</b>	158
	<b>BARCODE_MODULE_WIDTH_3</b>	237
	<b>BARCODE_MODULE_WIDTH_4</b>	316
	<b>BARCODE_MODULE_WIDTH_5</b>	395
	<b>BARCODE_MODULE_WIDTH_6</b>	474

<i>barcodeSymbol</i>	<i>moduleSize</i>	Width of Barcode Image
<b>BARCODE_GS1_EXPANDED</b> *1	<b>BARCODE_MODULE_WIDTH_2</b>	number of barcode module × 2
	<b>BARCODE_MODULE_WIDTH_3</b>	number of barcode module × 3
	<b>BARCODE_MODULE_WIDTH_4</b>	number of barcode module × 4
	<b>BARCODE_MODULE_WIDTH_5</b>	number of barcode module × 5
	<b>BARCODE_MODULE_WIDTH_6</b>	number of barcode module × 6

\*1: The number of barcode module is determined by the barcode data to be specified.

<i>barcodeSymbol</i>	<i>nwRatio</i>	<i>moduleSize</i>	Width of Barcode Image
<b>BARCODE_CODE39</b>	<b>NWRATIO_1TO2</b>	<b>BARCODE_MODULE_WIDTH_2</b>	26 × number of barcode data + 50
		<b>BARCODE_MODULE_WIDTH_3</b>	39 × number of barcode data + 75
		<b>BARCODE_MODULE_WIDTH_4</b>	52 × number of barcode data + 100
		<b>BARCODE_MODULE_WIDTH_5</b>	65 × number of barcode data + 125
		<b>BARCODE_MODULE_WIDTH_6</b>	78 × number of barcode data + 150
	<b>NWRATIO_1TO2_5</b>	<b>BARCODE_MODULE_WIDTH_2</b>	29 × number of barcode data + 56
		<b>BARCODE_MODULE_WIDTH_3</b>	45 × number of barcode data + 87
		<b>BARCODE_MODULE_WIDTH_4</b>	58 × number of barcode data + 112
		<b>BARCODE_MODULE_WIDTH_5</b>	74 × number of barcode data + 143
		<b>BARCODE_MODULE_WIDTH_6</b>	87 × number of barcode data + 168
	<b>NWRATIO_1TO3</b>	<b>BARCODE_MODULE_WIDTH_2</b>	32 × number of barcode data + 62
		<b>BARCODE_MODULE_WIDTH_3</b>	48 × number of barcode data + 93
		<b>BARCODE_MODULE_WIDTH_4</b>	64 × number of barcode data + 124
		<b>BARCODE_MODULE_WIDTH_5</b>	80 × number of barcode data + 155
		<b>BARCODE_MODULE_WIDTH_6</b>	96 × number of barcode data + 186

<i>barcodeSymbol</i>	<i>nwRatio</i>	<i>moduleSize</i>	<b>Width of Barcode Image</b>
<b>BARCODE_ITF</b>	<b>NWRATIO_1TO2</b>	<b>BARCODE_MODULE_WIDTH_2</b>	14 × number of barcode data + 16
		<b>BARCODE_MODULE_WIDTH_3</b>	21 × number of barcode data + 24
		<b>BARCODE_MODULE_WIDTH_4</b>	28 × number of barcode data + 32
		<b>BARCODE_MODULE_WIDTH_5</b>	35 × number of barcode data + 40
		<b>BARCODE_MODULE_WIDTH_6</b>	42 × number of barcode data + 48
	<b>NWRATIO_1TO2_5</b>	<b>BARCODE_MODULE_WIDTH_2</b>	16 × number of barcode data + 17
		<b>BARCODE_MODULE_WIDTH_3</b>	25 × number of barcode data + 26
		<b>BARCODE_MODULE_WIDTH_4</b>	32 × number of barcode data + 34
		<b>BARCODE_MODULE_WIDTH_5</b>	41 × number of barcode data + 43
		<b>BARCODE_MODULE_WIDTH_6</b>	48 × number of barcode data + 51
	<b>NWRATIO_1TO3</b>	<b>BARCODE_MODULE_WIDTH_2</b>	18 × number of barcode data + 18
		<b>BARCODE_MODULE_WIDTH_3</b>	27 × number of barcode data + 27
		<b>BARCODE_MODULE_WIDTH_4</b>	36 × number of barcode data + 36
		<b>BARCODE_MODULE_WIDTH_5</b>	45 × number of barcode data + 45
		<b>BARCODE_MODULE_WIDTH_6</b>	54 × number of barcode data + 54
<b>BARCODE_CODABAR*<sup>1</sup></b>	<b>NWRATIO_1TO2</b>	<b>BARCODE_MODULE_WIDTH_2</b>	20 × number of data + 2 × (2 + number of wide data) - 2
		<b>BARCODE_MODULE_WIDTH_3</b>	30 × number of data + 3 × (2 + number of wide data) - 3
		<b>BARCODE_MODULE_WIDTH_4</b>	40 × number of data + 4 × (2 + number of wide data) - 4
		<b>BARCODE_MODULE_WIDTH_5</b>	50 × number of data + 5 × (2 + number of wide data) - 5
		<b>BARCODE_MODULE_WIDTH_6</b>	60 × number of data + 6 × (2 + number of wide data) - 6



<i>barcodeSymbol</i>	<i>nwRatio</i>	<i>moduleSize</i>	<b>Width of Barcode Image</b>
<b>BARCODE_CODABAR*1</b>	<b>NWRATIO_1TO2_5</b>	<b>BARCODE_MODULE_WIDTH_2</b>	22 × number of data + 3 × (2 + number of wide data) - 2
		<b>BARCODE_MODULE_WIDTH_3</b>	34 × number of data + 5 × (2 + number of wide data) - 3
		<b>BARCODE_MODULE_WIDTH_4</b>	44 × number of data + 6 × (2 + number of wide data) - 4
		<b>BARCODE_MODULE_WIDTH_5</b>	56 × number of data + 8 × (2 + number of wide data) - 5
		<b>BARCODE_MODULE_WIDTH_6</b>	66 × number of data + 9 × (2 + number of wide data) - 6
	<b>NWRATIO_1TO3</b>	<b>BARCODE_MODULE_WIDTH_2</b>	24 × number of data + 4 × (2 + number of wide data) - 2
		<b>BARCODE_MODULE_WIDTH_3</b>	36 × number of data + 6 × (2 + number of wide data) - 3
		<b>BARCODE_MODULE_WIDTH_4</b>	48 × number of data + 8 × (2 + number of wide data) - 4
		<b>BARCODE_MODULE_WIDTH_5</b>	60 × number of data + 10 × (2 + number of wide data) - 5
		<b>BARCODE_MODULE_WIDTH_6</b>	72 × number of data + 12 × (2 + number of wide data) - 6

\*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 " : / . + ".

<i>barcodeSymbol</i>	<b>Number of Data</b>	<i>moduleSize</i>	<b>Width of Barcode Image</b>
<b>BARCODE_EAN13_ADDON</b>	14 or 15	<b>BARCODE_MODULE_WIDTH_2</b>	244
		<b>BARCODE_MODULE_WIDTH_3</b>	366
		<b>BARCODE_MODULE_WIDTH_4</b>	488
		<b>BARCODE_MODULE_WIDTH_5</b>	610
		<b>BARCODE_MODULE_WIDTH_6</b>	732
	17 or 18	<b>BARCODE_MODULE_WIDTH_2</b>	298
		<b>BARCODE_MODULE_WIDTH_3</b>	447
		<b>BARCODE_MODULE_WIDTH_4</b>	596
		<b>BARCODE_MODULE_WIDTH_5</b>	745
		<b>BARCODE_MODULE_WIDTH_6</b>	894
<b>BARCODE_JAN13_ADDON</b>	14 or 15	<b>BARCODE_MODULE_WIDTH_2</b>	244
		<b>BARCODE_MODULE_WIDTH_3</b>	366
		<b>BARCODE_MODULE_WIDTH_4</b>	488
		<b>BARCODE_MODULE_WIDTH_5</b>	610
		<b>BARCODE_MODULE_WIDTH_6</b>	732
	17 or 18	<b>BARCODE_MODULE_WIDTH_2</b>	298
		<b>BARCODE_MODULE_WIDTH_3</b>	447
		<b>BARCODE_MODULE_WIDTH_4</b>	596
		<b>BARCODE_MODULE_WIDTH_5</b>	745
		<b>BARCODE_MODULE_WIDTH_6</b>	894

## B.1.2 printPDF417, printPageModePDF417



### (1) Height of the barcode image

$$\text{Height of the barcode image}^{*1} = \text{moduleHeight} \times \text{row}^{*2}$$

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

\*2:  $\text{row} \neq 0$

### (2) Width of the barcode image

When *pdf417Symbol* is **PDF417\_STANDARD**:

$$\text{Width of the barcode image} = (17 \times \text{column}^{*1} + 69) \times \text{module size value}$$

\*1:  $\text{column} \neq 0$

When *pdf417Symbol* is **PDF417\_COMPACT**:

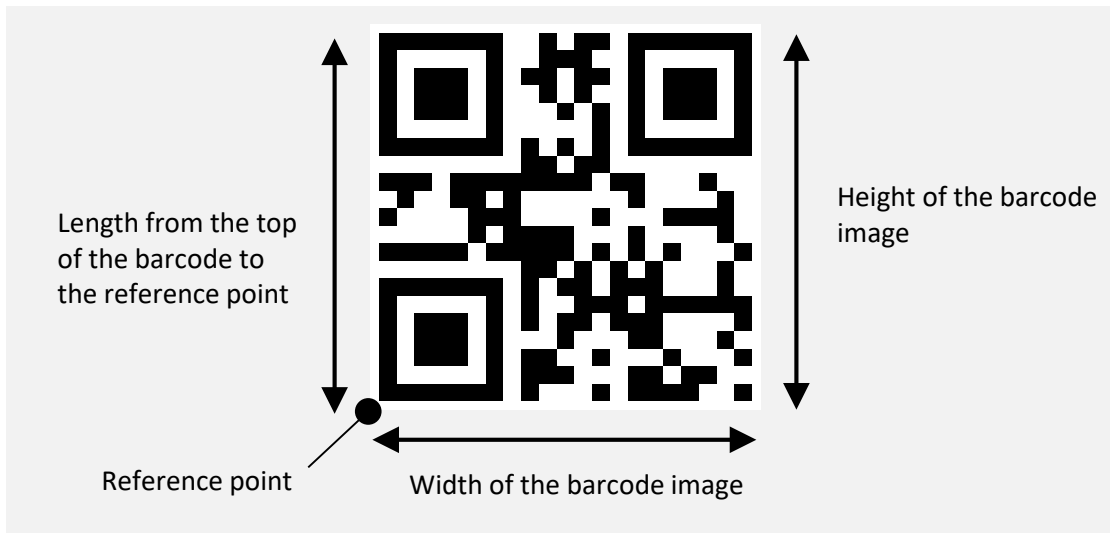
$$\text{Width of the barcode image} = (17 \times \text{column}^{*1} + 35) \times \text{module size value}$$

\*1:  $\text{column} \neq 0$

### Module Size Value

<i>moduleSize</i>	Module Size Value
<b>PDF417_MODULE_WIDTH_2</b>	2
<b>PDF417_MODULE_WIDTH_3</b>	3
<b>PDF417_MODULE_WIDTH_4</b>	4
<b>PDF417_MODULE_WIDTH_5</b>	5
<b>PDF417_MODULE_WIDTH_6</b>	6
<b>PDF417_MODULE_WIDTH_7</b>	7
<b>PDF417_MODULE_WIDTH_8</b>	8

### B.1.3 printQRCode, printPageModeQRCode



#### (1) Height and width of the barcode image

Height<sup>\*1</sup> and width of the barcode image =  $(4 \times \text{version}^2 + 17) \times \text{module size value}$

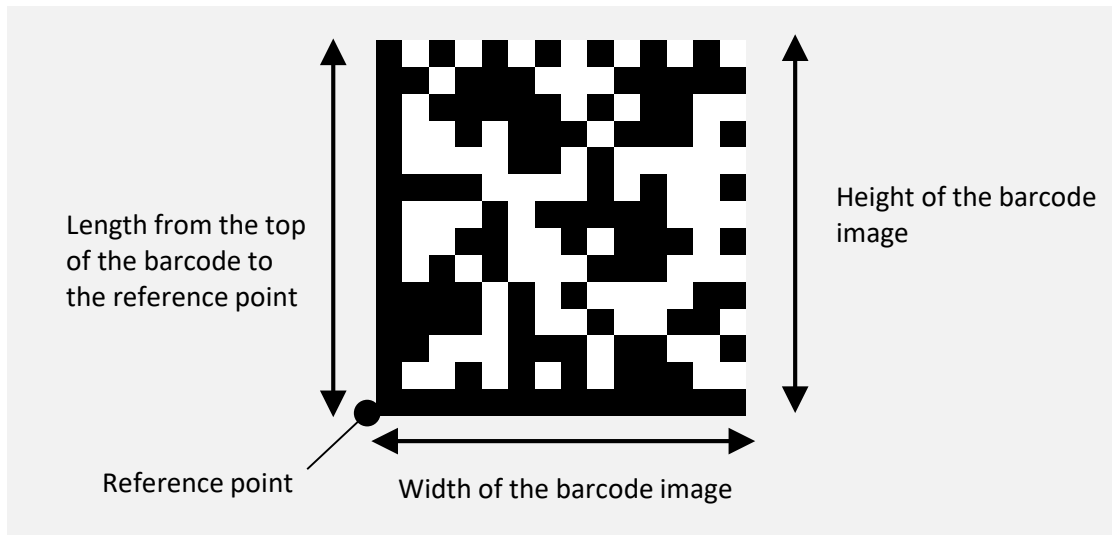
\*1: Height of the barcode image = Length from the top of the barcode to the reference point

\*2: The version is determined by the content of the barcode data and the error correction level.

#### Module Size Value

<i>moduleSize</i>	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 × module size value

Width of the barcode image = number of horizontal module × module size value

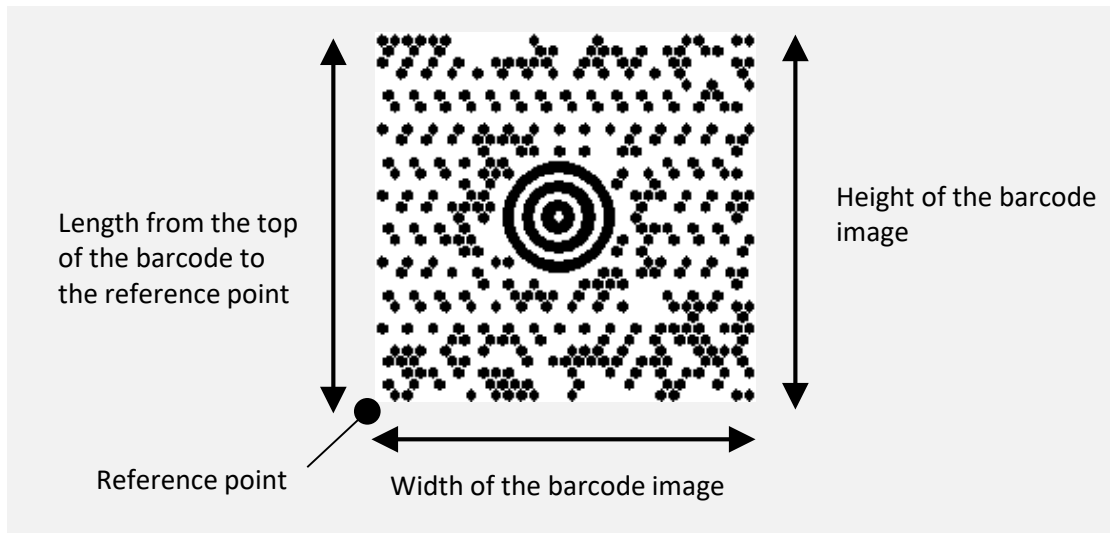
<i>dataMatrixModule</i>	<b>Number of Vertical Module</b>	<b>Number of Horizontal Module</b>
DATA_MATRIX_10_10	10	10
DATA_MATRIX_12_12	12	12
DATA_MATRIX_14_14	14	14
DATA_MATRIX_16_16	16	16
DATA_MATRIX_18_18	18	18
DATA_MATRIX_20_20	20	20
DATA_MATRIX_22_22	22	22
DATA_MATRIX_24_24	23	23
DATA_MATRIX_26_26	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

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

#### Module Size Value

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

### B.1.5 printMaxicode, printPageModeMaxicode



- (1) Height of the barcode image

Height of the barcode image<sup>\*1</sup> = 200

<sup>\*1</sup>: 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 = 210

### B.1.6 printGS1DataBarStacked, printPageModeGS1DataBarStacked



#### (1) Height and width of the barcode image

Height of the barcode image<sup>\*1</sup> = 13 × module size value

<sup>\*1</sup>: Height of the barcode image = Length from the top of the barcode to the reference point

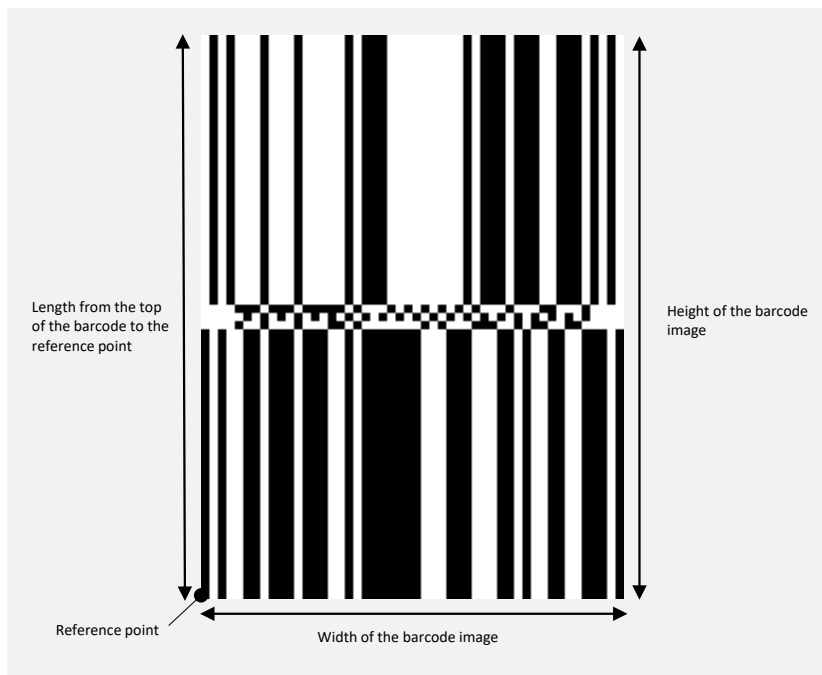
Width of the barcode image = 50 × module size value

#### Module Size Value

<i>moduleSize</i>	Module Size Value
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16



**B.1.7 printGS1DataBarStackedOmnidirectional ,  
printPageModeGS1DataBarStackedOmnidirectional**



(1) Height and width of the barcode image

Height of the barcode image<sup>\*1</sup> = (*moduleHeight* × 2 + 3) × module size value

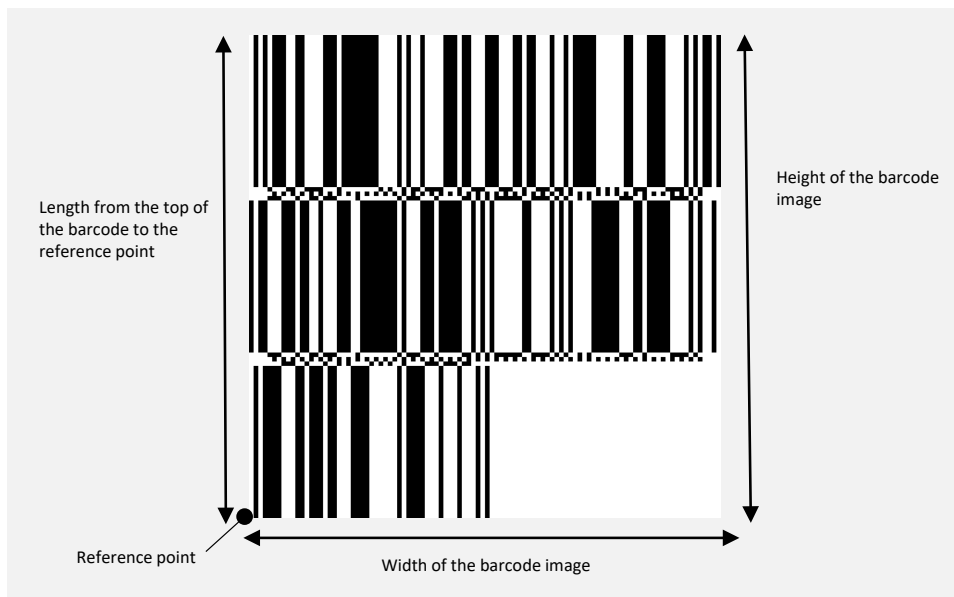
\*1: Height of the barcode image = Length from the top of the barcode to the reference point

Width of the barcode image = 50 × module size value

**Module Size Value**

<i>moduleSize</i>	Module Size Value
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16

### B.1.8 printGS1DataBarExpandedStacked , printPageModeGS1DataBarExpandedStacked



#### (1) Height and width of the barcode image

Height of the barcode image\*<sup>1</sup> = ((34 + 3) × number of row\*<sup>2</sup> + 34) × module size value

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

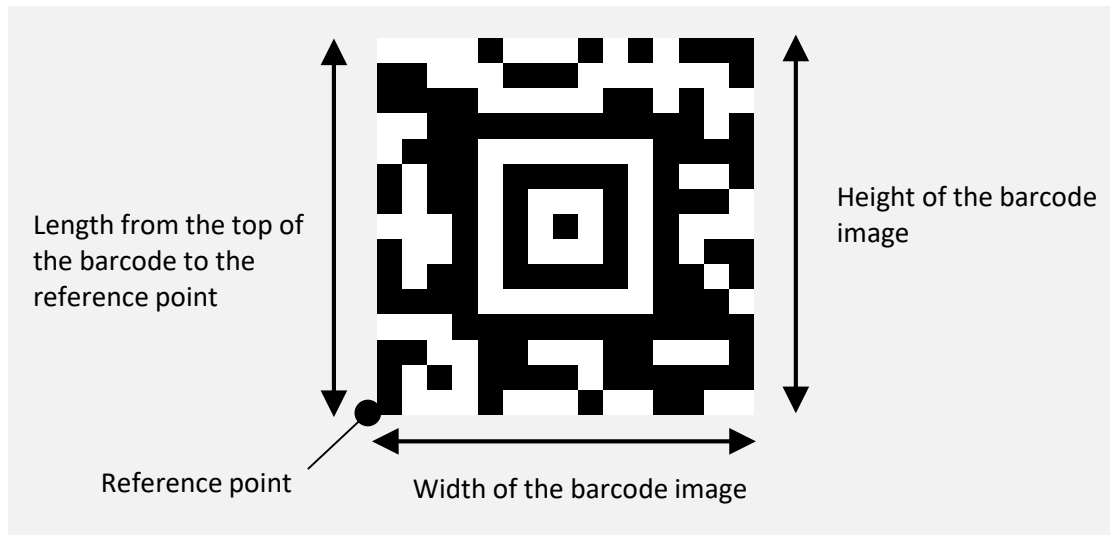
\*2: The number of row is determined by the barcode data.

Width of the barcode image = (4 + 49 × column / 2) × module size value

#### Module Size Value

<i>moduleSize</i>	Module Size Value
GS1DATABAR_MODULE_SIZE_2	2
GS1DATABAR_MODULE_SIZE_3	3
GS1DATABAR_MODULE_SIZE_4	4
GS1DATABAR_MODULE_SIZE_5	5
GS1DATABAR_MODULE_SIZE_6	6
GS1DATABAR_MODULE_SIZE_7	7
GS1DATABAR_MODULE_SIZE_8	8
GS1DATABAR_MODULE_SIZE_9	9
GS1DATABAR_MODULE_SIZE_10	10
GS1DATABAR_MODULE_SIZE_11	11
GS1DATABAR_MODULE_SIZE_12	12
GS1DATABAR_MODULE_SIZE_13	13
GS1DATABAR_MODULE_SIZE_14	14
GS1DATABAR_MODULE_SIZE_15	15
GS1DATABAR_MODULE_SIZE_16	16

### B.1.9 printAztecCode , printPageModeAztecCode



#### (1) Height and width of the barcode image

Height\*<sup>1</sup> and width of the barcode image = number of module size × module size value

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

Example: When *aztecSymbol* is **AZTECCODE\_COMPACT** and *layer* is 1 and *moduleSize* is **AZTECCODE\_MODULE\_SIZE\_6**:

Height and width of the barcode image = 15 × 6 = 90

#### Number of Module

<i>aztecSymbol</i>	<i>layer</i>	Number of Module
<b>AZTECCODE_FULLRANGE</b>	4	31
	5	37
	6	41
	7	45
	8	49
	9	53
	10	57
	11	61
	12	67
	13	71
	14	75
	15	79
	16	83
	17	87
	18	91
	19	95
	20	101
	21	105

<i>aztecSymbol</i>	<i>layer</i>	<b>Number of Module</b>
<b>AZTECCODE_FULLRANGE</b>	22	109
	23	113
	24	117
	25	121
	26	125
	27	131
	28	135
	29	139
	30	143
	31	147
	32	151
<b>AZTECCODE_COMPACT</b>	1	15
	2	19
	3	23
	4	27

#### Module Size Value

<i>moduleSize</i>	<b>Module Size Value</b>
<b>AZTECCODE_MODULE_SIZE_2</b>	2
<b>AZTECCODE_MODULE_SIZE_3</b>	3
<b>AZTECCODE_MODULE_SIZE_4</b>	4
<b>AZTECCODE_MODULE_SIZE_5</b>	5
<b>AZTECCODE_MODULE_SIZE_6</b>	6
<b>AZTECCODE_MODULE_SIZE_7</b>	7
<b>AZTECCODE_MODULE_SIZE_8</b>	8
<b>AZTECCODE_MODULE_SIZE_9</b>	9
<b>AZTECCODE_MODULE_SIZE_10</b>	10
<b>AZTECCODE_MODULE_SIZE_11</b>	11
<b>AZTECCODE_MODULE_SIZE_12</b>	12
<b>AZTECCODE_MODULE_SIZE_13</b>	13
<b>AZTECCODE_MODULE_SIZE_14</b>	14
<b>AZTECCODE_MODULE_SIZE_15</b>	15
<b>AZTECCODE_MODULE_SIZE_16</b>	16