SII SDK for .NET MAUI
Application Programmer's Guide

Rev.07

[Products]

SLP720RT Series
SLP721RT Series
RP-F10 Series
RP-G10 Series
RP-E10 Series
RP-D10 Series
MP-B30 Series
MP-B30L Series
MP-B21L Series
MP-B20 Series
MP-A40 Series
DSP-A01 Series

Rev.01	March 2024
Rev.02	May 2024
Rev.03	July 2024
Rev.04	November 2024
Rev.05	March 2025
Rev.06	March 2026
Rev.07	May 2026

Copyright © 2024-2026 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.
iPad®, iPad Air®, iPad mini®, iPhone®, and iPod® are trademarks of Apple Inc., registered in the U.S. and other countries and regions.
The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc.
IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.
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 SDK for .NET MAUI" (hereinafter referred to as "SDK") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

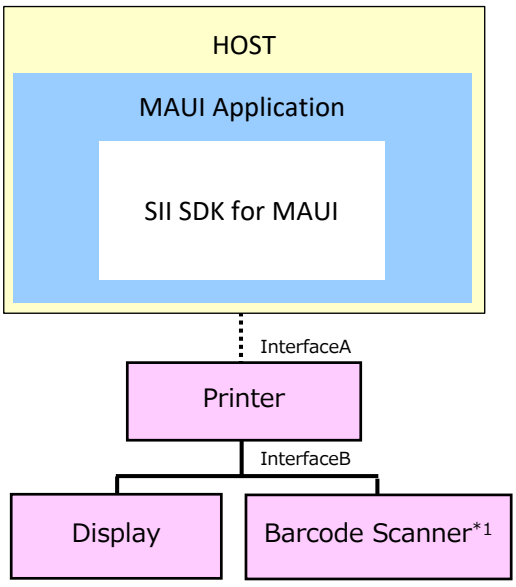
Target Devices

The printers supported by SDK are listed below.

iOS

Printer	Interface A	Display	Interface B
SLP720RT Series	TCP/IP	-	-
SLP721RT Series	Bluetooth	-	-
	TCP/IP		
RP-F10 Series	Bluetooth	DSP-A01 Series	USB
	USB		
	TCP/IP		
RP-E10 Series	Bluetooth	-	-
	TCP/IP*1		
RP-D10 Series	Bluetooth		
	TCP/IP*2		
MP-B30 Series MP-B30L Series	Bluetooth		
	TCP/IP		
MP-B21L Series	Bluetooth		
	TCP/IP		
MP-B20 Series	Bluetooth		
MP-A40 Series	Bluetooth		

*1: Use main firmware version 1.11, LAN interface firmware version 1.13.01 or later.
*2: Use main firmware version 1.05, LAN interface firmware version 1.13.01 or later.



Configuration Image

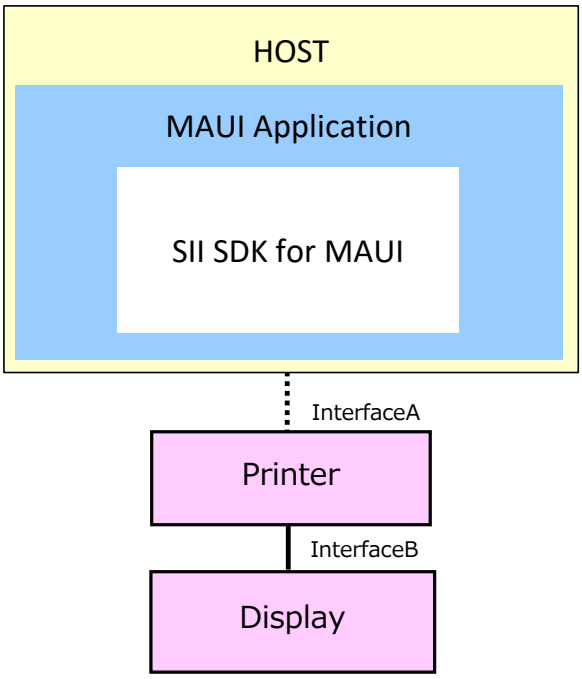
*1: See "RP-F10 SERIES Thermal Printer USER'S GUIDE" for details about the combination of peripherals including the barcode scanner.

Android

(1) When the display is used by connecting to SII printer.

Printer	Interface A	Display	Interface B
SLP720RT Series	USB	-	-
	TCP/IP		
SLP721RT Series	Bluetooth	-	-
	USB		
	TCP/IP		
RP-F10 Series	Bluetooth	DSP-A01 Series*1	USB
	USB		
	TCP/IP		
RP-G10 Series	USB	-	-
RP-E10 Series	Bluetooth		
	USB		
	TCP/IP*2		
RP-D10 Series	Bluetooth		
	USB		
	TCP/IP*3		
MP-B30 Series MP-B30L Series	Bluetooth		
	USB		
	TCP/IP		
MP-B21L Series	Bluetooth		
	USB		
	TCP/IP		
MP-B20 Series	Bluetooth		
	USB		
MP-A40 Series	Bluetooth		
	USB		

*1: When the display is used by connecting to SII printer, it is described as "DSP-A01 via the printer".
*2: Use main firmware version 1.11, LAN interface firmware version 1.13.01 or later.
*3: Use main firmware version 1.05, LAN interface firmware version 1.13.01 or later.

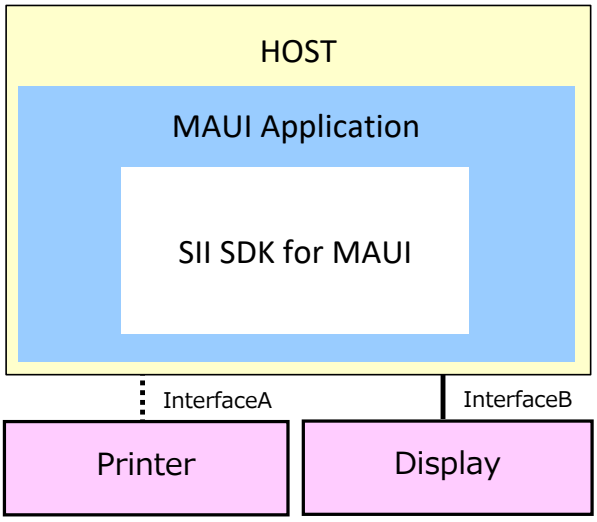


Configuration Image

(2) When the display is used alone.

Printer	Interface A	Display	Interface B
-	-	DSP-A01 series*1	USB

*1: When the display is used alone, it is described as "DSP-A01 alone".



Configuration Image

Terms

The terms used in this manual are described below.

Printer

Term	Description
Technical Reference for Printer	The following technical reference. ・SLP720RT SERIES THERMAL PRINTER TECHNICAL REFERENCE ・SLP721RT SERIES THERMAL PRINTER TECHNICAL REFERENCE ・RP-F10 SERIES THERMAL PRINTER TECHNICAL REFERENCE ・RP-G10 SERIES THERMAL PRINTER TECHNICAL REFERENCE ・RP-E10 SERIES THERMAL PRINTER TECHNICAL REFERENCE ・RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE ・MP-B30 SERIES THERMAL PRINTER TECHNICAL REFERENCE ・MP-B30-x02JK2x THERMAL PRINTER TECHNICAL REFERENCE ・MP-B30L SERIES THERMAL PRINTER TECHNICAL REFERENCE ・MP-B30L-x46JK2x THERMAL PRINTER TECHNICAL REFERENCE ・MP-B21L SERIES THERMAL PRINTER TECHNICAL REFERENCE ・MP-B20-B02JK1 THERMAL PRINTER TECHNICAL REFERENCE ・MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCE
Printer command	Command for controlling the printer described in "Technical Reference for Printer".
USER'S GUIDE	The following user's guide. ・SLP720RT SERIES Thermal Printer USER'S GUIDE ・SLP721RT SERIES Thermal Printer USER'S GUIDE ・RP-F10 SERIES Thermal Printer USER'S GUIDE ・RP-G10 SERIES Thermal Printer USER'S GUIDE ・RP-E10 SERIES Thermal Printer USER'S GUIDE ・RP-D10 SERIES Thermal Printer USER'S GUIDE ・MP-B30 SERIES Thermal Printer USER'S GUIDE ・MP-B30-x02JK2x Thermal Printer USER'S GUIDE ・MP-B30L SERIES Thermal Printer USER'S GUIDE ・MP-B30L-x46JK2x Thermal Printer USER'S GUIDE ・MP-B21L SERIES Thermal Printer USER'S GUIDE ・MP-B20-B02JK1 Thermal Printer USER'S GUIDE ・MP-A40 SERIES Thermal Printer USER'S GUIDE

Display

Term	Description
Technical Reference for Display	The following technical reference. ・DSP-A01 SERIES CUSTOMER DISPLAY TECHNICAL REFERENCE
Display command	Command for controlling the display described in "Technical Reference for Display".
Slide	The image data of the screen size (Width 480 px × Height 272 px). The displays as a standby screen and as a backscreen superimposed on a template.

Term	Description
Template	The stylized form having elements that can set attributes such as drawing areas and mapping positions. The elements include text elements (text data), img elements (image data), barcode elements (barcode data), and qr elements (QR Code data). To register templates, define a map ID each for an element to place. Registered image data or text data is shown on the display by updating the screen after selecting a template and specifying its map ID. The data in the template is required to be specified XML file format. The maximum size of template data is width 480 px × height 272 px.
Map ID	An ID defined to an element which is holding positional information or modification information when a template is registered.
Macro	A function to register multiple APIs in order of execution, and execute automatically when an event occurs.
Event	An event which is defined by "Event notification" in the display commands.

Table of Contents

Chapter 1 Product Overview 1-1

1.1 Functions Provided by SDK 1-1
1.2 SDK Overview 1-1
1.2.1 SDK Configuration 1-1
1.2.2 Functions Provided by Library 1-2
1.2.3 Development of Application that Performs Bluetooth Communication or USB Communication with SII Printer 1-2
1.2.4 Registered Data of Display at Factory 1-3

Chapter 2 Product Specifications 2-1

2.1 Operating Environment 2-1
2.1.1 iOS 2-1
2.1.2 Android 2-6
2.1.3 .NET MAUI 2-6
2.2 Printer Settings 2-7
2.2.1 iOS 2-7
2.2.2 Android 2-14
2.3 Precaution 2-18
2.3.1 iOS 2-18
2.3.2 Android 2-18

Chapter 3 How to Use Library 3-1

3.1 iOS and Android Application Development Environment 3-1
3.2 Provided Files 3-2
3.3 Build Library into Visual Studio 3-3
3.3.1 Windows 3-3
3.3.2 Mac 3-12
3.4 Precautions 3-20

Chapter 4 Library Functions of Printer 4-1

4.1 Standard Mode and Page Mode 4-1
4.1.1 Basic Operation 4-1
(1) Standard mode 4-1
(2) Page mode 4-2
4.1.2 Text Data Printing in Standard Mode 4-3
4.1.3 Mapping Position of Print Data in Page Mode 4-4
(1) Print area of page mode 4-4
(2) Print direction 4-4
(3) Reference point 4-5
4.1.4 Print Data Process at Out of Print Area of Page Mode 4-6

4.2 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
Method or Property
Method name or property name 4-13
4.4.1 Enumerated Constant List 4-14
① Device model (DeviceModel) 4-14
② Port type (PortType) 4-14
③ Drawer number (DrawerNum) 4-14
④ Pulse width (PulseWidth) 4-15
⑤ Buzzer pattern (BuzzerPattern) 4-15
⑥ Dithering (Dithering) 4-15
⑦ Printer response type (PrinterResponseId) 4-16
⑧ Device type (DeviceType) 4-16
⑨ Batch processing selection (TransactionFunction) 4-16
⑩ Alignment (PrintAlignment) 4-16
⑪ Bold character (CharacterBold) 4-17
⑫ Underline (CharacterUnderline) 4-17
⑬ Character scale (CharacterScale) 4-17
⑭ Character font (CharacterFont) 4-18
⑮ Module size (ModuleSize) 4-18
⑯ Error correction level (ErrorCorrection) 4-21
⑰ Reverse print (CharacterReverse) 4-22
⑱ Inversion print (CharacterInversion) 4-22
⑲ Barcode symbol (BarcodeSymbol) 4-22
⑳ HRI character print position (HriPosition) 4-23
㉑ N:W ratio (NwRatio) 4-23
㉒ PDF417 symbol (Pdf417Symbol) 4-23
㉓ QR Code Model (QrModel) 4-23
㉔ Data Matrix module (DataMatrixModule) 4-24
㉕ MaxiCode Mode (MaxiCodeMode) 4-25

㉖ Aztec symbol (AztecSymbol) 4-25
㉗ Cutting method (CuttingMethod) 4-25
㉘ Form feed position (FeedPosition) 4-26
㉙ Image rotation direction (Rotate) 4-28
㉚ Image scaling (ImageScale) 4-28
㉛ Print direction (Direction) 4-28
㉜ Line style (LineStyle) 4-29
㉝ International character set (InternationalCharacter) 4-29
㉞ Codepage (CodePage) 4-30
㉟ Paper selection with or without mark when printing label file (PrintSmartLabelMode) 4-30
4.4.2 PrinterManager Class 4-31
(1) Method List 4-31
① Common method to standard mode and page mode 4-31
② Dedicated method for standard mode 4-32
③ Dedicated method for page mode 4-33
(2) Common property list to standard mode and page mode 4-34
(3) Constant List 4-35
① Barcode and PDF417 4-35
② Battery remaining capacity level 4-35
(4) Method Details 4-36
① Common method to standard mode and page mode 4-36
PrinterManager Constructor 4-36
Connect Start communicating with device 4-36
Disconnect Stop communicating with device 4-38
OpenDrawer Open cash drawer 4-38
Buzzer Sound buzzer 4-38
ExternalBuzzer Sound external buzzer 4-39
GetStatus Get printer status 4-39
Abort Abort waiting state of printer 4-41
RegisterLogo Register logo 4-41
UnregisterLogo Delete registered logo 4-42
RegisterStyleSheet Register style sheet 4-42
UnregisterStyleSheet Delete registered style sheet 4-43
ResetPrinter Reset printer 4-43
GetPrinterResponse Get various responses from printer 4-44
StartDiscoveryDevice Start device search 4-45
CancelDiscoveryDevice Cancel device search 4-46
GetFoundDevice Get found device information list 4-46
ControlTransaction Start/Finish batch processing 4-47
SetStatusChangedEventHandler Start/Finish call back of printer status change 4-48

SetBarcodeScannerReadDataEventHandler
Start/Finish call back of receipt barcode data 4-49
SetBarcodeScannerChangedOnlineEventHandler
Start/Finish call back of barcode scanner connection 4-49
SetBarcodeScannerChangedOfflineEventHandler
Start/Finish call back of barcode scanner disconnection 4-50
② Dedicated method for standard mode 4-51
SendText Send text data 4-51
SendTextEx Send format specified text data 4-51
PrintBarcode Print barcode 4-52
PrintPDF417 Print PDF417 4-56
PrintQRcode Print QR Code 4-57
PrintDataMatrix Print Data Matrix 4-57
PrintMaxiCode Print MaxiCode 4-58
PrintGS1DataBarStacked
Print GS1 Databar Stacked 4-59
PrintGS1DataBarStackedOmnidirectional
Print GS1 Databar Stacked Omni-directional 4-59
PrintGS1DataBarExpandedStacked
Print GS1 Databar Expanded Stacked 4-60
PrintAztecCode Print Aztec Code 4-60
CutPaper Cut paper 4-61
FeedPosition Paper form feed 4-62
SendBinary Send binary data 4-62
SendDataFile Send specified file 4-62
SendImage Send image 4-64
PrintPDF Print PDF page 4-64
PrintLogo Print logo 4-66
PrintSmartLabelImageData
Print label 4-66
③ Dedicated method for page mode 4-67
EnterPageMode Start page mode 4-68
ExitPageMode End page mode 4-68
SetPageModeArea
Specify print area of page mode 4-68
SetPageModeDirection
Specify print direction of page mode 4-69
SetPageModeLineSpacing
Specify line spacing of page mode 4-70
PrintPageMode Print page mode 4-70
PrintPageModeText
Send text data of page mode 4-70
PrintPageModeTextEx
Send format specified text data of page mode 4-71
PrintPageModeBarcode
Print barcode of page mode 4-72

PrintPageModePDF417
Print PDF417 of page mode....4-76
PrintPageModeQRcode
Print QR Code of page mode....4-77
PrintPageModeDataMatrix
Print Data Matrix of page mode....4-77
PrintPageModeMaxiCode
Print MaxiCode of page mode....4-78
PrintPageModeGS1DataBarStacked
Print GS1 Databar Stacked of page mode....4-79
PrintPageModeGS1DataBarStackedOmnidirectional
Print GS1 Databar Stacked Omni-directional of page mode....4-79
PrintPageModeGS1DataBarExpandedStacked
Print GS1 Databar Expanded Stacked of page mode....4-80
PrintPageModeAztecCode
Print Aztec Code of page mode....4-81
SendPageModeBinary
Send binary data of page mode....4-82
PrintPageModeImageFile
Draw Image file of page mode....4-82
PrintPageModeRectangle
Draw rectangle image of page mode....4-83
PrintPageModeLine
Print ruled line of page mode....4-84
PrintPageModeLogo
Print logo of page mode....4-86
(5) Common property detail to standard mode and page mode....4-87
SendTimeout Get/Set send timeout period....4-87
ReceiveTimeout Get/Set receive timeout period....4-87
InternationalCharacter
Get/Set international character set....4-88
CodePage Get/Set codepage....4-88
DeviceModel Get device model....4-88
PortType Get connecting port type....4-89
IsConnect Verify connection state with printer....4-89
SocketKeepingTime
Get/Set socket keeping time....4-89
Version Get SDK version....4-90
PrintSmartLabelMode
Get/Set paper when printing label file....4-90
4.4.3 DeviceInfo Class....4-91
(1) Method List....4-91
(2) Property List....4-91
(3) Method Details....4-92
DeviceInfo Constructor....4-92

(4) Property Details 4-93
PortType Get connecting port type 4-93
DeviceName Get device name (printer model name) 4-93
BluetoothAddress Get Bluetooth address 4-93
MacAddress Get MAC address 4-93
IpAddress Get IP address 4-93
IsBonded Get paring state 4-94
DevicePath Get device path 4-94
4.4.4 PrinterIdentifier Class 4-95
(1) Method List 4-95
(2) Method Details 4-95
PrinterModelID Get printer model ID 4-95
TypeID Get type ID 4-95
RomVersionID Get ROM version ID 4-95
MainFirmwareVersion
Get firmware version (main) 4-96
Manufacturer Get manufacturer 4-96
ModelName Get model name 4-96
SerialNumber Get serial number 4-96
BootFirmwareVersion
Get firmware version (boot) 4-96
BootFirmwareChecksum
Get firmware checksum (boot) 4-97
MainFirmwareChecksum
Get firmware checksum (main) 4-97
FirmwareChecksum
Get firmware checksum (main + boot) 4-97
4.4.5 PrinterException Class 4-98
(1) Method List 4-98
(2) Property List 4-98
(3) Constant List 4-98
① Error code 4-98
(4) Method Details 4-100
PrinterException Constructor 4-100
(5) Property Details 4-100
HResult Get error code 4-100
Message Get error message 4-100
4.4.6 SmartLabelManager Class 4-101
(1) Method List 4-101
(2) Method Details 4-102
SmartLabelManager
Constractor 4-102
SelectSmartLabelFile
Specify label file 4-102
ReplaceSmartLabelTextData
Replace text data of label 4-103
ReplaceSmartLabelImageData
Replace image data of label 4-104

ReplaceSmartLabelBarcodeData
Replace barcode data of label....4-105

Chapter 5 Library Functions of Display 5-1

5.1 API Reference 5-1
Method or Property
Method name or property name 5-2
5.1.1 Enumerated Constant List....5-3
① Device model (DeviceModel)....5-3
② Port type (PortType) 5-3
③ Printer response type (PrinterResponseId)....5-3
④ Device type (DeviceType) 5-3
⑤ Memory area (MemoryArea) 5-4
⑥ Alignment (PrintAlignment)....5-4
⑦ Bold character (CharacterBold)....5-4
⑧ Underline (CharacterUnderline) 5-4
⑨ Character scale (CharacterScale)....5-4
⑩ Character font (CharacterFont) 5-5
⑪ Registered font (RegisterdFont)....5-5
⑫ Module size (ModuleSize) 5-5
⑬ Error correction level (ErrorCorrection) 5-6
⑭ QR data mode (QrDataMode) 5-6
⑮ QR quiet zone (QrQuietZone) 5-6
⑯ Macro registration processing (MacroRegistrationFunction)....5-7
⑰ Display response type (DisplayResponseId)....5-7
⑱ International character set (InternationalCharacter)....5-7
⑲ Codepage (CodePage)....5-8
5.1.2 PrinterManager Class 5-9
(1) Method List 5-9
(2) Property list....5-10
(3) Method Details....5-11
PrinterManager Constructor....5-11
Connect Start communicating with device....5-11
Disconnect Stop communicating with device 5-12
GetPrinterResponse
Get various responses from printer....5-13
StartDiscoveryDevice
Start device search....5-14
CancelDiscoveryDevice
Cancel device search 5-15
GetFoundDevice
Get found device information list....5-15
Defragment Optimize memory area 5-16
InitializeMemoryArea
Initialize memory area 5-16
ShowTemplate Display template....5-17
ShowSlide Display slide 5-18

EnterStandbyMode
Display standby 5-18
ExecuteMacro Execute macro 5-19
TurnOnScreen Turn on/off screen 5-19
SelectTemplate Select template 5-19
SetTemplateImageData
Set image data 5-20
SelectTemplateTextObject
Select text element 5-21
SetTemplateTextAlignment
Alignment of text data 5-21
SetTemplateTextLeftMargin
Set left margin of text data 5-22
SetTemplateTextLineSpacing
Set line spacing of text data 5-22
SetTemplateTextBold
Set bold character of text data 5-23
SetTemplateTextUnderline
Set underline of text data 5-23
SetTemplateTextSize
Set character size of text data 5-24
SetTemplateTextFont
Set character font of text data 5-24
SetTemplateTextRegisteredFont
Set registered font of text data 5-25
SetTemplateTextRightSpacing
Set right space of text data 5-25
SetTemplateTextColor
Set character color of text data 5-26
SetTemplateTextData
Input text data 5-26
SetTemplateBarcodeData
Input barcode data 5-27
SetTemplateQrCodeData
Input QR Code data 5-28
RegisterTemplate
Register template 5-29
UnregisterTemplate
Delete template 5-30
RegisterImageData
Register image data 5-31
UnregisterImageData
Delate image data 5-32
RegisterSlideData
Register slide data 5-32
UnregisterSlideData
Delete slide data 5-33

RegisterUserDefinedCharacter
Register user-defined character 5-34
UnregisterUserDefinedCharacter
Delete user-defined character 5-35
RegisterOptionFont
Register optional font 5-35
UnregisterOptionFont
Delete optional font 5-36
ControlMacroRegistration
Start/Finish of macro registration 5-36
GetDisplayResponse
Get various response from Display 5-38
SendBinary Send binary data 5-39
SendDataFile Send specified file 5-40
(4) Property Details 5-41
SendTimeout Get/Set send timeout period 5-41
ReceiveTimeout Get/Set receive timeout period 5-41
InternationalCharacter
Get/Set international character set 5-41
CodePage Get/Set codepage 5-42
DeviceModel Get device model 5-42
PortType Get connecting port type 5-42
IsConnect Verify connection state with printer 5-43
SocketKeepingTime
Get/Set socket keeping time 5-43
Version Get SDK version 5-43
5.1.3 DeviceInfo Class 5-44
(1) Method List 5-44
(2) Property List 5-44
(3) Method List 5-45
DeviceInfo Constructor 5-45
(4) Property Details 5-46
PortType Get connecting port type 5-46
DeviceName Get device name (printer model name) 5-46
BluetoothAddress
Get Bluetooth address 5-46
MacAddress Get MAC address 5-46
IpAddress Get IP address 5-46
IsBonded Get paring state 5-47
DevicePath Get device path 5-47
5.1.4 PrinterIdentifier Class 5-48
(1) Method List 5-48
(2) Method Details 5-48
PrinterModelID Get printer model ID 5-48
TypeID Get type ID 5-48
RomVersionID Get ROM version ID 5-49
MainFirmwareVersion
Get firmware version (main) 5-49

Manufacturer Get manufacturer 5-49
ModelName Get model name 5-49
SerialNumber Get serial number 5-49
BootFirmwareVersion
Get firmware version (boot) 5-49
BootFirmwareChecksum
Get firmware checksum (boot) 5-50
MainFirmwareChecksum
Get firmware checksum (main) 5-50
FirmwareChecksum
Get firmware checksum (main + boot) 5-50
5.1.5 PrinterException Class 5-51
(1) Method List 5-51
(2) Property List 5-51
(3) Constant List 5-51
① Error code 5-51
(4) Method Details 5-53
PrinterException Constructor 5-53
(5) Property Details 5-53
HResult Get error code 5-53
Message Get error message 5-53

Chapter 6 Chapter 6 Sample Program 6-1

6.1 Screen Layout 6-1
6.2 Precaution 6-2

Appendix A Character Set A-1

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

Appendix B Barcode Size List B-1

B.1 Barcode Size List (SLP720RT, SLP721RT, RP-F10, RP-G10, MP-B30, MP-B30L, MP-B21L) B-1
B.1.1 PrintBarcode, PrintPageModeBarcode B-1
B.1.2 PrintPDF417, PrintPageModePDF417 ..... B-7
B.1.3 PrintQRCode, PrintPageModeQRCode B-8
B.1.4 PrintDataMatrix, PrintPageModeDataMatrix B-9
B.1.5 PrintMaxicode, PrintPageModeMaxicode B-11
B.1.6 PrintGS1DataBarStacked, PrintPageModeGS1DataBarStacked B-12
B.1.7 PrintGS1DataBarStackedOmnidirectional, PrintPageModeGS1DataBarStackedOmnidirectional B-13
B.1.8 PrintGS1DataBarExpandedStacked, PrintPageModeGS1DataBarExpandedStacked B-14
B.2 Barcode Size List (MP-B30L, MP-B21L) B-15
B.2.1 PrintAztecCode, PrintPageModeAztecCode B-15

B.3 Barcode Size List (RP-E10, RP-D10, MP-B20) B-17
B.3.1 PrintBarcode, PrintPageModeBarcode B-17
B.3.2 PrintPDF417, PrintPageModePDF417 ..... B-22
B.3.3 PrintQRCode, PrintPageModeQRCode B-23
B.3.4 PrintDataMatrix, PrintPageModeDataMatrix B-25
B.3.5 PrintMaxicode, PrintPageModeMaxicode B-26
B.3.6 PrintGS1DataBarStacked, PrintPageModeGS1DataBarStacked B-27
B.3.7 PrintGS1DataBarStackedOmnidirectional, PrintPageModeGS1DataBarStackedOmnidirectional B-28
B.3.8 PrintGS1DataBarExpandedStacked, PrintPageModeGS1DataBarExpandedStacked B-29

Appendix C Open Source Software License C-1

C.1 MIT License C-1
C.2 Apache License 2.0 ..... C-2

Chapter 1
Product Overview

This chapter describes the overview of the SDK.

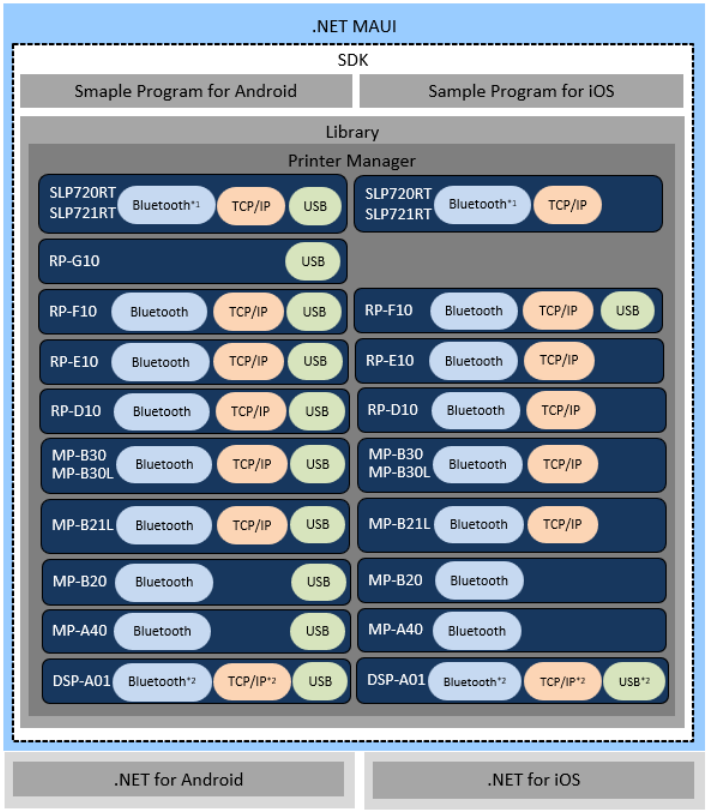
1.1 Functions Provided by SDK

The SDK including the library and the sample programs provides the functions to use SII printer (hereinafter referred to as "printer") described in "Target Printers" in "INTRODUCTION".
Moreover, the SDK provides the library sample programs in .NET MAUI application in Visual Studio.

1.2 SDK Overview

1.2.1 SDK Configuration

The SDK incorporates SII Print Class Library for Android and SII Print Class Library for iOS.
The library and the sample programs in the SDK are indicated with dashed lines in the figure below.



*1 : Supported only by SLP721RT.
*2 : It is valid only when using the display via the printer.

1.2.2 Functions Provided by Library

The applications of iOS or Android to be developed by .NET MAUI can easily send print data and printer commands to a printer through the communication port (Bluetooth, USB, or TCP/IP) on an iOS or Android device.
Also, the applications can get the printer status.

The library provides the following functions.
The functions to be provided differ depending on the devices. See "Chapter 4 Library Functions of Printer Library" or "Chapter 5 Library Functions of Display" for the functions to be provided.

- Connecting to / disconnecting from a device
- Sending data to a device (print data and/or printer commands[*1])
- Printing barcode and 2-dimensional barcode
- Sending a data file to a device (print data and/or printer commands[*1])
- Cutting paper
- Getting the printer status
- Aborting the waiting state of a printer
- Getting various responses from a device
- Bulk registration of print commands
- Registering a printer status call back function
- Searching the printer by Bluetooth or TCP/IP
- Printer hardware reset
- Drawer operation control
- Buzzer beeping control
- Screen display control
- Registering a barcode scanner call back function
- Printing a label file
- Replacing object data in a label file
- Outputting a log file

*1: Commands that read the response from the printer are not supported.
In order to read responses from the printer, use `GetStatus` or `GetPrinterResponse`.

(NOTE) See "RP-F10 SERIES Thermal Printer USER'S GUIDE" for details of the recommended barcode scanner and the barcode scanner setting.

1.2.3 Development of Application that Performs Bluetooth Communication or USB Communication with SII Printer

To register the application that communicates with the printer through Bluetooth or USB to App Store, the pre-application to Apple from SII is necessary. For details, please contact SII.

1.2.4 Registered Data of Display at Factory

Registered data of the display at the factory such as templates may be added or changed without prior notice for the quality improvement.
A template which is specified appropriate encode is required to use depending on language settings or character codes to specify. See SII's Website for details about the data to be registered at the factory.
https://www.sii-ps.com/dspa01/

Chapter 2
Product Specifications

This chapter describes the product specifications of the library.

2.1 Operating Environment

2.1.1 iOS

(1) Applicable iOS Devices

Applicable iOS devices for the library are shown in the following list.

(a) SLP721RT

iPhone models

- iPhone 15 Pro Max
- iPhone 15 Pro
- iPhone 15 Plus
- iPhone 15
- iPhone 14 Pro Max
- iPhone 14 Pro
- iPhone 14 Plus
- iPhone 14
- iPhone SE (3rd generation)
- iPhone 13 Pro Max
- iPhone 13 Pro
- iPhone 13
- iPhone 13 mini
- iPhone 12 Pro Max
- iPhone 12 Pro
- iPhone 12
- iPhone 12 mini
- iPhone SE (2nd generation)
- iPhone 11 Pro Max
- iPhone 11 Pro
- iPhone 11
- iPhone XS Max
- iPhone XS
- iPhone XR

iPad models

- iPad Pro 12.9-inch (6th generation)
- iPad Pro 11inch (4th generation)
- iPad (10th generation)
- iPad Air (5th generation)
- iPad mini (6th generation)
- iPad (9th generation)
- iPad Pro 12.9inch (5th generation)
- iPad Pro 11inch (3rd generation)
- iPad Air (4th generation)
- iPad (8th generation)
- iPad Pro 12.9inch (4th generation)
- iPad Pro 11inch (2nd generation)
- iPad (7th generation)
- iPad Air (3rd generation)
- iPad mini (5th generation)
- iPad Pro 12.9inch (3rd generation)
- iPad (6th generation)
- iPad Pro 12.9inch (2nd generation)

(b) RP-F10-x27J1-5 (USB Type-C + Bluetooth + USB host model), MP-B30-B02JK2-xx, MP-B30L-B46JK2-xx, MP-B20-B02Jx1-74, MP-A40-B06JK3x

iPhone models

- iPhone 16 Pro Max
- iPhone 16 Pro
- iPhone 16 Plus
- iPhone 16
- iPhone 15 Pro Max
- iPhone 15 Pro
- iPhone 15 Plus
- iPhone 15
- iPhone 14 Pro Max
- iPhone 14 Pro
- iPhone 14 Plus
- iPhone 14
- iPhone SE (3rd generation)
- iPhone SE (2nd generation)

iPad models

- iPad mini (A17 Pro)
- iPad Pro 13-inch (M4)
- iPad Pro 11-inch (M4)
- iPad Air 13-inch (M2)
- iPad Air 11-inch (M2)
- iPad Pro 12.9-inch (6th generation)
- iPad Pro 11-inch (4th generation)
- iPad (10th generation)
- iPad Air (5th generation)
- iPad mini (6th generation)

(c) RP-F10-x27J1-4 (Bluetooth + USB host model), MP-B30-B02JK1-xx

iPhone models

- iPhone XR
- iPhone XS
- iPhone XS Max
- iPhone X
- iPhone 8
- iPhone 8 Plus
- iPhone 7
- iPhone 7 Plus
- iPhone SE
- iPhone 6s
- iPhone 6s Plus

iPad models

- iPad Pro 11-inch
- iPad Pro 12.9-inch (3rd generation)
- iPad (6th generation)
- iPad Pro 12.9-inch (2nd generation)
- iPad Pro 10.5-inch
- iPad (5th generation)
- iPad Pro 9.7-inch
- iPad Pro 12.9-inch (1st generation)
- iPad mini 4

iPod models

- iPod touch (6th generation)

(d) MP-B30L-B46JK1-xx

iPhone models

- iPhone 12
- iPhone 12 Pro
- iPhone 12 Pro Max
- iPhone 12 mini
- iPhone SE (2nd generation)
- iPhone 11
- iPhone 11 Pro
- iPhone 11 Pro Max
- iPhone XR
- iPhone XS
- iPhone XS Max
- iPhone X
- iPhone 8
- iPhone 8 Plus
- iPhone 7
- iPhone 7 Plus
- iPhone SE
- iPhone 6s
- iPhone 6s Plus

iPad models

- iPad (8th generation)
- iPad Pro 11-inch (2nd generation)
- iPad Pro 12.9-inch (4th generation)
- iPad (7th generation)
- iPad Pro 11-inch
- iPad Pro 12.9-inch (3rd generation)
- iPad (6th generation)
- iPad Pro 12.9-inch (2nd generation)
- iPad Pro 10.5-inch
- iPad (5th generation)
- iPad Pro 9.7-inch
- iPad Pro 12.9-inch (1st generation)
- iPad mini 4

iPod models

- iPod touch (7th generation)
- iPod touch (6th generation)

(e) MP-B21L

iPhone models

- iPhone 14 Pro Max
- iPhone 14 Pro
- iPhone 14 Plus
- iPhone 14
- iPhone SE (3rd generation)
- iPhone 13 Pro Max
- iPhone 13 Pro
- iPhone 13
- iPhone 13 mini
- iPhone 12 Pro Max
- iPhone 12 Pro
- iPhone 12
- iPhone 12 mini
- iPhone SE (2nd generation)
- iPhone 11 Pro Max
- iPhone 11 Pro
- iPhone 11
- iPhone XS Max
- iPhone XS
- iPhone XR

iPad models

- iPad Pro 12.9-inch (6th generation)
- iPad Pro 11inch (4th generation)
- iPad (10th generation)
- iPad Air (5th generation)
- iPad mini (6th generation)
- iPad (9th generation)
- iPad Pro 12.9inch (5th generation)
- iPad Pro 11inch (3rd generation)
- iPad Air (4th generation)
- iPad (8th generation)
- iPad Pro 12.9inch (4th generation)
- iPad Pro 11inch (2nd generation)
- iPad (7th generation)
- iPad Air (3rd generation)
- iPad mini (5th generation)
- iPad Pro 12.9inch (3rd generation)
- iPad (6th generation)
- iPad Pro 12.9inch (2nd generation)

(f) RP-E10, RP-D10

iPhone models

- iPhone X
- iPhone 8
- iPhone 8 Plus
- iPhone 7
- iPhone 7 Plus
- iPhone SE
- iPhone 6s
- iPhone 6s Plus
- iPhone 6
- iPhone 6 Plus

iPad models

- iPad Pro 12.9-inch (2nd generation)
- iPad Pro 10.5-inch
- iPad (5th generation)
- iPad Pro 9.7-inch
- iPad Pro 12.9-inch (1st generation)
- iPad mini 4
- iPad Air 2
- iPad mini 3

iPod models

- iPod touch (6th generation)

(2) Applicable iOS Versions

The applicable iOS versions for this library are listed below.

- iOS 17-17.6.1
- iPadOS 17-17.6.1
- iOS 18-18.6
- iPadOS 18-18.6
- iOS 26-26.4
- iPadOS 26-26.4

2.1.2 Android

(1) Applicable OS

The applicable OSs for this library are listed below.

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

2.1.3 .NET MAUI

(1) Supported Environments

The supported .NET MAUI environments for this library are listed below.

- .NET 10.0 (net10.0-android, net10.0-ios)
- .NET 9.0 (net9.0-android, net9.0-ios)

2.2 Printer Settings

Set the memory switches of the printer to [Value] in the following table when using the library.
See "USER'S GUIDE" for details of the memory switches and the factory default settings.

2.2.1 iOS

(1) SLP720RT, SLP721RT

MS	Function	Value
1-2	Taken Mode Selection (Taken Mode)	0: Enable[*1] 1: Disable[*2]
1-3	Mark Mode Selection (Mark Mode)	0: Enable[*3] 1: Disable[*4]
4-6	Paper Auto Detection Selection (Paper Auto Detection)	0: Enable[*3] 1: Disable [*3*4]
5-1	Automatic Status Response Selection (Auto Status Back)	0: Enable
5-2	Initialized Response Selection (Init. Response)	0: Enable
5-3	Data Discard Selection When Error Occurs (Error Through)	0: Enable
5-4	Data Discard Selection When Output Buffer Full Occurs (Response Data Discarding)	1: Disable
7	Thermal Paper Selection (Thermal Paper)	00B: Receipt 01B: Linerless label 10B: SLP Label
13-3	Realtime Command Selection (Realtime Command)	1: Enable
17-3	Feed Backward Setting After Paper Cutting (Backfeed After Cut)	0: Enable 1: Disable[*5]

*1: When printing continuously on linerless label, set this value to "Enable".
The status response of the taken sensor is responded when this value is set to "Enable".
*2: When printing continuously on receipt or SLP Label, set this value to "Disable".
*3: When using `FeedPosition`, one of the following settings is necessary.
・To automatically detect paper, set the memory switch MS 4-6 (Paper Auto Detection Selection) of the printer to "Enable".
・To specify the paper, set MS 4-6 (Paper Auto Detection Selection) to "Disable" and set MS 1-3 (Mark Mode Selection) to "Enable".
In addition, select the paper to use as follows.
・For marked linerless label:
Set MS 7 (Thermal Paper Selection) to "Linerless label".
・For SLP Label:
Set MS 7 (Thermal Paper Selection) to "SLP Label".
*4: Set this value to "Disable" and select the paper to be used in the memory switch MS 7 (Thermal Paper Selection) of the printer when using the receipt (other than marked paper) or the Linerless label (other than the marked paper).
*5: When executing `CutPaper` under the following conditions and using `PrintSmartLabelImageData` immediately after, set this value to "Disable".
・MS1-3 (Mark Mode Selection) is set to "Disable".
・**`CUT_FULL`** is specified to `CuttingMethod` of `CutPaper`, or MS1-2 (Taken Mode Selectoin) is set to "Enable" and **`CUT_PARTIAL`** is specified to `CuttingMethod`.

(2) RP-F10

MS	Function	Value
5-1	Automatic Status Response Selection (Auto Status Back)	0: Enable
5-2	Initialized Response Selection (Init. Response)	0: Enable
5-3	Data Discard Selection When Error Occurs (Error Through)	0: Enable
5-4	Data Discard Selection When Output Buffer Full Occurs (Response Data Discarding)	1: Disable
13-3	Realtime Command Selection (Realtime Command)	1: Enable
38-1	Scanner Automatic Status Response Selection*1 (Scanner Auto Status Back)	0: Enable*2
39-1*3	iOS Auto Connection (Auto Connection)	1: Disable 0: Enable*4

*1: The firmware of the printer to support the barcode scanner is Ver.1.10 or later.
*2: Select "Enable" when using the barcode scanner.
See "USER'S GUIDE" for details about the combination of peripherals.
*3: Only for Bluetooth connection.
*4: Select "Enable" when using `ResetPrinter`.

(3) RP-E10, RP-D10

MS	Function	Value
5-2	Initialized Response Selection (Init. Response)	0: Enable
5-3	Data Discard Selection When Error Occurs (Error Through)	0: Enable
39-1*1	iOS Auto Connection (Auto Connection)	1: Disable 0: Enable*2

*1: Only for Bluetooth connection.
*2: Select "Enable" when using `ResetPrinter`.

(4) MP-B30

- Memory switch

MS	Function	Value
1-1	Communication Selection (Interface)	1: USB/Wireless
1-2	Mark Mode Selection (Mark Mode)	1 : Disable*1 0 : Enable
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 "Disable" when using `CutPaper`.

- Bluetooth communication setting

See the printer command "Set Bluetooth Communication" described in "MP-B30 SERIES THERMAL PRINTER TECHNICAL REFERENCES" or "MP-B30-x02JK2x THERMAL PRINTER TECHNICAL REFERENCES" for the Bluetooth communication setting.

Function	Value
iOS Auto Connection Selection (Auto Connection)	1: Enable*1 0: Disable

*1: Select "Enable" when using `ResetPrinter`.

(5) MP-B30L

- Memory switch

MS	Function	Value
1-1	Communication Selection (Interface)	1: USB/Wireless
1-2	Mark/Gap Mode Selection (Mark/Gap Mode)	1 : Disable*1 0 : Enable*2
1-3 to 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
2-1	Initialization Performance Selection After Paper Setting (Paper Set Handle)	0: Form Feed
2-2	Realtime Command Selection (Realtime Command)	0: Enable
2-7	Paper Form Feed Selection by Feed Switch (SW Paper Form Feed)	1: Cut Position Form Feed (Cut Position)
9-1	Automatic Status Response Selection (Auto Status Back)	0: Enable
9-2	Initialized Response Selection (Init. Response)	0: Enable

*1: Select "Disable" when using `CutPaper`.
*2: Select "Enable" when using `FeedPosition`.

- Bluetooth communication setting

See the printer command "Set Bluetooth Communication" described in "MP-B30L SERIES THERMAL PRINTER TECHNICAL REFERENCES" or "MP-B30L-x46JK2x THERMAL PRINTER TECHNICAL REFERENCES" for the Bluetooth communication setting.

Function	Value
iOS Auto Connection Selection (Auto Connection)	1: Enable*1 0: Disable

*1: Select "Enable" when using `ResetPrinter`.

(6) MP-B21L

- Memory switch

MS	Function	Value
1-1	Interface Selection (Interface)	1: USB/Wireless
1-2	Mark Gap Mode Selection (Mark/Gap Mode)	1: Disable*1 0: Enable*2
1-3 to 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 "Disable" when using `CutPaper`.
*2: Select "Enable" when using `FeedPosition`.

- Bluetooth communication setting

See the printer command "Set Bluetooth Communication" described in "MP-B21L SERIES THERMAL PRINTER TECHNICAL REFERENCES" for the Bluetooth communication setting.

Function	Value
iOS Auto Connection Selection (Auto Connection)	1: Enable*1 0: Disable

*1: Select "Enable" when using `ResetPrinter`.

(7) MP-B20

- Memory switch

MS	Function	Value
1-1	Communication Selection (Interface)	1: Wireless
3-1	Automatic Status Response Selection (Auto Status Back)	0: Enable
3-2	Initialized Response Selection (Init. Response)	0: Enable
3-3	Realtime Command Selection (Realtime Command)	0: Enable
3-4	Data Discard Selection When Error Occurs (Error Through)	0: Enable

- Bluetooth communication setting

  See the printer command "Set Bluetooth Communication" described in "MP-B20-B02JK1 THERMAL PRINTER TECHNICAL REFERENCES" for the Bluetooth communication setting.

Function	Value
iOS Auto Connection Selection (iOS Auto Connection)	1: Enable*1 0: Disable

*1: Select "Enable" when using `ResetPrinter`.

(8) MP-A40

- Memory switch

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

*1: Select "Disable" when using `CutPaper`.
*2: Select "Enable" when using `FeedPosition`.

- Bluetooth communication setting

See the printer command "Set Bluetooth Communication" described in "MP-A40 SERIES THERMAL PRINTER TECHNICAL REFERENCES" for the Bluetooth communication setting.

Function	Value
iOS Auto Connection Selection (Auto Connection)	1: Enable*1 0: Disable

*1: Select "Enable" when using `ResetPrinter`.

2.2.2 Android

(1) SLP720RT, SLP721RT

MS	Function	Value
1-2	Taken Mode Selection (Taken Mode)	0 : Enable[*1] 1 : Disable[*2]
1-3	Mark Mode Selection (Mark Mode)	0 : Enable[*3] 1 : Disable[*4]
4-6	Paper Auto Detection Selection (Paper Auto Detection)	0 : Enable[*3] 1 : Disable[*3*4]
5-1	Automatic Status Response Selection (Auto Status Back)	0 : Enable
5-2	Initialized Response Selection (Init. Response)	0 : Enable
5-3	Data Discard Selection When Error Occurs (Error Through)	0 : Enable
5-4	Data Discard Selection When Output Buffer Full Occurs (Response Data Discarding)	1 : Disable
7	Thermal Paper Selection (Thermal Paper)	00B : Receipt 01B : Linerless label 10B : SLP Label
13-3	Realtime Command Selection (Realtime Command)	1 : Enable
17-3	Feed Backward Setting After Paper Cutting (Backfeed After Cut)	0: Enable 1: Disable[*5]

*1: When printing continuously on linerless label, set this value to "Enable".
The status response of the taken sensor is responded when this value is set to "Enable".

*2: When printing continuously on receipt or SLP Label, set this value to "Disable".

*3: When using `FeedPosition`, one of the following settings is necessary.
・To automatically detect paper, set the memory switch MS 4-6 (Paper Auto Detection Selection) of the printer to "Enable".
・To specify the paper, set MS 4-6 (Paper Auto Detection Selection) to "Disable" and set MS 1-3 (Mark Mode Selection) to "Enable".
In addition, select the paper to use as follows.
・For marked linerless label:
Set MS 7 (Thermal Paper Selection) to "Linerless label".
・For SLP Label:
Set MS 7 (Thermal Paper Selection) to "SLP Label".

*4: Set this value to "Disable" and select the paper to be used in the memory switch MS 7 (Thermal Paper Selection) of the printer when using the receipt (other than marked paper) or the Linerless label (other than the marked paper).

*5: When executing `CutPaper` under the following conditions and using `PrintSmartLabelImageData` immediately after, set this value to "Disable".
・MS1-3 (Mark Mode Selection) is set to "Disable".
・**`CUT_FULL`** is specified to `CuttingMethod` of `CutPaper`, or MS1-2 (Taken Mode Selectoin) is set to "Enable" and **`CUT_PARTIAL`** is specified to `CuttingMethod`.

(2) RP-F10, RP-G10

MS	Function	Value
5-1	Automatic Status Response Selection (Auto Status Back)	0: Enable
5-2	Initialized Response Selection (Init. Response)	0: Enable
5-3	Data Discard Selection When Error Occurs (Error Through)	0: Enable
5-4	Data Discard Selection When Output Buffer Full Occurs (Response Data Discarding)	1: Disable
13-3	Realtime Command Selection (Realtime Command)	1: Enable
38-1	Scanner Automatic Status Response Selection*1 (Scanner Auto Status Back)	0: Enable*2

*1: The firmware of the printer to support the barcode scanner is Ver.1.10 or later.
*2: Select "Enable" when using the barcode scanner.
See "USER'S GUIDE" for details about the combination of peripherals.

(3) RP-E10, RP-D10

MS	Function	Value
5-2	Initialized Response Selection (Init. Response)	0: Enable

(4) MP-B30

MS	Function	Value
1-1	Communication Selection (Interface)	1: USB/Wireless*1 0: USB*2
1-2	Mark Mode Selection (Mark Mode)	1 : Disable*3 0 : Enable
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`.

(5) MP-B30L

MS	Function	Value
1-1	Communication Selection (Interface)	1: USB/Wireless*1 0: USB*2
1-2	Mark/Gap Mode Selection (Mark/Gap Mode)	1 : Disable*3 0 : Enable*4
1-3 to 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
2-1	Initialization Performance Selection After Paper Setting (Paper Set Handle)	0: Form Feed
2-2	Realtime Command Selection (Realtime Command)	0: Enable
2-7	Paper Form Feed Selection by Feed Switch (SW Paper Form Feed)	1: Cut Position Form Feed (Cut Position)
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`.

(6) MP-B21L

MS	Function	Value
1-1	Interface Selection (Interface)	1: USB/Wireless*1 0: USB*2
1-2	Mark Gap Mode Selection (Mark/Gap Mode)	1: Disable*3 0: Enable*4
1-3 to 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`.

(7) MP-B20

MS	Function	Value
1-1	Communication Selection (Interface)	1: Wireless*1 0: USB*2
3-1	Automatic Status Response Selection (Auto Status Back)	0: Enable
3-2	Initialized Response Selection (Init. Response)	0: Enable
3-3	Realtime Command Selection (Realtime Command)	0: Enable
3-4	Data Discard Selection When Error Occurs (Error Through)	0: Enable
3-5	Data Discard Selection When Output Buffer Full Occurs (Response Data Discarding)	1: Disable

*1: Select "Wireless" for Bluetooth connection.
*2: Select "USB" for USB connection.

(8) MP-A40

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

*1: Select "USB/Wireless" for Bluetooth connection or TCP/IP Connection.
*2: Select "Disable" when using `CutPaper`.
*3: Select "Enable" when using `FeedPosition`.

2.3 Precaution

2.3.1 iOS

This library is not thread safe. When this library is used on multiple threads, abnormal termination may occur.

When TCP/IP connection is used, the communication port cannot be shared with printer drivers or other libraries in this library.
When TCP/IP connection is used, wireless LAN access point which the iOS device is connected to and the printer need to be connected to the same network.

This library does not support a concurrent connection from multiple apps to one printer when multiple apps are started simultaneously by the Multitasking function for the iPad with iPadOS.

2.3.2 Android

When using Bluetooth, it is necessary that the connection has been established by SPP (Serial Port Profile).

When using USB, it is necessary that the Android device has supported the USB host function.

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

When TCP/IP connection is used, wireless LAN access point which the Android device is connected to and the printer need to be connected to the same network.

Chapter 3
How to Use Library

This chapter describes the development of an iOS and Android application and how to use the library.

3.1 iOS or Android Application Development Environment

In order to develop iOS or Android applications, the following tools are required.

- iOS and Android

  Windows: Visual Studio 2022 or Visual Studio 2026

  Mac: Visual Studio Code

- iOS

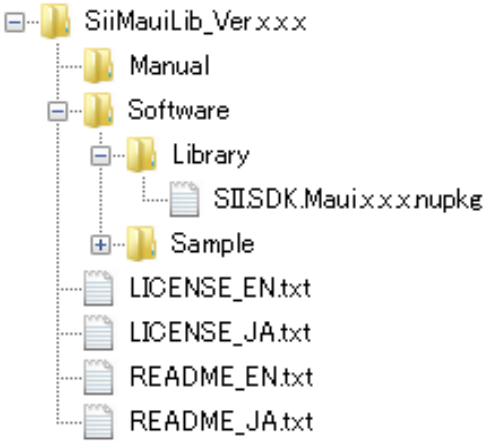
  Mac: Xcode16.1 or later

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

(NOTE) The Xcode is required to have been installed on Mac to build an iOS application.

3.2 Provided Files

The file configuration of the SDK is as follows.



The file format of the library is NuGet package. The file name of the library is SII.SDK.Maui.x.x.x.nupkg.

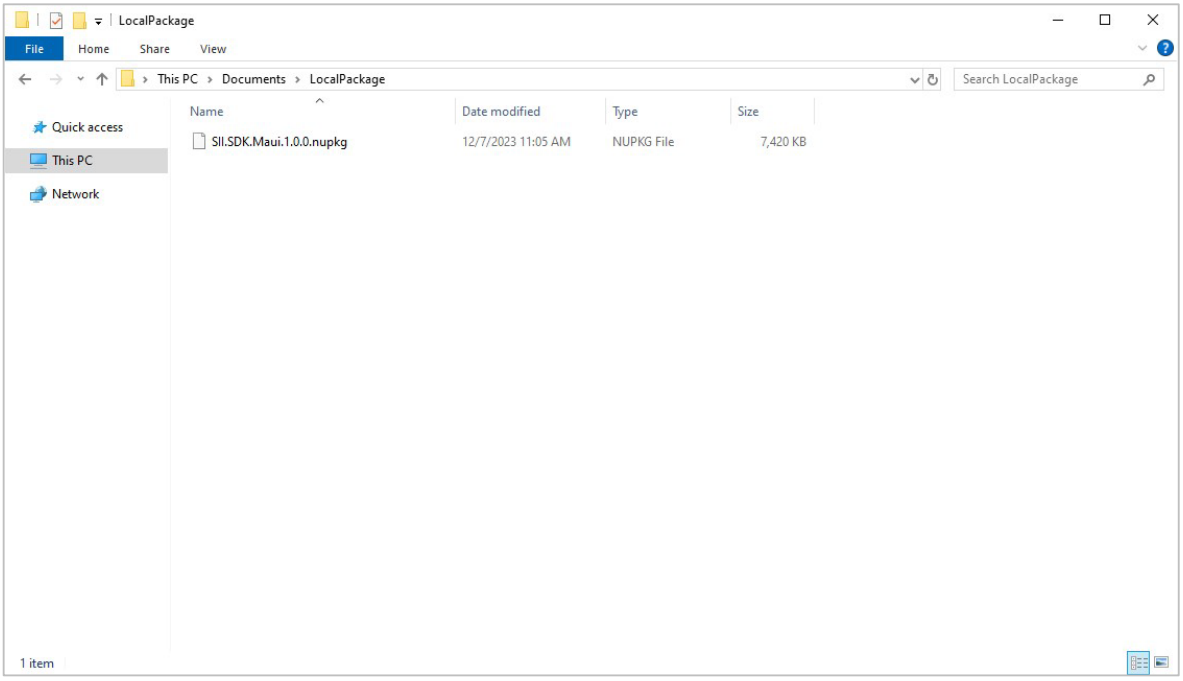
3.3 Build Library into Visual Studio

Using the project of the sample program (Sample) included in SDK as an example, this section describes how to build the library into the project.
See "Chapter 6 Sample Program" for the sample program included in the SDK.

3.3.1 Windows

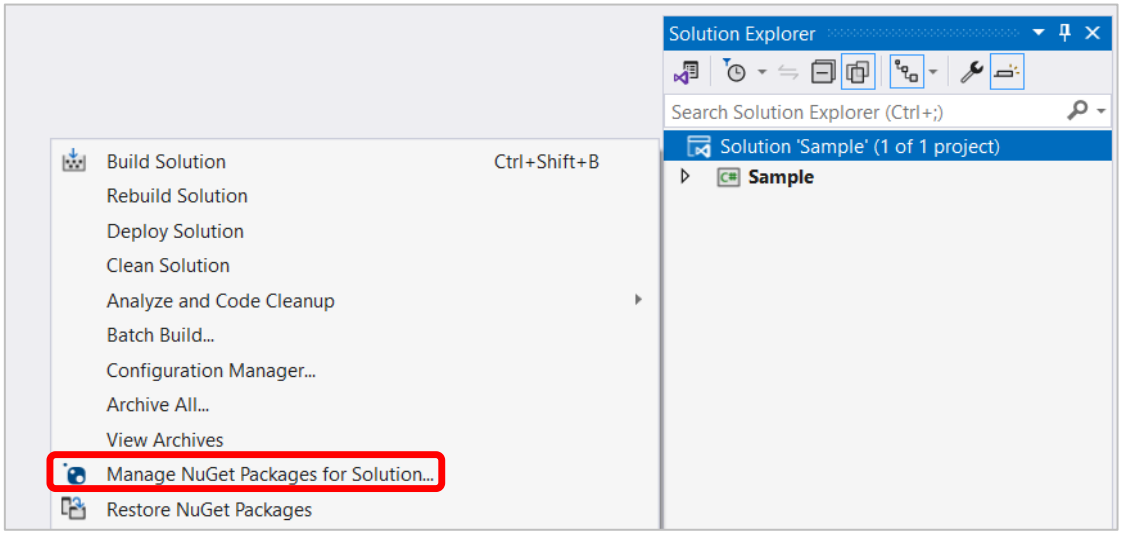
(1) Common to iOS and Android

(a) Allocate the library to an arbitrary folder.
As an example, create the "LocalPackage" folder and allocate SII.SDK.Maui.x.x.x.nupkg in the folder.

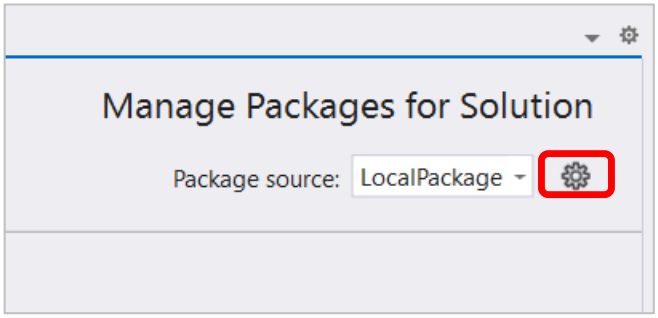


(b) Open Visual Studio

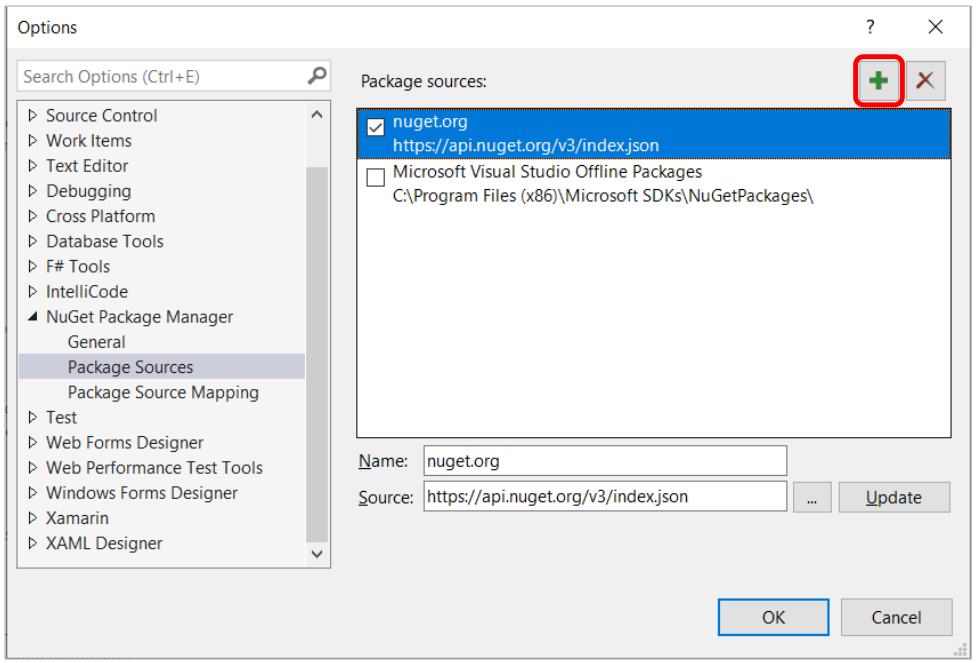
(c) Select [Manage NuGet Packages for Solution…] from the right-click menu of the [Solution 'Sample'] in the Solution Explorer.



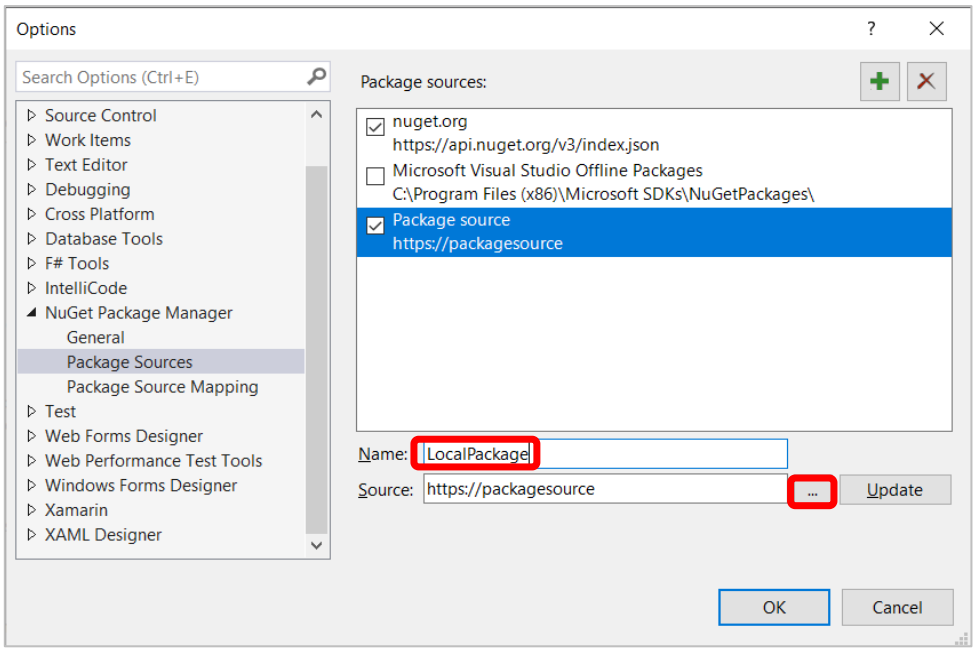
(d) Click the gear icon on the [Manage Packages for Solution].



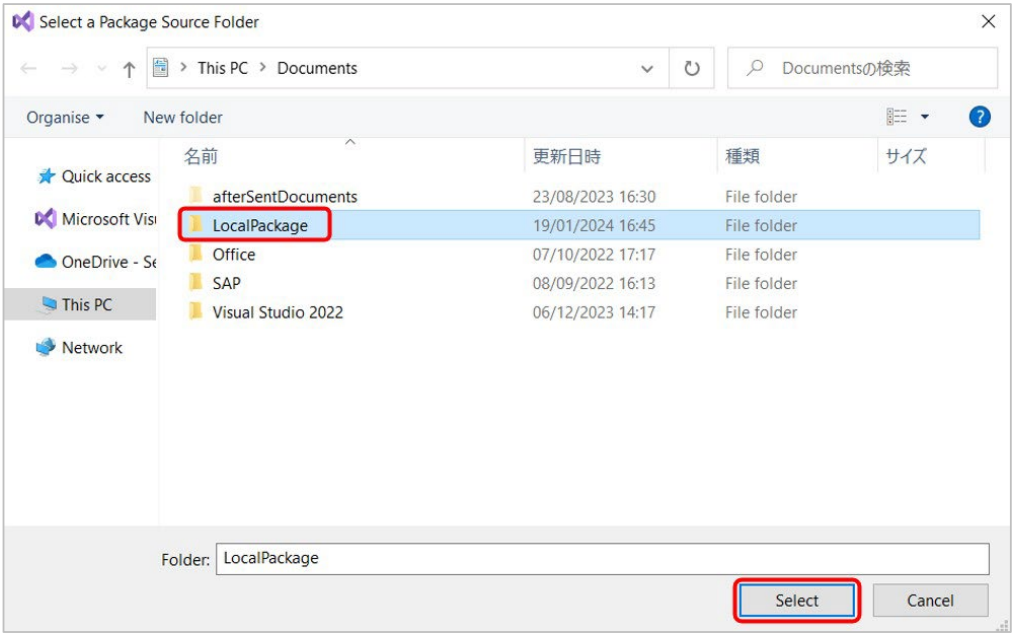
(e) Click the [+] button on the "Options" dialog.



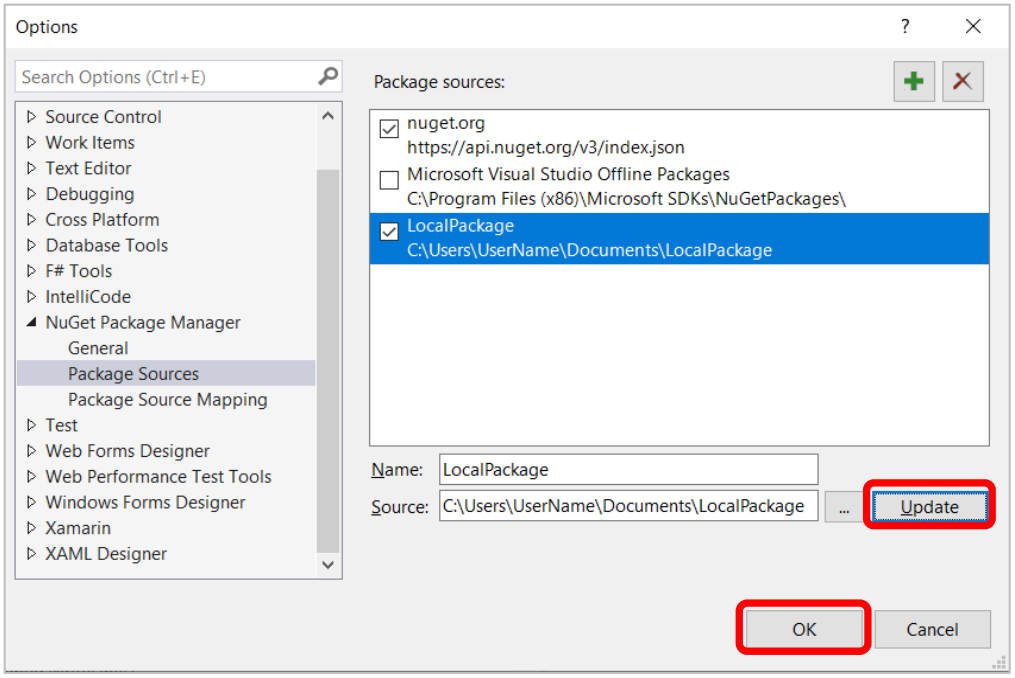
(f) Change the name in the "Name" to LocalPackage and click the [. . .] button.



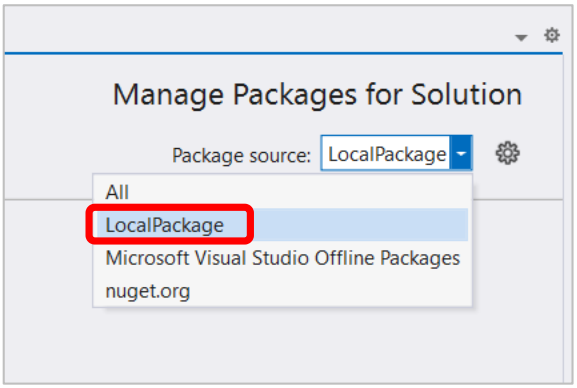
(g) On the "Select a Package Source Folder" dialog, select the "LocalPackage" folder created in the process (a) and click the [Select] button.



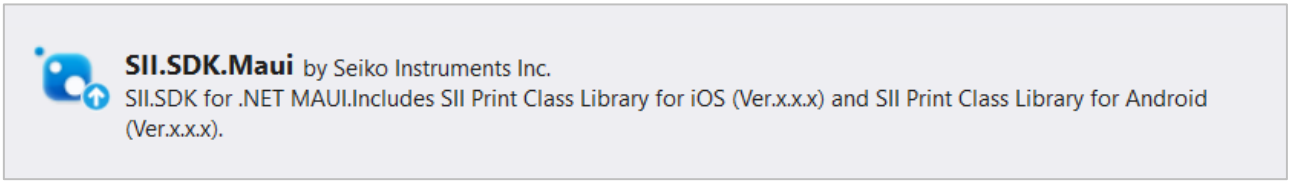
(h) The "Options" dialog is displayed. Click the [Update] button and then click the [OK] button.



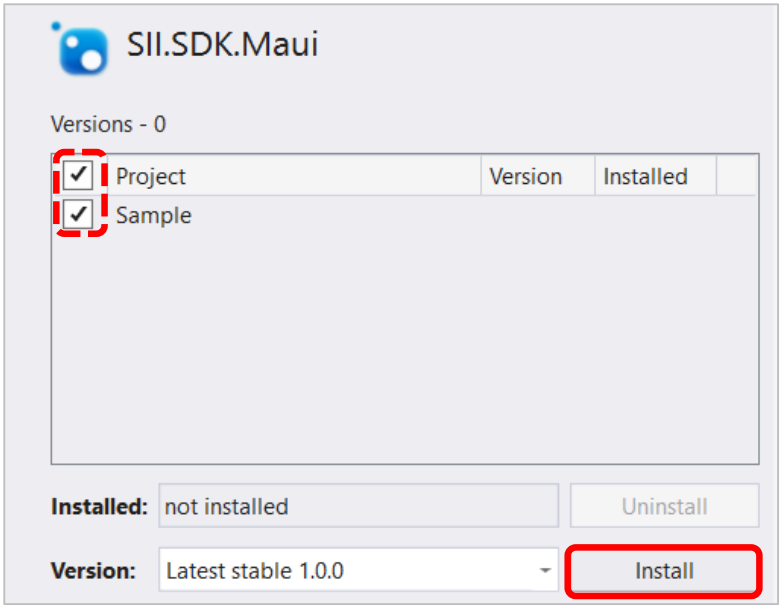
(i) Select [LocalPackage] in the pulldown menu on the "Manage Packages for Solution" - "Package source".



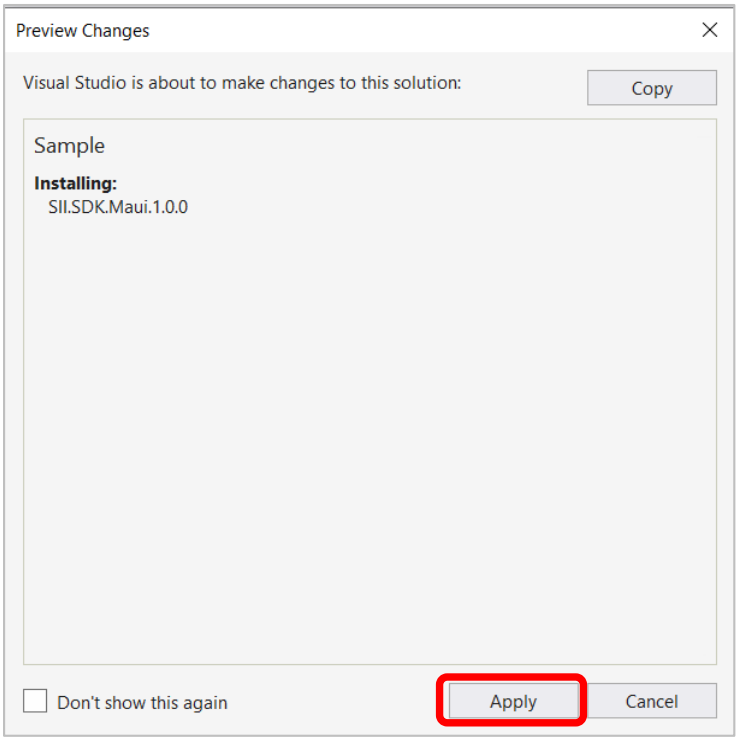
(j) Select the [SII.SDK.Maui].



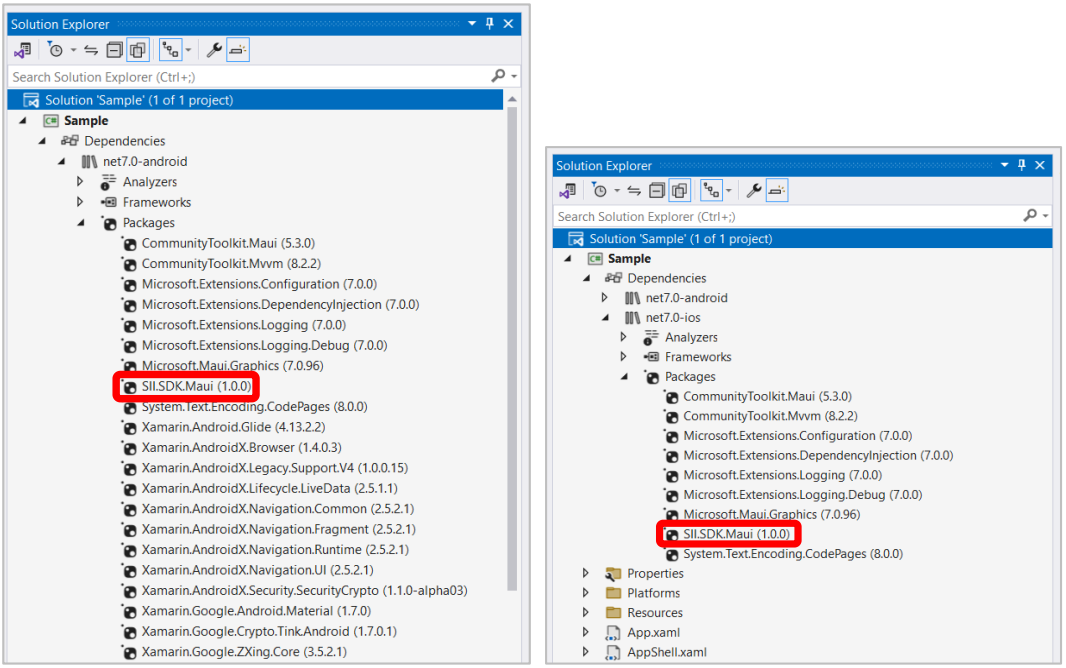
(k) Check the boxes of necessary projects on the [SII.SDK.Maui] property panel and then click the [Install] button.



(l) After the "Preview Changes" dialog is displayed, click the [Apply] button.



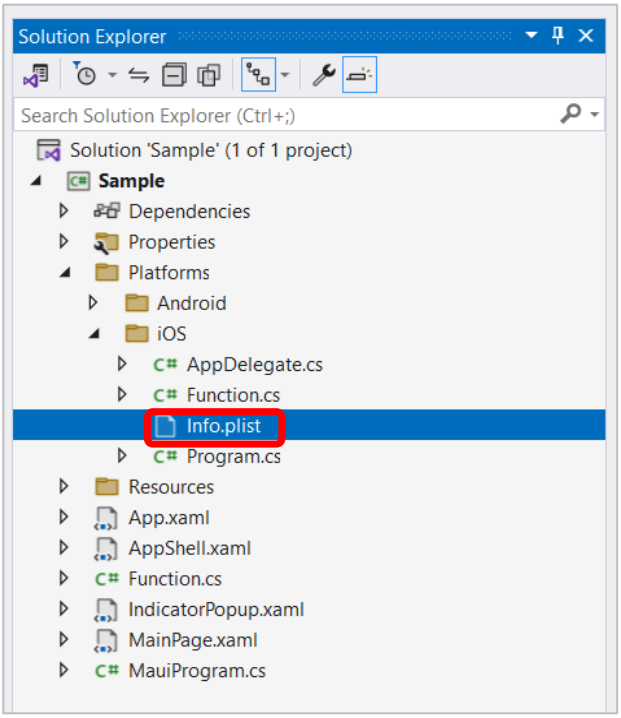
(m) Confirm that the project has been installed.



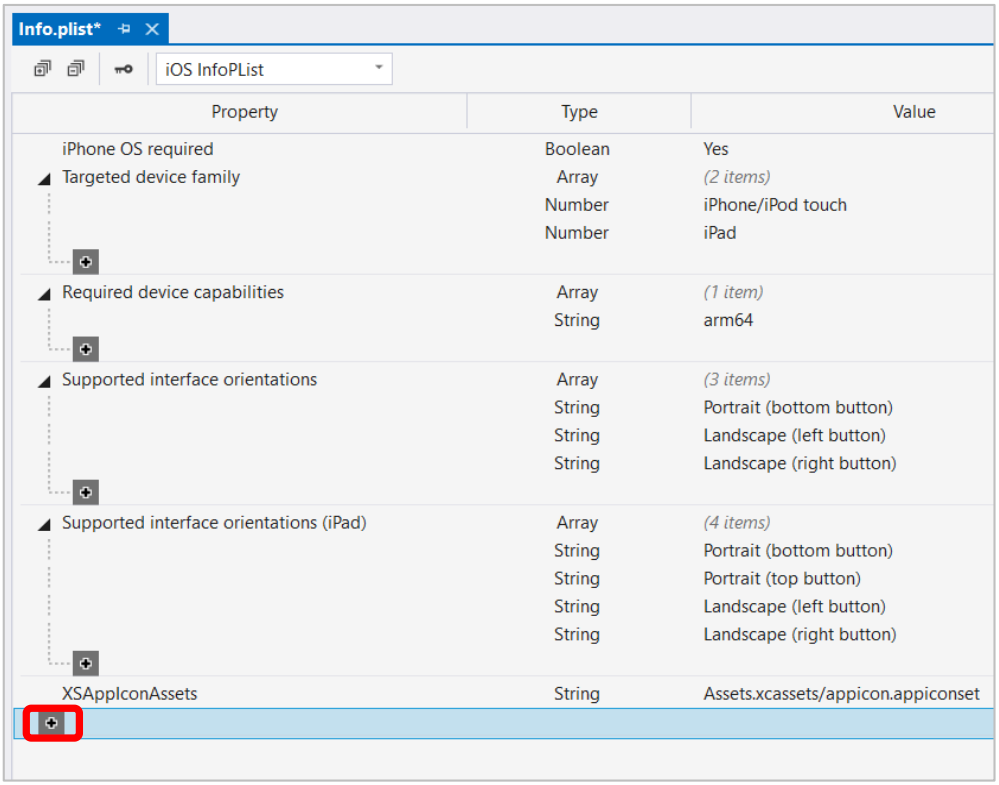
The following steps differ depending on the platform. See "3.3.1(2) iOS" for iOS. See "3.3.1(3) Android" for Android.

(2) iOS

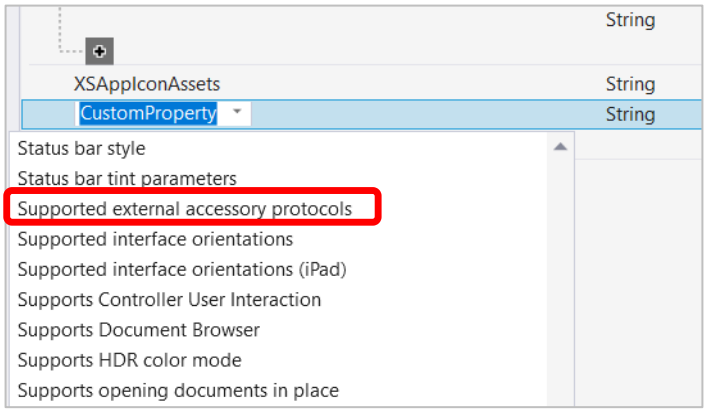
(n) Select [View Designer] from the right-click menu of the [Sample] - [Info.plist] on the Solution Explorer.



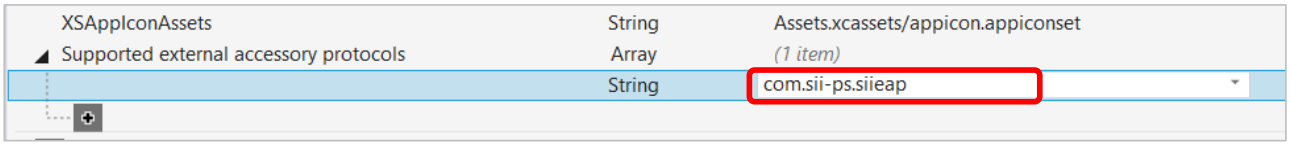
(o) Click the [+] on the [Info.plist] to expand the list.



(p) Select the [Supported external accessory protocols] in the list.



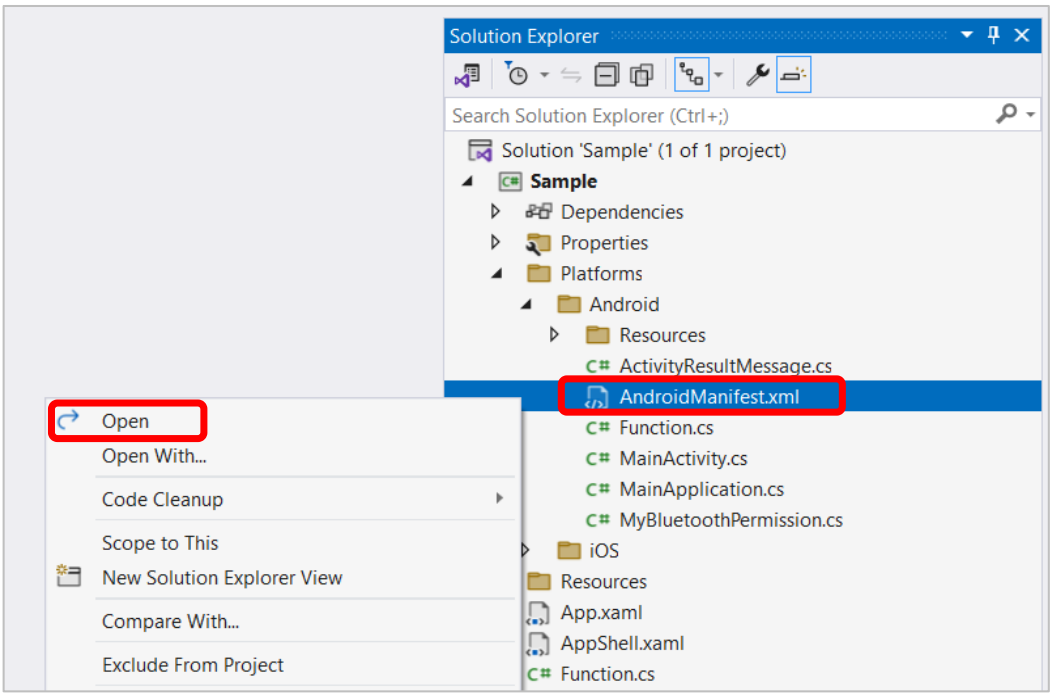
(q) Enter com.sii-ps.siieap in the expanded [Supported external accessory protocols] as a value.



By completing these procedures, the library function becomes available.

(3) Android

(n) Select [Open] from the right-click menu of [Sample] – [AndroidManifest.xml] in the Solution Explorer.



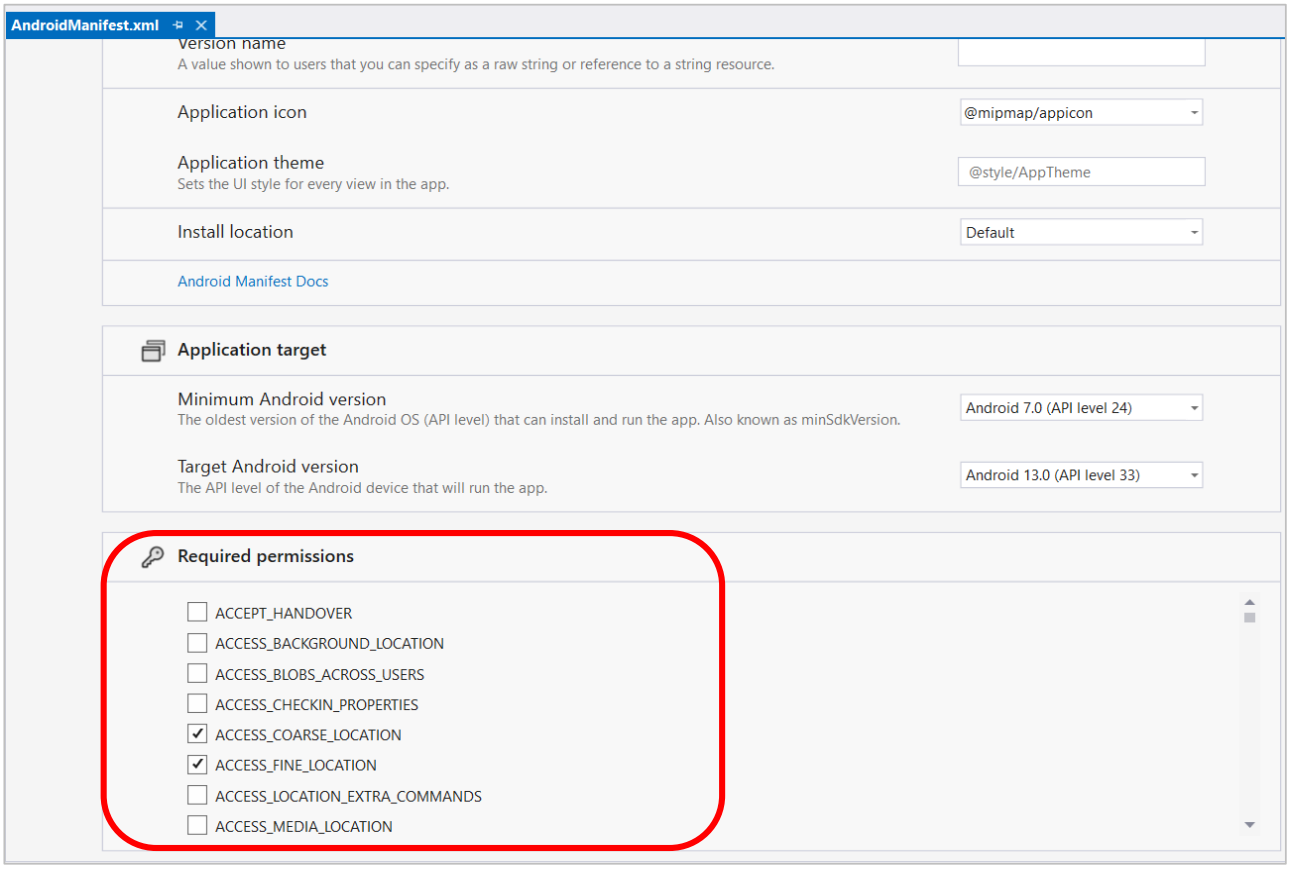
(o) Check the boxes on the "Required permissions" the following items.

[When using Bluetooth]

- ACCESS_FINE_LOCATION
- ACCESS_COARSE_LOCATION
- BLUETOOTH
- BLUETOOTH_ADMIN
- BLUETOOTH_CONNECT
- BLUETOOTH_SCAN

[When using TCP/IP]

- ACCESS_WIFI_STATE
- CHANGE_WIFI_STATE
- INTERNET



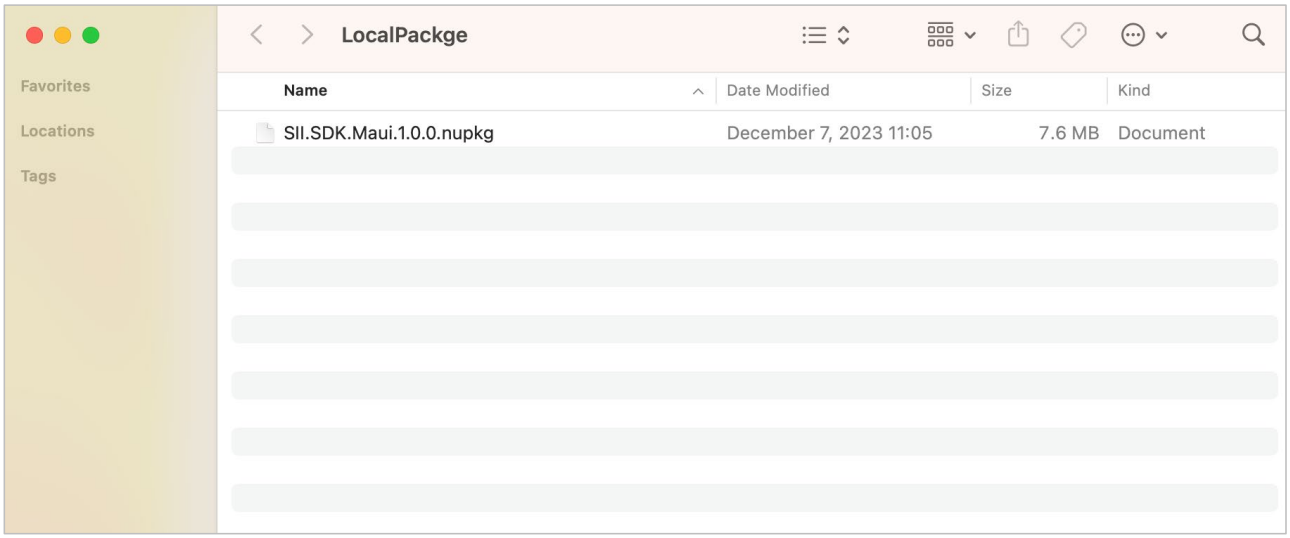
(p) When using SmartLabelManager Class, add "Xamarin.Google.ZXing.Core" package from "Manage NuGet Packages…" in Sample.

By completing these procedures, the library function becomes available.

3.3.2 Mac

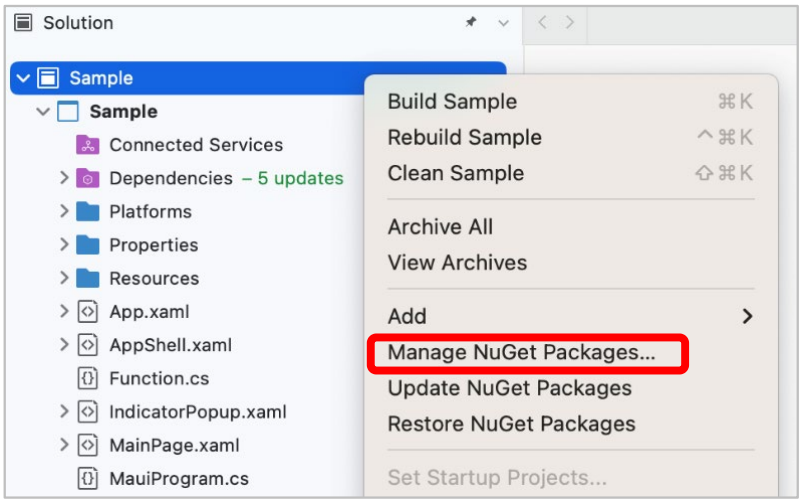
(1) Common to iOS and Android

(a) Allocate the library to an arbitrary folder.
As an example, create the "LocalPackage" folder and allocate SII.SDK.Maui.x.x.x.nupkg in the folder.

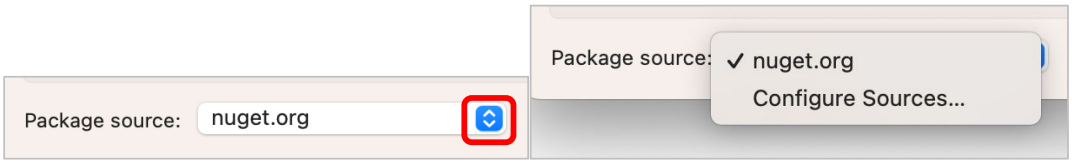


(b) Open Visual Studio.

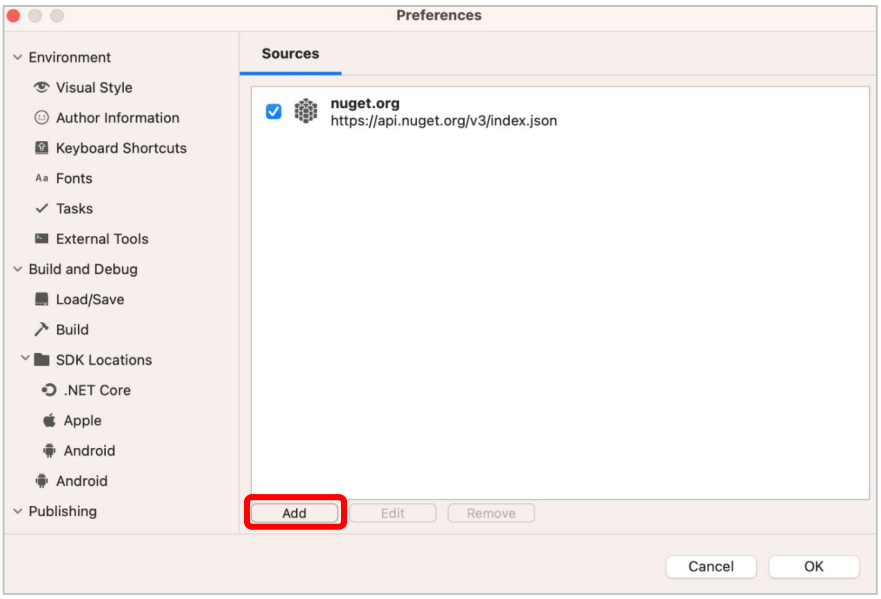
(c) Select [Manage NuGet Packages…] from the right-click menu of "Sample" in Solution Explorer.



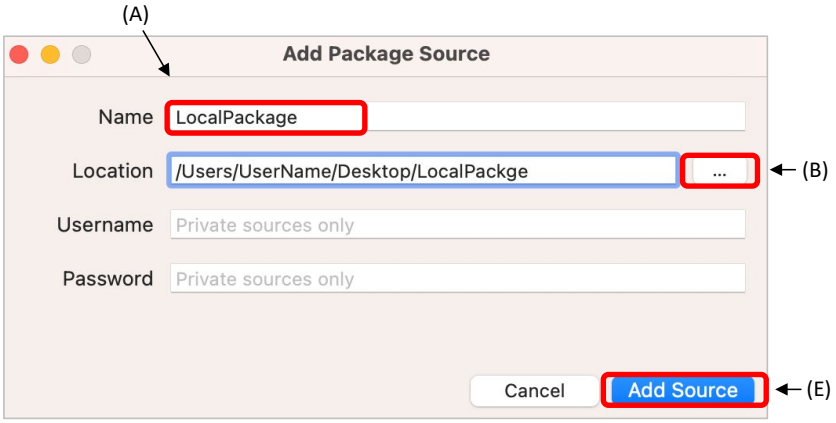
(d) Select [Configure Sources…] from the drop down menu in [Package Sources].

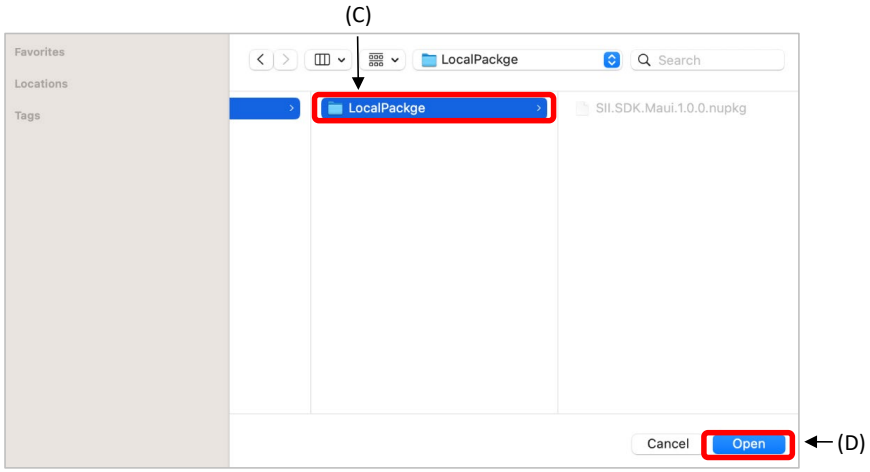


(e) Click the [Add] button on the "Preferences" dialog.

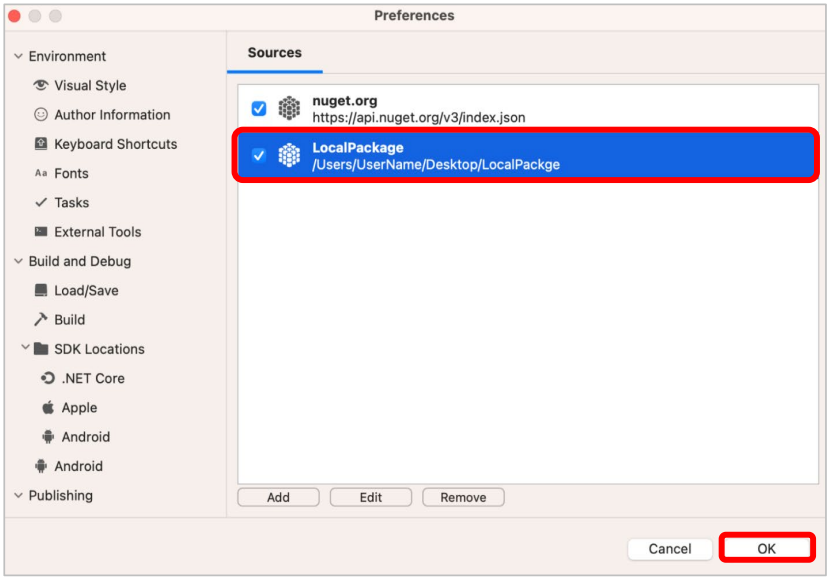


(f) Enter LocalPackage in the "Name" on the "Add Package Source" dialog (A) and click the […] button (B). Select the "LocalPackage" folder created in the process (a) on the "Select a Package Source Folder" dialog (C) and click the [Open] button (D).
The "Add Package Source" dialog is displayed. Click the [Add Source] button (E).

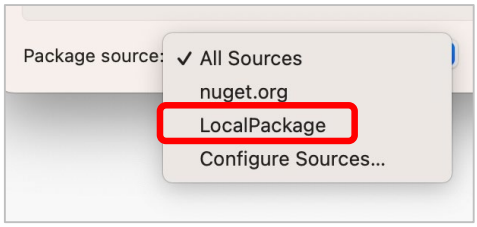




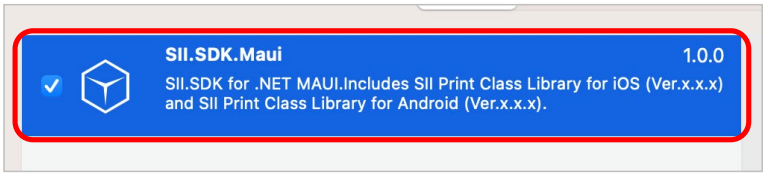
(g) Confirm the "Preferences" dialog that the "LocalPackage" has been added and click the [OK] button.



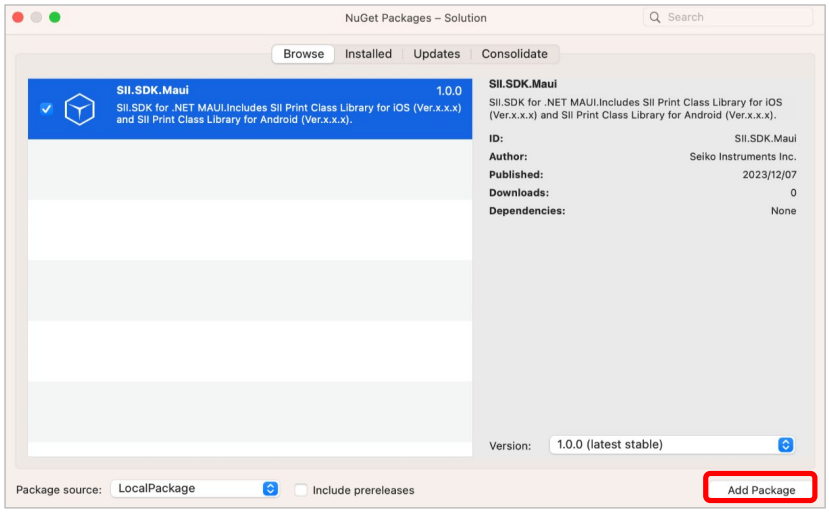
(h) Select [LocalPackage] from the drop-down menu on "Package sources".



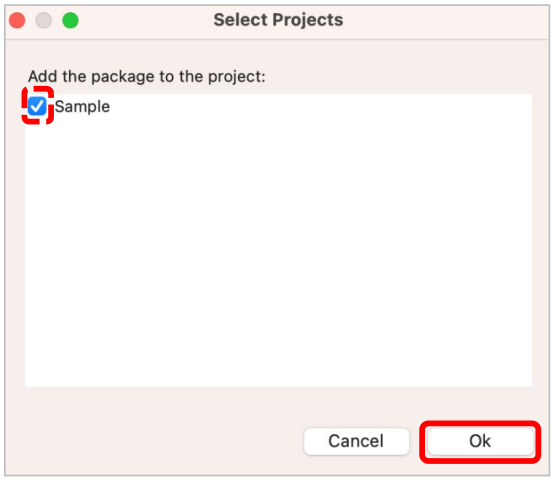
(i) Select the [SII.SDK.Maui].



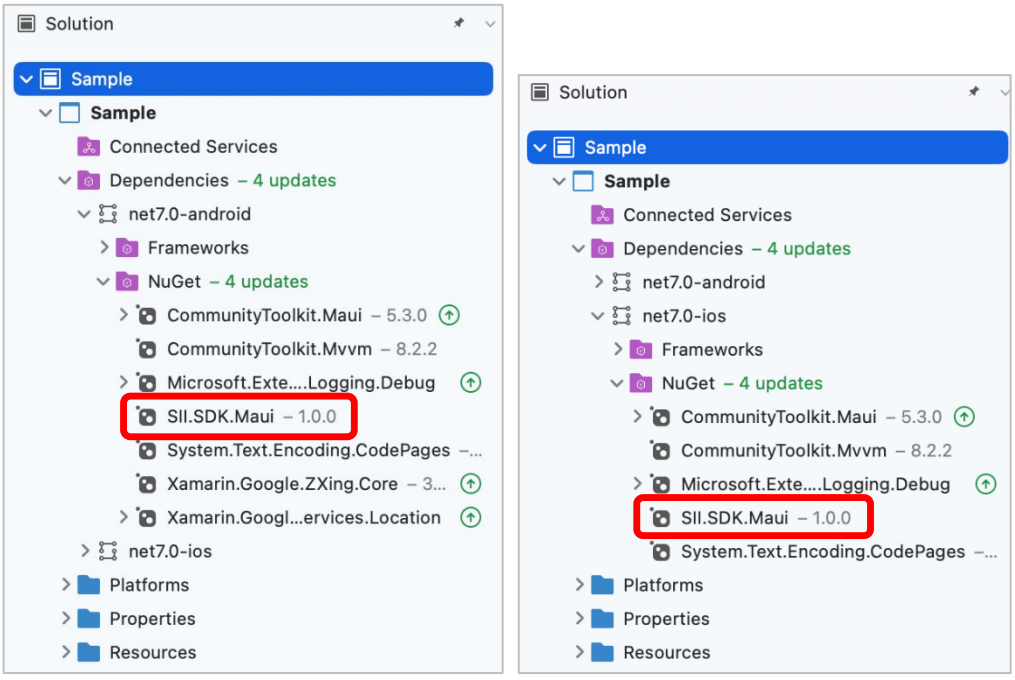
(j) Click the [Add Package] button.



(k) Check the boxes of necessary projects on the "Select Projects" dialog to add and then click the [Ok] button.



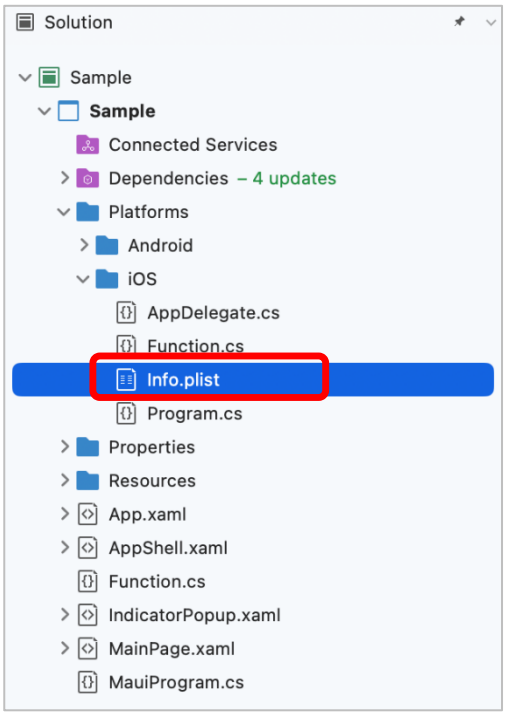
(l) Confirm that the projects have been installed.



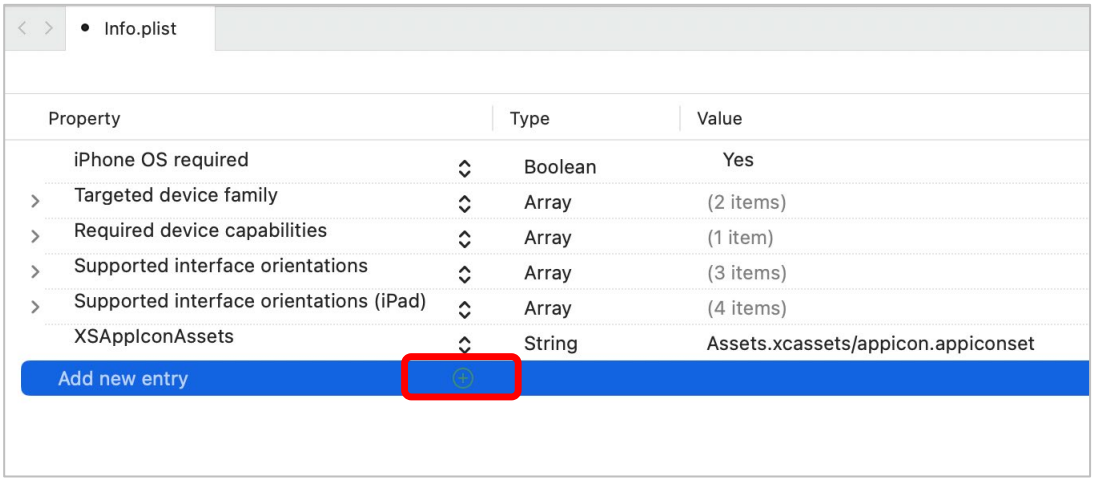
The following steps differ depending on the platform. See "3.3.2(2) iOS" for iOS. See "3.3.2(3) Android" for Android.

(2) iOS

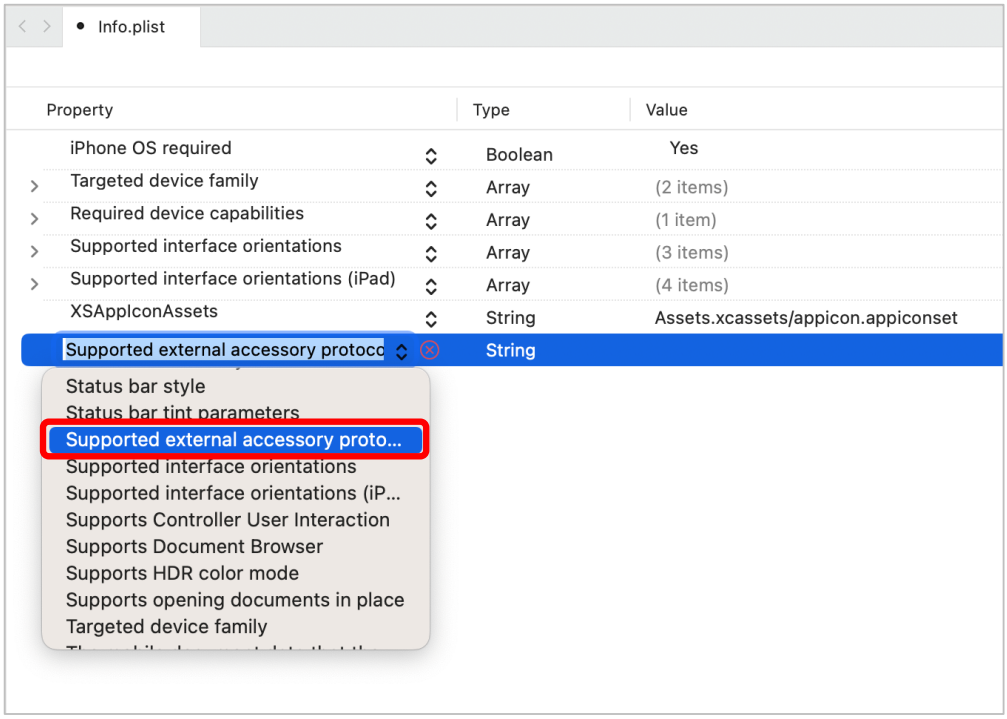
(m) Select the [Sample] - [Info.plist] in the Solution Explorer.



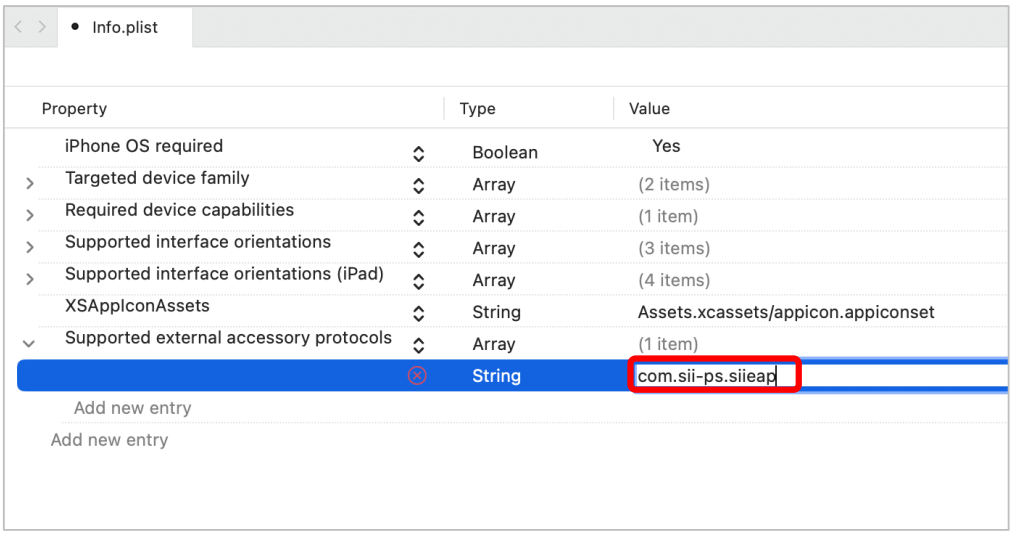
(n) Select the ⊕ of "Property" – "Add new entry".



(o) Select the [Supported external accessory protocols] from the list.



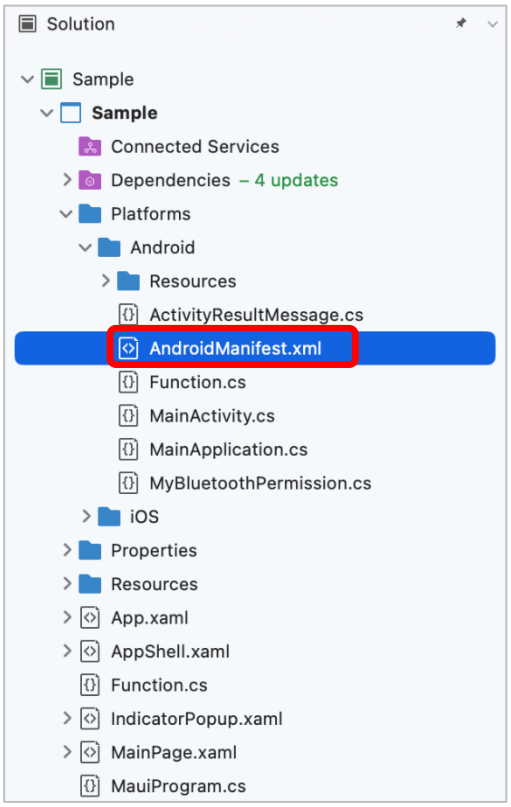
(p) Enter com.sii-ps.siieap in the expanded [Supported external accessory protocols] as a value.



By completing these procedures, the library function becomes available.

(3) Android

(m) Select the [Sample] – [AndroidManifest.xml] on Solution Explorer.



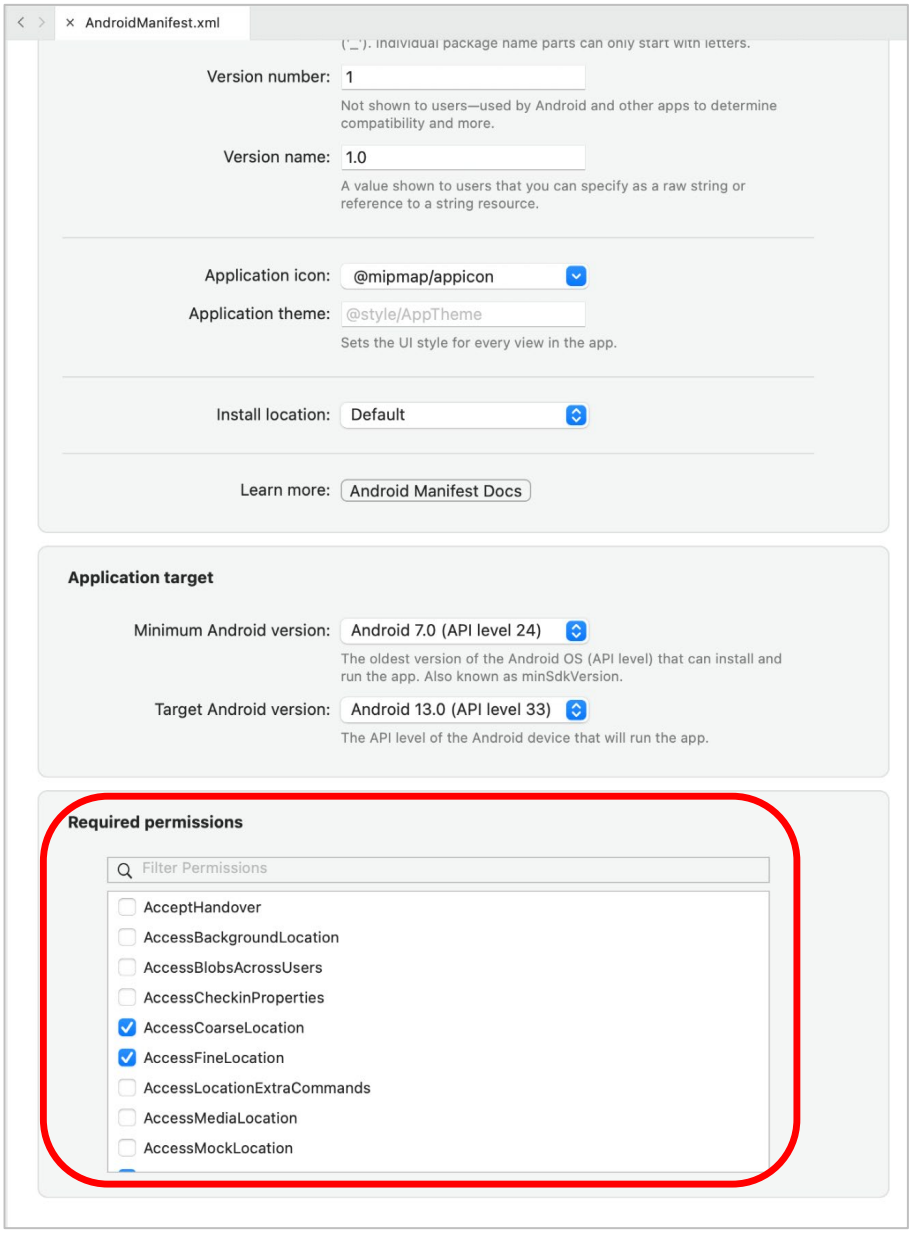
(n) Check the boxes on the "Required permissions" the following items.
Click the [OK] button after checking the boxes.

[When using Bluetooth]

- `AccessFineLocation`
- `AccessCoarseLocation`
- `Bluetooth`
- `BluetoothAdmin`
- `BluetoothConnect`
- `BluetoothScan`

[When using TCP/IP]

- `AccessWifiState`
- `ChangeWifiState`
- `Internet`



(o) When using `SmartLabelManager` Class, add "Xamarin.Google.ZXing.Core" package from "Manage NuGet Packages…" in Sample.

By completing these procedures, the library function becomes available.

3.4 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

- **About the TargetFrameworks Specified in the .csproj File**

  In TargetFrameworks, specify the .NET version supported by this library.
  If a version outside the supported range is specified, build errors (for example, NETSDK1202 or NU1202) may occur.

Chapter 4
Library Functions of Printer

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

4.1 Standard Mode and Page Mode

4.1.1 Basic Operation

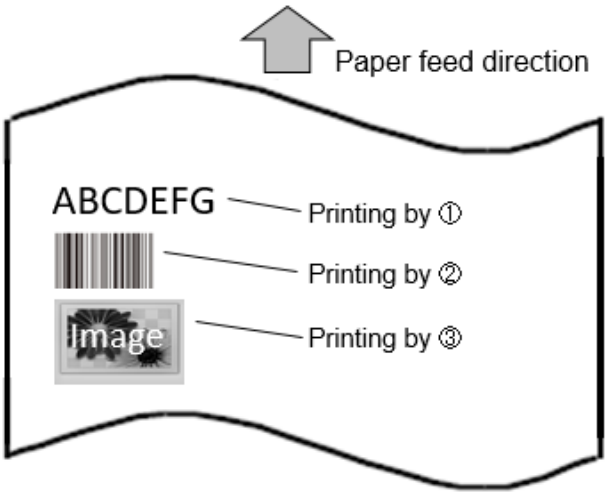
There are two printing modes "Standard mode" and "Page mode" in the library.
The "Standard mode" and "Page mode" are described below.

"Page mode" is not supported by SLP720RT, SLP721RT, and MP-B20.

(1) Standard mode

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

Sample print command
① Send text data
② Print barcode
③ Send specified file (Specify an image file)



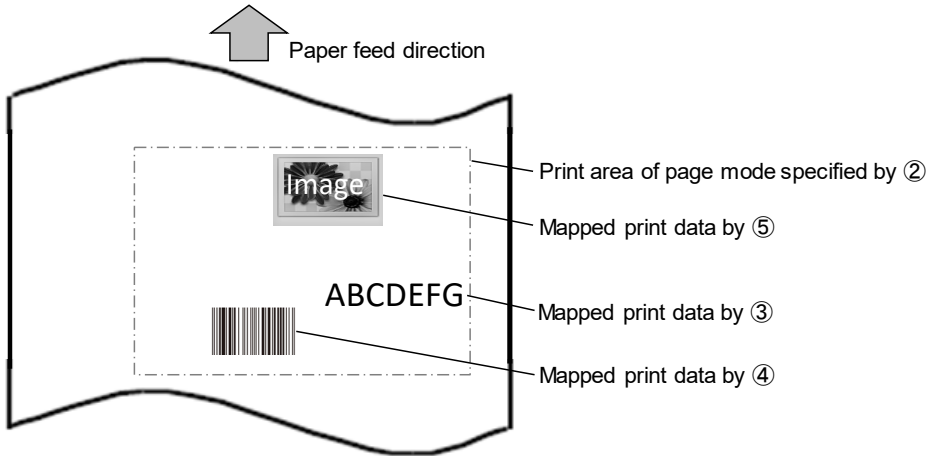
Standard mode suits the printing with an unfixed length such as a receipt.

(2) Page mode

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

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

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



Page mode suits the printing for the followings.
・The printing with a fixed length.
・The printing with the coordinate determination of the character starting position or the ruled line print position.

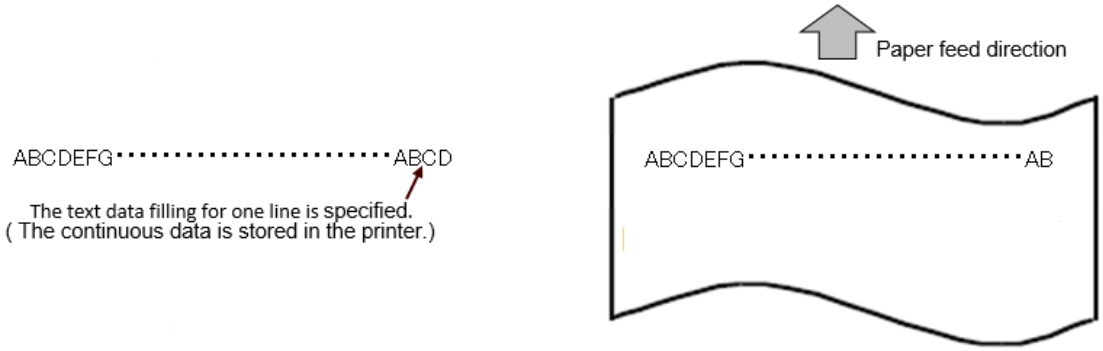
4.1.2 Text Data Printing in Standard Mode

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

The text data is stored in the printer when the text data less than one line is specified.
The stored text data is printed by either the following conditions.

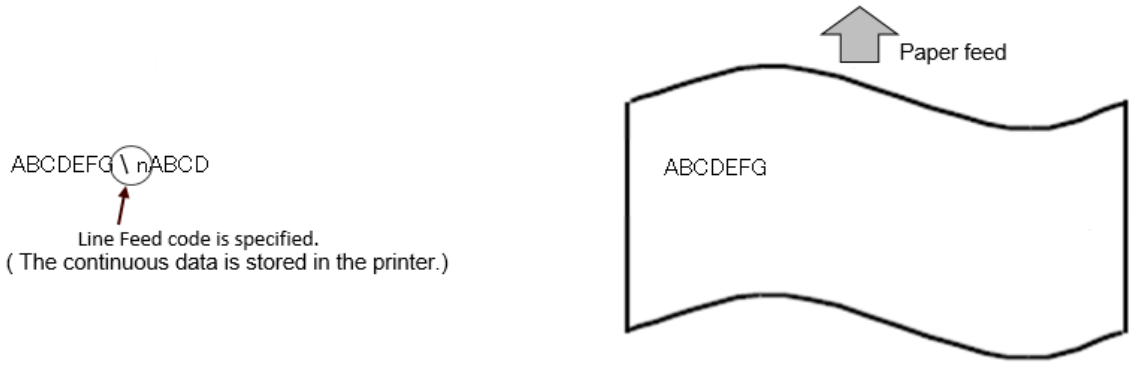
・The text data filling for one line is specified.
・Line Feed code is specified.

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



Input data **Print result**

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



Input data **Print result**

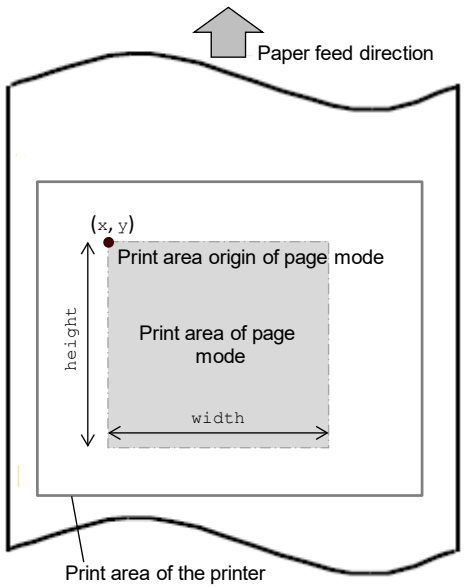
4.1.3 Mapping Position of Print Data in Page Mode

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

(1) Print area of page mode

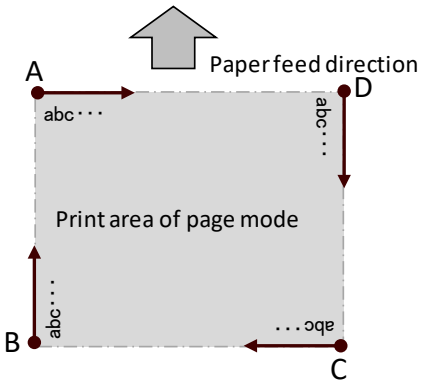
The print area of page mode is specified against the print area of the printer by the print area origin, and the width and the height of page mode. The view of the print area is shown in the following figures.

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



(2) Print direction

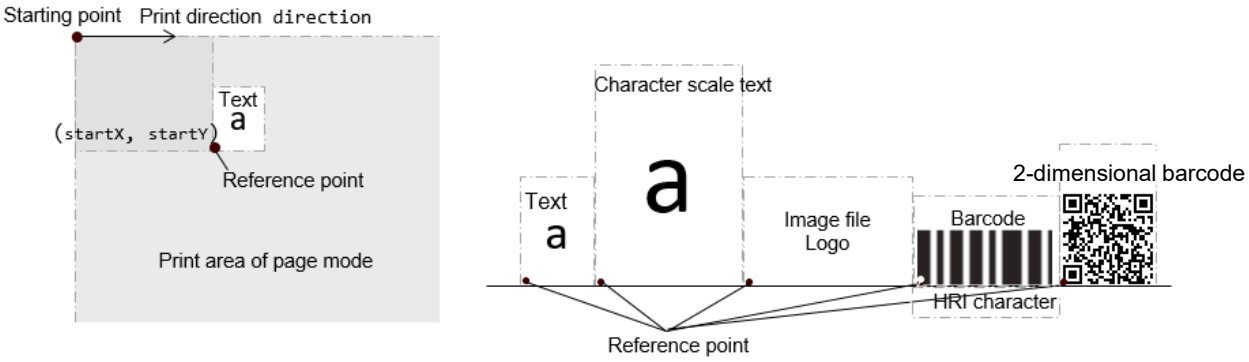
Specify the print direction at setting the print area of page mode.
The starting point is changed depending on specifying the print direction for each direction.
The relation between the print direction and the starting point is shown in the figure below.



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

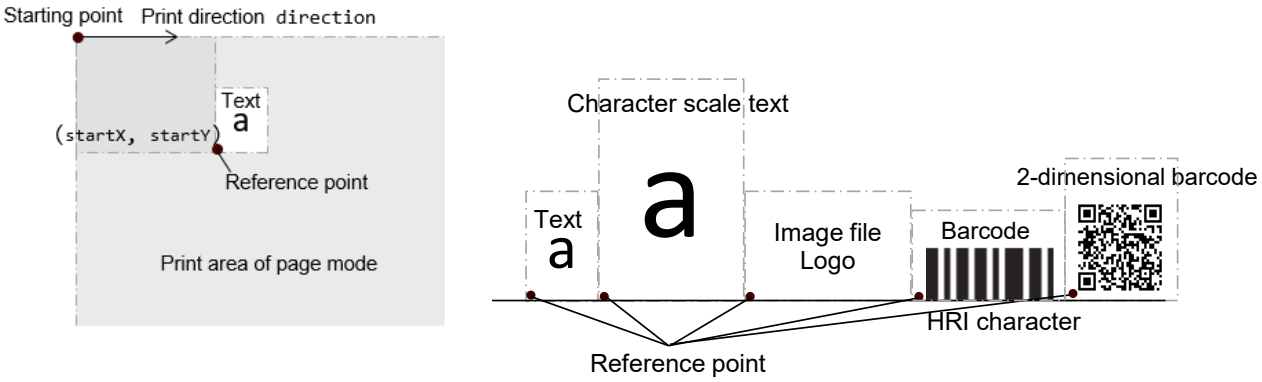
(3) Reference point

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



Relation Between Reference Point and Each Printing Element of RP-F10, RP-G10, MP-B30, MP-B30L, and MP-B21L

(NOTE) ・**The reference point cannot be specified out of the print area of page mode.**
・**The quiet zone is not included in the print element of barcode.**



Relation Between Reference Point and Each Printing Element of RP-E10, RP-D10, and MP-A40

(NOTE) ・**The reference point cannot be specified out of the print area of page mode.**
・**The quiet zone is included in the print element of barcode.**

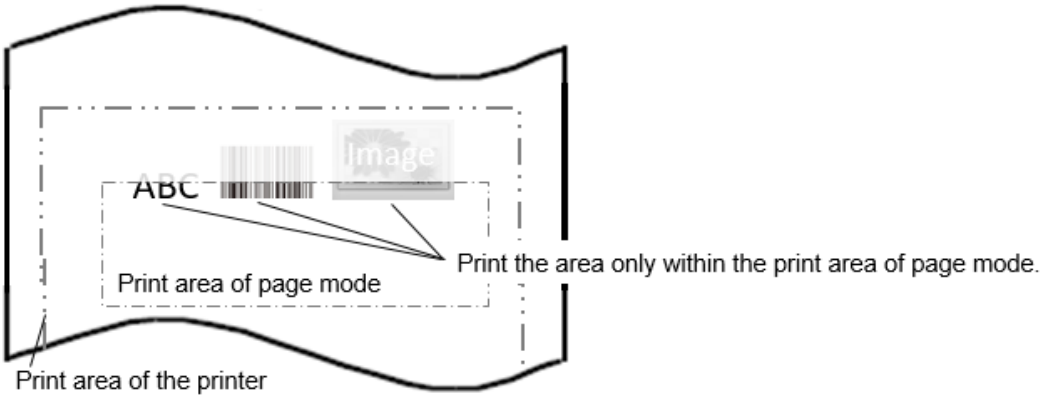
4.1.4 Print Data Process at Out of Print Area of Page Mode

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

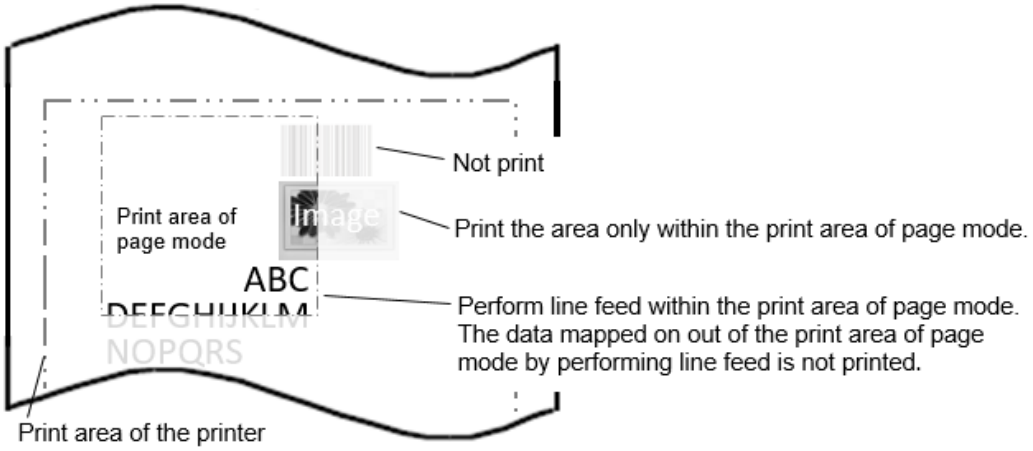
Text	Barcode, 2-dimensional Barcode	Image File, Logo, Rectangle, Ruled Line
ABC		

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



(NOTE) The print data specified by `BARCODE_CUSTOR_BAR_CODE_JP` is not printed.

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



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

4.2 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".

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

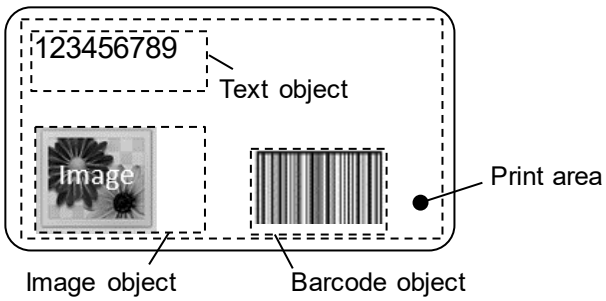
- **iOS**

- **Android**

Reference It is possible to search "SII Layout Editor" and install in App Store or Google Play.

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

Supported objects in the library 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.
- When the font set in the app is not included in the library, the text data will be printed using the system’s 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
  - CODE128
  - ITF (support only Android)
  - UPC-A
  - EAN13
  - CODABAR
  - UPC-E
  - EAN8
  - PDF417
  - Data Matrix
  - QR Code

- The "Serialization" is not supported.

- The barcode setting shown in the 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.
- When the font set in the app is not included in the library, the text data will be printed using the system’s standard font.

⑤ Contact object

- When the font set in the app is not included in the library, the text data will be printed using the system’s standard font.

⑥ DateTime object

- When the font set in the app is not included in the library, the text data will be printed using the system’s standard font.

⑦ Drawing object

- When "Line Width" is too thin, dashed, long dashed, or double lines may be squished.
- The drawing position of the drawing object may differ between the app and the library.

4.2.2 Method for Using Label File

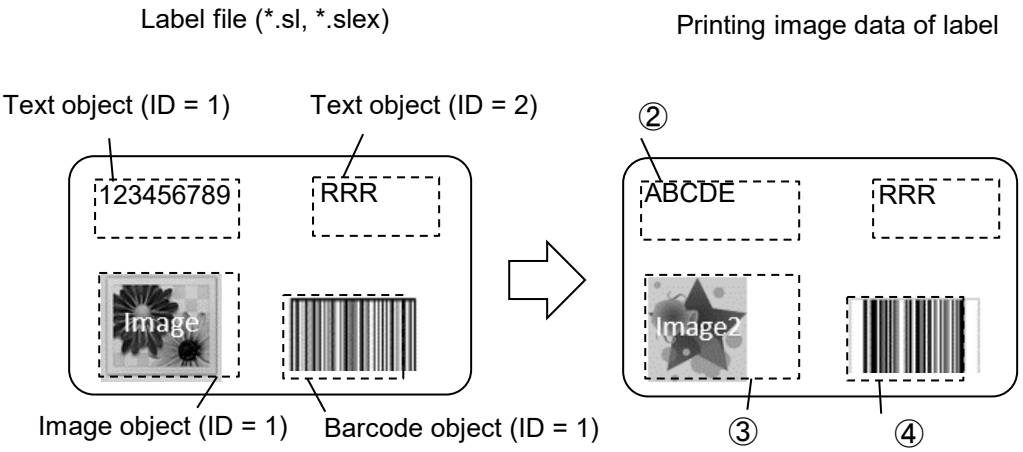
The printing method using the label file is described below.

(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 **PrinterException** is thrown. 2 : Records API execution history.
LOGSIZEMAX	Log file maximum size	1MB : Log file maximum size is 1 MB 5MB : Log file maximum size is 5 MB 10MB : Log file maximum size is 10 MB 50MB : Log file maximum size is 50 MB
LOGOUTPUT	System log output enabled/disabled*1	0 : System log output is disabled 1 : System log output is enabled

*1: System log output
- For iOS: Console
- For Android: Logcat

4.3.3 Log File

Log files are saved as local files in the app 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

This library includes the following classes.

✓: Supported, - : Not supported

Name	Description	SLP720RT SLP721RT MP-B30L MP-B21L	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B20 MP-A40
`PrinterManager`	Provides the APIs used for communication with the printer and for printing. See "**4.4.2 `PrinterManager` Class**". See "**4.4.1 Enumerated Constant List**" for enumerated constants necessary for APIs provided in the `PrinterManager` class.	✓	✓	✓	✓
`DeviceInfo`	Stores the device information found by `StartDiscoveryPrinter`. See "**4.4.3 `DeviceInfo` Class**". See "**4.4.1 Enumerated Constant List**" for enumerated constants necessary for APIs provided in the `DeviceInfo` class.	✓	✓	✓	✓
`PrinterIdentifier`	Stores the printer ID of the printer retrieved by `GetPrinterResponse`. See "**4.4.4 `PrinterIdentifier` Class** ".	✓	✓	✓	✓
`PrinterException`	Exception class that is thrown at API call. See "**4.4.5 `PrinterException` Class**".	✓	✓	✓	✓
`SmartLabelManager`	Provides the API to specify label files or replace data. See "**4.4.6 `SmartLabelManager` Class**" for more details.	✓	–	–	–

Reference See "Chapter 5 Library Functions of Display" for the API reference when using DSP-A01 via the printer or alone.

The following is the description of the described contents in each class of the methods and properties.

Method or Property　Method name or property name

The outline of the method or property is described.

Target　The supported platform and device are described.

- Platform

	Supported All Platforms	**Supported Different Device Depending on Platform**
Description in "Target"	All Platforms	iOS
		Android

- Device

	Supported Multiple Devices		**Supported Individual Device**
Description in "Target"	All devices	All printers	SLP720RT
			SLP721RT
			RP-F10
			RP-G10
			RP-E10
			RP-D10
			MP-B30
			MP-B30L
			MP-B21L
			MP-B20
			MP-A40
		-	DSP-A01 via the printer
			DSP-A01 alone

Syntax　Describe syntax.

Parameter　Parameter and the description of the parameter are described.

Valid range　The valid range is described when there is a valid range.

Error　The description of when an error occurred is described.

Initial value　The initial value is described when there is an initial value.

Description　Description of the method is described.

Return value　The return value is described when there is a return value.

4.4.1 Enumerated Constant List

Constants of enumerated type are described in the following.

① Device model (`DeviceModel`)

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

Constant Name	Description
DEVICE_MODEL_RP_D10	RP-D10
DEVICE_MODEL_RP_E10	RP-E10
DEVICE_MODEL_MP_B20	MP-B20
DEVICE_MODEL_MP_B30	MP-B30
DEVICE_MODEL_MP_B30L	MP-B30L
DEVICE_MODEL_MP_B21L	MP-B21L
DEVICE_MODEL_MP_A40	MP-A40
DEVICE_MODEL_RP_F10	RP-F10/RP-G10*1
DEVICE_MODEL_SLP720RT	SLP720RT/SLP721RT

*1: Supported only Android.

② Port type (`PortType`)

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

Constant Name	Description
PORT_TYPE_BLUETOOTH	Bluetooth
PORT_TYPE_USB	USB
PORT_TYPE_TCP	TCP/IP

③ Drawer number (`DrawerNum`)

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

Constant Name	Description
DRAWER_1	Drawer 1
DRAWER_2	Drawer 2

④ Pulse width (`PulseWidth`)

Constants of enumerated type used for the pulse width are shown in the following table. As for the drawer control time, follow the specification of your drawer.

Constant Name	Description
`ON_OFF_TIME_100`	ON/OFF time 100 milliseconds
`ON_OFF_TIME_200`	ON/OFF time 200 milliseconds
`ON_OFF_TIME_300`	ON/OFF time 300 milliseconds
`ON_OFF_TIME_400`	ON/OFF time 400 milliseconds
`ON_OFF_TIME_500`	ON/OFF time 500 milliseconds
`ON_OFF_TIME_600`	ON/OFF time 600 milliseconds
`ON_OFF_TIME_700`	ON/OFF time 700 milliseconds
`ON_OFF_TIME_800`	ON/OFF time 800 milliseconds

⑤ Buzzer pattern (`BuzzerPattern`)

Constants of enumerated type used for the buzzer pattern of the external buzzer are shown in the following table.

Constant Name	Description
`BUZZER_PATTERN_1`	Pattern 1
`BUZZER_PATTERN_2`	Pattern 2
`BUZZER_PATTERN_3`	Pattern 3
`BUZZER_PATTERN_4`	Pattern 4

⑥ 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

⑦ Printer response type (`PrinterResponseId`)

Constants of enumerated type used for getting various responses from the printer are shown in the following table.

Constant Name	Description
PRINTER_RESPONSE_REQUEST	Execution response request
PRINTER_RESPONSE_USER_AREA	Send remaining capacity of user area
PRINTER_RESPONSE_ARRANGE_USER_AREA	Send remaining capacity of user area after defragment
PRINTER_RESPONSE_NV_GRAPHICS	Send NV graphics memory capacity
PRINTER_RESPONSE_KEY_CODE	Send key code list of defined NV graphics
PRINTER_RESPONSE_BATTERY_STATUS[*1]	Battery remaining capacity level
PRINTER_RESPONSE_MAINTENANCE_COUNTER	Send maintenance counter
PRINTER_RESPONSE_PRINTER_IDENTIFIER	Send printer ID

*1: Supported only by MP-B30, MP-B30L, MP-B21L, MP-B20, and MP-A40.

⑧ Device type (`DeviceType`)

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

Constant Name	Description
TYPE_PRINTER	Printer
TYPE_DISPLAY	Display

⑨ 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

⑩ 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

⑪ Bold character (`CharacterBold`)

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

Constant Name	Description
`BOLD_CANCEL`	Cancel bold character
`BOLD`	Specify bold character

⑫ Underline (`CharacterUnderline`)

Constants of enumerated type used for underline 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

⑬ Character scale (`CharacterScale`)

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

Constant Name	Description
`VARTICAL_1_HORIZONTAL_1`	Height × 1 and width × 1
`VARTICAL_1_HORIZONTAL_2`	Height × 1 and width × 2
`VARTICAL_1_HORIZONTAL_3`	Height × 1 and width × 3
`VARTICAL_1_HORIZONTAL_4`	Height × 1 and width × 4
`VARTICAL_2_HORIZONTAL_1`	Height × 2 and width × 1
`VARTICAL_2_HORIZONTAL_2`	Height × 2 and width × 2
`VARTICAL_2_HORIZONTAL_3`	Height × 2 and width × 3
`VARTICAL_2_HORIZONTAL_4`	Height × 2 and width × 4
`VARTICAL_2_HORIZONTAL_6`	Height × 2 and width × 6
`VARTICAL_3_HORIZONTAL_1`	Height × 3 and width × 1
`VARTICAL_3_HORIZONTAL_2`	Height × 3 and width × 2
`VARTICAL_3_HORIZONTAL_3`	Height × 3 and width × 3
`VARTICAL_3_HORIZONTAL_4`	Height × 3 and width × 4
`VARTICAL_4_HORIZONTAL_1`	Height × 4 and width × 1
`VARTICAL_4_HORIZONTAL_2`	Height × 4 and width × 2
`VARTICAL_4_HORIZONTAL_3`	Height × 4 and width × 3
`VARTICAL_4_HORIZONTAL_4`	Height × 4 and width × 4
`VARTICAL_4_HORIZONTAL_6`	Height × 4 and width × 6
`VARTICAL_4_HORIZONTAL_8`	Height × 4 and width × 8
`VARTICAL_6_HORIZONTAL_2`	Height × 6 and width × 2
`VARTICAL_6_HORIZONTAL_4`	Height × 6 and width × 4

Constant Name	Description
VARTICAL_6_HORIZONTAL_6	Height × 6 and width × 6
VARTICAL_6_HORIZONTAL_8	Height × 6 and width × 8
VARTICAL_8_HORIZONTAL_4	Height × 8 and width × 4
VARTICAL_8_HORIZONTAL_6	Height × 8 and width × 6
VARTICAL_8_HORIZONTAL_8	Height × 8 and width × 8

⑭ Character font (`CharacterFont`)

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

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

⑮ Module size (`ModuleSize`)

Constants of enumerated type used for width, nominal fine element width, and module size of barcode are shown in the following table.

Constant Name	Description	Method to Use
BARCODE_MODULE_WIDTH_2	Fine element 2 dots Module width 0.250 mm	● `PrintBarcode` ● `PrintPageModeBarcode`
BARCODE_MODULE_WIDTH_3	Fine element 3 dots Module width 0.375 mm	
BARCODE_MODULE_WIDTH_4	Fine element 4 dots Module width 0.500 mm	
BARCODE_MODULE_WIDTH_5	Fine element 5 dots Module width 0.625 mm	
BARCODE_MODULE_WIDTH_6	Fine element 6 dots Module width 0.750 mm	
CUSTOMERBARCODEJP_MODULE_4	Customer Bar Code_JP size 4 points*1	
CUSTOMERBARCODEJP_MODULE_5	Customer Bar Code_JP size 5 points*1	
CUSTOMERBARCODEJP_MODULE_6	Customer Bar Code_JP size 6 points*1	
CUSTOMERBARCODEJP_MODULE_7	Customer Bar Code_JP size 7 points*1	
CUSTOMERBARCODEJP_MODULE_8	Customer Bar Code_JP size 8 points*1	
CUSTOMERBARCODEJP_MODULE_9	Customer Bar Code_JP size 9 points*1	
CUSTOMERBARCODEJP_MODULE_10	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
CUSTOMERBARCODEJP_MODULE_4	1.500 (12 dots)	0.500 (4 dots)	0.500 (4 dots)	0.250 (2 dots)	0.250 (2 dots)
CUSTOMERBARCODEJP_MODULE_5	1.875 (15 dots)	0.625 (5 dots)	0.625 (5 dots)	0.375 (3 dots)	0.250 (2 dots)
CUSTOMERBARCODEJP_MODULE_6	2.250 (18 dots)	0.750 (6 dots)	0.750 (6 dots)	0.375 (3 dots)	0.375 (3 dots)
CUSTOMERBARCODEJP_MODULE_7	2.625 (21 dots)	0.875 (7 dots)	0.875 (7 dots)	0.500 (4 dots)	0.375 (3 dots)
CUSTOMERBARCODEJP_MODULE_8	3.000 (24 dots)	1.000 (8 dots)	1.000 (8 dots)	0.500 (4 dots)	0.500 (4 dots)
CUSTOMERBARCODEJP_MODULE_9	3.375 (27 dots)	1.125 (9 dots)	1.125 (9 dots)	0.625 (5 dots)	0.500 (4 dots)
CUSTOMERBARCODEJP_MODULE_10	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	● PrintPDF417 ● PrintPageModePDF417
PDF417_MODULE_WIDTH_3	Nominal fine element width 3 dots	
PDF417_MODULE_WIDTH_4	Nominal fine element width 4 dots	
PDF417_MODULE_WIDTH_5*1	Nominal fine element width 5 dots	
PDF417_MODULE_WIDTH_6*1	Nominal fine element width 6 dots	
PDF417_MODULE_WIDTH_7*1	Nominal fine element width 7 dots	
PDF417_MODULE_WIDTH_8*1	Nominal fine element width 8 dots	
QR_MODULE_SIZE_2	2 dots	● PrintQRcode ● PrintPageModeQRcode
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*1	12 dots	
QR_MODULE_SIZE_13*1	13 dots	
QR_MODULE_SIZE_14*1	14 dots	
QR_MODULE_SIZE_15*1	15 dots	
QR_MODULE_SIZE_16*1	16 dots	

Constant Name	Description	Method to Use
DATAMATRIX_MODULE_SIZE_2	2 dots	● PrintDataMatrix ● PrintPageModeDataMatrix
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*1	12 dots	
DATAMATRIX_MODULE_SIZE_13*1	13 dots	
DATAMATRIX_MODULE_SIZE_14*1	14 dots	
DATAMATRIX_MODULE_SIZE_15*1	15 dots	
DATAMATRIX_MODULE_SIZE_16*1	16 dots	
GS1DATABAR_MODULE_SIZE_2*1	2 dots	● PrintGS1DataBarStacked ● PrintGS1DataBarStackedOmnidirectional ● PrintGS1DataBarExpandedStacked ● PrintPageModeGS1DataBarStacked ● PrintPageModeGS1DataBarStackedOmnidirectional ● PrintPageModeGS1DataBarExpandedStacked
GS1DATABAR_MODULE_SIZE_3*1	3 dots	
GS1DATABAR_MODULE_SIZE_4*1	4 dots	
GS1DATABAR_MODULE_SIZE_5*1	5 dots	
GS1DATABAR_MODULE_SIZE_6*1	6 dots	
GS1DATABAR_MODULE_SIZE_7*1	7 dots	
GS1DATABAR_MODULE_SIZE_8*1	8 dots	
GS1DATABAR_MODULE_SIZE_9*1	9 dots	
GS1DATABAR_MODULE_SIZE_10*1	10 dots	
GS1DATABAR_MODULE_SIZE_11*1	11 dots	
GS1DATABAR_MODULE_SIZE_12*1	12 dots	
GS1DATABAR_MODULE_SIZE_13*1	13 dots	
GS1DATABAR_MODULE_SIZE_14*1	14 dots	
GS1DATABAR_MODULE_SIZE_15*1	15 dots	
GS1DATABAR_MODULE_SIZE_16*1	16 dots	

Constant Name	Description	Method to Use
AZTECCODE_MODULE_SIZE_2[*2]	2 dots	● PrintAzteccode ● PrintPageModeAzteccode
AZTECCODE_MODULE_SIZE_3[*2]	3 dots	
AZTECCODE_MODULE_SIZE_4[*2]	4 dots	
AZTECCODE_MODULE_SIZE_5[*2]	5 dots	
AZTECCODE_MODULE_SIZE_6[*2]	6 dots	
AZTECCODE_MODULE_SIZE_7[*2]	7 dots	
AZTECCODE_MODULE_SIZE_8[*2]	8 dots	
AZTECCODE_MODULE_SIZE_9[*2]	9 dots	
AZTECCODE_MODULE_SIZE_10[*2]	10 dots	
AZTECCODE_MODULE_SIZE_11[*2]	11 dots	
AZTECCODE_MODULE_SIZE_12[*2]	12 dots	
AZTECCODE_MODULE_SIZE_13[*2]	13 dots	
AZTECCODE_MODULE_SIZE_14[*2]	14 dots	
AZTECCODE_MODULE_SIZE_15[*2]	15 dots	
AZTECCODE_MODULE_SIZE_16[*2]	16 dots	

*1: Not supported by RP-E10 nor RP-D10.
*2: Supported only by MP-B30L and MP-B21L.

⑯ 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	● PrintPDF417 ● PrintPageModePDF417
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	● PrintQRcode ● PrintPageModeQRcode
QR_ERROR_CORRECTION_M	Error correction level M	
QR_ERROR_CORRECTION_H	Error correction level H	
QR_ERROR_CORRECTION_Q	Error correction level Q	

⑰ 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

⑱ 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

⑲ Barcode symbol (`BarcodeSymbol`)

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

Constant Name	Description	Syntax*1
BARCODE_UPC_A	UPC-A	(a)
BARCODE_UPC_E	UPC-E	(a)
BARCODE_EAN13	EAN13	(a)
BARCODE_JAN13	JAN13	(a)
BARCODE_EAN8	EAN8	(a)
BARCODE_JAN8	JAN8	(a)
BARCODE_CODE39	CODE39	(a), (b)
BARCODE_CODE93	CODE93	(c)
BARCODE_CODE128	CODE128	(c)
BARCODE_ITF	ITF	(a), (b)
BARCODE_CODABAR	CODABAR	(a), (b)
BARCODE_EAN13_ADDON	EAN13 add-on	(a)
BARCODE_JAN13_ADDON	JAN13 add-on	(a)
BARCODE_CUSTOMER_BAR_CODE_JP*2	Customer Bar Code_JP	(d)
BARCODE_GS1_OMNI_DIRECTIONAL*3	GS1 Databar Omni-directional	(a)
BARCODE_GS1_TRUNCATED*3	GS1 Databar Truncated	(a)
BARCODE_GS1_LIMITED*3	GS1 Databar Limited	(a)
BARCODE_GS1_EXPANDED*3	GS1 Databar Expanded	(a)

*1: See `PrintBarcode` or `PrintPageModeBarcode` for details of syntax.
*2: Supports only MP-B21L.
*3: Not supported by RP-E10 nor RP-D10.

⑳ 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)

㉑ 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

㉒ PDF417 symbol (`Pdf417Symbol`)

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

Constant Name	Description
PDF417_STANDARD	PDF417
PDF417_COMPACT	Compact PDF417

㉓ 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

㉔ 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

㉕ MaxiCode Mode (`MaxiCodeMode`)

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

Constant Name	Description
MAXI_CODE_2	Mode2
MAXI_CODE_3	Mode3
MAXI_CODE_4	Mode4
MAXI_CODE_5	Mode5

㉖ Aztec symbol (`AztecSymbol`)

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

Constant Name	Description
AZTECCODE_FULLRANGE	Full-range mode
AZTECCODE_COMPACT	Compact mode

㉗ Cutting method (`CuttingMethod`)

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

- SLP720RT, SLP721RT, RP-F10, RP-G10, RP-E10

Constant Name	Description	
	Paper Feed to Cut Position	Cutting Method
CUT_FULL	Enabled	Full cut
CUT_FULL_NO_FEED	Disabled	
CUT_PARTIAL	Enabled	Partial cut
CUT_PARTIAL_NO_FEED	Disabled	
CUT_NONE[*1]	Disabled	No cut

*1: Supported only by `PrintPageMode`.

- MP-B30, MP-B30L, MP-B21L, MP-B20, MP-A40

Constant Name	Description
CUT_FULL	No cut Paper feed operation to the paper cut position
CUT_PARTIAL	
CUT_NONE[*1]	No cut

*1: Supported only by `PrintPageMode`.

㉘ Form feed position (FeedPosition)

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

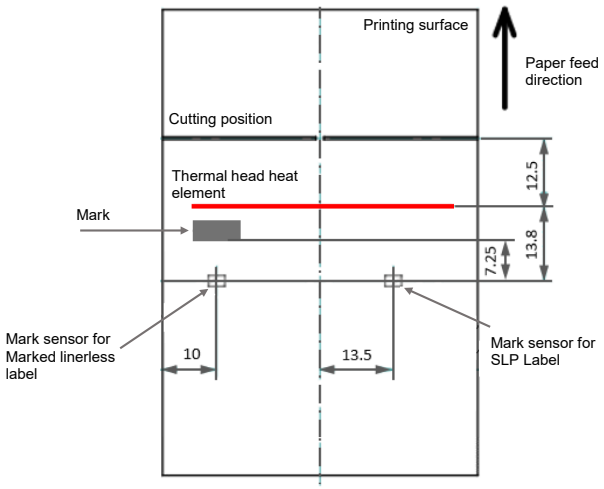
Constant Name	Description
FEED_CUTTER[*1]	After detecting the mark or gap, feeds the paper to the cutting position. The paper feed length is the length of the memory switches[*2*3*4] of the printer.
FEED_NEXT_TOF[*5]	After detecting the next mark or next gap, feeds the paper to the print position. The paper feed length is the length of the memory switches[*6] of the printer.

*1: Supported only by SLP720RT, SLP721RT, MP-B30L, MP-B21L, and MP-A40.
*2: For SLP720RT and SLP721RT, the paper feed length is the length of the memory switches MS 8 to 9 (Mark Position Correction) of the printer.
The default of the paper feed is 58 dots (7.25 mm).
*3: For MP-B30L and MP-B21L, 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 MP-B30L: 125 dots (15.6 mm) and MP-B21L: 122 dots (15.2 mm).
*4: For MP-A40, the paper feed length is the length of the memory switches MS 21 to 22 (Mark Position Correction) of the printer.
*5: Supported only by MP-B30L.
*6: For MP-B30L and MP-B21L, 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 MP-B30L: 125 dots (15.6 mm) and MP-B21L: 122 dots (15.2 mm).

Reference

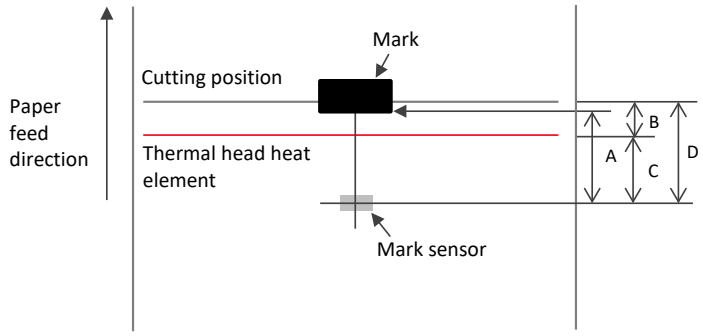
See "USER'S GUIDE" for the details of the memory switch of the printer.
The memory switch of the printer can be changed in the iOS app "SII Printer Utility" on the App Store and the Android app "SII Printer Utility" on the Google Play.

In SLP720RT and SLP721RT, the relation between the sensor position and the defaults of memory switches MS 8 to 9 (Mark Correction) of the printer is shown in the following figure.



Unit: mm

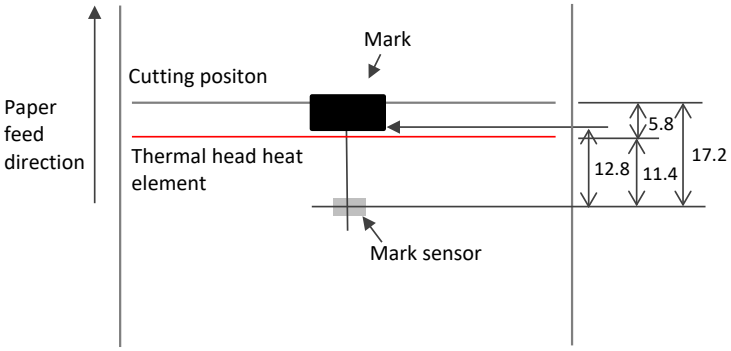
In MP-B30L and MP-B21L, the relation between the sensor position and the defaults of memory
switches MS 21 to 22 (Mark Detection Cut Position Correction) of the printer and memory switches MS 26 to 27 (Mark Detection Print Position Correction) of the printer are shown in the following figure.
When the memory switch of the printer is set to the default, the cutting position of FEED_CUTTER and the next print position of FEED_NEXT_TOF is the same paper feed length.



Unit: mm

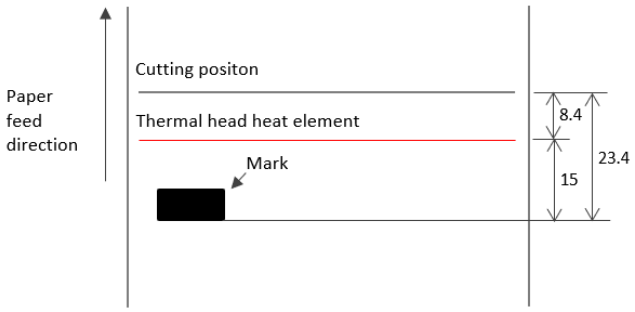
MP-B30L: A = 15.6, B = 5.8, C = 11.4, D = 17.2
MP-B21L: A = 15.2, B = 5.4, C = 11.3, D = 16.7

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 using MP-B30L as an example.



Unit: mm

In MP-A40, the relation between the mark position and the initial values of the printer memory switches MS21–22 (Mark Position Correction) is shown in the following figure.



Unit: mm

Note When using label in MP-B30L and MP-B21L, 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.

㉙ 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

㉚ Image scaling (`ImageScale`)

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

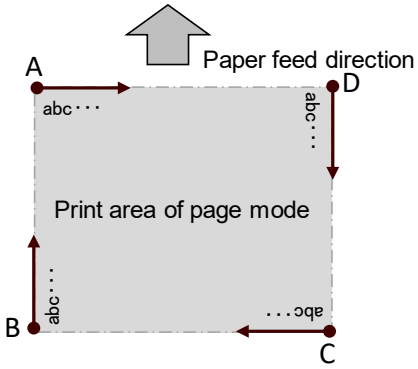
Constant Name	Description
`IMAGE_SCALE_NONE`	No scaling
`IMAGE_SCALE_WIDTH_FIT`	Scale to fit print width

㉛ 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

Constant Name	Description
DIRECTION_TOP_TO_BOTTOM	Starting point: Upper right (D on the figure), Print direction: Upper to Below



㉜ Line style (`LineStyle`)

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

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

㉝ International character set (`InternationalCharacter`)

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

Constant Name	Description
COUNTRY_USA	USA
COUNTRY_FRANCE	France
COUNTRY_GERMANY	Germany
COUNTRY_ENGLAND	England
COUNTRY_DENMARK_1	Denmark I
COUNTRY_SWEDEN	Sweden
COUNTRY_ITALY	Italy
COUNTRY_SPAIN	Spain I
COUNTRY_JAPAN	Japan
COUNTRY_NORWAY	Norway
COUNTRY_DENMARK_2	Denmark II
COUNTRY_SPAIN_2	Spain II
COUNTRY_LATIN_AMERICA	Latin America
COUNTRY_ARABIA	Arabia

㉞ Codepage (CodePage)

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

Constant Name	Description
CODE_PAGE_437	USA, Standard Europe (Code Page437)
CODE_PAGE_KATAKANA	Katakana
CODE_PAGE_850	Multilingual (Code Page850)
CODE_PAGE_860	Portuguese (Code Page860)
CODE_PAGE_863	Canadian-French (Code page863)
CODE_PAGE_865	Nordic (Code Page865)
CODE_PAGE_857[*1*2]	Turkish (Code Page857)
CODE_PAGE_737[*2]	Greek (Code Page737)
CODE_PAGE_1252	Latin (Code Page1252)
CODE_PAGE_866[*2]	Russian (Code Page866)
CODE_PAGE_852	Eastern Europe (Code Page852)
CODE_PAGE_858	Euro (Code Page858)
CODE_PAGE_855[*2]	Cyrillic (Code Page855)
CODE_PAGE_864[*1*3]	Arabic (Code Page864)
CODE_PAGE_1250	Central European (Code Page1250)
CODE_PAGE_1251	Cyrillic (Code Page1251)
CODE_PAGE_1253[*4]	Greek (Code Page1253)
CODE_PAGE_1254	Turkish (Code Page1254)

*1: 20ACh of the Unicode cannot be printed or displayed.
*2: Not supported by RP-E10 nor RP-D10.
*3: Font B cannot be printed or displayed.
*4: 00AAh of the Unicode cannot be printed or displayed.

㉟ Paper selection with or without mark when printing label file (PrintSmartLabelMode)

Constants of enumerated type used for selecting the paper when printing label file are shown in the following table.

Constant Name	Description
PRINTSMARTLABEL_MODE_MARK	Marked paper
PRINTSMARTLABEL_MODE_NONEMARK	Paper without mark

4.4.2 `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 "4.4.2(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 "4.4.2(1)② Dedicated method for standard mode" for the methods.
Dedicated method for page mode	The valid methods in page mode. See "4.4.2(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 "4.4.2(4)① Common method to standard mode and page mode" for details of the common methods.

✓: Supported, - : Not supported

Name	Description	SLP720RT SLP721RT	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B30L MP-B21L MP-B20 MP-A40
`PrinterManager`	Constructor	✓	✓	✓	✓
`Connect`	Start communicating with device	✓	✓	✓	✓
`Disconnect`	Stop communicating with device	✓	✓	✓	✓
`OpenDrawer`	Open cash drawer	✓*1	✓	✓	–
`Buzzer`	Sound buzzer	–	–	✓	–
`ExternalBuzzer`	Sound external buzzer	✓*1	✓	–	–
`GetStatus`	Get printer status	✓	✓	✓	✓
`Abort`	Abort waiting state of printer	✓	✓	✓	✓
`RegisterLogo`	Register logo	✓	✓	✓	✓
`UnregisterLogo`	Delete registered logo	✓	✓	✓	✓
`RegisterStyleSheet`	Register style sheet	–	–	✓	✓*2
`UnregisterStyleSheet`	Delete registered style sheet	–	–	✓	✓*2
`ResetPrinter`	Reset printer	✓	✓	✓	✓
`GetPrinterResponse`	Get various responses from printer	✓	✓	✓	✓
`StartDiscoveryDevice`	Start device search	✓	✓	✓	✓
`CancelDiscoveryDevice`	Cancel device search	✓	✓	✓	✓
`GetFoundDevice`	Get found device information list	✓	✓	✓	✓
`ControlTransaction`	Start/Finish batch processing	✓	✓	✓	✓

Name	Description	SLP720RT SLP721RT	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B30L MP-B21L MP-B20 MP-A40
SetStatusChangedEventHandler	Start/Finish call back of printer status change	✓	✓	✓	✓
SetBarcodeScannerReadDataEventHandler	Start/Finish call back of receiving barcode data	–	✓*3	–	–
SetBarcodeScannerChangedOnlineEventHandler	Start/Finish call back of barcode scanner connection	–	✓*3	–	–
SetBarcodeScannerChangedOfflineEventHandler	Start/Finish call back of barcode scanner disconnection	–	✓*3	–	–

*1: Supported only by SLP721RT.
*2: Supported only by MP-A40.
*3: Supported only by RP-F10.

② Dedicated method for standard mode

Methods provided by the dedicated method for standard mode are shown in the following table.
See "4.4.2(4)② Dedicated method for standard mode" for details of the specified methods.

✓: Supported, - : Not supported

Name	Description	SLP720RT SLP721RT	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B30L MP-B21L MP-B20 MP-A40
SendText	Send text data	✓	✓	✓	✓
SendTextEx	Send format specified text data	✓	✓	✓	✓
PrintBarcode	Print barcode	✓	✓	✓	✓
PrintPDF417	Print PDF417	✓	✓	✓	✓
PrintQRcode	Print QR Code	✓	✓	✓	✓
PrintDataMatrix	Print Data Matrix	✓	✓	✓	✓
PrintMaxiCode	Print MaxiCode	✓	✓	✓	✓
PrintGS1DataBarStacked	Print GS1 Databar Stacked	✓	✓	–	✓
PrintGS1DataBarStackedOmnidirectional	Print GS1 Databar Stacked Omni-directional	✓	✓	–	✓
PrintGS1DataBarExpandedStacked	Print GS1 Databar Expanded Stacked	✓	✓	–	✓
PrintAztecCode	Print Aztec Code	–	–	–	✓*1
CutPaper	Cut paper	✓	✓	✓	✓*2
Feed Position	Paper form feed	✓	–	–	✓*3
SendBinary	Send binary data	✓	✓	✓	✓
SendDataFile	Send specified file	✓	✓	✓	✓
SendImage	Send image	✓	✓	✓	✓
PrintPDF	Print PDF page	✓	✓	✓	✓
PrintLogo	Print logo	✓	✓	✓	✓
PrintSmartLabelImageData	Print label	✓	–	–	–

*1: Supported only by MP-B30L and MP-B21L.

*2: Only the paper feed operation to the paper cut position is performed.
*3: Supported only by MP-B30L, MP-B21L and MP-A40.

③ Dedicated method for page mode

Methods provided by the dedicated method for page mode are shown in the following table.
See "4.4.2(4)③ Dedicated method for page mode" for details of the specified methods.

✓: Supported, - : Not supported

Name	Description	SLP720RT SLP721RT	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B30L MP-B21L MP-A40	MP-B20
EnterPageMode	Start page mode	-	✓	✓	✓	-
ExitPageMode	End page mode	-	✓	✓	✓	-
SetPageModeArea	Specify print area of page mode	-	✓	✓	✓	-
SetPageModeDirection	Specify print direction of page mode	-	✓	✓	✓	-
SetPageModeLineSpacing	Specify line spacing of page mode	-	✓	✓	✓	-
PrintPageMode	Print page mode	-	✓	✓	✓	-
PrintPageModeText	Send text data of page mode	-	✓	✓	✓	-
PrintPageModeTextEx	Send format specified text data of page mode	-	✓	✓	✓	-
PrintPageModeBarcode	Print barcode of page mode	-	✓	✓	✓	-
PrintPageModePDF417	Print PDF417 of page mode	-	✓	✓	✓	-
PrintPageModeQRcode	Print QR Code of page mode	-	✓	✓	✓	-
PrintPageModeDataMatrix	Print Data Matrix of page mode	-	✓	✓	✓	-
PrintPageModeMaxiCode	Print MaxiCode of page mode	-	✓	✓	✓	-
PrintPageModeGS1DataBar Stacked	Print GS1 Databar Stacked of page mode	-	✓	✓	✓	-
PrintPageModeGS1DataBar StackedOmnidirectional	Print GS1 Databar Stacked Omni-directional of page mode	-	✓	✓	✓	-
PrintPageModeGS1DataBar ExpandedStacked	Print GS1 Databar Expanded Stacked of page mode	-	✓	✓	✓	-
PrintPageModeAztecCode	Print Aztec Code of page mode	-	–	–	✓*1	-
SendPageModeBinary	Send binary data of page mode	-	✓	✓	✓	-
PrintPageModeImageFile	Draw image file of page mode	-	✓	✓	✓	-
PrintPageModeRectangle	Draw rectangle image of page mode	-	✓	✓	✓	-
PrintPageModeLine	Print ruled line of page mode	-	✓	✓	✓	-
PrintPageModeLogo	Print logo of page mode	-	✓	✓	✓	-

*1: Supported only by MP-B30L and MP-B21L.

(2) Common property list to standard mode and page mode

Properties provided by `PrinterManager` class are shown in the following table.

✓: Supported, - : Not supported

Name	Access	Description	SLP720RT SLP721RT	RP-F10 RP-G10	RP-E10 RP-D10	MP-B30 MP-B30L MP-B21L MP-B20 MP-A40
`SendTimeout`	R/W	Get/Set send timeout period	✓	✓	✓	✓
`ReceiveTimeout`	R/W	Get/Set receive timeout period	✓	✓	✓	✓
`InternationalCharacter`	R/W	Get/Set international character set	✓	✓	✓	✓
`CodePage`	R/W	Get/Set codepage	✓	✓	✓	✓
`DeviceModel`	R	Get device model	✓	✓	✓	✓
`PortType`	R	Get connecting port type	✓	✓	✓	✓
`IsConnect`	R	Verify connection state with printer	✓	✓	✓	✓
`SocketKeepingTime`	R/W	Get/Set socket keeping time	✓	✓	✓	✓*1
`Version`	R	Get SDK version	✓	✓	✓	✓
`PrintSmartLabelMode`	R/W	Get/Set paper when printing label file	✓	–	–	✓*2

*1: Not supported by MP-B20.
*2: Supported only by MP-B30L and MP-B21L.

(3) Constant List

① Barcode and PDF417

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

Constant Name	Description	Value
BARCODE_HEIGHT_DEFAULT	Default value of barcode height	162
PDF417_MODULE_HEIGHT_DEFAULT	Default value 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

② Battery remaining capacity level

Constants used for battery remaining capacity level 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
BATTERY_STATUS_AC_POWERED	Powered by AC	4

(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 the `SII.SDK.Maui.Printer.PrinterManager` class.

Target	All platforms	All devices
Syntax	`public PrinterManager(object context);`	
Parameter	`context`	Context For iOS: Specify null. For Android: Specify application context to call this method.

`Connect` Start communicating with device

Starts communicating with the device.

Target	All platforms	All devices
Syntax	(a) `public void Connect(DeviceModel deviceModel, DeviceInfo deviceInfo);`	
	(b) `public void Connect(DeviceModel deviceModel, PortType portType, string param);`	
Parameter	`deviceModel`	Device model See "4.4.1① Device model (`DeviceModel`)" for available constants.
	`deviceInfo`	Device information Specify the object of the `DeviceInfo` class. The object of `DeviceInfo` class can be retrieved from `StartDiscoveryDevice` and `GetFoundDevice`.
	`portType`	Port type See "4.4.1② Port type (`PortType`)" for available constants.
	`param`	Depends on the setting of the `portType`. See the "Target devices" described in "INTRODUCTION" for the corresponding interface.

・portType is **PORT_TYPE_BLUETOOTH**
Specify the Bluetooth address or Bluetooth device name (Bluetooth Accessary).
Example: "00:11:22:AA:BB:CC", "RP-F10"
・portType is **PORT_TYPE_USB**
For iOS:
Specify the printer name.
Example: "RP-F10"
For Android:
Specify null.
・portType is **PORT_TYPE_TCP**
Specify the IP address of the printer.
Example: "192.168.0.190", "192.168.0.1"

Error

`PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Description

Call this method before using other `PrinterManager` class methods.
In order to make this library work properly, this method may change the printer settings when connecting.

For Bluetooth connection:
In iOS, communication with a printer paired with iOS device starts through Bluetooth connection.
Connect to the paired Bluetooth address or Bluetooth device (Bluetooth Accessory) specified by `param`.
In Android, connect the printer specified by `deviceModel` to the Bluetooth address specified by `param`.

For USB connection:
In iOS, communication with a printer connected with the device through the USB cable starts through the USB connection.
In Android, connect the printer specified by `deviceModel`.

For TCP/IP connection:
Communication with a printer connected to the same network as the device starts through TCP/IP connection. Connection is made to the IP address specified by `param`.
TCP port 9100 and 26100 are used for communication.

- **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 the socket keeping time set by `SocketKeepingTime`.
  Then the new socket is created immediately and used for the next connection.
  If the printer is receiving a connection request from another host on the same network at the time of discarding the socket, the printer establishes communication with that host, so the reconnection may fail.

Note

This method does not support a concurrent connection from multiple apps to one printer.

Disconnect — Stop communicating with device

Finishes communicating with the device.

Target	All platforms	All devices

Syntax
```
public void Disconnect();
```

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Note It is recommended to get execution response by **`PRINTER_RESPONSE_REQUEST`** of `GetPrinterResponse` before executing this method. If not, the following problems may occur.

- The communication is disconnected before the print data sending from device to the printer is completed, and a part of the data may be lost.
- In Bluetooth connection, when either `Disconnect` or `Connect` is executed while the printer is in the buffer full state[*1], the communication between the devise and the printer may be disconnected.

  *1: The state of buffer full means that the buffer of the printer is filled with print data. The size to be in buffer full state is approximately 4KB.

If you do not execute `GetPrinterResponse` in your program, please fully evaluate your program to confirm no problems arise.

OpenDrawer — Open cash drawer

Opens the specified cash drawer.

Target	All platforms	SLP721RT, RP-F10, RP-G10, RP-E10, RP-D10

Syntax
```
public void OpenDrawer(DrawerNum drawerNum, PulseWidth onOffTime);
```

Parameter

`drawerNum`	Drawer number See "4.4.1③ Drawer number (`DrawerNum`)" for available constants.
`onOffTime`	Pulse width See "4.4.1④ Pulse width (`PulseWidth`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Buzzer — Sound buzzer

Sounds the buzzer.

Target	All platforms	RP-E10, RP-D10

Syntax
```
public void Buzzer(int onTime, int offTime);
```

Parameter

`onTime`	Buzzer On time (millisecond) Set the time the buzzer on. The valid range is 0 to 510.

	`offTime`	Buzzer Off time (millisecond) Set the time the buzzer off. The valid range is 0 to 510.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

`ExternalBuzzer` Sound external buzzer

Sounds the external buzzer.

Target All platforms SLP721RT, RP-F10, RP-G10

Syntax
```
public void ExternalBuzzer(BuzzerPattern buzzerPattern,
                           int buzzerCount);
```

Parameter

`buzzerPattern`	Buzzer pattern See "4.4.1⑤ Buzzer pattern (`BuzzerPattern`)" for available constants. The external buzzer sound stops under one of the following conditions: ・Sounding for the number of times set by `buzzerCount` ・Opening the cover ・Executing the printer command "Stop External Buzzer"
`buzzerCount`	Buzzer sound count (times) The external buzzer sounds for the number of times set by `buzzerCount`. The valid range is 1 to 255.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

`GetStatus` Get printer status

Gets the latest printer status.

Target All platforms All printers

Syntax
```
public void GetStatus(out int[] buf);
```

Parameter `buf` Status retrieved from the printer

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Description The status retrieved from the printer is stored in an int type array.

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[*1]	No error	Error
1	Hardware error[*1]	No error	Error
2	Head temperature error[*1]	No error	Error
3	Autocutter error[*2*3*4]	No error[*12]	Error

Bit	Function	Value	
		0	1
4	Out-of-paper error[*1]	No error	Error
5	Near end sensor error[*3]	No error[*12]	Error
6	Jam error at mark detection[*3*5*8*9*11] Jam error at mark and gap detection[*6*10]	No error[*12]	Error
7	Cover open error[*2*3*4*5*6*8*9*10*11]	No error[*12]	Error
8	FEED Switch status[*1]	OFF	ON
9	Reserved	Fixed	-
10	Paper feed status[*1]	Stop	Operating
11	Return-waiting status[*1]	Not waiting	Waiting
12	Reserved	Fixed	-
13	Taken sensor status[*8*9]	Paper removed	Paper removal waiting
13	Reserved	-	Fixed
14	Reserved	-	Fixed
15	Drawer switch input status[*2*3*4*9]	Low[*13]	High[*14]
16	FLASH memory rewriting[*1]	Not rewriting	Rewriting
17	Peripherals selection[*2*3*4]	Printer	Others
17	Reserved	Fixed[*13]	Fixed[*14]
18	Reserved	Fixed[*15]	Fixed[*16]
19	Reserved	-	Fixed
20 to 22[*14]	Battery remaining capacity level[*5*6*7*10]	000: No battery 001: Low (Battery remaining capacity: approx. 10%) 011: Middle (Battery remaining capacity: approx. 40%) 111: Full (Battery remaining capacity: approx. 80%)	
20 to 22[*14]	Battery remaining capacity[*11]	000: No battery 001: Level 4, 5 (Need to charge) 010: Level 3 (Battery remaining capacity: approx. 10%) 011: Level 2 (Battery remaining capacity: approx. 40%) 100: Level 1 (Battery remaining capacity: approx. 80%)	
23	Battery error[*5*6*7*10*11]	No error	Error[*14]
24 to 31	Reserved	-	Fixed

*1: Supported by all printers.
*2: Supported by RP-F10 and RP-G10.
*3: Supported by RP-E10.
*4: Supported by RP-D10.
*5: Supported by MP-B30.
*6: Supported by MP-B30L.
*7: Supported by MP-B20.
*8: Supported by SLP720RT.
*9: Supported by SLP721RT.
*10: Supported by MP-B21L.
*11: Supported by MP-A40.
*12: The [Value] of the printer that does not support [Function] is fixed to 0.
*13: The [Value] of SLP720RT is fixed to 0.

*14: The [Value] of the printer that does not support [Function] is fixed to 1.
*15: The [Value] of SLP720RT, SLP721RT, RP-F10, and RP-G10 is fixed to 0.
*16: The [Value] of RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-B20, and MP-A40 is fixed to 1.

Abort — Abort waiting state of printer

Aborts the waiting state of the printer.

Target: All platforms, All printers

Syntax:
```
public void Abort();
```

Error: `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Description: When sending of image file by `SendDataFile` is interrupted, the printer does not accept other processes until the specified image file is received completely. (Methods and transmission data are misinterpreted and recognized as a part of the image file.) To solve this situation, use this method to abort the waiting state of the printer.
Note that when this method is executed, a part of unprinted image file may be printed.

RegisterLogo — Register logo

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

Target: All platforms, All printers

Syntax:
```
public void RegisterLogo(string fileName,
                         string logoId,
                         Dithering dithering);
```

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

For iOS:
File path strings to use

For Android:

- 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 "3.4 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.

Supported image file file extensions are .bmp, .jpg, .jpeg, and .png. Colored image is converted to monochrome image by binarization and registered.

	`logoId`	ID of the logo to be registered (key code) Specify the ID of the logo to be registered as a two-character string. Valid characters are ASCII character codes from 20h (space) to 7Eh (tilde) such as alphanumeric ('0' to '9', 'A' to 'Z', 'a' to 'z').
	`dithering`	Dithering See "4.4.1⑥ Dithering (`Dithering`)" for available constants.
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.	

`UnregisterLogo` Delete registered logo

Deletes the registered logo.

Target	All platforms	All printers
Syntax	`public void UnregisterLogo(string logoId);`	
Parameter	`logoId`	ID of the logo to be deleted (key code) Specify the ID of the registered logo as a character string.
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.	

`RegisterStyleSheet` Register style sheet

Registers CSS file to the printer as a style sheet.

Target	All platforms	RP-E10, RP-D10, MP-A40
Syntax	`public void RegisterStyleSheet(string fileName, int cssId);`	
Parameter	`filename`	File path of CSS file to be registered as a style sheet The formats that can be entered are described below. For iOS: File path strings to use For Android: • 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 "3.4 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. Supported image file file extension is .css.

	`cssId`	ID of the style sheet to be registered Set the ID of the style sheet to be registered. The valid range is 1 to 4.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Description Write the style sheets in CSS (cascading style sheets) language.
The number of the registerable style sheet is a maximum of 4.
The number of the registerable style in a single CSS file is a maximum of 64.

`UnregisterStyleSheet` Delete registered style sheet

Deletes the registered style sheet.

Target All platforms RP-E10, RP-D10, MP-A40

Syntax `public void UnregisterStyleSheet(int cssId);`

Parameter	`cssId`	ID of the style sheet to be deleted Specify the ID of the registered style sheet

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

`ResetPrinter` Reset printer

Performs a hardware reset of the printer.

Target All platforms All printers

Syntax `public void ResetPrinter();`

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Description For Bluetooth connection:
The printer hardware reset is performed by the printer command "Printer Reset".
For iOS (RP-F10, RP-E10 and RP-D10), enable the printer memory switch "iOS Auto Connection" before executing this method.
For iOS (MP-B30, MP-B30L, MP-B21L, MP-B20, and MP-A40), enable the iOS Auto Connection by the printer command "Bluetooth communication setting" before executing this method.
When it is disabled, this method fails to reconnect after reset and `PrinterException` is thrown.
This method takes about 10 seconds to complete reconnection with the printer after performing the reset. Use this method after setting a sufficient receive timeout period.

For USB connection:
For iOS, the printer hardware reset is performed by the printer command "Printer Reset".
For Android, the printer reset is performed by the SOFT_RESET function in the 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.

GetPrinterResponse — Get various responses from printer

Gets response data from the printer.

Target: All platforms, All printers

Syntax:

```
public void GetPrinterResponse<T>(PrinterResponseId responseId,
                                  object param,
                                  out T[] response);
```

Parameter:

- `responseId` — Response type constant
  See "4.4.1⑦ Printer response type (`PrinterResponseId`)" for available constants.
- `param` — Command parameter
  The value to be specified varies with the response type constant.
  See the following table for description of the value to be specified.
- `response` — Buffer for storing the retrieved response data
  The buffer type varies with the response type constant.
  See the following table for the buffer type.

Response Type Constant

Parameter	Description
PRINTER_RESPONSE_REQUEST (Execution response request)	
`param`	Specify 0 to 15 (00h to 0Fh) in an int type.
`response`	Specify an int type array. When the response is retrieved successfully, the response code of the execution response request is stored with 128 to 143 (80h to 8Fh).
PRINTER_RESPONSE_USER_AREA (Send remaining capacity of user area)	
`param`	Specify null.
`response`	Specify an int type array. When the response is retrieved successfully, the remaining capacity (unit: byte) of the user area is stored numerically.
PRINTER_RESPONSE_ARRANGE_USER_AREA (Send remaining capacity of user area after defragment)	
`param`	Specify null.
`response`	Specify an int type array. When the response is retrieved successfully, the remaining capacity (unit: byte) of the user area after defragment is stored numerically.
PRINTER_RESPONSE_NV_GRAPHICS (Send NV graphics memory capacity)	
`param`	Specify null.
`response`	Specify an int type array. When the response is retrieved successfully, the NV graphics memory capacity (unit: byte) is stored numerically.
PRINTER_RESPONSE_KEY_CODE (Send key code list of defined NV graphics)	
`param`	Specify null.
`response`	Specify a string type array. When the response is retrieved successfully, the key code of NV graphics is stored in an array of strings.

Response Type Constant		
	Parameter	**Description**
PRINTER_RESPONSE_BATTERY_STATUS (Battery remaining capacity level)		
	param	Specify null.
	response	Supported only by MP-B30, MP-B30L, MP-B21L, MP-B20, and MP-A40. Specify an int type array. When the response is retrieved successfully, the battery remaining capacity level is stored numerically. See "4.4.2(3)② Battery remaining capacity level" for details of the value. Battery remaining capacity level **BATTERY_STATUS_FULL** : Full (Battery remaining capacity: approx. 80%) **BATTERY_STATUS_MIDDLE** : Middle (Battery remaining capacity: approx. 40%) **BATTERY_STATUS_LOW** : Low (Battery remaining capacity: approx. 10%) **BATTERY_STATUS_EMPTY** : No battery **BATTERY_STATUS_AC_POWERED** : Powered by AC*1
PRINTER_RESPONSE_MAINTENANCE_COUNTER (Send maintenance counter)		
	param	Specify null.
	response	When the response is retrieved successfully, the maintenance counter is stored as shown below. buf[0] : Number of paper feed lines (unit: 100 dot-line) buf[1] : Number of thermal head activation times (unit: 100 dot-line) buf[2] : Number of autocutter drive times*1 buf[3] : Drive time of printer (unit: minute) buf[4] : Number of paper feed lines (integrated value) (unit: 100 dot-line) buf[5] : Number of thermal head activation times (integrated value) (unit: 100 dot-line) buf[6] : Number of autocutter drive times (integrated value)*1 buf[7] : Drive time of printer (integrated value) (unit: minute)
PRINTER_RESPONSE_PRINTER_IDENTIFIER (Send printer ID)		
	param	Specify null.
	response	When the response is retrieved successfully, a PrinterIdentifier class object is stored in the first element of buf. Example: buf.size() = 1, buf.get[0] = "PrinterIdentifier class object" See "4.4.4 **PrinterIdentifier** Class" for the PrinterIdentifier class object.

*1: Supported by MP-A40.
*2: Not supported by MP-B30, MP-B30L, MP-B21L, MP-B20, and MP-A40. Always returns 0.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

`StartDiscoveryDevice` Start device search

Searches the device.

Target All platforms All devices

Syntax
```
public void StartDiscoveryDevice(DeviceType deviceType,
                                  PortType portType,
                                  int timeout,
                                  Action completion);
```

Parameter	`deviceType`	Device Type See "4.4.1⑧ Device type (`DeviceType`)" for available constants. For iOS, only **`TYPE_PRINTER`** is available.
	`portType`	Port type See "4.4.1② Port type (`PortType`)" for available constants.
	`timeout`	Search timeout period (millisecond: ms) Sets the timeout period per search. 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. ● For **`PORT_TYPE_TCP`** Each time the local broadcast packet is sent, this method waits for a response from the printer until the period specified by `timeout` elapses. ● For **`PORT_TYPE_BLUETOOTH`** When the device is being searched, this method searches the device until the period specified by `timeout` elapses. When the device search is completed, the search is completed without waiting for the period of `timeout`. For iOS, specifying of `timeout` is ignored. ● For **`PORT_TYPE_USB`** Specifying of `timeout` is ignored.
	`completion`	The block executed when completing the device search
Error	`PrinterException` is thrown when an error occurs while this method is being called.	
Description	This method searches for the device. The device information of the discovered device can be retrieved by `GetFoundDevice`.	

`CancelDiscoveryDevice` Cancel device search

Cancels `StartDiscoveryDevice` under execution.

Target	All platforms	All printers
Syntax	`public void CancelDiscoveryDevice();`	
Description	The cancellation of the search is notified as an event to the block set to `completion` of `StartDiscoveryPrinter`.	

`GetFoundDevice` Get found device information list

Gets the device information list found by `StardDiscoveryDevice` from the destination `DeviceInfo` class.

Target	All platforms	All printers
Syntax	`public DeviceInfo[] GetFoundDevice();`	
Description	For the Bluetooth connection on iOS, only the device information of the connected printer can be retrieved.	
Return value	See "**4.4.3 `DeviceInfo` Class**".	

ControlTransaction — Start/Finish batch processing

Starts or finishes batch processing.

Target: All platforms, All printers

Syntax:
```
public void ControlTransaction(TransactionFunction control);
```

Parameter:

control	Batch processing selection	See "4.4.1⑨ Batch processing selection (TransactionFunction)" for available constants.

Error: PrinterException is thrown when an error occurs while this method is being called. See "**4.4.5 PrinterException Class**" for details on the error.

Description: The procedure of batch processing is as follows:

(1) Starts batch processing.
Specify **TRANSACTION_START**.

(2) Executes 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) Finishes 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
・OpenDrawer
・Buzzer
・ExternalBuzzer
・SendBinary
・SendDataFile
・SendImage
・PrintPDF

・PrintLogo[*1]
・PrintSmartLabelImageData
・EnterPageMode
・ExitPageMode
・SetPageModeArea
・SetPageModeDirection
・SetPageModeLineSpacing
・PrintPageMode
・PrintPageModeText
・PrintPageModeTextEx
・PrintPageModeBarcode
・PrintPageModePDF417
・PrintPageModeQRcode
・PrintPageModeDataMatrix
・PrintPageModeMaxiCode
・PrintPageModeGS1DataBarStacked
・PrintPageModeGS1DataBarStackedOmnidirectional
・PrintPageModeGS1DataBarExpandedStacked
・PrintPageModeAztecCode
・SendPageModeBinary
・PrintPageModeImageFile
・PrintPageModeRectangle
・PrintPageModeLine
・PrintPageModeLogo[*1]

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

SetStatusChangedEventHandler　Start/Finish call back of printer status change

Starts or finishes the call back of printer status change.

Target　All platforms　All printers

Syntax
```
public void SetStatusChangedEventHandler(
                                    Action<int> statusChangedEventHandler);
```

Parameter　statusChangedEventHandler
Delegate object to receive notification of printer status change

Description　Starts to notify printer status change by executing this method after specifying the enabled delegate object other than null in statusChangedEventHandler.

The printer status change is notified only when IsConnect is true.

This method is called the latest status at the following timing.
・When the printer status is changed.
・When this method is executed after specifying the enabled delegate object other than null in statusChangedEventHandler.

The notification of the printer status change finishes at the following timing.
・When Disconnect is executed.
・When this method is executed by specifying null in statusChangedEventHandler.

When communication with the printer is disconnected, this method notifies 0x80000000. After disconnection from the printer, the library attempts to resume communication with the printer until Disconnect is executed. When communication with the printer becomes possible, this method notifies the latest printer status.
See GetStatus for description of the printer status.

Do not execute the APIs of PrinterManager class in the delegate object to specify in statusChangeEventHandler.

SetBarcodeScannerReadDataEventHandler
Start/Finish call back of receipt barcode data

Starts or finishes the call back of receipt of the barcode data.

Target All platforms RP-F10

Syntax
```
public void SetBarcodeScannerReadDataEventHandler(
                Action<byte[]> barcodeScannerReadDataEventHandler);
```

Parameter
barcodeScannerReadDataEventHandler
Delegate object to receive notification of barcode data receipt

Description The delegate object specified in the barcodeScannerReadDataEventHandler is called when the barcode scanner connected to the printer scans a barcode.

Receipt of barcode data is notified when IsConnect is true and communicating with the printer.

The receipt of barcode data finishes at the following timing.
・When Disconnect is executed.
・When this method is executed by specifying null in barcodeScannerReadDataEvent Handler.

Do not execute the APIs of PrinterManager class in the delegate object to specify in barcodeScannerReadDataEventHandler.

SetBarcodeScannerChangedOnlineEventHandler
Start/Finish call back of barcode scanner connection

Notifies the connection of the barcode scanner.

Target All platforms RP-F10

Syntax
```
public void SetBarcodeScannerChangedOnlineEventHandler(
                    Action barcodeScannerChangedOnlineEventHandler);
```

Parameter
barcodeScannerChangedOnlineEventHandler
Delegate object to receive the notification of barcode scanner connection

Description This method is called at the following timing.
・When the barcode scanner is connected with the printer and this method is executed after specifying the enabled delegate object other than null in barcodeScanner ChangedOnlineEventHandler.
・When the barcode scanner is connected to the printer.

The connection of the barcode scanner is notified when IsConnect is true and the printer is connecting.

The call back of the barcode scanner finishes at the following timing.
・When Disconnect is executed.
・When this method is executed by specifying null in barcodeScannerChangedOnline EventHandler.

Do not execute the APIs of PrinterManager class in the delegate object to specify in barcodeScannerChangedOnlineEventHandler.

SetBarcodeScannerChangedOfflineEventHandler
Start/Finish call back of barcode scanner disconnection

Notifies the disconnection of the barcode scanner.

Target: All platforms RP-F10

Syntax:
```
public void SetBarcodeScannerChangedOfflineEventHandler(
                        Action barcodeScannerChangedOfflineEventHandler);
```

Parameter:
barcodeScannerChangedOfflineEventHandler
Delegate object to receive the notifications of barcode scanner disconnection

Description:
This method is called at the following timing.
・When the barcode scanner is not connected to the printer and this method is executed after specifying the enabled delegate object other than null in barcodeScanner ChangedOfflineEventHandler.
・When the barcode scanner is disconnected from the printer.

The disconnection of the barcode scanner is notified when IsConnect is true.

The call back of barcode scanner disconnection finishes at the following timing.
・When Disconnect is executed.
・When this method is executed by specifying null in barcodeScannerChangedOffline EventHandler.

Do not execute the APIs of PrinterManager class in the delegate object to specify in barcodeScannerChangedOfflineEventHandler.

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.

Target	All platforms	All printers

Syntax

```
public void SendText(string text);
```

Parameter	`text`	Text data to send to the printer Data size that can be specified at 1 time is 16 KB (16384 bytes).
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.	
Description	This method encodes the specified text data to printable text data based on `InternationalCharacter` and `CodePage`, and 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 formatted text data to the printer.

Target	All platforms	All printers

Syntax

(a)
```
public void SendTextEx(string text,
                       CharacterBold bold,
                       CharacterUnderline underline,
                       CharacterReverse reverse,
                       CharacterFont font,
                       CharacterScale scale,
                       PrintAlignment alignment);
```

(b)
```
public void SendTextEx(string text,
                       CharacterBold bold,
                       CharacterUnderline underline,
                       CharacterReverse reverse,
                       CharacterInversion inversion,
                       CharacterFont font,
                       CharacterScale scale,
                       PrintAlignment alignment);
```

Parameter	`text`	Text data to send to the printer Data size that can be specified at 1 time is 16 KB (16384 bytes).
	`bold`	Bold character See "4.4.1⑪ Bold character (`CharacterBold`)" for available constants.
	`underline`	Underline See "4.4.1⑫ Underline (`CharacterUnderline`)" for available constants.

	`reverse`	Reverse print See "4.4.1⑰ Reverse print (`CharacterReverse`)" for available constants.
	`inversion`	Inversion print See "4.4.1⑱ Inversion print (`CharacterInversion`)" for available constants.
	`font`	Character font See "4.4.1⑭ Character font (`CharacterFont`)" for available constants.
	`scale`	Character scale See "4.4.1⑬ Character scale (`CharacterScale`)" for available constants.
	`alignment`	Alignment See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Description This method encodes the specified text data to printable text data based on `InternationalCharacter` and `CodePage`, and 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.

`PrintBarcode` Print barcode

Prints the barcode.
In syntax (a), barcode data can be specified as a character string.
In syntax (b), barcode data can be specified as a character string, and the alignment and the N:W ratio can also be specified.
In syntax (c), barcode data can be specified as a byte array.
In syntax (d), the Customer Bar Code_JP can be specified as a character string.

Target All platforms Syntax (a) to syntax (c): All printers
Syntax (d): MP-B21L

Syntax (a)
```
public void PrintBarcode(BarcodeSymbol barcodeSymbol,
                         string text,
                         ModuleSize moduleSize,
                         int moduleHeight,
                         HriPosition hriPosition,
                         CharacterFont hriFont,
                         PrintAlignment alignment);
```

(b)
```
public void PrintBarcode(BarcodeSymbol barcodeSymbol,
                         string text,
                         ModuleSize moduleSize,
                         int moduleHeight,
                         HriPosition hriPosition,
                         CharacterFont hriFont,
                         PrintAlignment alignment,
                         NwRatio nwRatio);
```

(c)
```
public void PrintBarcode(BarcodeSymbol barcodeSymbol,
                         byte[] data,
                         ModuleSize moduleSize,
                         int moduleHeight,
                         HriPosition hriPosition,
                         CharacterFont hriFont,
                         PrintAlignment alignment);
```

(d)
```
public void PrintBarcode(BarcodeSymbol barcodeSymbol,
                         string text,
                         ModuleSize moduleSize,
                         PrintAlignment alignment);
```

Parameter

`barcodeSymbol` Barcode symbol
See "4.4.1⑲ 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'	
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'	

Barcode	Number of Data	Inputtable Data Character String (Data)	Remarks
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 "4.4.1⑮ Module size (`ModuleSize`)" for available constants.

`moduleHeight` Barcode height (dot)

- When `barcodeSymbol` is below, the valid range is 1 to 255.
  `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`

- When `barcodeSymbol` is below, the valid range varies depending on `barcodeSymbol` and `moduleSize`.

barcodeSymbol		
	moduleSize	**Valid Range**
`BARCODE_GS1_OMNI_DIRECTIONAL`		
	`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

barcodeSymbol		
	moduleSize	**Valid Range**
BARCODE_GS1_TRUNCATED		
	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
BARCODE_GS1_LIMITED		
	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
BARCODE_GS1_EXPANDED		
	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 "4.4.1⑳ HRI character print position (`HriPosition`)" for available constants.

`hriFont` HRI character font
See "4.4.1⑭ Character font (`CharacterFont`)" for available constants.

`alignment` Alignment
See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants

`nwRatio` N:W ratio
See "4.4.1㉑ N:W ratio (`NwRatio`)" for available constants.
Depending on the specified `nwRatio` and `moduleSize`, the width of the wide element is set as shown in the following table.

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

moduleSize	nwRatio		
	NWRATIO_1TO2	**NWRATIO_1TO2_5**	**NWRATIO_1TO3**
BARCODE_MODULE_WIDTH_6	1.500 mm (12 dots)	1.875 mm (15 dots)	2.250 mm (18 dots)

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Note In SLP720RT, SLP721RT, RP-F10, RP-G10, MP-B30, MP-B30L, and MP-B21L, 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.

Target All platforms All printers

Syntax
```
public void PrintPDF417(string text,
                        ErrorCorrection errorCorrection,
                        int row,
                        int column,
                        ModuleSize moduleSize,
                        int moduleHeight,
                        PrintAlignment alignment,
                        Pdf417Symbol pdf417Symbol);
```

Parameter

`text`	Barcode data to send to the printer
`errorCorrection`	Error correction level See "4.4.1⑯ Error correction level (`ErrorCorrection`)" for available constants.
`row`	Number of rows (row) The valid range is 0, 3 to 90. When 0 is specified, the number of rows is automatically set.
`column`	Number of columns in data area The valid range is 0 to 30. When 0 is specified, the number of columns in the data area is automatically set.
`moduleSize`	Nominal fine element width See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.
`moduleHeight`	Module height (dot) The valid range is 2 to 127. When the module height is set smaller, some barcode scanners may not read it. Set 3 or more for normal use.
`alignment`	Alignment See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.
`pdf417Symbol`	PDF417 symbol See "4.4.1㉒ PDF417 symbol (`Pdf417Symbol`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Note In SLP720RT, SLP721RT, RP-F10, RP-G10, MP-B30, MP-B30L, and MP-B21L, 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.

Target All platforms All printers

Syntax
```
public void PrintQRcode(string text,
                        ErrorCorrection errorCorrection,
                        ModuleSize moduleSize,
                        PrintAlignment alignment,
                        QrModel model);
```

Parameter

`text`	Barcode data to send to the printer The version is automatically set depending on the number of data bytes set with `text`.
`errorCorrection`	Error correction level See "4.4.1⑯ Error correction level (`ErrorCorrection`)" for available constants.
`moduleSize`	Module size See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.
`alignment`	Alignment See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.
`model`	QR Code Model See "4.4.1㉓ QR Code Model (`QrModel`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Note In SLP720RT, SLP721RT, RP-F10, RP-G10, MP-B30, MP-B30L, and MP-B21L, 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.

Target All platforms All printers

Syntax
```
public void PrintDataMatrix(string text,
                            DataMatrixModule dataMatrixModule,
                            ModuleSize moduleSize,
                            PrintAlignment alignment);
```

Parameter

`text`	Barcode data to send to the printer

	`dataMatrixModule`	Number of Data Matrix modules See "4.4.1㉔ Data Matrix module (`DataMatrixModule`)" for available constants.
	`moduleSize`	Module size See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.
	`alignment`	Alignment See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Note In SLP720RT, SLP721RT, RP-F10, RP-G10, MP-B30, MP-B30L, and MP-B21L, 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.

`PrintMaxiCode` Print MaxiCode

Prints MaxiCode.

Target All platforms All printers

Syntax
```
public void PrintMaxiCode(string text,
                          MaxiCodeMode maxiCodeMode,
                          PrintAlignment alignment);
```

Parameter	`text`	Barcode data to send to the printer • When `maxiCodeMode` is **`MAXI_CODE_2`**: Add service class (3 digits), country code (3 digits), and postal code (9 digits) to the beginning of the data. • When `maxiCodeMode` is **`MAXI_CODE_3`**: Add service class (3 digits), country code (3 digits), and postal code (6 digits) to the beginning of the data.
	`maxiCodeMode`	MaxiCode Mode See "4.4.1㉕ MaxiCode Mode (`MaxiCodeMode`)" for available constants.
	`alignment`	Alignment See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Note In SLP720RT, SLP721RT, RP-F10, RP-G10, MP-B30, MP-B30L, and MP-B21L, 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.

PrintGS1DataBarStacked — Print GS1 Databar Stacked

Prints GS1 Databar Stacked.

Target	All platforms	SLP720RT, SLP721RT, RP-F10, RP-G10, MP-B30, MP-B30L, MP-B21L, MP-B20, MP-A40

Syntax

```
public void PrintGS1DataBarStacked(string text,
                                   ModuleSize moduleSize,
                                   PrintAlignment alignment);
```

Parameter	`text`	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.
	`moduleSize`	Module size See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.
	`alignment`	Alignment See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.

Error — `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

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.

Target	All platforms	SLP720RT, SLP721RT, RP-F10, RP-G10, MP-B30, MP-B30L, MP-B21L, MP-B20, MP-A40

Syntax

```
public void PrintGS1DataBarStackedOmnidirectional(string text,
                                                  int moduleHeight,
                                                  ModuleSize moduleSize,
                                                  PrintAlignment alignment);
```

Parameter	`text`	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.
	`moduleHeight`	Barcode module height (number of modules) The valid range is 33 to 255.
	`moduleSize`	Module size See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.
	`alignment`	Alignment See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.

Error — `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

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

PrintGS1DataBarExpandedStacked Print GS1 Databar Expanded Stacked

Prints GS1 Databar Expanded Stacked.

Target All platforms SLP720RT, SLP721RT, RP-F10, RP-G10, MP-B30, MP-B30L, MP-B21L, MP-B20, MP-A40

Syntax
```
public void PrintGS1DataBarExpandedStacked(string text,
                                           int column,
                                           ModuleSize moduleSize,
                                           PrintAlignment alignment);
```

Parameter

`text`	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' for FNC1.
`column`	Number of columns Specify the number of columns in 1 line. An even number from 2 to 20 is valid.
`moduleSize`	Module size See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.
`alignment`	Alignment See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

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

PrintAztecCode Print Aztec Code

Prints Aztec Code.

Target All platforms MP-B30L, MP-B21L

Syntax
```
public void PrintAztecCode(string text,
                           int layer,
                           int errorCorrection
                           ModuleSize moduleSize,
                           AztecSymbol aztecSymbol,
                           PrintAlignment alignment);
```

Parameter `text` Barcode data to send to the printer
Encodes the data to binary in UTF-8 and sends to the printer.
When the data to be sent cannot be specified as a string, it can be specified using the following escape sequence.

Escape sequence	Description
\u00nn	Control code (Specify nn as a hexadecimal number)
\\	Backslash

To specify FNC1, enter "\u001b\u0030" in the data.
To specify 1Bh, enter "\u001b\u001b" in the data.

`layer` Number of layer
The valid range of full-range mode is 0, 4 to 32.
The valid range of compact mode is 0, 1 to 4.
When 0 is specified, layer is automatically set.

`errorCorrection` Error correction level (%)
The valid range is 0, 5 to 95.
When 0 is specified, the level is 23%.

`moduleSize` Module size
See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.

`aztecSymbol` Aztec symbol
See "4.4.1㉖ Aztec symbol (`AztecSymbol`)" for available constants.

`alignment` Alignment
See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

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

`CutPaper` Cut paper

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

Target All platforms All printers

Syntax
```
public void CutPaper(CuttingMethod cuttingMethod);
```

Parameter `cuttingMethod` Cutting method
See "4.4.1㉗ Cutting method (`CuttingMethod`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Description For MP-B30, MP-B30L, MP-B21L, MP-B20, or MP-A40, the paper is not cut and the paper is fed to the paper cut position.

FeedPosition — Paper form feed

Performs the paper form feed of marked paper or label.

Target	All platforms	SLP720RT, SLP721RT, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void FeedPosition(FeedPosition feedPosition);
```

Parameter	`feedPosition`	Form feed position See "4.4.1㉘ Form feed position (`FeedPosition`)" for available constants.
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.	
Description	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.

Target	All platforms	All devices

Syntax
```
public void SendBinary(byte[] data);
```

Parameter	`data`	Binary data to send to the printer Data size that can be specified at 1 time is 3MB (3145728 bytes).
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.	
Description	This method sends the specified binary data to the printer without conversion. By sending printer command as binary data by this method, printer functions not supported in the library become available. However, this method does not support commands to get the response from the printer.	

SendDataFile — Send specified file

Sends file data.

Target	All platforms	All devices

Syntax
```
public void SendDataFile(string fileName,
                         PrintAlignment alignment,
                         Dithering dithering);
```

Parameter	`fileName`	File path of the data to send to the printer The formats that can be entered are described below. For iOS: File path strings to use

For Android:

- 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 "3.4 Precautions - About Scoped Storage" for details.

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

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

The maximum file size that can be specified is 16MB (167772161 bytes).
The file extensions that can be sent and the file transmission are described below.

- .bmp, .jpg, .jpeg, .png
  Data is sent to the printer as image file. Colored image file is converted to monochrome image by binarization and sent to the printer. Printing is performed in batch after mapping the image file on the memory of the printer.

- .txt
  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 `InternationalCharacter` and `CodePage`, 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 at the end of the text data.

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

- .htm, .html
  Supported only by RP-E10, RP-D10 and MP-A40.
  Data is sent to the printer as html data without conversion.

`alignment` Alignment
It is valid when the extension of the file specified by `fileName` is .bmp, .jpg, .jpeg, .png, or .txt.
See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.

`dithering` Dithering
It is valid when the extension of the file specified by `fileName` is .bmp, .jpg, .jpeg, or .png.
See "4.4.1⑥ Dithering (`Dithering`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

SendImage — Send image

Prints an specified image.

Target: All platforms, All devices

Syntax:
```
public void SendImage(object image,
                      PrintAlignment alignment,
                      Dithering dithering);
```

Parameter:

`image` — Image Data to sent to the printer

For iOS:
Specify image data that conforms to the UIKit.UIImage class provided by iOS.

For Android:
Specify image data that conforms to the Android.Graphics.Bitmap class provided by Android.

`alignment` — Alignment
See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.

`dithering` — Dithering
See "4.4.1⑥ Dithering (`Dithering`)" for available constants.

Error: `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

PrintPDF — Print PDF page

Prints the specified page of the selected PDF file.

Target: All platforms, All printers

Syntax:
```
public void PrintPDF(string fileName,
                     int startIndex,
                     int endIndex,
                     Rotate rotate,
                     ImageScale imageScale,
                     int bottomMargin,
                     Dithering dithering,
                     PrintAlignment alignment)
```

Parameter:

`fileName` — File path of the PDF file
The file extension for supporting PDF is .pdf.
The maximum file size that can be specified is 16MB (167772161 bytes).
The formats that can be entered are described below.

For iOS:
File path strings to use

	For Android: • 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 "3.4 Precautions - About Scoped Storage" for details. • URI string of the following scheme name handled by the class "android.net.Uri" prepared for Android ・file:// ・content:// It is necessary to specify the URI string obtained from "Storage Access Framework" for this parameter. Please note that URI created without being obtained from "Storage Access Framework" may not be able to open the file.
`startIndex`	Start number of the printed page The range is -1, 1 to the number of pages in the PDF file. When -1 is specified for `startIndex`, all pages are printed. When -1 is specified for `startIndex`, the value of `endIndex` is ignored. When 1 or more is specified for `startIndex`, pages from the page number specified in `startIndex` to the page number specified in `endIndex` are printed. When a value more than the value specified for `endIndex` is specified for `startIndex`, an error occurs. When an out-of-range value is specified for `startIndex` or `endIndex`, an error occurs.
`endIndex`	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 `endIndex`, pages from the page number specified in `startIndex` to the last page of the PDF file are printed.
`rotate`	Rotation direction of the image See "4.4.1(3)㉙ Image rotation direction (`Rotate`)" for available constants.
`imageScale`	Image scaling See "4.4.1(3)㉚ Image scaling (`ImageScale`)" for available constants. When **`IMAGE_SCALE_WIDTH_FIT`** is specified for `imageScale`, the image width is converted to the print width of the printer while maintaining the aspect ratio.
`bottomMargin`	Bottom margin (dot) The range is -1, 0 to 2400. When -1 is specified for `bottomMargin`, the image is created and printed while maintaining the bottom margin. When a value between 0 and 2400 is specified for `bottomMargin`, the bottom margin is changed to the specified size. When a value between 0 and 2400 is specified for `bottomMargin`, blank pages are not printed.
`dithering`	Dithering See "4.4.1⑥ Dithering (`Dithering`)" for available constants.
`alignment`	Alignment See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Description PDF is converted to printable format for the printer and is sent to the printer.

The color image data is converted to monochrome image by binarization.

When the following memory switches is set to "Enable", form feed of marked paper or label is performed after printing the PDF page.
・MP-B21L, MP-B30L : MS 1-2 (Mark/Gap Mode Selection)
・MP-B30, MP-A40 : MS 1-2 (Mark Mode Selection)
・SLP720RT, SLP721RT, RP-E10 : MS 1-3 (Mark Mode Selection)

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

`PrintLogo` Print logo

Prints the registered logo.

Target All platforms All printers

Syntax
```
public void PrintLogo(string logoId, PrintAlignment alignment);
```

Parameter

`logoId`	ID of the logo to be printed (key code) Specify the ID of the registered logo as a character string.
`alignment`	Alignment See "4.4.1⑩ Alignment (`PrintAlignment`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

`PrintSmartLabelImageData` Print label

Prints labels.

Target All platforms SLP720RT, SLP721RT, MP-B30L, MP-B21L

Syntax
```
public void PrintSmartLabelImageData(SmartLabelManager labelManager);
```

Parameter

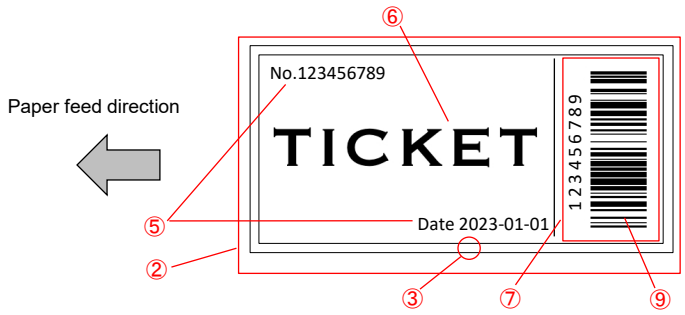
`labelManager`	Class that provides the function to convert the label file (*.sl, *.slex) into the printable data

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Description See "**4.4.6 `SmartLabelManager` Class**" 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

```
printerManager.EnterPageMode();
```

② Specify print area of page mode

```
printerManager.SetPageModeArea(0, 0, 355, 576);
```

③ Specify a rectangle and a ruled line

```
printerManager.PrintPageModeRectangle(0, 0, 344, 575, LineStyle.LINESTYLE_THIN);
printerManager.PrintPageModeRectangle(7, 7, 336, 567, LineStyle.LINESTYLE_THIN);
printerManager.PrintPageModeLine(11, 404, 334, 404, LineStyle.LINESTYLE_THIN);
```

④ Specify print direction of page mode

```
printerManager.SetPageModeDirection(Direction.DIRECTION_TOP_TO_BOTTOM );
```

⑤ Specify a character

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

⑥ Specify an image file

```
printerManager.PrintPageModeImageFile(
            10,
            222,
            System.IO.Path.Combine(FileSystem.Current.AppDataDirectory, "TicketImage.jpg"),
            Dithering.DITHERING_DISABLE);
```

⑦ Specify print area of page mode

```
printerManager.SetPageModeArea(0, 404, 345, 163);
```

⑧ Specify print direction

```
printerManager.SetPageModeDirection(Direction.DIRECTION_LEFT_TO_RIGHT);
```

⑨ Specify a barcode

```
printerManager.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

```
printerManager.PrintPageMode(CuttingMethod.CUT_PARTIAL);
```

⑪ Ends page mode

```
printerManager.ExitPageMode();
```

EnterPageMode — Start page mode

Starts page mode.

Target	All platforms	RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void EnterPageMode();
```

Error
`PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on 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.

Target	All platforms	RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void ExitPageMode();
```

Error
`PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on 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.

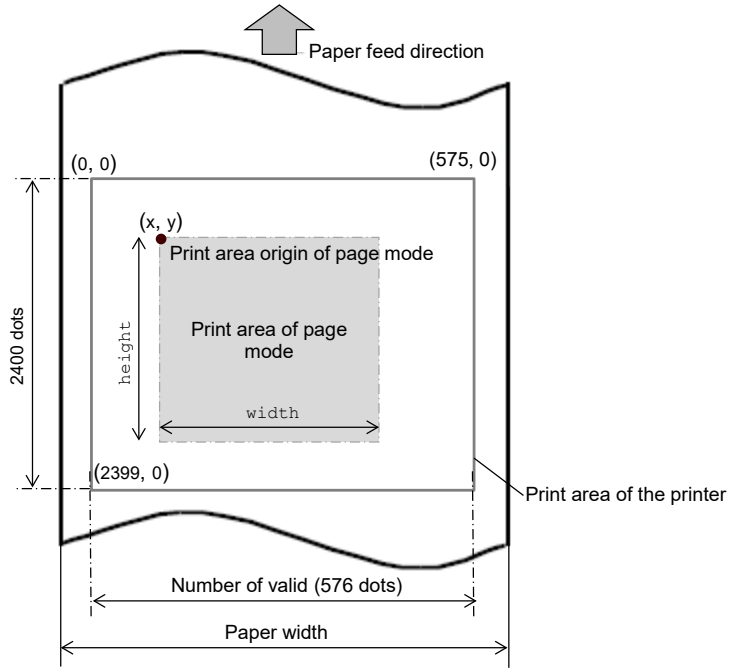
Target	All platforms	RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void SetPageModeArea(int x, int y, int width, int height);
```

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 576 dots.



Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on 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` = print width, `height` = 2400 after executing `EnterPageMode`.

The print width for MP-B30 is 576.
The print width for MP-B30L and MP-B21L is the width of the memory switch MS 3 (Print Width) of the printer.
The print width for MP-A40 is the printing width of the memory switch MS 3-1 to 6 (Paper Width) of the printer.
The print width for RP-F10, RP-G10, RP-E10, and RP-D10 is the effective number of dots applied to the memory switch MS 4-5 (Number of Effective Dots).
See "USER'S GUIDE" for details of the memory switches and the factory default settings.

`SetPageModeDirection` Specify print direction of page mode

Specifies print direction of page mode.

Target All platforms RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void SetPageModeDirection(Direction direction);
```

Parameter	`direction`	Print direction See "4.4.1㉛ Print direction (`Direction`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Description Start page mode by `EnterPageMode` before executing this method.

The print direction is left to right after executing `EnterPageMode`.

`SetPageModeLineSpacing` Specify line spacing of page mode

Specifies line spacing of page mode.

Target All platforms RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void SetPageModeLineSpacing(int lineSpacing);
```

Parameter	`lineSpacing`	Line spacing (dot) of page mode The valid range is 0 to 255.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Description Start page mode by `EnterPageMode` before executing this method.

The line spacing is 34 dots after executing `EnterPageMode`.

`PrintPageMode` Print page mode

Prints the print data kept in page data buffer.

Target All platforms RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void PrintPageMode(CuttingMethod cuttingMethod);
```

Parameter	`cuttingMethod`	Cutting method See "4.4.1㉗ Cutting method (`CuttingMethod`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Description The print data is kept after printing. The print data is discarded at the timing of the following:
- ・Execute `EnterPageMode`
- ・Execute `Disconnect`
- ・Execute `ExitPageMode`

`PrintPageModeText` Send text data of page mode

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

Target All platforms RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void PrintPageModeText(int startX, int startY, string text);
```

Parameter	`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	`text`	Text data Data size that can be specified at 1 time is 16 KB (16384 bytes).

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Description This method encodes the specified text data to printable text data based on `InternationalCharacter` and `CodePage`.

Start page mode by `EnterPageMode` before executing this method.

`PrintPageModeTextEx` Send format specified text data of page mode

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

Target All platforms RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax

```
public void PrintPageModeTextEx(int startX,
                                int startY,
                                string text,
                                CharacterBold bold,
                                CharacterUnderline underline,
                                CharacterReverse reverse,
                                CharacterFont font,
                                CharacterScale scale);
```

Parameter	`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	`text`	Text data Data size that can be specified at 1 time is 16 KB (16384 bytes).
	`bold`	Bold character See "4.4.1⑪ Bold character (`CharacterBold`)" for available constants.
	`underline`	Underline See "4.4.1⑫ Underline (`CharacterUnderline`)" for available constants.
	`reverse`	Reverse print See "4.4.1⑰ Reverse print (`CharacterReverse`)" for available constants.
	`font`	Font See "4.4.1⑭ Character font (`CharacterFont`)" for available constants.
	`scale`	Character scale See "4.4.1⑬ Character scale (`CharacterScale`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Description This method encodes the specified text data to printable text data based on `InternationalCharacter` and `CodePage`.

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.
In syntax (a), barcode data can be specified as a character string.
In syntax (b), barcode data can be specified as a character string, and the N:W ratio can also be specified.
In syntax (c), barcode data can be specified as a byte array.
In syntax (d), the Customer Bar Code_JP can be specified as a character string.

Target All platforms
Syntax (a) to syntax (c): RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40
Syntax (d): MP-B21L

Syntax (a)
```
public void PrintPageModeBarcode(int startX,
                                 int startY,
                                 BarcodeSymbol barcodeSymbol,
                                 string text,
                                 ModuleSize moduleSize,
                                 int moduleHeight,
                                 HriPosition hriPosition,
                                 CharacterFont hriFont);
```

(b)
```
public void PrintPageModeBarcode(int startX,
                                 int startY,
                                 BarcodeSymbol barcodeSymbol,
                                 string text,
                                 ModuleSize moduleSize,
                                 int moduleHeight,
                                 HriPosition hriPosition,
                                 CharacterFont hriFont,
                                 NwRatio nwRatio);
```

(c)
```
public void PrintPageModeBarcode(int startX,
                                 int startY,
                                 BarcodeSymbol barcodeSymbol,
                                 byte[] data,
                                 ModuleSize moduleSize,
                                 int moduleHeight,
                                 HriPosition hriPosition,
                                 CharacterFont hriFont);
```

(d)
```
public void PrintPageModeBarcode(int startX,
                                 int startY,
                                 BarcodeSymbol barcodeSymbol,
                                 string text,
                                 ModuleSize moduleSize);
```

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

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

`barcodeSymbol` BarcodeSymbol
See "4.4.1⑲ Barcode symbol (`BarcodeSymbol`)" for available constants and correspondent 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'	
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 "4.4.1⑮ 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_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

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

barcodeSymbol		
	moduleSize	**Valid Range**
BARCODE_GS1_OMNI_DIRECTIONAL		
	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
BARCODE_GS1_TRUNCATED		
	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

barcodeSymbol		
	moduleSize	**Valid Range**
BARCODE_GS1_LIMITED		
	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
BARCODE_GS1_EXPANDED		
	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 "4.4.1⑳ HRI character print position (`HriPosition`)" for available constants.

`hriFont` HRI character font
See "4.4.1⑭ Character font (`CharacterFont`)" for available constants.

`nwRatio` N:W ratio
See "4.4.1㉑ 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.

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

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on 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.
In RP-F10, RP-G10, MP-B30, MP-B30L, and MP-B21L, 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.

PrintPageModePDF417 Print PDF417 of page mode

Maps PDF417 on the print area of page mode.

Target	All platforms	RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax

```
public void PrintPageModePDF417(int startX,
                                int startY,
                                string text,
                                ErrorCorrection errorCorrection,
                                int row,
                                int column,
                                ModuleSize moduleSize,
                                int moduleHeight,
                                Pdf417Symbol pdf417Symbol);
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`text`	Barcode data
`errorCorrection`	Error correction level See "4.4.1⑯ Error correction level (`ErrorCorrection`)" for available constants.
`row`	The number of rows (row) The valid range is 0, 3 to 90. When 0 is specified, the number of rows is automatically set.
`column`	The number of columns in data area The valid range is 0 to 30. When 0 is specified, the number of columns in the data area is automatically set.
`moduleSize`	Barcode width See"4.4.1⑮ Module size (`ModuleSize`)" for available constants.
`moduleHeight`	Module height (dot) The valid range is 2 to 127. When the module height is set smaller, some barcode scanners may not read it. Set 3 or more for normal use.
`pdf417Symbol`	Symbol of PDF417 See "4.4.1㉒ PDF417 symbol (`Pdf417Symbol`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on 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.
In RP-F10, RP-G10, MP-B30, MP-B30L, and MP-B21L, 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.

Target: All platforms — RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax:

```
public void PrintPageModeQRcode(int startX,
                                 int startY,
                                 string text,
                                 ErrorCorrection errorCorrection,
                                 ModuleSize moduleSize,
                                 QrModel model);
```

Parameter:

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`text`	Barcode data The version is automatically set depending on the number of data specified on `text`.
`errorCorrection`	Error correction level See "4.4.1⑯ Error correction level (`ErrorCorrection`)" for available constants.
`moduleSize`	Barcode width See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.
`model`	QR Code Model See "4.4.1㉓ QR Code Model (`QrModel`)" for available constants.

Error: `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on 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.
In RP-F10, RP-G10, MP-B30, MP-B30L, and MP-B21L, 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.

PrintPageModeDataMatrix — Print Data Matrix of page mode

Maps Data Matrix on the print area of page mode.

Target: All platforms — RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax:

```
public void PrintPageModeDataMatrix(int startX,
                                     int startY,
                                     string text,
                                     DataMatrixModule dataMatrixModule,
                                     ModuleSize moduleSize);
```

Parameter:

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

	`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	`text`	Barcode data
	`dataMatrixModule`	The number of Data Matrix modules See "4.4.1㉔ Data Matrix module (`DataMatrixModule`)" for available constants.
	`moduleSize`	Barcode width See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on 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.
In RP-F10, RP-G10, MP-B30, MP-B30L, and MP-B21L, 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.

`PrintPageModeMaxiCode` Print MaxiCode of page mode

Maps MaxiCode on the print area of page mode.

Target All platforms RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void PrintPageModeMaxiCode(int startX,
                                  int startY,
                                  string text,
                                  MaxiCodeMode maxiCodeMode);
```

Parameter	`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	`text`	Barcode data • When `maxiCodeMode` is **`MAXI_CODE_2`**: Add the service class (3 digits), the country code (3 digits), and the postal code (9 digits) to the beginning of the data. • When `maxiCodeMode` is **`MAXI_CODE_3`**: Add the service class (3 digits), the country code (3 digits), and the postal code (6 digits) to the beginning of the data.
	`maxiCodeMode`	MaxiCode Mode See "4.4.1㉕ MaxiCode Mode (`MaxiCodeMode`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on 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.
In RP-F10, RP-G10, MP-B30, MP-B30L, and MP-B21L, 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.

PrintPageModeGS1DataBarStacked Print GS1 Databar Stacked of page mode

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

Target All platforms RP-F10, RP-G10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void PrintPageModeGS1DataBarStacked(int startX,
                                           int startY,
                                           string text,
                                           ModuleSize moduleSize);
```

Parameter

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`text`	Barcode data 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.
`moduleSize`	Barcode width See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on 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.

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

PrintPageModeGS1DataBarStackedOmnidirectional Print GS1 Databar Stacked Omni-directional of page mode

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

Target All platforms RP-F10, RP-G10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void PrintPageModeGS1DataBarStackedOmnidirectional(int startX,
                                                          int startY,
                                                          string text,
                                                          int moduleHeight,
                                                          ModuleSize moduleSize);
```

Parameter

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

	`text`	Barcode data 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.
	`moduleHeight`	Barcode module height (the number of the modules) The valid range is 33 to 255.
	`moduleSize`	Module size See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on 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.

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.

Target All platforms RP-F10, RP-G10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax
```
public void PrintPageModeGS1DataBarExpandedStacked(int startX,
                                                   int startY,
                                                   string text,
                                                   int column,
                                                   ModuleSize moduleSize);
```

Parameter	`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	`text`	Barcode data Enter any number of characters using the following: ' ', '!', '"', '%', '&', '', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '=', '>', '?', '_', '0' to '9', 'A' to 'Z', 'a' to 'z'. Enter '{1' to FNC1.
	`column`	The number of columns Specify the number of the columns in 1 line. The valid range is the even number from 2 to 20.
	`moduleSize`	Barcode width See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on 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.

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

PrintPageModeAztecCode — Print Aztec Code of page mode

Maps Aztec Code on the print area of page mode.

Target: All platforms — MP-B30L, MP-B21L

Syntax:

```
public void PrintPageModeAztecCode(int startX,
                                   int startY,
                                   string text,
                                   int layer,
                                   int errorCorrection
                                   ModuleSize moduleSize,
                                   AztecSymbol aztecSymbol);
```

Parameter:

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

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

`text` — Barcode data
Encodes the data to binary in UTF-8 and sends to the printer.
When the data to be sent cannot be specified as a string, it can be specified using the following escape sequence.

Escape sequence	Description
\u00nn	Control code (Specify nn as a hexadecimal number)
\\	Backslash

To specify FNC1, enter "\u001b\u0030" in the data.
To specify 1Bh, enter "\u001b\u001b" in the data.

`layer` — Number of layer
The valid range of full-range mode is 0, 4 to 32.
The valid range of compact mode is 0, 1 to 4.
When 0 is specified, layer is automatically set.

`errorCorrection` — Error correction level (%)
The valid range is 0, 5 to 95.
When 0 is specified, the level is 23%.

`moduleSize` — Module size
See "4.4.1⑮ Module size (`ModuleSize`)" for available constants.

`aztecSymbol` — Aztec symbol
See "4.4.1㉖ Aztec symbol (`AztecSymbol`)" for available constants.

Error: `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on 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.

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.

Target	All platforms	RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax

```
public void SendPageModeBinary(byte[] data);
```

Parameter	`data`	Binary data Data size that can be specified at 1 time is 3MB (3145728 bytes).

Error

`PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on 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 by 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.

Target	All platforms	RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax

```
public void PrintPageModeImageFile(int startX,
                                   int startY,
                                   string fileName,
                                   Dithering dithering);
```

Parameter	`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
	`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
	`fileName`	File path of the data The formats that can be entered are described below. For iOS: File path strings to use For Android: • 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 "3.4 Precautions - About Scoped Storage" for details.

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

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

The maximum file size that can be specified is 16MB (167772161 bytes).
The image files that can be sent are .bmp, .jpg, .jpeg, .png.
Colored image file is converted to monochrome image by binarization and registered.

`dithering`	Dithering See "4.4.1⑥ Dithering (`Dithering`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Description Start page mode by `EnterPageMode` before executing this method.

`PrintPageModeRectangle` Draw rectangle image of page mode

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

Target All platforms RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

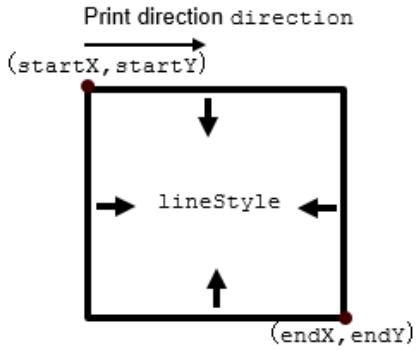
Syntax
```
public void PrintPageModeRectangle(int startX,
                                   int startY,
                                   int endX,
                                   int endY,
                                   LineStyle lineStyle);
```

`startX`	The horizontal reference point (dot) from the starting point The valid range is 0 to 2399.
`startY`	The vertical reference point (dot) from the starting point The valid range is 0 to 2399.
`endX`	The horizontal reference point (dot) from the ending point The valid range is 0 to 2399.
`endY`	The vertical reference point (dot) from the ending point The valid range is 0 to 2399.
`lineStyle`	Line style See "4.4.1㉜ Line style (`LineStyle`)" for available constants.

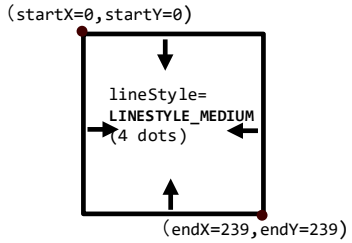
Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Description Start page mode by `EnterPageMode` before executing this method.

The rectangle is mapped to the `direction` of `SetPageModeDirection` as shown in the figure below.



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

Image	Parameter
	`startX` 0 `startY` 0 `endX` 239 `endY` 239 `lineStyle` **`LINESTYLE_MEDIUM`**

`PrintPageModeLine` Print ruled line of page mode

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

Target: All platforms — RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax:

```
public void PrintPageModeLine(int startX,
                              int startY,
                              int endX,
                              int endY,
                              LineStyle lineStyle);
```

Parameter:

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

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

`endX` The horizontal reference point (dot) from the ending point
The valid range is 0 to 2399.

`endY` The vertical reference point (dot) from the ending point
The valid range is 0 to 2399.

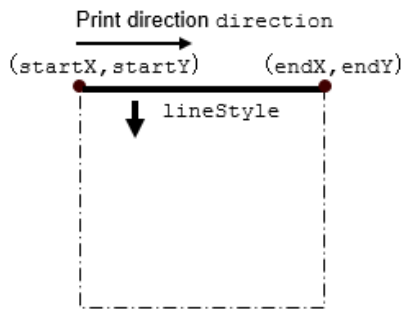
	`lineStyle`	Line style See "4.4.1㉜ Line style (`LineStyle`)" for available constants.

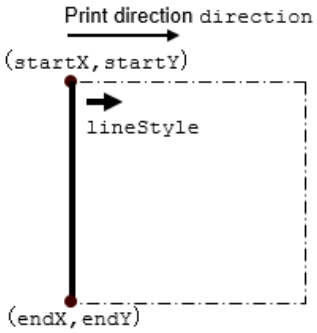
Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details on the error.

Description Start page mode by `EnterPageMode` before executing this method.

A diagonal stroke cannot be drawn by this method.

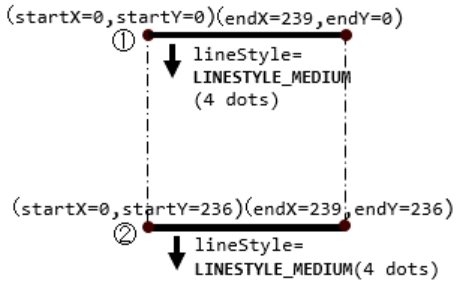
The ruled line is mapped to the `direction` of `SetPageModeDirection` as shown in the figure below.



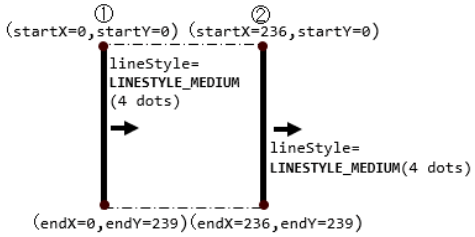


Mapping direction of horizontal ruled line **Mapping direction of vertical ruled line**

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

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

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

PrintPageModeLogo — Print logo of page mode

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

Target: All platforms — RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax:

```
public void PrintPageModeLogo(int startX,
                              int startY,
                              string logoId);
```

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.
- `logoId` — Logo ID to print (key code). Specify the ID of the registered logo as a character string

Error: `PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `PrinterException` Class**" for details on the error.

Description: Start page mode by `EnterPageMode` before executing this method.

(5) Common property detail to standard mode and page mode

SendTimeout — Get/Set send timeout period

Gets or sets the timeout period in sending data.

Target	All platforms	All devices

Syntax

```
public int SendTimeout {get; set;}
```

Valid range

For iOS:
100 to 300000 (millisecond: ms)
When the value is specified less than 100, the period is set to 100 ms.
When the value is specified more than 300000, the period is set to 300000 ms.
For Android:
100 to 900000 (millisecond: ms)
When the value is specified out of the range, the period is set to 10000 ms.

Default: 10000

Description: Getting or setting is possible by this property regardless of whether `IsConnect` is true or false.
The set timeout period becomes effective at the next data sending.

ReceiveTimeout — Get/Set receive timeout period

Gets or sets the timeout period in receiving data.

Target	All platforms	All devices

Syntax

```
public int ReceiveTimeout {get; set;}
```

Valid range

For iOS:
100 to 300000 (millisecond: ms)
When the value is specified less than 100, the period is set to 100 ms.
When the value is specified more than 300000, the period is set to 300000 ms.
For Android:
100 to 900000 (millisecond: ms)
When the value is specified out of the range, the period is set to 10000 ms.

Default: 10000

Description: Getting or setting is possible by this property regardless of whether `IsConnect` is true or false.
The set timeout period becomes effective at the next data receiving.

InternationalCharacter — Get/Set international character set

Gets or sets the value of international character set.

Target: All platforms, All devices

Syntax:
```
public InternationalCharacter {get; set;}
```

Description: See “4.4.1㉝ International character set (InternationalCharacter)" for configurable constants. When an invalid value is specified, it is ignored.

When this property is not set, the international character set is as follows depending on the language setting of device.

When the language setting of the device is Japanese: **COUNTRY_JAPAN**
When the language setting of the device is other than Japanese: **COUNTRY_USA**

When 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 of the 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

CodePage — Get/Set codepage

Gets or sets the value of codepage.

Target: All platforms, All devices

Syntax:
```
public CodePage CodePage {get; set;}
```

Description: See "4.4.1㉞ Codepage (CodePage)" for configurable constants. When an invalid value is specified, it is ignored.

When this property is not set, the codepage follows depending on the language setting of device.

When the language setting of the device is Japanese:
CODE_PAGE_KATAKANA
When the language setting of the device is other than Japanese:
CODE_PAGE_1252

The encoder used for sending the text data by SendText, SendTextEx, SendDataFile, PrintPageModeText, or PrintPageModeTextEx is changed.
See "Appendix A Character Set" for characters to be printed.

DeviceModel — Get device model

Gets the value of the connecting printer model.

Target: All platforms, All devices

Syntax:
```
public DeviceModel DeviceModel {get;}
```

Default: DeviceModel.DEVICE_MODEL_RP_E10

Return value: See "4.4.1① Device model (DeviceModel)" for available constants.

PortType — Get connecting port type

Gets the value of the port type used for connecting with the printer.

Target	All platforms	All printers

Syntax

```
public PortType PortType {get;}
```

Default: PortType.PORT_TYPE_BLUETOOTH

Return value: See "4.4.1② Port type (PortType)" for available constants.

IsConnect — Verify connection state with printer

Verifies connection state with the printer.

Target	All platforms	All devices

Syntax

```
public bool IsConnect {get;}
```

Return value:
- true — Connected to the printer
- false — Not connected to the printer

Description: This property retains the Connect state as a BOOL value.
When Connect succeeds, this property is true. After Connect, when Disconnect succeeds, this property becomes false.

SocketKeepingTime — Get/Set socket keeping time

Gets or sets the socket keeping time.

Target	All platforms	SLP720RT, SLP721RT, RP-F10, RP-G10, RP-E10, RP-D10, MP-B30, MP-B30L, MP-B21L, MP-A40

Syntax

```
public int SocketKeepingTime {get; set;}
```

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: Getting or setting is possible by this property 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 iOS app "SII Printer Utility" on the App Store and the Android app "SII Printer Utility" on the Google Play.

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

Version — Get SDK version

Gets the SDK version as a character string.

Target	All platforms	All devices
Syntax	`public string Version {get;}`	
Return value	Returns 2 SDK version strings. (e.g. "1.0.0 / 1.0.0") 1st: SII SDK for .NET MAUI version 2nd: SII Print Class Library version compatible with the operating environment	
Description	Getting is possible by this property regardless of whether `IsConnect` is true or false.	

PrintSmartLabelMode — Get/Set paper when printing label file

Gets or sets the paper when printing label file.

Target	All platforms	SLP720RT, SLP721RT, MP-B30L, MP-B21L
Syntax	`public PrintSmartLabelMode PrintSmartLabelMode {get; set;}`	
Description	See “4.4.1㉟ Paper selection with or without mark when printing label file (`PrintSmartLabelMode`)" for configurable constants. When an invalid value is specified, it is ignored. When this property is not set, **`PRINTSMARTLABRL_MODE_MARK`** is set. Getting or setting is possible by this property regardless of whether `IsConnect` is true or false. When specifying the marked paper, feeds the paper to the print start position at `printSmartLabelImageData` execution. When specifying the paper without the mark, the paper is not fed to the print start position at `printSmartLabelImageData` execution.	

4.4.3 `DeviceInfo` Class

This class stores the device information found by printer searching method. See "4.4.3(2) Property List" for device information to be able to get.

(1) Method List

Methods provided by `DeviceInfo` class are shown in the following table.

Name	Description
`DeviceInfo`	Constructor of the device information class

(2) Property List

Properties provided by `DeviceInfo` class are shown in the following table.

Name	Access	Description
`PortType`	R	Get connecting port type
`DeviceName`	R	Get device name (printer model name)
`BluetoothAddress`	R	Get Bluetooth address
`MacAddress`	R	Get MAC address
`IpAddress`	R	Get IP address
`IsBonded`	R	Get paring state
`DevicePath`	R	Get device path

(3) Method Details

DeviceInfo — Constructor

In syntax (a), this is the constructor of the device information class used for Bluetooth connections.
In syntax (b), this is the constructor of the device information class used for TCP/IP connections.
In syntax (c), this is the constructor of the device information class used for USB connections.

Target	All platforms	All devices

Syntax

(a)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string bluetoothAddress,
                  bool isBonded);
```

(b)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string macAddress,
                  string ipAddress);
```

(c)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string devicePath);
```

Description: This method stores the printer information found by `StartDiscoveryPrinter`.

(4) Property Details

PortType — Get connecting port type

Target	All platforms	All devices
Syntax	`public PortType PortType {get;}`	
Description	This property gets the connecting port type from the device information found by `StartDiscoveryDevice`.	

DeviceName — Get device name (printer model name)

Target	All platforms	All devices
Syntax	`public string DeviceName {get;}`	
Description	This property gets the device name (printer model name) from the device information found by `StartDiscoveryDevice`.	

BluetoothAddress — Get Bluetooth address

Target	All platforms	All printers
Syntax	`public string BluetoothAddress {get;}`	
Description	This property gets the Bluetooth address from the printer information found by `StartDiscoveryDevice`.	

MacAddress — Get MAC address

Target	All platforms	All printers
Syntax	`public string MacAddress {get;}`	
Description	This property gets the MAC address from the device information found by `StartDiscoveryDevice`.	

IpAddress — Get IP address

Target	All platforms	All printers
Syntax	`public string IpAddress {get;}`	
Description	This property gets the IP address from the device information found by `StartDiscoveryDevice`.	

IsBonded — Get paring state

Target	All platforms	All printers

Syntax

```
public bool IsBonded {get;}
```

Description

This property gets the paring state from the device information found by `StartDiscoveryDevice`.

In iOS, this property always returns true.

DevicePath — Get device path

Target	Android	All printers

Syntax

```
public string DevicePath {get;}
```

Description

This property gets the USB device file path string from the device information found by `StartDiscoveryDevice`.

4.4.4 `PrinterIdentifier` Class

`PrinterIdentifier` class stores the printer ID of the printer retrieved by `GetPrinterResponse`.

(1) Property List

Properties provided by **`PrinterIdentifier`** class are shown in the following table.

Name	Access	Description
`PrinterModelID`	R	Get printer model ID
`TypeID`	R	Get type ID
`RomVersionID`	R	Get ROM version ID
`MainFirmwareVersion`	R	Get firmware version (main)
`Manufacturer`	R	Get manufacturer
`ModelName`	R	Get model name
`SerialNumber`	R	Get serial number
`BootFirmwareVersion`	R	Get firmware version (boot)
`BootFirmwareChecksum`	R	Get firmware checksum (boot)
`MainFirmwareChecksum`	R	Get firmware checksum (main)
`FirmwareChecksum`	R	Get firmware checksum (main + boot)

(2) Property Details

`PrinterModelID` — Get printer model ID

Target	All platforms	All printers
Syntax	`public byte PrinterModelID()`	
Description	This property gets the printer model ID from the printer ID of the printer retrieved by `GetPrinterResponse`.	

`TypeID` — Get type ID

Target	All platforms	All printers
Syntax	`public byte TypeID()`	
Description	This property gets the type ID from the printer ID of the printer retrieved by `GetPrinterResponse`.	

`RomVersionID` — Get ROM version ID

Target	All platforms	All printers
Syntax	`public byte RomVersionID()`	

Description This property gets the ROM version ID from the printer ID of the printer retrieved by `GetPrinterResponse`.

MainFirmwareVersion — Get firmware version (main)

Target All platforms All printers

Syntax `public String MainFirmwareVersion()`

Description This property gets the firmware version (main) as a string from the printer ID of the printer retrieved by `GetPrinterResponse`.

Manufacturer — Get manufacturer

Target All platforms All printers

Syntax `public String Manufacturer()`

Description This property gets the manufacturer name as a string from the printer ID of the printer retrieved by `GetPrinterResponse`.

ModelName — Get model name

Target All platforms All printers

Syntax `public String ModelName()`

Description This property gets the model name as a string from the printer ID of the printer retrieved by `GetPrinterResponse`.

SerialNumber — Get serial number

Target All platforms All printers

Syntax `public String SerialNumber()`

Description This property gets the serial number as a string from the printer ID of the printer retrieved by `GetPrinterResponse`.

BootFirmwareVersion — Get firmware version (boot)

Target All platforms All printers

Syntax `public String BootFirmwareVersion()`

Description This property gets the firmware version (boot) as a string from the printer ID of the printer retrieved by `GetPrinterResponse`.

BootFirmwareChecksum — Get firmware checksum (boot)

Target	All platforms	All printers

Syntax

```
public byte[] BootFirmwareChecksum()
```

Description

This property gets the firmware checksum (boot) from the printer ID of the printer retrieved by `GetPrinterResponse`.

MainFirmwareChecksum — Get firmware checksum (main)

Target	All platforms	All printers

Syntax

```
public byte[] MainFirmwareChecksum()
```

Description

This property gets the firmware checksum (main) from the printer ID of the printer retrieved by `GetPrinterResponse`.

FirmwareChecksum — Get firmware checksum (main + boot)

Target	All platforms	All printers

Syntax

```
public byte[] FirmwareChecksum()
```

Description

This property gets the firmware checksum (main + boot) from the printer ID of the printer retrieved by `GetPrinterResponse`.

4.4.5 `PrinterException` Class

(1) Method List

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

Name	Description
`PrinterException`	Constructor

(2) Property List

Properties provided by `PrinterException` class are shown in the following table.

Name	Access	Description
`HResult`	R	Get error code
`Message`	R	Get error message

(3) Constant List

① Error code

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

Constant Name	Description	Value
ERROR_ACCESS_DENIED	Failed to get the handle.[*1]	-1
	An unavailable port was specified.	
	An unsupported method was specified.	
ERROR_SHARING_VIOLATION	An already opened port was specified.	-11
ERROR_PORT_NOT_OPENED	The port is not open.	-12
ERROR_DEVICE_NOT_CONNECTED	There is a problem with connection between the device and printer.	-21
ERROR_OFFLINE	Disconnected state or the printer is offline.	-22
ERROR_EXTERNAL_DEVICE_NOT_CONNECTED	Display is not connected.	-23
ERROR_DEVICE_INITIALIZE_FAILED	Failed to change the printer settings. 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
ERROR_DATA_SIZE_ZERO	0-byte data was specified.	-101
ERROR_OVER_MAX_DATA_SIZE	Maximum data size is exceeded.	-102
ERROR_DATA_SIZE_INVALID	Data size is invalid.	-103
ERROR_ENCODE_FAILED	An error occurred in encoding text data.[*1]	-111
ERROR_TIMEOUT	Send timeout occurred.	-201
	Receive timeout occurred.	
ERROR_FILE_NOT_FOUND	The specified file is not found.	-301

Constant Name	Description	Value
ERROR_FILE_USED	The specified file is in use by another process.	-302
ERROR_FILE_INVALID	The specified file is invalid.	-303
ERROR_LOW_MEMORY	Memory shortage occurred when loading image file.	-311
ERROR_OVER_MAX_IMAGE	Either or both of width and height of image file exceeds the number of printable maximum dots.	-312
ERROR_LOGO_NOT_DEFINED	The logo is not registered.	-313
ERROR_LOW_USER_AREA	Remaining user area is insufficient.	-401
ERROR_LOW_EXTERNAL_RAM	Remaining RAM capacity is insufficient.	-402
ERROR_NOT_REGISTERD	The template is not registered. The image data is not registered. The slide data is not registered. The optional font is not registered. The user-defined character is not registered.	-403
ERROR_NOT_UNREGISTERD	The template is not deleted. The image data is not deleted. The slide data is not deleted. The optional font is not deleted. The user-defined character is not deleted.	-404
ERROR_INVALID_NO	The specified value for the logo ID is invalid.	-501
ERROR_OVER_STYLE_NUM	The number of style registered in the specified file exceeds rated value (64).	-502
ERROR_INVALID_DATA	The specified data is invalid.	-503
ERROR_PAGE_MODE_SPECIFIED	Page mode is specified.	-511
ERROR_PAGE_MODE_NOT_SPECIFIED	Page mode is not specified.	-512
ERROR_LABEL_FILE_NOT_SELECTED	The label file is not selected.	-521
ERROR_GET_LABEL_IMAGE	Failed to create the label image.	-522
ERROR_INVALID_PARAM	The specified parameter is invalid.	-9999

*1: Abnormal processing might have occurred.

(4) Method Details

PrinterException — Constructor

This is the exception class that is thrown when API of PrinterManager class is called.

Syntax `public PrinterException(int code, string message);`

(5) Property Details

HResult — Get error code

Target	All platforms	All printers

Syntax `public int HResult {get;}`

Description This property gets the error code in an int type.

Return value See "4.4.5(3) Constant List".

Message — Get error message

Target	All platforms	All printers

Syntax `public string Message {get;}`

Description This property gets a string that supplements the contents of the HResult.

4.4.6 `SmartLabelManager` Class

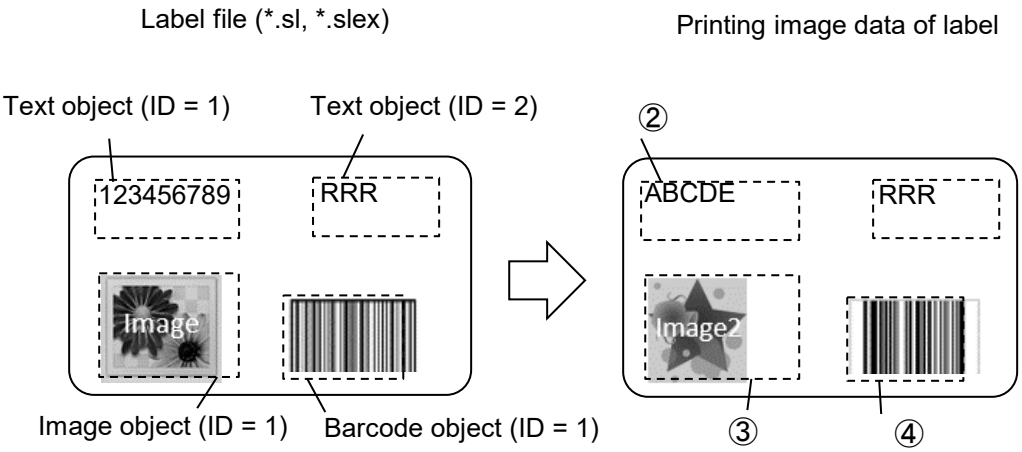
`SmartLabelManager` class provides the function to convert the label file (*.sl, *.slex) created using the app into the printable data from the printer.

(1) Method List

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

Name	Description
`SmartLabelManager`	Constructor
`SelectSmartLabelFile`	Specify label file
`ReplaceSmartLabelTextData`	Replace text data of label
`ReplaceSmartLabelImageData`	Replace image data of label
`ReplaceSmartLabelBarcodeData`	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);
```

(2) Method Details

SmartLabelManager — Constractor

Constructor for `SII.SDK.Maui.Printer.SmartLabelManager` class.

Target	All platforms	SLP720RT, SLP721RT, MP-B30L, MP-B21L
Syntax	`public SmartLabelManager(Context context);`	
Parameter	`context`	Context For iOS: Specify null. For Android Specify application context to call this method.

SelectSmartLabelFile — Specify label file

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

Target	All platforms	SLP720RT, SLP721RT, MP-B30L, MP-B21L
Syntax	`public void SelectSmartLabelFile(string filePath);`	
Parameter	`filePath`	File path of label file (*.sl, *.slex) to use (*.sl is deprecated.) The formats that can be entered are described below. For iOS: Name of label file (*.sl, *.slex) to use Specify the label file (*.sl, *.slex) created using the app. For Android: •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 "3.4 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.
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**4.4.5 `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).

Target　All platforms　SLP720RT, SLP721RT, MP-B30L, MP-B21L

Syntax
```
public void ReplaceSmartLabelTextData(int mapID, string text);
```

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

Error　`PrinterException`
`PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `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).

Target　All platforms　SLP720RT, SLP721RT, MP-B30L, MP-B21L

Syntax
```
public void ReplaceSmartLabelImageData(int mapID,object image);
```

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

`image`　Image data to replace
For iOS:
Specify image data conforming to the iOS class "UIKit.UIImage".
For Android:
Specify image data conforming to the Android class "Android.graphics.Bitmap".

Error　`PrinterException`
`PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details of the error.

ReplaceSmartLabelBarcodeData　Replace barcode data of label

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

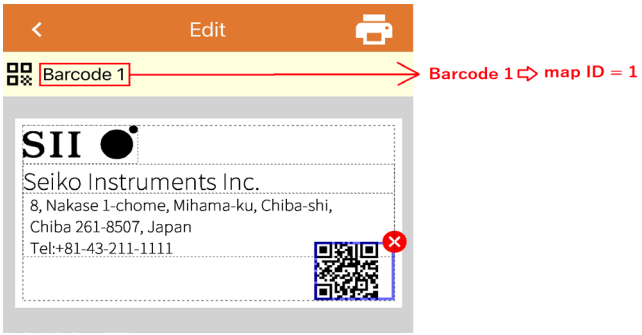
Target　All platforms　SLP720RT, SLP721RT, MP-B30L, MP-B21L

Syntax

```
public void ReplaceSmartLabelBarcodeData(int mapID, string text);
```

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.

Error　`PrinterException`
`PrinterException` is thrown when an error occurs while this method is being called.
See "**4.4.5 `PrinterException` Class**" for details of the error.

Chapter 5
Library Functions of Display

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

5.1 API Reference

This library includes the following classes.

✓: Supported, - : Not supported

Name	Description	Support
`PrinterManager`	Provides the APIs used for communication with the device. See "**5.1.2 `PrinterManager` Class**". See "**5.1.1 Enumerated Constant List**" for enumerated constants necessary for APIs provided in `PrinterManager` class.	✓
`DeviceInfo`	Stores the device information found by `StartDiscoveryDevice`. See "**5.1.3 `DeviceInfo` Class**". See "**5.1.1 Enumerated Constant List**" for enumerated constants necessary for APIs provided in **`DeviceInfo`** class.	✓
`PrinterIdentifier`	Stores the printer ID of the printer retrieved by `GetPrinterResponse`. See "**5.1.4 `PrinterIdentifier` Class**". See "**5.1.1 Enumerated Constant List**" for enumerated constants necessary for APIs provided in **`PrinterIdentifier`** class.	✓
`PrinterException`	Exception class that is thrown at API call. See "**5.1.5 `PrinterException` Class**".	✓

(NOTE) See "Target Devices" in "INTRODUCTION" and confirm the device combination for each platform.

The following is the description of the described contents in each class of the method and properties.

Method or Property　Method name or property name

The outline of the method or property is described.

Target　The supported platform and device are described.

- Platform

	Supported All Platforms	Supported Different Device Depending on Platform
Description in "Target"	All platforms	iOS
		Android

- Device

	Supported Multiple Devices		Supported Individual Device
Description in "Target"	All devices	All printers	SLP720RT
			SLP721RT
			RP-F10
			RP-G10
			RP-E10
			RP-D10
			MP-B30
			MP-B30L
			MP-B21L
			MP-B20
			MP-A40
		-	DSP-A01 via the printer
			DSP-A01 alone

Syntax　Describe syntax.

Parameter　Parameter and the description of the parameter are described.

Valid range　The valid range is described when there is a valid range.

Error　The description of when an error occurred is described.

Initial value　The initial value is described when there is an initial value.

Description　Description of the method is described.

Return value　The return value is described when there is a return value.

5.1.1 Enumerated Constant List

Constants of enumerated type are described in the following.

① Device model (`DeviceModel`)

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

Constant Name	Description
DEVICE_MODEL_RP_F10[*1]	RP-F10
DEVICE_MODEL_DSP_A01	DSP-A01

*1: This is available only when using via the printer.

② Port type (`PortType`)

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

Constant Name	Description
PORT_TYPE_BLUETOOTH[*1]	Bluetooth
PORT_TYPE_USB	USB
PORT_TYPE_TCP[*1]	TCP/IP

*1: This is available for only when using via the printer on Android.

③ Printer response type (`PrinterResponseId`)

Constants of enumerated type used for getting various responses from the printer are shown in the following table.

Constant Name	Description
PRINTER_RESPONSE_REQUEST	Execution response request
PRINTER_RESPONSE_USER_AREA	Send remaining capacity of user area
PRINTER_RESPONSE_ARRANGE_USER_AREA	Send remaining capacity of user area after defragment
PRINTER_RESPONSE_NV_GRAPHICS	Send NV graphics memory capacity
PRINTER_RESPONSE_KEY_CODE	Send key code list of defined NV graphics
PRINTER_RESPONSE_MAINTENANCE_COUNTER	Send maintenance counter
PRINTER_RESPONSE_PRINTER_IDENTIFIER	Send printer ID

④ Device type (`DeviceType`)

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

Constant Name	Description
TYPE_PRINTER	Printer
TYPE_DISPLAY	Display

⑤ Memory area (`MemoryArea`)

Constants of enumerated type used for operating memory area are shown in the following table.

Constant Name	Description
`MEMORY_DISPLAY_USERMEMORY`	User area

⑥ Alignment (`PrintAlignment`)

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

Constant Name	Description
`ALIGNMENT_LEFT`	Aligned left
`ALIGNMENT_CENTER`	Centered
`ALIGNMENT_RIGHT`	Aligned right

⑦ Bold character (`CharacterBold`)

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

Constant Name	Description
`BOLD_CANCEL`	Cancel bold character
`BOLD`	Specify bold character

⑧ Underline (`CharacterUnderline`)

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

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

⑨ Character scale (`CharacterScale`)

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

Constant Name	Description
`VARTICAL_1_HORIZONTAL_1`	Height × 1 and width × 1
`VARTICAL_1_HORIZONTAL_2`	Height × 1 and width × 2
`VARTICAL_1_HORIZONTAL_3`	Height × 1 and width × 3
`VARTICAL_1_HORIZONTAL_4`	Height × 1 and width × 4
`VARTICAL_2_HORIZONTAL_1`	Height × 2 and width × 1

Constant Name	Description
VARTICAL_2_HORIZONTAL_2	Height × 2 and width × 2
VARTICAL_2_HORIZONTAL_3	Height × 2 and width × 3
VARTICAL_2_HORIZONTAL_4	Height × 2 and width × 4
VARTICAL_3_HORIZONTAL_1	Height × 3 and width × 1
VARTICAL_3_HORIZONTAL_2	Height × 3 and width × 2
VARTICAL_3_HORIZONTAL_3	Height × 3 and width × 3
VARTICAL_3_HORIZONTAL_4	Height × 3 and width × 4
VARTICAL_4_HORIZONTAL_1	Height × 4 and width × 1
VARTICAL_4_HORIZONTAL_2	Height × 4 and width × 2
VARTICAL_4_HORIZONTAL_3	Height × 4 and width × 3
VARTICAL_4_HORIZONTAL_4	Height × 4 and width × 4

⑩ Character font (`CharacterFont`)

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

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

⑪ Registered font (`RegisterdFont`)

Constants of enumerated type used for registered fonts are shown in the following table

Constant Name	Description
FONT_STANDARD	Standard font
FONT_OPTION	Optional font

⑫ Module size (`ModuleSize`)

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

Constant Name	Description	Method to Use
QR_MODULE_SIZE_2	2 dots	`SetTemplateQRCodeData`
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	

Constant Name	Description	Method to Use
QR_MODULE_SIZE_9	9 dots	SetTemplateQRCodeData
QR_MODULE_SIZE_10	10 dots	
QR_MODULE_SIZE_11	11 dots	

⑬ 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
QR_ERROR_CORRECTION_L	Error correction level L	SetTemplateQRCodeData
QR_ERROR_CORRECTION_M	Error correction level M	
QR_ERROR_CORRECTION_H	Error correction level H	
QR_ERROR_CORRECTION_Q	Error correction level Q	

⑭ QR data mode (QrDataMode)

Constants of enumerated type used for QR data mode are shown in the following table.

Constant Name	Description
QRDATAMODE_NUMERIC	Numeric mode
QRDATAMODE_ALPHANUMERIC	Alphanumeric mode
QRDATAMODE_8BITBYTE	8-bit byte mode
QRDATAMODE_KANJI	Kanji mode
QRDATAMODE_MIXTURE	Mixed mode

⑮ QR quiet zone (QrQuietZone)

Constants of enumerated type used for QR quiet zone are shown in the following table.

Constant Name	Description
QRQUIETZONE_EXIST	Set QR quiet zone on
QRQUIETZONE_NONE	Set QR quiet zone off

⑯ Macro registration processing (`MacroRegistrationFunction`)

Constants of enumerated type used for macro registration processing are shown in the following table.

Constant Name	Description
MACRO_REGISTRATION_CLEAR	Cancel macro registration processing
MACRO_REGISTRATION_START	Start macro registration processing
MACRO_REGISTRATION_REGIST	Finish macro registration and macro registration processing

⑰ Display response type (`DisplayResponseId`)

Constants used for display response type are shown in the following table.

Constant Name	Description
DISPLAY_RESPONSE_REQUEST	Execution response request
DISPLAY_RESPONSE_USER_AREA	Send remaining capacity of user area
DISPLAY_RESPONSE_TEMPLATE_ID_LIST	Send template ID
DISPLAY_RESPONSE_IMAGE_ID_LIST	Send image ID
DISPLAY_RESPONSE_SLIDE_ID_LIST	Send slide ID
DISPLAY_RESPONSE_TEMPLATE_LABEL	Send template name
DISPLAY_RESPONSE_IMAGE_LABE	Send image name
DISPLAY_RESPONSE_SLIDE_LABEL	Send slide name

⑱ International character set (`InternationalCharacter`)

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

Constant Name	Description
COUNTRY_USA	USA
COUNTRY_FRANCE	France
COUNTRY_GERMANY	Germany
COUNTRY_ENGLAND	England
COUNTRY_DENMARK_1	Denmark I
COUNTRY_SWEDEN	Sweden
COUNTRY_ITALY	Italy
COUNTRY_SPAIN	Spain I
COUNTRY_JAPAN	Japan
COUNTRY_NORWAY	Norway
COUNTRY_DENMARK_2	Denmark II
COUNTRY_SPAIN_2	Spain II
COUNTRY_LATIN_AMERICA	Latin America
COUNTRY_ARABIA	Arabia

⑲ Codepage (CodePage)

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

Constant Name	Description
CODE_PAGE_437	USA, Standard Europe (Code Page437)
CODE_PAGE_KATAKANA	Katakana
CODE_PAGE_850	Multilingual (Code Page850)
CODE_PAGE_860	Portuguese (Code Page860)
CODE_PAGE_863	Canadian-French (Code page863)
CODE_PAGE_865	Nordic (Code Page865)
CODE_PAGE_857[*1]	Turkish (Code Page857)
CODE_PAGE_737	Greek (Code Page737)
CODE_PAGE_1252	Latin (Code Page1252)
CODE_PAGE_866	Russian (Code Page866)
CODE_PAGE_852	Eastern Europe (Code Page852)
CODE_PAGE_858	Euro (Code Page858)
CODE_PAGE_855	Cyrillic (Code Page855)
CODE_PAGE_864[*1][*2]	Arabic (Code Page864)
CODE_PAGE_1250	Central European (Code Page1250)
CODE_PAGE_1251	Cyrillic (Code Page1251)
CODE_PAGE_1253[*3]	Greek (Code Page1253)
CODE_PAGE_1254	Turkish (Code Page1254)

*1: 20ACh of the Unicode cannot be displayed.
*2: Font B cannot be printed or displayed.
*3: 00AAh of the Unicode cannot be displayed.

5.1.2 `PrinterManager` Class

(1) Method List

Methods provided by the `PrinterManager` class are shown in the following table.
See "5.1.2(3) Method Details" for details of the methods.

✓: Supported, - : Not supported

Name	Description	Support
`PrinterManager`	Constructor	✓
`Connect`	Start communicating with device	✓
`Disconnect`	Stop communicating with device	✓
`GetPrinterResponse`	Get various responses from printer	✓
`StartDiscoveryDevice`	Start device search	✓
`CancelDiscoveryDevice`	Cancel device search	✓
`GetFoundDevice`	Get found device information list	✓
`Defragment`	Optimize memory area	✓
`InitializeMemoryArea`	Initialize memory area	✓
`ShowTemplate`	Display template	✓
`ShowSlide`	Display slide	✓
`EnterStandbyMode`	Display standby	✓
`ExecuteMacro`	Execute macro	✓
`TurnOnScreen`	Turn on/off screen	✓
`SelectTemplate`	Select template	✓
`SetTemplateImageData`	Set image data	✓
`SelectTemplateTextObject`	Select text element	✓
`SetTemplateTextAlignment`	Alignment of text data	✓
`SetTemplateTextLeftMargin`	Set left margin of text data	✓
`SetTemplateTextLineSpacing`	Set line spacing of text data	✓
`SetTemplateTextBold`	Set bold character of text data	✓
`SetTemplateTextUnderline`	Set underline of text data	✓
`SetTemplateTextSize`	Set character size of text data	✓
`SetTemplateTextFont`	Set character font of text data	✓
`SetTemplateTextRegisteredFont`	Set registered font of text data	✓
`SetTemplateTextRightSpacing`	Set right space of text data	✓
`SetTemplateTextColor`	Set character color of text data	✓
`SetTemplateTextData`	Input text data	✓
`SetTemplateBarcodeData`	Input barcode data	✓
`SetTemplateQRCodeData`	Input QR Code data	✓
`RegisterTemplate`	Register template	✓
`UnregisterTemplate`	Delete template	✓
`RegisterImageData`	Register image data	✓
`UnregisterImageData`	Delete image data	✓
`RegisterSlideData`	Register slide data	✓

Name	Description	Support
UnregisterSlideData	Delete slide data	✓
RegisterUserDefinedCharacter	Register user-defined character	✓
UnregisterUserDefinedCharacter	Delete user-defined character	✓
RegisterOptionFont	Register optional font	✓
UnregisterOptionFont	Delete optional font	✓
ControlMacroRegistration	Start/Finish of macro registration	✓
GetDisplayResponse	Get various responses from the display	✓
SendBinary	Send binary data	✓
SendDataFile	Send specified file	✓

(2) Property list

Properties provided by PrinterManager class are shown in the following table.

✓: Supported, - : Not supported

Name	Access	Description	Support
SendTimeout	R/W	Get/Set send timeout period	✓
ReceiveTimeout	R/W	Get/Set receive timeout period	✓
InternationalCharacter	R/W	Get/Set international character set	✓
CodePage	R/W	Get/Set codepage	✓
DeviceModel	R	Get device model	✓
PortType	R	Get connecting port type	✓
IsConnect	R	Verify connection state with printer	✓
SocketKeepingTime	R/W	Get/Set socket keeping time	✓
Version	R	Get SDK version	✓

(3) Method Details

PrinterManager — Constructor

Constructor for SII.SDK.Maui.Printer.PrinterManager class.

Target	All platforms	All devices
Syntax	`public PrinterManager(object context);`	
Parameter	`context`	Context For iOS: Specify null. For Android: Specify application context to call this method.

Connect — Start communicating with device

Starts communicating with the device.

Target	All platforms	All devices
Syntax	(a) `public void Connect(DeviceModel deviceModel, DeviceInfo deviceInfo);`	
	(b) `public void Connect(DeviceModel deviceModel, PortType portType, string param);`	
Parameter	`deviceModel`	Device model See "5.1.1① Device model (`DeviceModel`)" for available constants.
	`deviceInfo`	Device information Specify the object of `DeviceInfo` class. The object of DeviceInfo class can be retrieved from `StartDiscoveryDevice` and `GetFoundDevice`.
	`portType`	Port type See "5.1.1② Port type (`PortType`)" for available constants.
	`param`	Depends on the setting of `portType`. See the "Target Devices" described in "INTRODUCTION" for the corresponding interface. ・`portType` is **`PORT_TYPE_BLUETOOTH`** Specify the Bluetooth address or Bluetooth device name (Bluetooth Accessary). Example: "00:11:22:AA:BB:CC", "RP-F10" ・`portType` is **`PORT_TYPE_USB`** For iOS: Specify the printer name. Example: "RP-F10" For Android: Specify null. ・`portType` is **`PORT_TYPE_TCP`** Specify the IP address of the printer. Example: "192.168.0.190", "192.168.0.1"

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.

Description Call this method before using other `PrinterManager` class methods.
In order to make this library work properly, this method may change the printer settings when connecting.

For Bluetooth connection:
Connect to the paired Bluetooth address or Bluetooth device (Bluetooth Accessory) specified by `param` only when using the display via the printer.
In iOS, communication with the printer paired with an iOS device starts through the Bluetooth connection.
In Android, connect the printer specified by `deviceModel` to the Bluetooth address specified by `param`.

For USB connection:
In iOS, communication with a printer connected with the iOS device through the USB cable starts through the USB connection.
In Android, connect to the printer specified by `deviceModel` when using the display via the printer. Connect to the display specified by `deviceModel` when using the display alone.

For TCP/IP connection:
Communication with the printer connected to the same network as the device starts through TCP/IP connection only when using the display via the printer.
In both iOS and Android, connection is made to the IP address specified by `param`.

- **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 the socket keeping time set by `SocketKeepingTime`.
  Then the new socket is created immediately and used for the next connection.
  If the printer is receiving a connection request from another host on the same network at the time of discarding the socket, the printer establishes communication with that host, so the reconnection may fail.

Note This method does not support a concurrent connection from multiple apps to one printer.

`Disconnect` Stop communicating with device

Finishes communicating with the device.

Target All platforms All devices

Syntax
```
public void Disconnect();
```

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.

Note It is recommended to get execution response by **`PRINTER_RESPONSE_REQUEST`** of `GetPrinterResponse` before executing this method. If not, the communication is disconnected before completing to be sent the data of the iOS or Android device to the device, and part of the data may be lost. If you do not execute `GetPrinterResponse` or `GetDisplayResponse` in your program, please fully evaluate your program to confirm no problems arise.

GetPrinterResponse — Get various responses from printer

Gets response data from the printer.

Target
- iOS: All printers, DSP-A01 via the printer
- Android: All devices

Syntax

```
public void GetPrinterResponse<T>(PrinterResponseId responseId,
                                  object param,
                                  out T[] response);
```

Parameter

- `responseId`: Response type constant
  See "5.1.1③ Printer response type (`PrinterResponseId`)" for available constants.
- `param`: Command parameter
  The value to be specified varies with the response type constant.
  See the following table for description of the value to be specified.
- `response`: Buffer for storing the retrieved response data
  The buffer type varies with the response type constant.
  See the following table for the buffer type.

Response Type Constant		
	Parameter	**Description**
PRINTER_RESPONSE_REQUEST (Execution response request)		
	`param`	Specify 0 to 15 (00h to 0Fh) in an int type.
	`response`	Specify an int type array. When the response is retrieved successfully, the response code of the execution response request is stored with 128 to 143 (80h to 8Fh).
PRINTER_RESPONSE_USER_AREA (Send remaining capacity of user area)		
	`param`	Specify null.
	`response`	Specify an int type array. When the response is retrieved successfully, the remaining capacity (unit: byte) of the user area is stored numerically.
PRINTER_RESPONSE_ARRANGE_USER_AREA (Send remaining capacity of user area after defragment)		
	`param`	Specify null.
	`response`	Specify an int type array. When the response is retrieved successfully, the remaining capacity (unit: byte) of the user area after defragment is stored numerically.
PRINTER_RESPONSE_NV_GRAPHICS (Send NV graphics memory capacity)		
	`param`	Specify null.
	`response`	Specify an int type array. When the response is retrieved successfully, the NV graphics memory capacity (unit: byte) is stored numerically.

Response Type Constant		
	Parameter	**Description**
PRINTER_RESPONSE_KEY_CODE (Send key code list of defined NV graphics)		
	`param`	Specify null.
	`response`	Specify a string type array. When the response is retrieved successfully, the key code of NV graphics is stored in a character string.
PRINTER_RESPONSE_MAINTENANCE_COUNTER (Send maintenance counter)		
	`param`	Specify null.
	`response`	Specify an int type array. When the response is retrieved successfully, the maintenance counter is stored as shown below. buf[0] : Number of paper feed lines (unit: 100 dot-line) buf[1] : Number of thermal head activation times (unit: 100 dot-line) buf[2] : Number of autocutter drive times*1 buf[3] : Drive time of printer (unit: minute) buf[4] : Number of paper feed lines (integrated value) (unit: 100 dot-line) buf[5] : Number of thermal head activation times (integrated value) (unit: 100 dot-line) buf[6] : Number of autocutter drive times (integrated value)*1 buf[7] : Drive time of printer (integrated value) (unit: minute)
PRINTER_RESPONSE_PRINTER_IDENTIFIER (Send printer ID)		
	`param`	Specify null.
	`response`	When the response is retrieved successfully, a PrinterIdentifier class object is stored in the first element of buf. Example: buf.size() = 1, buf.get[0] = "PrinterIdentifier class object" See "**5.1.4 `PrinterIdentifier` Class**" for the PrinterIdentifier class object.

*1: Not supported by MP-B30, MP-B30L, MP-B21L, MP-B20, and MP-A40. Always returns 0.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

`StartDiscoveryDevice` Start device search

Searches the device.

Target
- iOS: All printers, DSP-A01 via the printer
- Android: All devices

Syntax
```
public void StartDiscoveryDevice(DeviceType deviceType,
                                 PortType portType,
                                 int timeout,
                                 Action completion);
```

Parameter
- `deviceType`: Device Type
  See "5.1.1④ Device type (`DeviceType`)" for available constants.
  For iOS, only **`TYPE_PRINTER`** is available.
- `portType`: Port type
  See "5.1.1② Port type (`PortType`)" for available constants.

	`timeout`	Search timeout period (millisecond: ms) Sets the timeout period per search. 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. ● For **`PORT_TYPE_TCP`** Each time the local broadcast packet is sent, this method waits for a response from the printer until the period specified by `timeout` elapses. ● For **`PORT_TYPE_BLUETOOTH`** When the device is being searched, this method searches the device until the period specified by `timeout` elapses. When the device search is completed, the search is completed without waiting for the period of `timeout`. For iOS, specifying of `timeout` is ignored. ● For **`PORT_TYPE_USB`** Specifying of `timeout` is ignored.
	`completion`	The block executed when completing the device search

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

Description This method searches for the device.
The device information of the discovered device can be retrieved by `GetFoundDevice`.

`CancelDiscoveryDevice` Cancel device search

Cancels `StartDiscoveryDevice` under execution.

Target All platforms All printers, DSP-A01 via the printer

Syntax
```
public void CancelDiscoveryDevice();
```

Description The cancellation of the search is notified as an event to the block set to `completion` of `StartDiscoveryDevice`.

`GetFoundDevice` Get found device information list

Gets the device information list found by `StartDiscoveryDevice` from the destination `DeviceInfo` class.

Target	iOS	All printers, DSP-A01 via the printer
	Android	All devices

Syntax
```
public DeviceInfo[] GetFoundDevice();
```

Return value See "5.1.3 `DeviceInfo` Class".

Defragment — Optimize memory area

Optimizes the memory area.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void Defragment(MemoryArea memoryArea);`	
Parameter	`memoryArea`	Memory area See "5.1.1⑤ Memory area (`MemoryArea`)" for available constants.
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.	
Description	It may take several minutes for optimization. Do not turn the printer power off during optimization. The display is changed to Standby mode when this method is executed. A selecting template is deselected.	

InitializeMemoryArea — Initialize memory area

Initializes the memory area.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void InitializeMemoryArea(MemoryArea memoryArea);`	
Parameter	`memoryArea`	Memory area See "5.1.1⑤ Memory area (`MemoryArea`)" for available constants.
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.	
Description	It may take several minutes for initialization. Do not turn the printer power off during the initialization. The display is changed to Standby mode when this method is executed. A selecting template is deselected.	
Note	Registered data in following methods is deleted when the memory area is initialized after specifying **`MEMORY_DISPLAY_USERMEMORY.`** ・`RegisterTemplate` ・`RegisterImageData` ・`ControlMacroRegistration` ・`RegisterUserDefinedCharacter` ・`RegisterOptionFont` In addition, part of data which is registered at the factory to use for the system is deleted either. Therefore the display becomes impossible to change to Guide mode when an error occurs in the printer. The used memory can be reused after executing `Defragment`.	

ShowTemplate Display template

Displays the template on the screen.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax

```
public void ShowTemplate(int time_ms);
```

Parameter

`time_ms` Display time (ms: millisecond)
Specify display time on the screen with `time_ms` (ms)
The valid range is 0 to 25500.
When the value exceeds 0 and less than 100 is specified, the time is set to 100 ms.
When the value exceeding 25500 is specified, the time is set to 25500 ms.

For macro registration:
When 0 is specified with `time_ms`, the template is shown continuously.
When other than 0 is specified with `time_ms`, a next template is shown after the display time is elapsed.

For other than macro registration:
When 0 is specified with `time_ms`, the template is shown continuously.
When other than 0 is specified with `time_ms`, the template returns to a previous template after the display time is elapsed.
In case of the previous screen has been updated with the display time other than 0, the screen is traced back to the template which was updated with the display time 0.

Error

`PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description

Updates the screen, and displays data being specified by the following methods.

- `SelectTemplate`
- `SetTemplateImageData`
- `SelectTemplateTextObject`
- `SetTemplateTextAlignment`
- `SetTemplateTextLeftMargin`
- `SetTemplateTextLineSpacing`
- `SetTemplateTextBold`
- `SetTemplateTextUnderline`
- `SetTemplateTextSize`
- `SetTemplateTextFont`
- `SetTemplateTextRightSpacing`
- `SetTemplateTextColor`
- `SetTemplateTextData`
- `SetTemplateBarcodeData`
- `SetTemplateQrCodeData`

ShowSlide — Display slide

Displays the slide on the screen.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void ShowSlide(int slideID, int time_ms);`	
Parameter	`slideID`	Slide ID Specify the ID of the slide data which is registered at `RegisterSlideData`. The valid range is 0 to 91. This method is ignored when slide data is not registered in the specified ID.
	`time_ms`	Display time (ms: millisecond) Specify display time on the screen with `time_ms` (ms) The valid range is 0 to 25500. When the value exceeds 0 and less than 100 is specified, the display time is set to 100 ms. When the value exceeds 25500 is specified, the display time is set to 25500 ms. For macro registration: When 0 is specified with `time_ms`, the slide is shown continuously. When other than 0 is specified with `time_ms`, a next slide is shown after the display time is elapsed. For other than macro registration: When 0 is specified with `time_ms`, the slide is shown continuously. When other than 0 is specified with `time_ms`, the slide returns to a previous slide after the display time is elapsed. In case of the previous screen has been updated with the display time other than 0, the screen is traced back to the slide which was updated with the display time 0.
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.	
Description	A selecting template is deselected.	

EnterStandbyMode — Display standby

Changes the display to Standby mode.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void EnterStandbyMode();`	
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.	
Description	This method is ignored during Standby mode or Guide mode is being displayed. A selecting template is deselected.	

ExecuteMacro — Execute macro

Executes the macro.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void ExecuteMacro(int macroID, int repeatCount);`	
Parameter	`macroID`	Macro ID Specify the macro ID which is registered at `ControlMacroRegistration`. The valid range is 0 to 127. This method is ignored when the macro is not registered in the specified ID.
	`repeatCount`	The number of execution times Specify times to execute the macro. The valid range is 0 to 255. Continues the repeating when 0 is specified.
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.	
Description	A selecting template is deselected.	

TurnOnScreen — Turn on/off screen

Sets the screen backlight on/off.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void TurnOnScreen(bool isOn);`	
Parameter	`isOn`	Screen status Specify the screen status from following. true: backlight on false: backlight off
Error	`PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.	

SelectTemplate — Select template

Selects the template to show on the display.
In syntax (a), the template and the slide data used as the template background can be selected.
In syntax (b), the template can be selected.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	(a) `public void SelectTemplate(int templateID, int slideID);`	
	(b) `public void SelectTemplate(int templateID);`	

Parameter	`templateID`	Template ID Specify the ID of template to select. The valid range is 0 to 127. This method is ignored when the template is not registered in the specified ID.
	`slideID`	Slide ID Specify the ID of slide data to use for the background of the template. The valid range is 0 to 91. This method is ignored when slide data is not registered in the specified ID.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description The data on the template is cleared when this method is executed.
The selected template is displayed to the screen when `ShowTemplate` is executed.

The selecting template is deselected when `ShowSlide`, `EnterStandbyMode`, or `ExecuteMacro` is executed.
The selecting template is deselected when the specified display time is executed at `ShowTemplate`.

Use following templates depends on the values of `CodePage` when characters other than 20h to 7Eh of ASCII character are input by `SetTemplateTextData`.

・When `CodePage` is **`CODE_PAGE_KATAKANA`**:
Use the template which encoding specifying is Shift_JIS.
・When `CodePage` is other than **`CODE_PAGE_KATAKANA`**:
Use the template which encoding specifying is ISO-2022-JP.

`SetTemplateImageData` Set image data

Sets image data to show on a selecting template.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void SetTemplateImageData(int mapID, int imageID);
```

Parameter	`mapID`	Map ID The valid range is 0 to 63. This method is ignored when a specified map ID is not defined in the template.
	`imageID`	Image ID Specify the ID of image data which was registered at `RegisterImageData`. The valid range is 0 to 63. This method is ignored when image data is not registered in the specified ID.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description After specifying the map ID of the selecting template by this method, specify the image ID to map.
The specified image data is displayed to the screen when `ShowTemplate` is executed.

This method setting is cleared under the following conditions.
・When SelectTemplate is executed.
・When other than 0 is specified at time_ms of ShowTemplate, and the specified display time has elapsed.
・When ShowtTemplate registered in ExecuteMacro is executed.

This method is ignored when a template is not selected.

SelectTemplateTextObject — Select text element

Selects the text element to start editing.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void SelectTemplateTextObject(int mapID);
```

Parameter	mapID	Map ID The valid range is 0 to 63. This method is ignored when a specified map ID is not defined in the template.

Error — PrinterException is thrown when an error occurs while this method is being called. See "**5.1.5 PrinterException Class**" for details on the error.

Description — After specifying a map ID of the selecting template by this method, start to edit characters.

When a scroll is set with the text element of the specified map ID and this method is executed after ShowTemplate, the scroll is executed.
When a scroll is not set with the text element of the set map ID and this method is executed after ShowTemplate, input text data is discarded.

This method setting is cleared under the following conditions.
・When SelectTemplate is executed.
・When ShowtTemplate is executed.

This method is ignored when a template is not selected.

SetTemplateTextAlignment — Alignment of text data

Sets alignment to text data shown on the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void SetTemplateTextAlignment(PrintAlignment alignment);
```

Parameter	alignment	Alignment See "5.1.1⑥ Alignment (PrintAlignment)" constants.

Error — PrinterException is thrown when an error occurs while this method is being called. See "**5.1.5 PrinterException Class**" for details on the error.

Description — Alignment is valid only as following cases.
・Text data is not entered in the specified map ID at SelectTemplateTextObject.
・Text data entered in the map ID which is specified at SelectTemplateTextObject is registered right after a line feed.

Specify the map ID at SelectTemplateTextObject before executing this method.
Input text data at SetTemplateTextData after executing this method.
The input text data is displayed to the screen when ShowTemplate is executed.

This method setting is cleared under the following conditions.
・When SelectTemplate is executed.
・When other than 0 is specified at time_ms of ShowTemplate, and the specified display time has elapsed.
・When ShowtTemplate registered in ExecuteMacro is executed.

SetTemplateTextLeftMargin — Set left margin of text data

Sets left margin to text data shown on the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void SetTemplateTextLeftMargin(int margin);
```

Parameter	margin	Left margin (pixel: px) The valid range is 0 to 479.

Error
PrinterException is thrown when an error occurs while this method is being called.
See "**5.1.5 PrinterException Class**" for details on the error.

Description
The left margin setting is valid only as following conditions.
・Text data is not entered in a specified map ID at SelectTemplateTextObject.
・Text data entered in a map ID which is specified at SelectTemplateTextObject is registered right after a line feed.

Specify the map ID at SelectTemplateTextObject before executing this method.
Input text data at SelectTemplateTextData after executing this method.
The input text data is displayed to the screen when ShowTemplate is executed.

This method setting is cleared under the following conditions.
・When SelectTemplate is executed.
・When other than 0 is specified at time_ms of ShowTemplate, and the specified display time has elapsed.
・When ShowtTemplate registered in ExecuteMacro is executed.

SetTemplateTextLineSpacing — Set line spacing of text data

Sets line spacing to text data shown on the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void SetTemplateTextLineSpacing(int spacing);
```

Parameter	spacing	Line spacing (pixel: px) The valid range is 0 to 255.

Error
PrinterException is thrown when an error occurs while this method is being called.
See "**5.1.5 PrinterException Class**" for details on the error.

Description
The line spacing setting is valid only as following cases.
・Text data is not entered in a specified map ID at SelectTemplateTextObject.
・Text data entered in a map ID which is specified at SelectTemplateTextObject is registered right after a line feed.

When a scroll is set to a text element of the specified map ID, the line spacing to text data is not reflected.

Specify the map ID at `SelectTemplateTextObject` before executing this method.
Input text data at `SelectTemplateTextData` after executing this method.
The input text data is displayed to the screen when `ShowTemplate` is executed.

This method setting is cleared under the following conditions.
・When `SelectTemplate` is executed.
・When other than 0 is specified at `time_ms` of `ShowTemplate`, and the specified display time has elapsed.
・When `ShowtTemplate` registered in `ExecuteMacro` is executed.

`SetTemplateTextBold` Set bold character of text data

Sets bold characters to text data shown on the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void SetTemplateTextBold(CharacterBold bold);
```

Parameter	`bold`	Bold character See "5.1.1⑦ Bold character (`CharacterBold`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description From text data after this method is executed, the bold characters are applied.
The bold character can be set one by one.

Specify the map ID at `SelectTemplateTextObject` before executing this method.
Input text data at `SelectTemplateTextData` after executing this method.
The input text data is displayed to the screen when `ShowTemplate` is executed.

This method setting is cleared under the following conditions.
・When `SelectTemplate` is executed.
・When other than 0 is specified at `time_ms` of `ShowTemplate`, and the specified display time has elapsed.
・When `ShowtTemplate` registered in `ExecuteMacro` is executed.

`SetTemplateTextUnderline` Set underline of text data

Sets underline to text data shown on the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void SetTemplateTextUnderline(CharacterUnderline underline);
```

Parameter	`underline`	Underline See "5.1.1⑧ Underline (`CharacterUnderline`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description From text data after this method is executed, the underlines are applied.
The underline can be set one by one.

Specify the map ID at SelectTemplateTextObject before executing this method.
Input text data at SelectTemplateTextData after executing this method.
The input text data is displayed to the screen when ShowTemplate is executed.

This method setting is cleared under the following conditions.
・When SelectTemplate is executed.
・When other than 0 is specified at time_ms of ShowTemplate, and the specified display time has elapsed.
・When ShowtTemplate registered in ExecuteMacro is executed.

SetTemplateTextSize — Set character size of text data

Sets character size to text data shown on the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void SetTemplateTextSize(CharacterScale scale);`	
Parameter	`scale`	Character scale See "5.1.1⑨ Character scale (CharacterScale)" for available constants.
Error	PrinterException is thrown when an error occurs while this method is being called. See "**5.1.5 PrinterException Class**" for details on the error.	
Description	From text data after this method is executed, the character sizes are applied. The character size can be set one by one. Specify the map ID at SelectTemplateTextObject before executing this method. Input text data at SelectTemplateTextData after executing this method. The input text data is displayed to the screen when ShowTemplate is executed. This method setting is cleared under the following conditions. ・When SelectTemplate is executed. ・When other than 0 is specified at time_ms of ShowTemplate, and the specified display time has elapsed. ・When ShowtTemplate registered in ExecuteMacro is executed.	

SetTemplateTextFont — Set character font of text data

Sets a character font to text data shown on the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void SetTemplateTextFont(CharacterFont font);`	
Parameter	`font`	Character font See "5.1.1⑩ Character font (CharacterFont)" for available constants.
Error	PrinterException is thrown when an error occurs while this method is being called. See "**5.1.5 PrinterException Class**" for details on the error.	
Description	From text data after this method is executed, the character fonts are applied. The character font can be set one by one.	

Specify the map ID at SelectTemplateTextObject before executing this method.
Input text data at SelectTemplateTextData after executing this method.
The input text data is displayed to the screen when ShowTemplate is executed.

This method setting is cleared under the following conditions.
・When SelectTemplate is executed.
・When other than 0 is specified at time_ms of ShowTemplate, and the specified display time has elapsed.
・When ShowtTemplate registered in ExecuteMacro is executed.

SetTemplateTextRegisteredFont — Set registered font of text data

Sets the registered font used for text data shown on the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void SetTemplateTextRegisteredFont(RegisteredFont font);`	
Parameter	font	Registered font See "5.1.1⑪ Registered font (RegisterdFont)" for available constants. This registered font is ignored when the optional font is not registered.
Error	PrinterException is thrown when an error occurs while this method is being called. See "**5.1.5 PrinterException Class**" for details on the error.	
Description	From text data after this method is executing, the registered fonts are applied. The registered fonts can be set one by one. Input text data by SetTemplateTextData after executing this method. The input text data is displayed to the screen when ShowTemplate is executed. This method setting is cleared under the following conditions. ・When SelectTemplate is executed. ・When other than 0 is specified at time_ms of ShowTemplate, and the specified display time has elapsed. ・When ShowtTemplate registered in ExecuteMacro is executed.	

SetTemplateTextRightSpacing — Set right space of text data

Sets the amount of right space to text data shown on the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void SetTemplateTextRightSpacing(int spacing);`	
Parameter	spacing	The amount of character space (pixel: px) The valid range is 0 to 255.
Error	PrinterException is thrown when an error occurs while this method is being called. See "**5.1.5 PrinterException Class**" for details on the error.	
Description	From text data after this method is executed, the amount of character right space is applied. The amount of right space can be set one by one.	

Specify the map ID at SelectTemplateTextObject before executing this method.
Input text data at SelectTemplateTextData after executing this method.
The input text data is displayed to the screen when ShowTemplate is executed.

This method setting is cleared under the following conditions.
・When SelectTemplate is executed.
・When other than 0 is specified at time_ms of ShowTemplate, and the specified display time has elapsed.
・When ShowtTemplate registered in ExecuteMacro is executed.

SetTemplateTextColor — Set character color of text data

Sets the character color used for text data shown on the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void SetTemplateTextColor(int color);`	
Parameter	color	Character color The valid range is 0 to 0xffffff.
Error	PrinterException is thrown when an error occurs while this method is being called. See "**5.1.5 PrinterException Class**" for details on the error.	
Description	This method can set character colors to text data. The character colors can be set in RGB24 bit color. The set color is displayed in the color converted to 16 bit RGB555. From text data after this method is executed, the character colors are applied. The character color can be set one by one. Specify the map ID at SelectTemplateTextObject before executing this method. Input text data at SelectTemplateTextData after executing this method. The input text data is displayed to the screen when ShowTemplate is executed. This method setting is cleared under the following conditions. ・When SelectTemplate is executed. ・When other than 0 is specified at time_ms of ShowTemplate, and the specified display time has elapsed. ・When ShowtTemplate registered in ExecuteMacro is executed.	

SetTemplateTextData — Input text data

Inputs text data to show on the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone
Syntax	`public void SetTemplateTextData(string text);`	
Parameter	text	Text data to show on the display Data size which is able to be specified it once is 1 to 1020 bytes.
Error	PrinterException is thrown when an error occurs while this method is being called. See "**5.1.5 PrinterException Class**" for details on the error.	
Description	This method encodes input text data into text data which is enable to show on the basis of settings at InternationalCharacter or CodePage, and displays.	

After specifying a map ID of the selecting template by `SelectTemplateTextObject`, input text data by this method.
This method is ignored when a map ID is not specified in `SelectTemplateTextObject`.
The input text data is displayed to the screen when `ShowTemplate` is executed.

The input text data is cleared under the following conditions.
- ・When `SelectTemplate` is executed.
- ・When other than 0 is specified at `time_ms` of `ShowTemplate`, and the specified display time has elapsed.
- ・When `ShowtTemplate` registered in `ExecuteMacro` is executed.

`SetTemplateBarcodeData` Input barcode data

Specifies a map ID of the barcode element on the selecting template, and inputs barcode data.
In syntax (a), the data for the barcode to be displayed can be specified as a character string.
In syntax (b), the data for the barcode to be displayed can be specified as a byte array.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax

(a)
```
public void SetTemplateBarcodeData(int mapID, string text);
```

(b)
```
public void SetTemplateBarcodeData(int mapID, byte[] data);
```

Parameter

`mapID`	Map ID The valid range is 0 to 7. This method is ignored when the specified map ID is not defined in the selecting template.
`text`	Barcode data The characters that can be input are ASCII characters 20h (space) to 7Eh (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z'). The available number of characters is 1 to 150. Barcode data which is not complying with barcode specification is ignored.
`data`	Barcode data The value that can be input is 00h to 7Fh. The available number of data is 1 to 150. Barcode data which is not comply with barcode specification is ignored.

Error

`PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description

After specifying a map ID of the selecting template by this method, input barcode data.
The input barcode data is displayed to the screen when `ShowTemplate` is executed.

The input barcode data is cleared under the following conditions.
- ・When `SelectTemplate` is executed.
- ・When other than 0 is specified at `time_ms` of `ShowTemplate`, and the specified display time has elapsed.
- ・When `ShowtTemplate` registered in `ExecuteMacro` is executed.

This method is ignored when a template is not selected.

SetTemplateQrCodeData — Input QR Code data

Specifies a map ID of a qr element on a selecting template, and inputs QR Code data.
In syntax (a), QR code data can be specified as a character string.
In syntax (b), QR code data can be specified as a character string, and the selected template settings are applied for `moduleSize`, `errorCorrection`, `mode`, and `qrQuietZone`.
In syntax (c), QR code data can be specified as a byte array.
In syntax (d), QR code data can be specified as a byte array, and the selected template settings are applied for `moduleSize`, `errorCorrection`, `mode`, and `qrQuietZone`.

Target

iOS	DSP-A01 via the printer
Android	DSP-A01 via the printer, DSP-A01 alone

Syntax

(a)
```
public void SetTemplateQRCodeData(int mapID,
                                   ModuleSize moduleSize,
                                   ErrorCorrection errorCorrection,
                                   QrDataMode mode,
                                   QrQuietZone qrQuietZone,
                                   string text);
```

(b)
```
public void SetTemplateQRCodeData(int mapID, string text);
```

(c)
```
public void SetTemplateQRCodeData(int mapID,
                                   ModuleSize moduleSize,
                                   ErrorCorrection errorCorrection,
                                   QrDataMode mode,
                                   QrQuietZone qrQuietZone,
                                   byte[] data);
```

(d)
```
public void SetTemplateQRCodeData(int mapID, byte[] data);
```

Parameter

`mapID`	Map ID The valid range is 0 to 7. This method is ignored when the specified map ID is not defined in the selecting template.
`moduleSize`	Module size See "5.1.1⑫ Module size (`ModuleSize`)" for available constants.
`errorCorrection`	Error correction level See "5.1.1⑬ Error correction level (`ErrorCorrection`)" for available constants.
`mode`	Data mode See "5.1.1⑭ QR data mode (`QrDataMode`)" for available constants.
`qrQuietZone`	Quiet zone See "5.1.1⑮ QR quiet zone (`QrQuietZone`)" for available constants.
`text`	QR Code data The characters that can be input are as follows. ・ASCII character codes from 20h (space) to 7Eh (tilde) such as alphanumeric ('0' to '9', 'A' to 'Z', 'a' to 'z'). ・8 bits Latin / Katakana based on JIS X 0201 ・Shift_JIS code based on JIS X 0208 The available data size is 1 to 3909 bytes. QR Code data which is not complied with QR Code specification is ignored.

	`data`	QR Code data QR Code data shown on the display. The values that can be input is 00h to FFh. The available number of data is 1 to 3909. QR Code data which is not complied with QR Code specification is ignored.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description After specifying a map ID of the selecting template by this method, input QR Code data.
The input QR Code data is displayed to the screen when `ShowTemplate` is executed.

The input QR Code data is cleared under the following conditions.
- When `SelectTemplate` is executed.
- When other than 0 is specified at `time_ms` of `ShowTemplate`, and the specified display time has elapsed.
- When `ShowtTemplate` registered in `ExecuteMacro` is executed.

This method is ignored when a template is not registered.

`RegisterTemplate` Register template

Registers a template in the display.

Target
iOS	DSP-A01 via the printer
Android	DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void RegisterTemplate(int templateID,
                             string label,
                             string fileName);
```

Parameter
`templateID`	Template ID Specify the ID of template to register. The valid range is 0 to 127. Do not specify the template ID of 127 because it is being used for the system.
`label`	Template name A name for identification can be specified to the template to be registered. The characters that can be specified are ASCII code characters 20h (space) to 7Eh (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z'). Do not use Unicode 00A5h ('¥'). The available number of characters is 0 to 32. This `label` is optional. Specify null when the template name is not registered. The specified template name can be retrieved by `GetDisplayResponse`.
`fileName`	File path of the template data to register in the display The formats that can be entered are described below. For iOS: File path strings to use

For Android:

- 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 "3.4 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.

Supported file extension is .xml.
The maximum data size that can be registered is 8192 bytes.
See "Technical Reference for Display" for details on registration of the template data.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.

Description The display is changed to Standby mode when this method is executed. A selecting template is deselected.

Note Registered data at the factory may be added or changed without prior notice for quality improvement.

`UnregisterTemplate` Delete template

Deletes the registered template in the display.

Target
- iOS: DSP-A01 via the printer
- Android: DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void UnregisterTemplate(int templateID);
```

Parameter

`templateID` Template ID
Specify the ID of template to delete.
The valid range is 0 to 127.
This method is ignored when a template is not registered in the specified ID.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.

Description The display is changed to Standby mode when this method is executed. A selecting template is deselected.

Used memory is not released even the template is deleted. The used memory can be reused after executing `Defragment`.

RegisterImageData — Register image data

Registers image data in the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax

```
public void RegisterImageData(int imageID,
                              string label,
                              string fileName);
```

Parameter

`imageID`

Image ID
The valid range is 0 to 63.
Do not select image IDs 49 to 63 because they are being used for the system.

`label`

Image name
A name for identification can be specified to image data to be registered.
The characters that can be specified are ASCII code characters 20h (space) to 7Eh (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z').
Do not use Unicode 00A5h ('¥').
The available number of characters is 0 to 32.
This `label` is optional. Specify null when the template name is not registered.
The specified image name can be retrieved at `GetDisplayResponse`.

`fileName`

File path
Specify the file name of image data to register.
The formats that can be entered are described below.

For iOS:
File path strings to use

For Android:

- 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 "3.4 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.

Supported file extensions are .jpg, .jpeg, and .png.
However, even the supported extensions may not be registered depending on the format.
The maximum file size that can be registered is 786432 bytes.
The maximum data size that can be registered is 480 horizontal × 272 vertical pixels(px).

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.

Description The display is changed to Standby mode when this method is executed. A selecting template is deselected.

Note Registered data at the factory may be added or changed without prior notice for quality improvement.

`UnregisterImageData` Delate image data

Deletes registered image data in the display.

Target
- iOS: DSP-A01 via the printer
- Android: DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void UnregisterImageData(int imageID);
```

Parameter

`imageID` Image ID
The valid range is 0 to 63.
This method is ignored when image data is not registered in the specified ID.

Error `PrinterException` is thrown when an error occurs while this method is being called. See "**5.1.5 `PrinterException` Class**" for details on the error.

Description The display is changed to Standby mode when this method is executed. A selecting template is deselected.

Used memory is not released even image data is deleted. The used memory can be reused after executing `Defragment`.

`RegisterSlideData` Register slide data

Registers slide data in the display.

Target
- iOS: DSP-A01 via the printer
- Android: DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void RegisterSlideData(int slideID,
                              string label,
                              string fileName);
```

Parameter

`slideID` Slide ID
The valid range is 0 to 91.
Do not specify slide IDs of 80 to 90 because they are being used for the system.

`label` Slide name
A name for identification can be specified to slide data to be registered.
The characters that can be specified are ASCII code characters 20h (space) to 7Eh (tilde) such as alphanumeric characters ('0' to '9', 'A' to 'Z', 'a' to 'z').
Do not use Unicode 00A5h ('¥').
The available number of characters is 0 to 32.
This `label` is optional. Specify null when the template name is not registered.

The specified slide name can be retrieved by `GetDisplayResponse`.

`fileName` File path
Specify the file name of slide data to register.
The formats that can be entered are described below.

For iOS:
File path strings to use

For Android:

- 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 "3.4 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.

Supported file name extensions are .jpg, jpeg, and .png.
However, even the supported extensions may not be registered depending on the format.
The maximum file size that can be registered is 786432 bytes.
The maximum data size that can be registered is 480 horizontal × 272 vertical pixels (px).

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description The display is changed to Standby mode when this method is executed. A selecting template is deselected.

Execute `ShowSlide` to show registered slide data.
Specify the slide ID at `SelectTemplate` to use a registered slide data as a backscreen of the template.

Note Registered data at the factory may be added or changed without prior notice for quality improvement.

`UnregisterSlideData` Delete slide data

Deletes registered slide data in the display.

Target iOS DSP-A01 via the printer
Android DSP-A01 via the printer, DSP-A01 alone

Syntax `public void UnregisterSlideData(int slideID);`

Parameter `slideID` Slide ID
The valid range is 0 to 91.
This method is ignored when slide data is not registered in a specified ID.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description The display is changed to Standby mode when this method is executed. A selecting template is deselected.

`RegisterUserDefinedCharacter` Register user-defined character

Registers user-defined characters in the display.

Target

iOS	DSP-A01 via the printer
Android	DSP-A01 via the printer, DSP-A01 alone

Syntax

```
public void RegisterUserDefinedCharacter(string fileName);
```

Parameter

`fileName` File path
Specify the file name of the user-defined characters to register.
The formats that can be entered are described below.

For iOS:
File path strings to use

For Android:

- 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 "3.4 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.

Supported file extension is .bin.
See "Register User-Defined Character" of the display command in "Technical Reference for Display" for details on the user-defined character data.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description The display is changed to Standby mode when this method is executed. A selecting template is deselected.
User-defined characters are overwritten when this method is executed with user-defined character registered status.

Use a template which encoding specifying is Shift_JIS for displaying user-defined characters.
Specify **`CODE_PAGE_KATAKANA`** for `CodePage` before user-defined characters are displayed.
Specify character codes that can be specified for `text` of `SetTemplateTextData` before user-defined characters are displayed.
The character codes that can be specified are 0xE000 to 0xE05D.

UnregisterUserDefinedCharacter — Delete user-defined character

Deletes registered user-defined characters in the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax

```
public void UnregisterUserDefinedCharacter();
```

Error

`PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description

The display is changed to Standby mode when this method is executed. A selecting template is deselected.

All registered user-defined characters are deleted.
Used memory is not released even the user-defined characters are deleted. The used memory can be reused after executing `Defragment`.

RegisterOptionFont — Register optional font

Registers optional fonts in the display.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax

```
public void RegisterOptionFont(int startCode,
                               int endCode,
                               int width,
                               int height,
                               string fileName);
```

Parameter

`startCode`	Character code for registration starting The valid range is 20h to FFh of ASCII character code.
`endCode`	Character code for registration finishing The valid range is 20h to FFh of ASCII character code.
`width`	Character width (pixel: px) The valid range is 1 to 255.
`height`	Character height (pixel: px) The valid range is 1 to 255.
`fileName`	File path Specify the file name of the optional font to register. The formats that can be entered are described below. For iOS: File path strings to use For Android: • 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 "3.4 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.

Supported file name extension is .bin.
See "Register Optional Font" of the display command in "Technical Reference for Display" for details on optional font data.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description The display is changed to Standby mode when this method is executed. A selecting template is deselected.

When this method is executed with optional font registered status, the registered optional fonts are deleted, and a new memory area is allocated to register optional fonts.
Used memory is not released even the registered optional fonts are deleted. The used memory can be reused after executing `Defragment`.

`UnregisterOptionFont` Delete optional font

Deletes registered optional fonts in the display.

Target

iOS	DSP-A01 via the printer
Android	DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void UnregisterOptionFont();
```

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description The display is changed to Standby mode when this method is executed. A selecting template is deselected.

All registered optional fonts are deleted.
Used memory is not released even the optional fonts are deleted. The used memory can be reused after executing `Defragment`.

`ControlMacroRegistration` Start/Finish of macro registration

Specifies start or finish of macro registration to use in the display.

Target

iOS	DSP-A01 via the printer
Android	DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void ControlMacroRegistration(int macroID,
                                     MacroRegistrationFunction control);
```

Parameter

`macroID`	Macro ID The valid range is -1 to 127. Do not select macro IDs of 120 to 126 because they are being used for the system.

`control`	Macro registration processing See "5.1.1⑯ Macro registration processing (`MacroRegistrationFunction`)" for available constants.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description The procedures of the macro registration processing are as follows:

(1) Starts macro registration processing.
Specify -1 for `macroID`.
Specify **`MACRO_REGISTRATION_START`** for `control`.

(2) Executes methods.
Starts buffering of transmit data when methods are targeted in macro registration processing.
Transmit data of a macro registration processing target method which is executed during the buffering is not sent to the printer, buffered in macro data buffer. The maximum transmit data size to be able to buffer is 1024 bytes.
When the buffered transmit data exceeds the maximum size, the macro registration processing target method at the point of exceeding is to be error.
When the error occurs, data under the registration is discarded and canceled the macro mode.
Regarding transmit data which is held, finish the macro registration processing by procedure (3).
When a method is out of the macro registration processing target, it is executed immediately without buffering the transmit data.

Methods for the macro registration processing target are shown below.
- `ShowTemplate`
- `ShowSlide`
- `SelectTemplate`
- `SetTemplateImageData`
- `SelectTemplateTextObject`
- `SetTemplateTextAlignment`
- `SetTemplateTextLeftMargin`
- `SetTemplateTextLineSpacing`
- `SetTemplateTextBold`
- `SetTemplateTextUnderline`
- `SetTemplateTextSize`
- `SetTemplateTextFont`
- `SetTemplateTextRegisteredFont`
- `SetTemplateTextRightSpacing`
- `SetTemplateTextColor`
- `SetTemplateTextData`
- `SetTemplateBarcodeData`
- `SetTemplateQrCodeData`

(3) Finishes macro registration processing.
Specify the macro ID (0 to 127) to register at `macroID`.
When **`MACRO_REGISTRATION_REGIST`** is specified at `control`, buffered transmit data is sent to the printer. The buffered transmit data is held even after transmitting to the printer. The display is changed to Standby mode when this method is executed.
A selecting template is being deselected.

The holding transmit data is discarded by following processes.
- Specify **`MACRO_REGISTRATION_CLEAR`**
- Specify **`MACRO_REGISTRATION_START`**
- Execute `Disconnect`

The registered macro can be executed at `ExecuteMacro`.

A process to delete the registered macro is as follow.
Specify **MACRO_REGISTRATION_START** at `control` and specify -1 for `macroID` to call this method.
Specify **MACRO_REGISTRATION_START** at `control` and specify the macro ID to delete, and then call this method.
The display is changed to Standby mode when this method is executed. A selecting template is being deselected.

Note Registered data at the factory may be added or changed without prior notice for quality improvement.

GetDisplayResponse Get various response from Display

Gets response data from the display.

Target
iOS DSP-A01 via the printer
Android DSP-A01 via the printer, DSP-A01 alone

Syntax
```
public void GetDisplayResponse<T>(DisplayResponseId responseId,
                                  object param,
                                  out T[] response);
```

Parameter
`responseId` Display response type constant
See "5.1.1⑰ Display response type (`DisplayResponseId`)" for available constants.

`param` Command parameter
The value to be specified varies with on the display response type constant.
See the following table for description of the value to be specified.

`response` Buffer for storing the retrieved response data
The buffer type varies with the display response type constant.
See the following table for the buffer type.

Response Type Constant		
	Parameter	**Description**
DISPLAY_RESPONSE_REQUEST (Execution response request)		
	`param`	Specify 0 to 15 (00h to 0Fh) in an int type.
	`response`	Specify an int type array. When the response is retrieved successfully, the response code of the execution response request is stored with 64 to 79 (40h to 4Fh).
DISPLAY_RESPONSE_USER_AREA (Send remaining capacity of user area)		
	`param`	Specify null.
	`response`	Specify an int type array. When the response is retrieved successfully, the remaining capacity (unit: byte) of the user area is stored numerically.
DISPLAY_RESPONSE_TEMPLATE_ID_LIST (Send template ID)		
	`param`	Specify null.
	`response`	Specify an int type array. When the response is retrieved successfully, the registered template ID is stored in an array of NSInteger.

Response Type Constant	
Parameter	**Description**
DISPLAY_RESPONSE_IMAGE_ID_LIST (Send image ID)	
param	Specify null.
response	Specify an int type array. When the response is retrieved successfully, the registered image ID is stored in an array of NSInteger.
DISPLAY_RESPONSE_SLIDE_ID_LIST (Send slide ID)	
param	Specify null.
response	Specify an int type array. When the response is retrieved successfully, the registered slide ID is stored in an array of NSInteger.
DISPLAY_RESPONSE_TEMPLATE_LABEL (Send template name)	
param	Specify 0 to 127 (00h to 7Fh) in an int type.
response	Specify a string type array. When the response is retrieved successfully, the registered template name is stored as a character string.
DISPLAY_RESPONSE_IMAGE_LABEL (Send image name)	
param	Specify 0 to 63 (00h to 3Fh) in an int type.
response	Specify a string type array. When the response is retrieved successfully, the image name specified at the image file registration is stored in a character string.
DISPLAY_RESPONSE_SLIDE_LABEL (Send slide name)	
param	Specify 0 to 91 (00h to 5bh) in an int type.
response	Specify a string type array. When the response is retrieved successfully, the slide name specified at slide data registration is stored in a character string.

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

`SendBinary` Send binary data

Sends binary data to the printer.

Target All platforms All devices

Syntax `public void SendBinary(byte[] data);`

Parameter `data` Binary data to send to the printer
Data size that can be specified at 1 time is 3MB (3145728 bytes).

Error `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

Description This method sends the specified binary data to the printer without conversion.

By sending display command as binary data by this method, display functions not supported in the library become available. However, this method does not support commands to get the response from the display.

SendDataFile — Send specified file

Sends file data.

Target: All platforms, All devices

Syntax:

```
public void SendDataFile(string fileName,
                         PrintAlignment alignment,
                         Dithering dithering);
```

Parameter:

`fileName` — File path of the data to send to the device
The formats that can be entered are described below.

For iOS:
File path strings to use

For Android:

- 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 "3.4 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 file size that can be specified is maximum 16MB (16777216 bytes).
The file extensions that can be sent and the file transmission are described below.

- .bin, .dat
  Data is sent to the device as binary data without conversion.

`alignment` — Alignment
The alignment is not reflected to the display function.

`dithering` — Dithering
The dithering is not reflected to the display function.

Error: `PrinterException` is thrown when an error occurs while this method is being called.
See "**5.1.5 `PrinterException` Class**" for details on the error.

(4) Property Details

SendTimeout — Get/Set send timeout period

Gets or sets the timeout period in sending data.

Target	All platforms	All devices

Syntax
```
public int SendTimeout {get; set;}
```

Valid range
For iOS:
100 to 300000 (millisecond: ms)
When the value is specified less than 100, the period is set to 100 ms.
When the value is specified more than 300000, the period is set to 300000 ms.
For Android:
100 to 900000 (millisecond: ms)
When the value is specified out of the range, the period is set to 10000 ms.

Default: 10000

Description: Getting or setting is possible by this property regardless of whether `IsConnect` is true or false.
The set timeout period becomes effective at the next data sending.

ReceiveTimeout — Get/Set receive timeout period

Gets or sets the timeout period in receiving data.

Target	All platforms	All devices

Syntax
```
public int ReceiveTimeout {get; set;}
```

Valid range
For iOS:
100 to 300000 (millisecond: ms)
When the value is specified less than 100, the period is set to 100 ms.
When the value is specified more than 300000, the period is set to 300000 ms.
For Android:
100 to 900000 (millisecond: ms)
When the value is specified out of the range, the period is set to 10000 ms.

Default: 10000

Description: Getting or setting is possible by this property regardless of whether `IsConnect` is true or false.
The set timeout period becomes effective at the next data receiving.

InternationalCharacter — Get/Set international character set

Gets or sets the value of international character set.

Target	All platforms	All devices

Syntax
```
public InternationalCharacter InternationalCharacter {get; set;}
```

Description: See "5.1.1⑱ International character set (`InternationalCharacter`)" for configurable constants.
When an invalid value is specified, it is ignored.

When this property is not set, the international character set is as follows depending on the language setting of device.

When the language setting of device is Japanese: **COUNTRY_JAPAN**
When the language setting of device is other than Japanese: **COUNTRY_USA**

When text data is sent by `SendDataFile` or `SetTemplateTextData`, the display result of the following character codes varies. See "Appendix A Character Set" for details of the characters to be printed.

Character codes with the varying display result depending on the configuration of the international character:
0x23, 0x24, 0x40, 0x5B, 0x5C, 0x5D, 0x5E, 0x60, 0x7B, 0x7C, 0x7D, 0x7E

`CodePage` Get/Set codepage

Gets or sets the value of codepage.

Target	All platforms	All devices

Syntax `public CodePage CodePage {get; set;}`

Description See "5.1.1⑲ Codepage (`CodePage`)" for configurable constants. When an invalid value is specified, it is ignored.

When this property is not set, the codepage follows depending on the language setting of device.
When the language setting of device is Japanese:
CODE_PAGE_KATAKANA
When the language setting of device is other than Japanese:
CODE_PAGE_1252

The encoder used for sending the text data by `SendDataFile` or `SetTemplateTextData` is changed.
See "Appendix A Character Set" for characters to be displayed.

`DeviceModel` Get device model

Gets the value of the connecting printer model.

Target	All platforms	All devices

Syntax `public DeviceModel DeviceModel {get;}`

Default DeviceModel.DEVICE_MODEL_RP_E10

Return value See "5.1.1① Device model (`DeviceModel`)" for available constants.

`PortType` Get connecting port type

Gets the value of the port type used for connecting with the printer.

Target	All platforms	All devices

Syntax `public PortType PortType {get;}`

Default PortType.PORT_TYPE_BLUETOOTH

Return value See "5.1.1② Port type (`PortType`)" for available constants.

IsConnect — Verify connection state with printer

Verifies connection state with the printer.

Target	All platforms	All devices

Syntax

```
public bool IsConnect {get;}
```

Return value

true	Connected to the printer
false	Not connected to the printer

Description

This property retains the `Connect` state as a BOOL value.
When `Connect` succeeds, this property is true. After `Connect`, when `Disconnect` succeeds, this property becomes false.

SocketKeepingTime — Get/Set socket keeping time

Gets or sets the socket keeping time.

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer

Syntax

```
public int SocketKeepingTime {get; set;}
```

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

Getting or setting is possible by this property 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 iOS app "SII Printer Utility" on the App Store and the Android app "SII Printer Utility" on the Google Play.

The set socket keeping time becomes effective at the next `Connect` execution.

Version — Get SDK version

Gets the SDK version as a character string.

Target	All platforms	All devices

Syntax

```
public string Version {get;}
```

Return value

Returns 2 SDK version strings. (e.g. "1.0.0 / 1.0.0")
1st: SII SDK for .NET MAUI version
2nd: SII Print Class Library version compatible with the operating environment

Description

Getting is possible by this property regardless of whether `IsConnect` is true or false.

5.1.3 `DeviceInfo` Class

This class stores the device information found by printer searching method. See "5.1.3(2) Property List" for device information to be able to get.

(1) Method List

Methods provided by `DeviceInfo` class are shown in the following table.

Name	Description
`DeviceInfo`	Constructor of the device information class

(2) Property List

Properties provided by `DeviceInfo` class are shown in the following table.

Name	Access	Description
`PortType`	R	Get connecting port type
`DeviceName`	R	Get device name (printer model name)
`BluetoothAddress`	R	Get Bluetooth address
`MacAddress`	R	Get MAC address
`IpAddress`	R	Get IP address
`IsBonded`	R	Get paring state
`DevicePath`	R	Get device path

(3) Method List

DeviceInfo Constructor

In syntax (a), this is the constructor of the device information class used for Bluetooth connections.
In syntax (b), this is the constructor of the device information class used for TCP/IP connections.
In syntax (c), this is the constructor of the device information class used for USB connections.

Target	All platforms	All devices

Syntax

(a)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string bluetoothAddress,
                  bool isBonded);
```

(b)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string macAddress,
                  string ipAddress);
```

(c)
```
public DeviceInfo(PortType portType,
                  string deviceName,
                  string devicePath);
```

Description

This method stores the printer information found by `StartDiscoveryDevice`.

In iOS, the syntax (a) and (b) are valid.

In Android, the syntax (a), (b) and (c) are valid when using DSP-A01 via the printer.
When using DSP-A01 alone, only the syntax (c) is valid.

(4) Property Details

PortType — Get connecting port type

Target	All platforms	All devices
Syntax	`public PortType PortType {get;}`	
Description	This property gets the connecting port type from the device information found by `StartDiscoveryDevice`.	

DeviceName — Get device name (printer model name)

Target	All platforms	All devices
Syntax	`public string DeviceName {get;}`	
Description	This property gets the device name (printer model name) from the device information found by `StartDiscoveryDevice`.	

BluetoothAddress — Get Bluetooth address

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public string BluetoothAddress {get;}`	
Description	This property gets the Bluetooth address from the printer information found by `StartDiscoveryDevice`.	

MacAddress — Get MAC address

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public string MacAddress {get;}`	
Description	This property gets the MAC address from the device information found by `StartDiscoveryDevice`.	

IpAddress — Get IP address

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public string IpAddress {get;}`	
Description	This property gets the IP address from the device information found by `StartDiscoveryDevice`.	

IsBonded — Get paring state

Target	All platforms	All printers, DSP-A01 via the printer

Syntax

```
public bool IsBonded {get;}
```

Description

This property gets the paring state from the device information found by `StartDiscoveryDevice`.

In iOS, this property always returns true.

DevicePath — Get device path

Target	All platforms	DSP-A01 via the printer, DSP-A01 alone

Syntax

```
public string DevicePath {get;}
```

Description

This property gets the USB device file path string from the device information found by `StartDiscoveryDevice`.

5.1.4 `PrinterIdentifier` Class

`PrinterIdentifier` class stores the printer ID of the printer retrieved by `GetPrinterResponse`.

(1) Property List

Properties provided by PrinterIdentifier class are shown in the following table.

Name	Access	Description
`PrinterModelID`	R	Get printer model ID
`TypeID`	R	Get type ID
`RomVersionID`	R	Get ROM version ID
`MainFirmwareVersion`	R	Get firmware version (main)
`Manufacturer`	R	Get manufacturer
`ModelName`	R	Get model name
`SerialNumber`	R	Get serial number
`BootFirmwareVersion`	R	Get firmware version (boot)
`BootFirmwareChecksum`	R	Get firmware checksum (boot)
`MainFirmwareChecksum`	R	Get firmware checksum (main)
`FirmwareChecksum`	R	Get firmware checksum (main + boot)

(2) Property Details

`PrinterModelID` — Get printer model ID

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public byte PrinterModelID()`	
Description	Gets the printer model ID from the printer ID of the printer retrieved by `GetPrinterResponse`.	

`TypeID` — Get type ID

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public byte TypeID()`	
Description	Gets the type ID from the printer ID of the printer retrieved by `GetPrinterResponse`.	

RomVersionID — Get ROM version ID

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public byte RomVersionID()`	
Description	Gets the ROM version ID from the printer ID of the printer retrieved by `GetPrinterResponse`.	

MainFirmwareVersion — Get firmware version (main)

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public String MainFirmwareVersion()`	
Description	Gets the firmware version (main) as a string from the printer ID of the printer retrieved by `GetPrinterResponse`.	

Manufacturer — Get manufacturer

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public String Manufacturer()`	
Description	Gets the manufacturer name as a string from the printer ID of the printer retrieved by `GetPrinterResponse`.	

ModelName — Get model name

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public String ModelName()`	
Description	Gets the model name as a string from the printer ID of the printer retrieved by `GetPrinterResponse`.	

SerialNumber — Get serial number

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public String SerialNumber()`	
Description	Gets the serial number as a string from the printer ID of the printer retrieved by `GetPrinterResponse`.	

BootFirmwareVersion — Get firmware version (boot)

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public String BootFirmwareVersion()`	
Description	Gets the firmware version (boot) as a string from the printer ID of the printer retrieved by `GetPrinterResponse`.	

BootFirmwareChecksum — Get firmware checksum (boot)

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public byte[] BootFirmwareChecksum()`	
Description	Gets the firmware checksum (boot) from the printer ID of the printer retrieved by `GetPrinterResponse`.	

MainFirmwareChecksum — Get firmware checksum (main)

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public byte[] MainFirmwareChecksum()`	
Description	Gets the firmware checksum (main) from the printer ID of the printer retrieved by `GetPrinterResponse`.	

FirmwareChecksum — Get firmware checksum (main + boot)

Target	All platforms	All printers, DSP-A01 via the printer
Syntax	`public byte[] FirmwareChecksum()`	
Description	Gets the firmware checksum (main + boot) from the printer ID of the printer retrieved by `GetPrinterResponse`.	

5.1.5 `PrinterException` Class

(1) Method List

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

Name	Description
`PrinterException`	Constructor

(2) Property List

Properties provided by `PrinterException` class are shown in the following table.

Name	Access	Description
`HResult`	R	Get error code
`Message`	R	Get error message

(3) Constant List

① Error code

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

Constant Name	Description	Value
ERROR_ACCESS_DENIED	Failed to get the handle.[*1]	-1
	An unavailable port was specified.	
	An unsupported method was specified.	
ERROR_SHARING_VIOLATION	An already opened port was specified.	-11
ERROR_PORT_NOT_OPENED	The port is not open.	-12
ERROR_DEVICE_NOT_CONNECTED	There is a problem with the Bluetooth connection.	-21
	There is a problem with the USB connection.	
	The printer with the specified IP address does not present.	
ERROR_OFFLINE	Disconnected state or the printer is offline.	-22
ERROR_EXTERNAL_DEVICE_NOT_CONNECTED	The display is not connected.	-23
ERROR_DEVICE_INITIALIZE_FAILED	Failed to change the printer settings. 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
ERROR_DATA_SIZE_ZERO	0-byte data was specified.	-101
ERROR_OVER_MAX_DATA_SIZE	Maximum data size is exceeded.	-102
ERROR_DATA_SIZE_INVALID	Data size is invalid.	-103
ERROR_ENCODE_FAILED	An error occurred in encoding text data.[*1]	-111

Constant Name	Description	Value
ERROR_TIMEOUT	Send timeout occurred. Receive timeout occurred.	-201
ERROR_FILE_NOT_FOUND	The specified file is not found.	-301
ERROR_FILE_USED	The specified file is in use by another process.	-302
ERROR_FILE_INVALID	The specified file is invalid.	-303
ERROR_LOW_MEMORY	Memory shortage occurred when loading image file.	-311
ERROR_OVER_MAX_IMAGE	Either or both of width and height of image file exceeds the number of printable maximum dots.	-312
ERROR_LOW_USER_AREA	Remaining user area is insufficient.	-401
ERROR_LOW_EXTERNAL_RAM	Remaining RAM capacity is insufficient.	-402
ERROR_NOT_REGISTERD	The template is not registered. The image data is not registered. The slide data is not registered. The optional font is not registered. The user-defined character is not registered.	-403
ERROR_NOT_UNREGISTERD	The template is not deleted. The image data is not deleted. The slide data is not deleted. The optional font is not deleted. The user-defined character is not deleted.	-404
ERROR_INVALID_DATA	The specified data is invalid.	-503
ERROR_INVALID_PARAM	The specified parameter is invalid.	-9999

*1: Abnormal processing might have occurred.

(4) Method Details

PrinterException — Constructor

This is the exception class that is thrown when API of `PrinterManager` class is called.

Syntax `public PrinterException(int code, string message);`

(5) Property Details

HResult — Get error code

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax `public int HResult {get;}`

Description This property gets the error code in int type.

Return value See "5.1.5(3) Constant List".

Message — Get error message

Target	iOS	DSP-A01 via the printer
	Android	DSP-A01 via the printer, DSP-A01 alone

Syntax `public string Message {get;}`

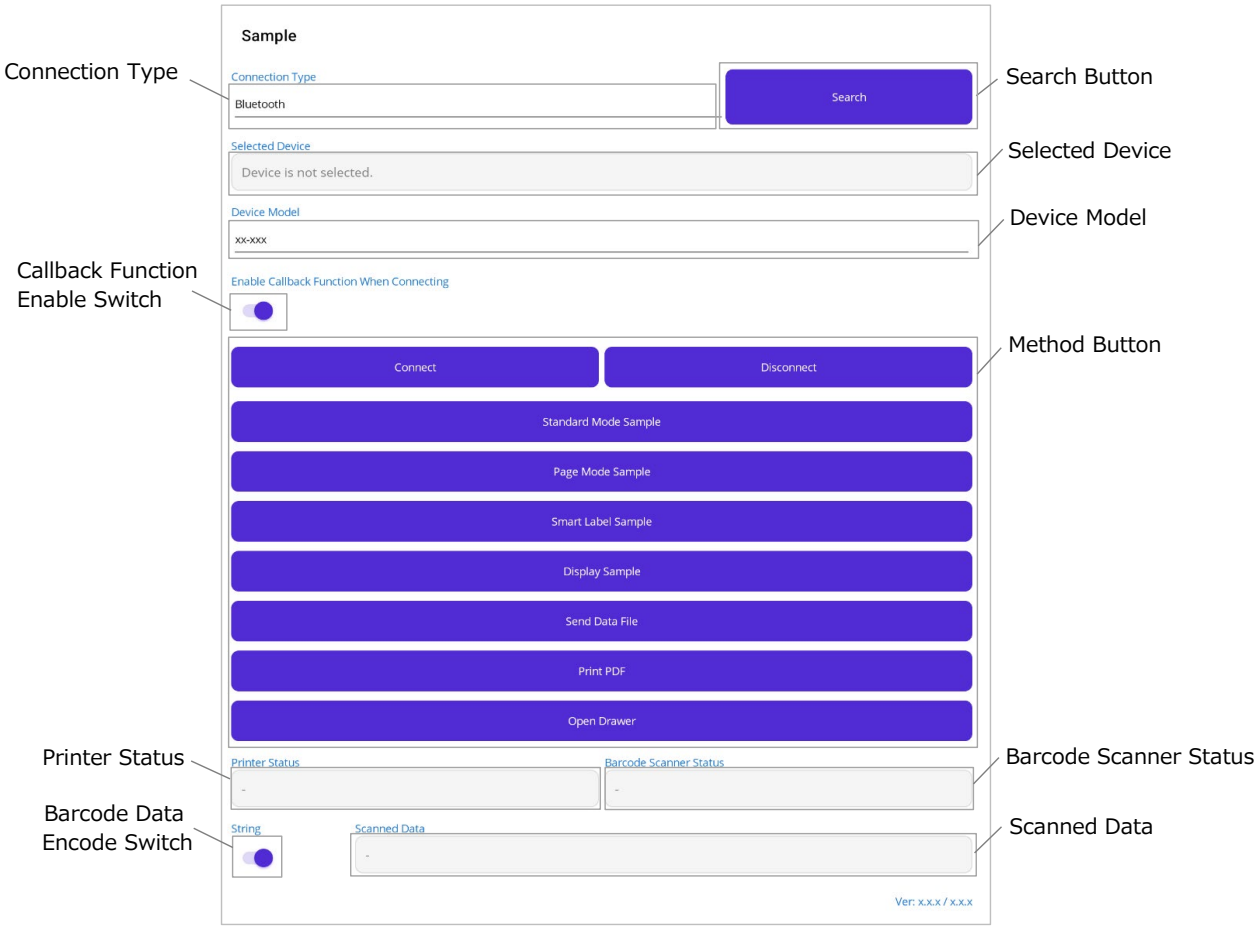
Description This property gets a string that supplements the contents of the `HResult`.

Chapter 6
Sample Program

This chapter describes the sample program provided by the SDK.

6.1 Screen Layout

The SDK includes a sample program in .NET MAUI application project format. This section describes the sample screen.



Item	Description
Connection type	Selects the connection type to the printer.
Search Button	Starts to search the selected device. A list of detected devices is displayed after completing the search. By tapping the device from the list, the device can be selected. For Bluetooth connection: A list of Bluetooth connection device is displayed. For iOS, only a paired Bluetooth device is displayed. For USB connection: The device connected to the USB is displayed. For TCP/IP connection: A list of the TCP/IP connectable device is displayed.
Selected Device	The selecting device is displayed.
Device Model	The device to connect is displayed. Tap the field to connect the device type to connect.
Callback Function Enable Switch	Selects whether to enable a callback function when connecting to a device. On: Starts the callback function upon connection. Off: The callback function does not respond.
Method Button*1	In addition to the method buttons to execute `Connect` and `Disconnect`, sample printing combined with each method and acquisition of their status can be executed.
Printer Status	Displays the printer status. When [Callback Function Enable Switch] is On, the latest printer status is displayed.
Barcode Scanner Status	Displays the connection status of the barcode scanner. When [Callback Function Enable Switch] is On, the connection status of the barcode scanner is displayed. In the device not support the barcode scanner, Offline is displayed.
Barcode Data Encode Switch	Selects the conversion of barcode data read by the barcode scanner. On: Converts the read binary data to the character strings and displays. Off: Displays the read binary data as it is.
Scanned Data	Displays the barcode data read by the barcode scanner. When [Callback Function Enable Switch] is On, the barcode data read by the barcode scanner is displayed. In the device not support the barcode scanner, nothing is displayed.

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

6.2 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 or display results vary depending on the setting of the international character set.
See "A.2 International Character Set" for the specific character codes.

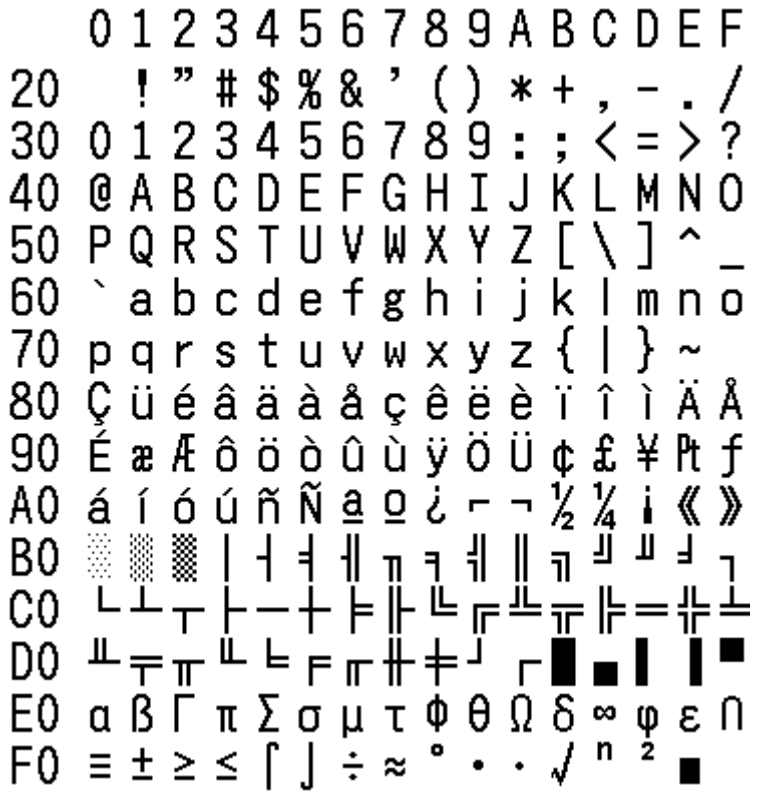


Figure A-1 CODE_PAGE_437 (USA, Standard Europe)

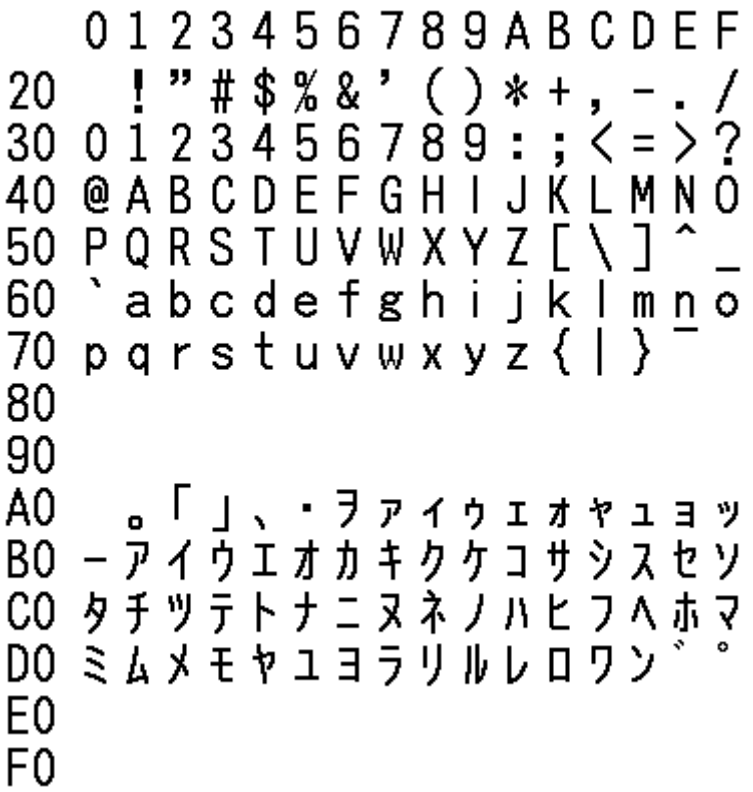


Figure A-2 CODE_PAGE_KATAKANA

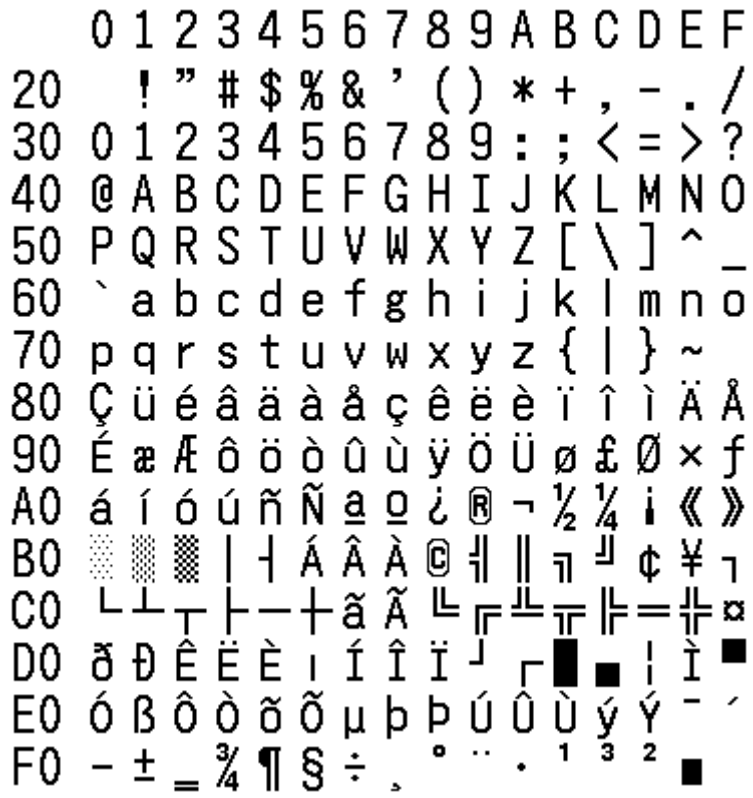


Figure A-3 CODE_PAGE_850 (Multilingual)

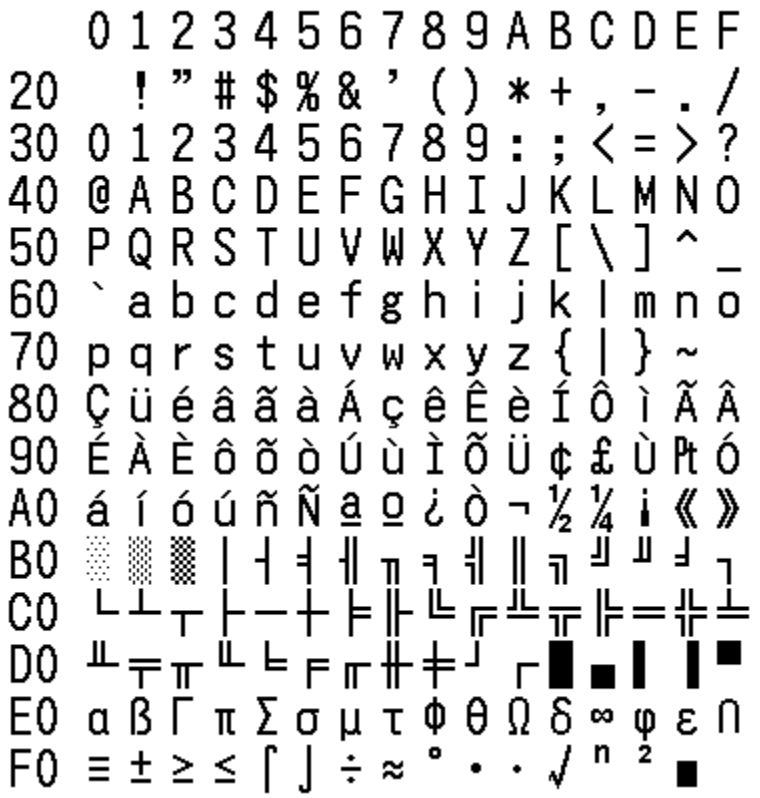


Figure A-4 CODE_PAGE_860 (Portuguese)

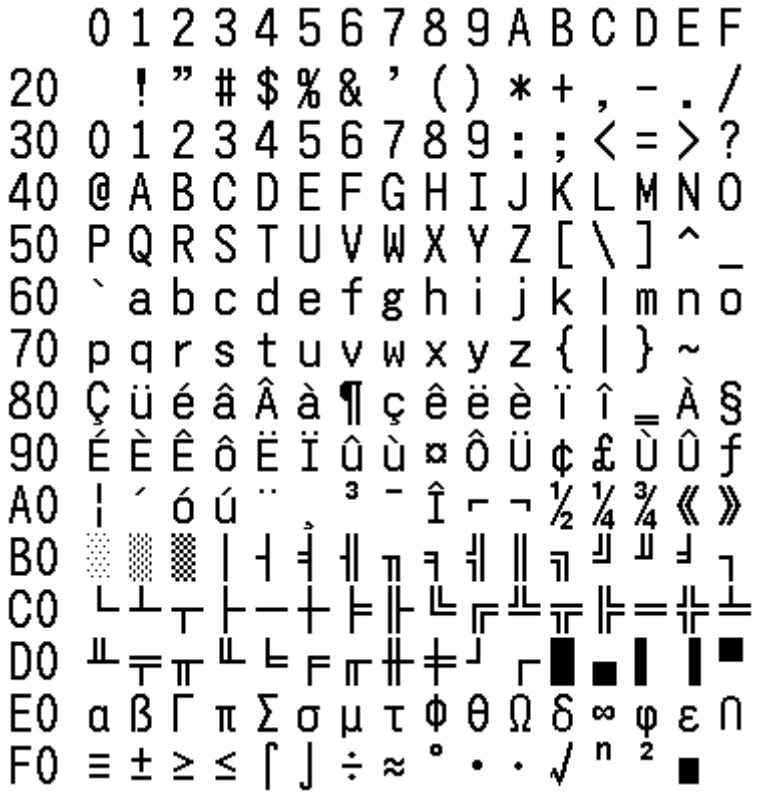


Figure A-5 CODE_PAGE_863 (Canadian-French)

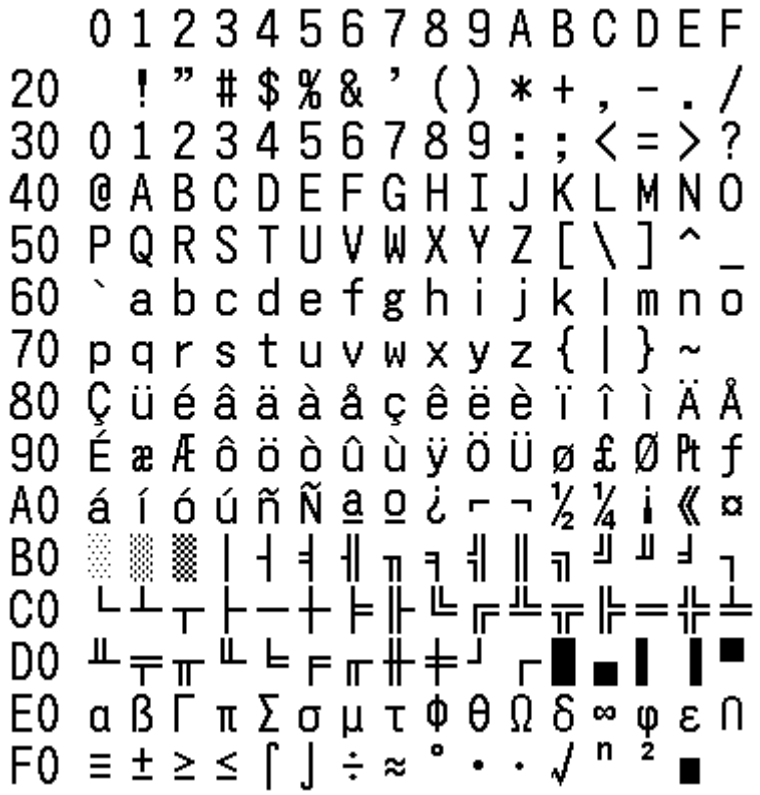


Figure A-6 CODE_PAGE_865 (Nordic)

CODE_PAGE_857 is not supported by RP-E10 nor RP-D10.

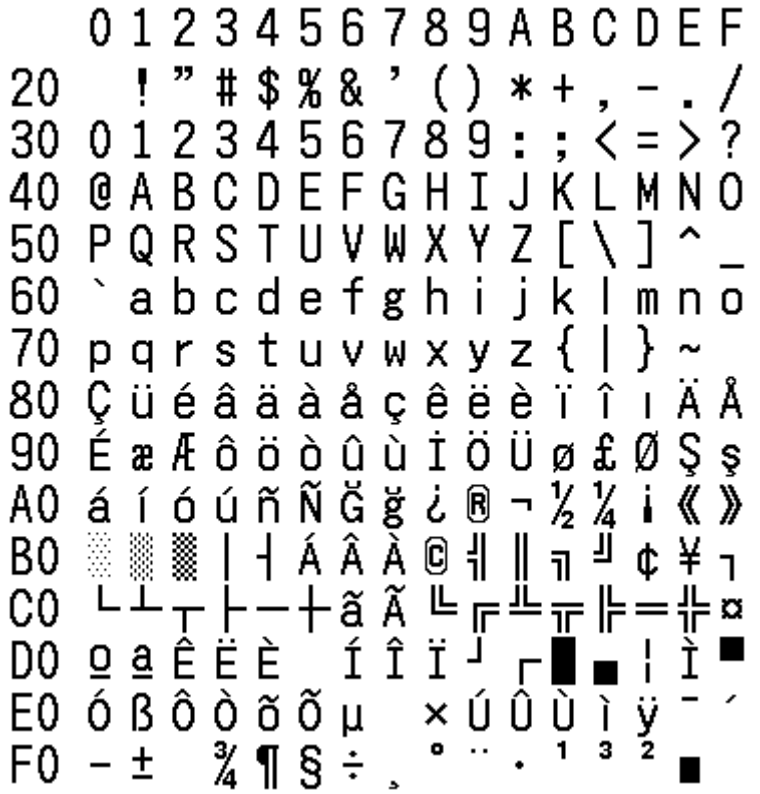


Figure A-7 CODE_PAGE_857 (Turkish)

CODE_PAGE_737 is not supported by RP-E10 nor RP-D10.

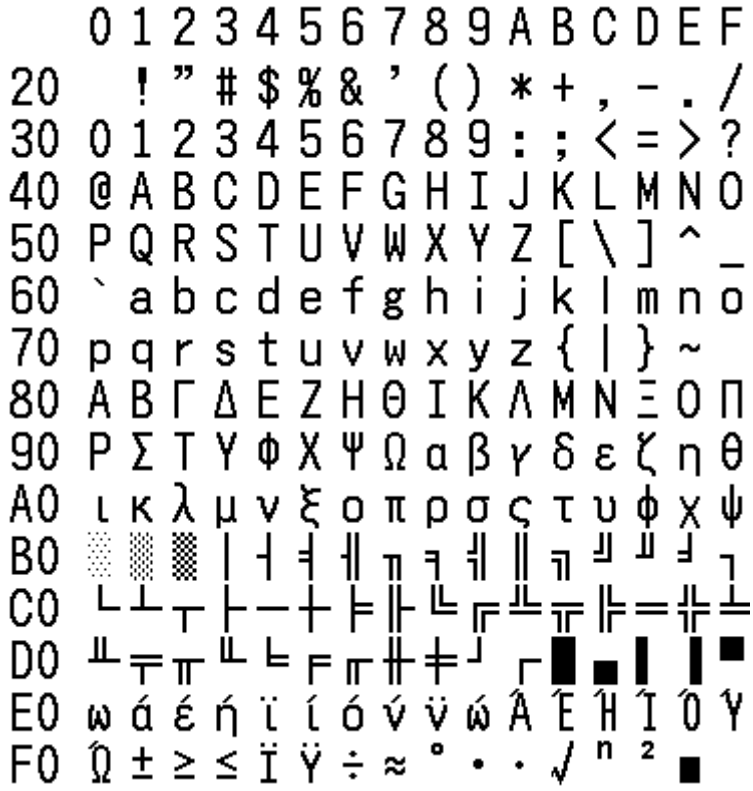


Figure A-8 CODE_PAGE_737 (Greek)



Figure A-9 CODE_PAGE_1252 (Latin)

CODE_PAGE_866 is not supported by RP-E10 nor RP-D10.

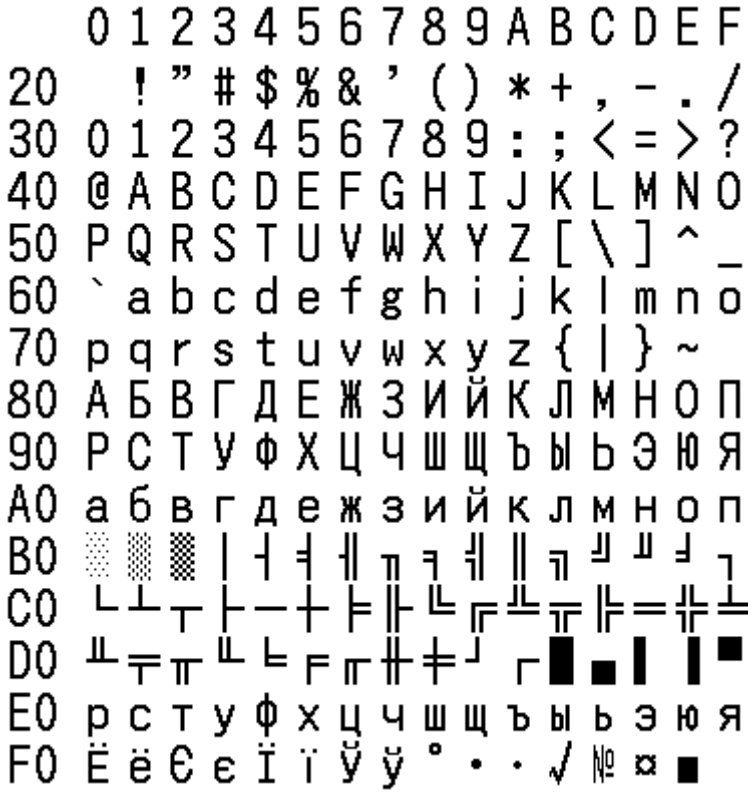


Figure A-10 CODE_PAGE_866 (Russian)

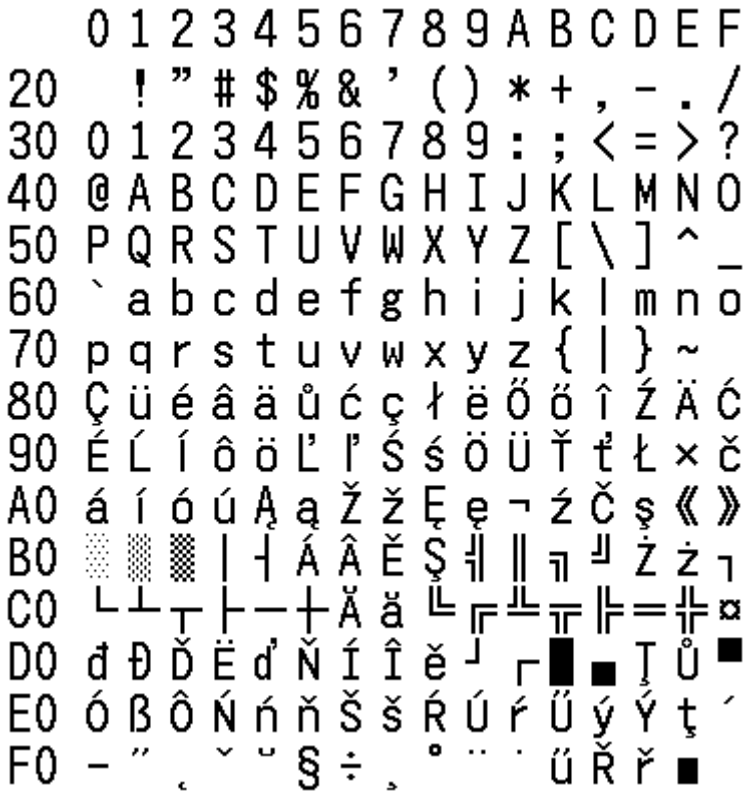


Figure A-11 CODE_PAGE_852 (Eastern Europe)

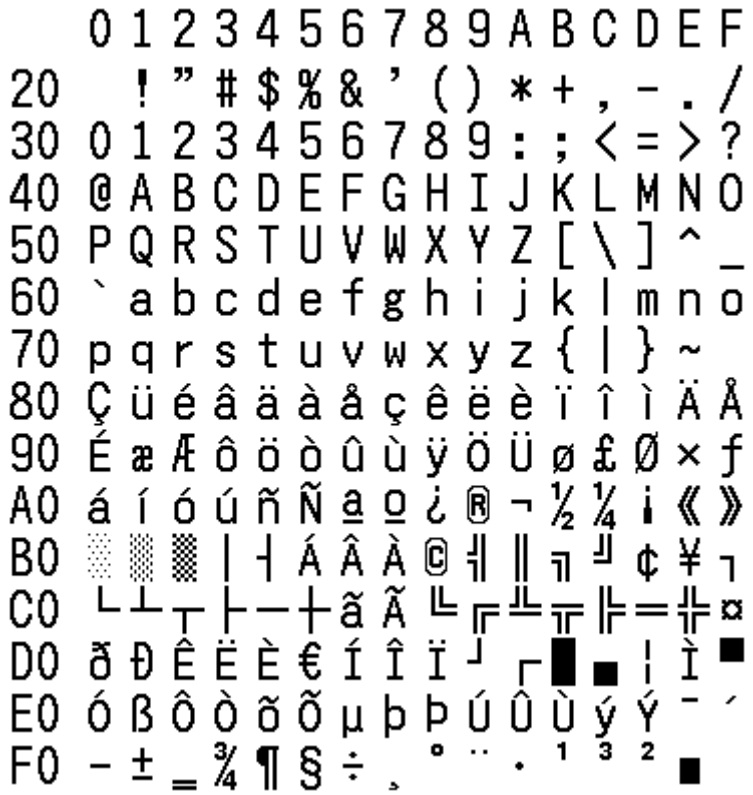


Figure A-12 CODE_PAGE_858 (Euro)

CODE_PAGE_855 is not supported by RP-E10 nor RP-D10.



Figure A-13 CODE_PAGE_855 (Cyrillic)



Figure A-14 CODE_PAGE_864 (Arabic)



Figure A-15 CODE_PAGE_1250 (Central European)



Figure A-16 CODE_PAGE_1251 (Cyrillic)

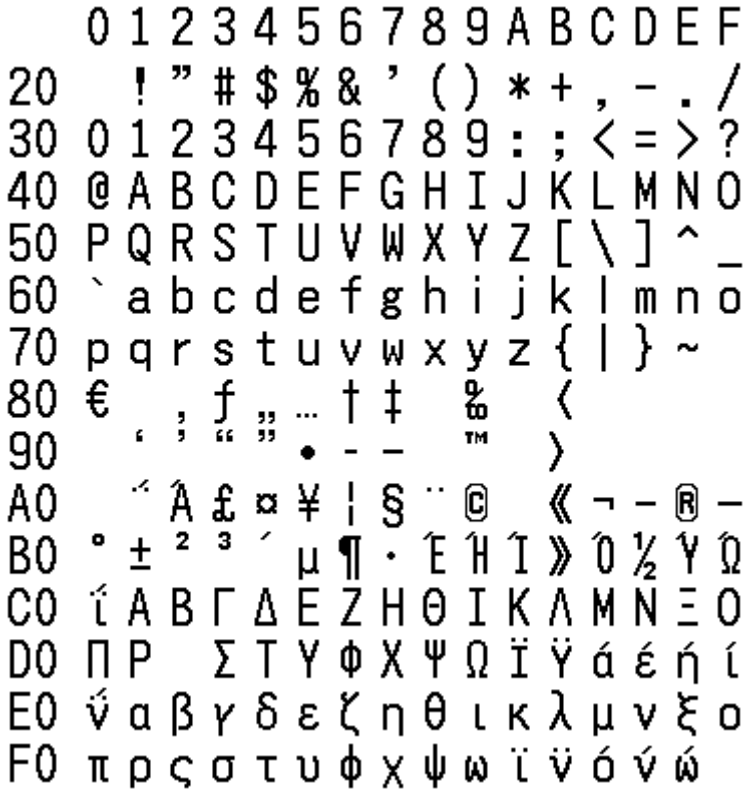


Figure A-17 CODE_PAGE_1253 (Greek)



Figure A-18 CODE_PAGE_1254 (Turkish)

A.2 International Character Set

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

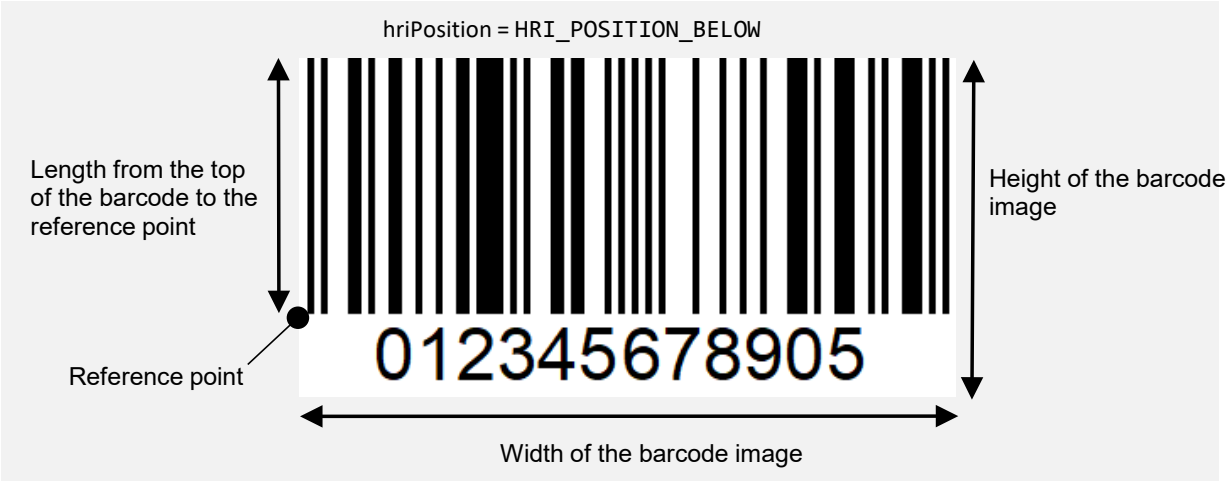
	23	24	40	5B	5C	5D	5E	60	7B	7C	7D	7E
COUNTRY_USA	#	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_FRANCE	#	$	à	°	ç	§	^	`	é	ù	è	¨
COUNTRY_GERMANY	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
COUNTRY_ENGLAND	£	$	@	[	\	]	^	`	{	|	}	~
COUNTRY_DENMARK_1	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
COUNTRY_SWEDEN	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
COUNTRY_ITALY	#	$	@	°	\	é	^	ù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

Figure A-19 International Character Set

Appendix B
Barcode Size List

B.1 Barcode Size List (SLP720RT, SLP721RT, RP-F10, RP-G10, MP-B30, MP-B30L, MP-B21L)

B.1.1 PrintBarcode, PrintPageModeBarcode



(1) Height of the barcode image

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

(2) Width of the barcode image

barcodeSymbol	moduleSize	**Width of Barcode Image**
BARCODE_UPC_A	**BARCODE_MODULE_WIDTH_2**	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
BARCODE_CODE93	**BARCODE_MODULE_WIDTH_2**	18 × number of barcode data + 56
	BARCODE_MODULE_WIDTH_3	27 × number of barcode data + 84
	BARCODE_MODULE_WIDTH_4	36 × number of barcode data + 112
	BARCODE_MODULE_WIDTH_5	45 × number of barcode data + 140
	BARCODE_MODULE_WIDTH_6	54 × number of barcode data + 168
BARCODE_CODE128	**BARCODE_MODULE_WIDTH_2**	22 × number of barcode data + 26
	BARCODE_MODULE_WIDTH_3	33 × number of barcode data + 39
	BARCODE_MODULE_WIDTH_4	44 × number of barcode data + 52

barcodeSymbol	moduleSize	**Width of Barcode Image**
BARCODE_CODE128	**BARCODE_MODULE_WIDTH_5**	55 × number of barcode data + 65
	BARCODE_MODULE_WIDTH_6	66 × number of barcode data + 78
BARCODE_GS1_OMNI_DIRECTIONAL	**BARCODE_MODULE_WIDTH_2**	192
	BARCODE_MODULE_WIDTH_3	288
	BARCODE_MODULE_WIDTH_4	384
	BARCODE_MODULE_WIDTH_5	480
	BARCODE_MODULE_WIDTH_6	576
BARCODE_GS1_TRUNCATED	**BARCODE_MODULE_WIDTH_2**	192
	BARCODE_MODULE_WIDTH_3	288
	BARCODE_MODULE_WIDTH_4	384
	BARCODE_MODULE_WIDTH_5	480
	BARCODE_MODULE_WIDTH_6	576
BARCODE_GS1_LIMITED	**BARCODE_MODULE_WIDTH_2**	158
	BARCODE_MODULE_WIDTH_3	237
	BARCODE_MODULE_WIDTH_4	316
	BARCODE_MODULE_WIDTH_5	395
	BARCODE_MODULE_WIDTH_6	474
BARCODE_GS1_EXPANDED[*1]	**BARCODE_MODULE_WIDTH_2**	number of barcode module × 2
	BARCODE_MODULE_WIDTH_3	number of barcode module × 3
	BARCODE_MODULE_WIDTH_4	number of barcode module × 4
	BARCODE_MODULE_WIDTH_5	number of barcode module × 5
	BARCODE_MODULE_WIDTH_6	number of barcode module × 6

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

barcodeSymbol	nwRatio	moduleSize	**Width of Barcode Image**
BARCODE_CODE39	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	26 × number of barcode data + 50
		BARCODE_MODULE_WIDTH_3	39 × number of barcode data + 75
		BARCODE_MODULE_WIDTH_4	52 × number of barcode data + 100
		BARCODE_MODULE_WIDTH_5	65 × number of barcode data + 125
		BARCODE_MODULE_WIDTH_6	78 × number of barcode data + 150
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	29 × number of barcode data + 56
		BARCODE_MODULE_WIDTH_3	45 × number of barcode data + 87
		BARCODE_MODULE_WIDTH_4	58 × number of barcode data + 112
		BARCODE_MODULE_WIDTH_5	74 × number of barcode data + 143
		BARCODE_MODULE_WIDTH_6	87 × number of barcode data + 168
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	32 × number of barcode data + 62
		BARCODE_MODULE_WIDTH_3	48 × number of barcode data + 93
		BARCODE_MODULE_WIDTH_4	64 × number of barcode data + 124
		BARCODE_MODULE_WIDTH_5	80 × number of barcode data + 155
		BARCODE_MODULE_WIDTH_6	96 × number of barcode data + 186
BARCODE_ITF	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	14 × number of barcode data + 16
		BARCODE_MODULE_WIDTH_3	21 × number of barcode data + 24
		BARCODE_MODULE_WIDTH_4	28 × number of barcode data + 32
		BARCODE_MODULE_WIDTH_5	35 × number of barcode data + 40
		BARCODE_MODULE_WIDTH_6	42 × number of barcode data + 48
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	16 × number of barcode data + 17
		BARCODE_MODULE_WIDTH_3	25 × number of barcode data + 26
		BARCODE_MODULE_WIDTH_4	32 × number of barcode data + 34

barcodeSymbol	nwRatio	moduleSize	**Width of Barcode Image**
BARCODE_ITF	**NWRATIO_1TO2_5**	**BARCODE_MODULE_WIDTH_5**	41 × number of barcode data + 43
		BARCODE_MODULE_WIDTH_6	48 × number of barcode data + 51
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	18 × number of barcode data + 18
		BARCODE_MODULE_WIDTH_3	27 × number of barcode data + 27
		BARCODE_MODULE_WIDTH_4	36 × number of barcode data + 36
		BARCODE_MODULE_WIDTH_5	45 × number of barcode data + 45
		BARCODE_MODULE_WIDTH_6	54 × number of barcode data + 54
BARCODE_CODABAR*1	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	20 × number of data + 2 × (2 + number of wide data) - 2
		BARCODE_MODULE_WIDTH_3	30 × number of data + 3 × (2 + number of wide data) - 3
		BARCODE_MODULE_WIDTH_4	40 × number of data + 4 × (2 + number of wide data) - 4
		BARCODE_MODULE_WIDTH_5	50 × number of data + 5 × (2 + number of wide data) - 5
		BARCODE_MODULE_WIDTH_6	60 × number of data + 6 × (2 + number of wide data) - 6
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	22 × number of data + 3 × (2 + number of wide data) - 2
		BARCODE_MODULE_WIDTH_3	34 × number of data + 5 × (2 + number of wide data) - 3
		BARCODE_MODULE_WIDTH_4	44 × number of data + 6 × (2 + number of wide data) - 4
		BARCODE_MODULE_WIDTH_5	56 × number of data + 8 × (2 + number of wide data) - 5
		BARCODE_MODULE_WIDTH_6	66 × number of data + 9 × (2 + number of wide data) - 6
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	24 × number of data + 4 × (2 + number of wide data) - 2
		BARCODE_MODULE_WIDTH_3	36 × number of data + 6 × (2 + number of wide data) - 3
		BARCODE_MODULE_WIDTH_4	48 × number of data + 8 × (2 + number of wide data) - 4
		BARCODE_MODULE_WIDTH_5	60 × number of data + 10 × (2 + number of wide data) - 5
		BARCODE_MODULE_WIDTH_6	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 ": / . +".

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

B.1.2 PrintPDF417, PrintPageModePDF417



(1) Height of the barcode image

Height of the barcode image[*1] = moduleHeight × row[*2]

*1: Height of the barcode image = Length from the top of the barcode to the reference point
*2: row≠0

(2) Width of the barcode image

When pdf417Symbol is **PDF417_STANDARD**:

Width of the barcode image = (17 × column[*1] + 69) × module size value

*1: column≠0

When pdf417Symbol is **PDF417_COMPACT**:

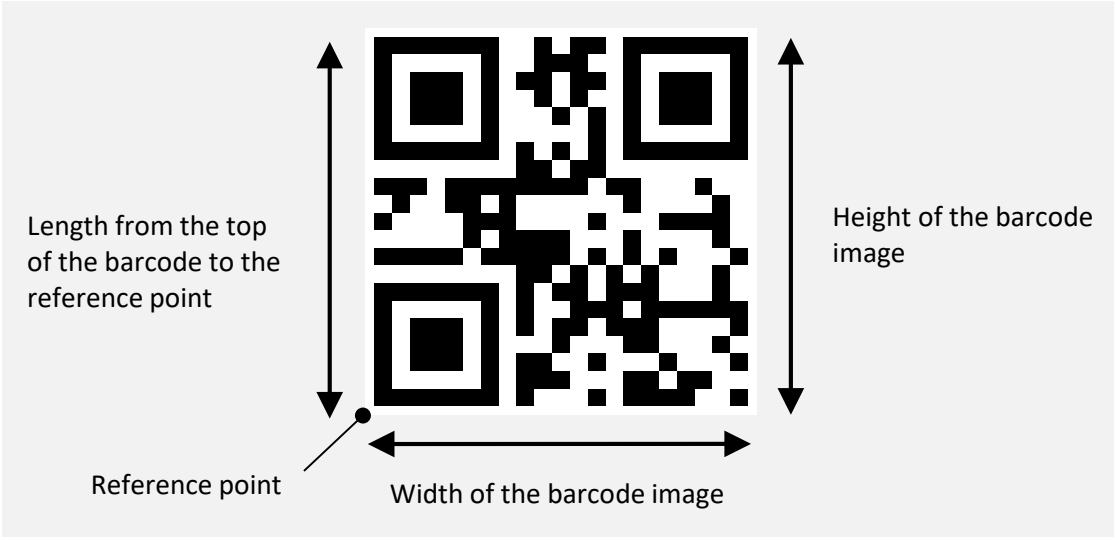
Width of the barcode image = (17 × column[*1] + 35) × module size value

*1: column≠0

Module Size Value

moduleSize	Module Size Value
PDF417_MODULE_WIDTH_2	2
PDF417_MODULE_WIDTH_3	3
PDF417_MODULE_WIDTH_4	4
PDF417_MODULE_WIDTH_5	5
PDF417_MODULE_WIDTH_6	6
PDF417_MODULE_WIDTH_7	7
PDF417_MODULE_WIDTH_8	8

B.1.3 PrintQRCode, PrintPageModeQRCode



(1) Height and width of the barcode image

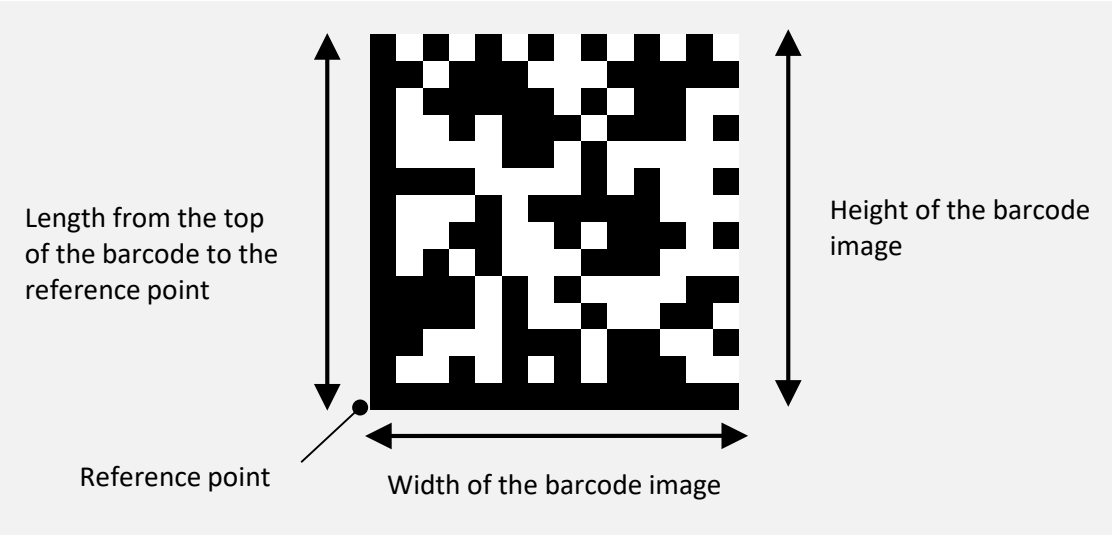
Height[*1] and width of the barcode image = (4 × version[*2] + 17) × module size value

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

Module Size Value

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

B.1.4 PrintDataMatrix, PrintPageModeDataMatrix



(1) Height and width of the barcode image

Height of the barcode image = number of vertical module × module size value

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

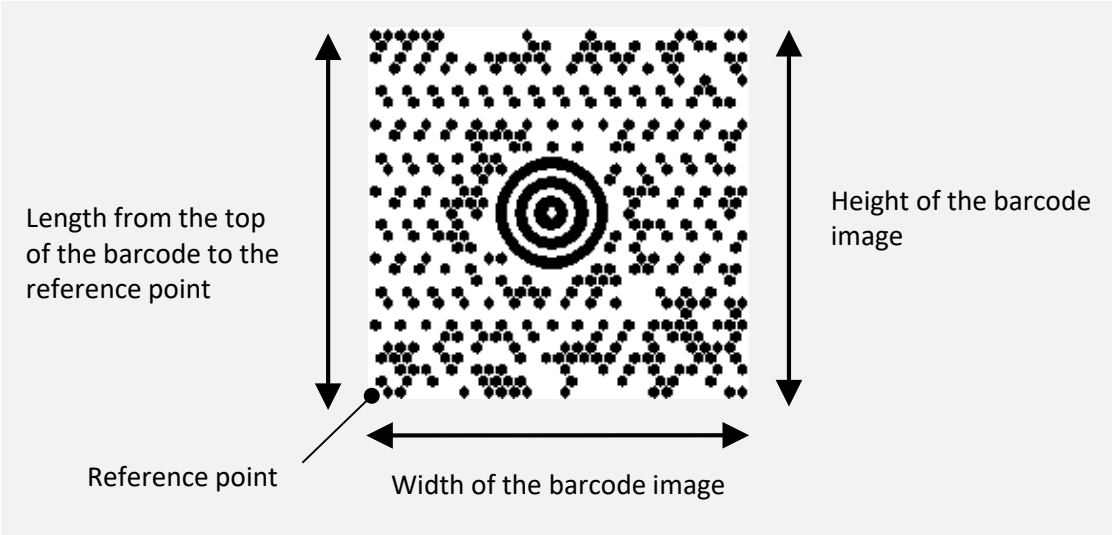
dataMatrixModule	Number of Vertical Module	Number of Horizontal Module
DATA_MATRIX_10_10	10	10
DATA_MATRIX_12_12	12	12
DATA_MATRIX_14_14	14	14
DATA_MATRIX_16_16	16	16
DATA_MATRIX_18_18	18	18
DATA_MATRIX_20_20	20	20
DATA_MATRIX_22_22	22	22
DATA_MATRIX_24_24	23	23
DATA_MATRIX_26_26	26	26
DATA_MATRIX_32_32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

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

Module Size Value

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

B.1.5 PrintMaxicode, PrintPageModeMaxicode



(1) Height of the barcode image

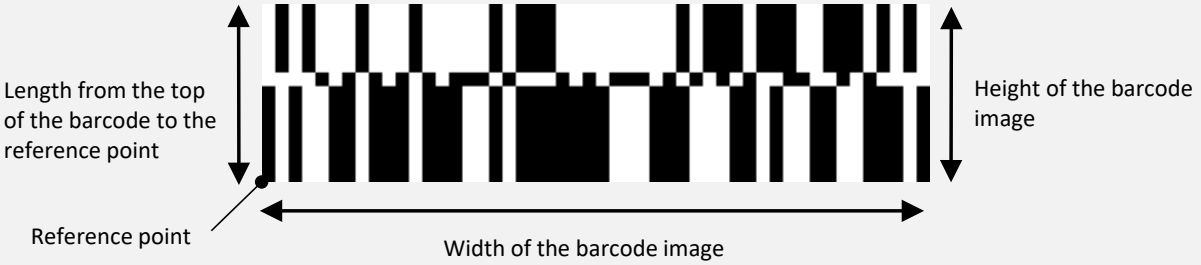
Height of the barcode image*1 = 200

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

(2) Width of the barcode image

Width of the barcode image = 210

B.1.6 PrintGS1DataBarStacked, PrintPageModeGS1DataBarStacked



(1) Height and width of the barcode image

Height of the barcode image[*1] = 13 × module size value

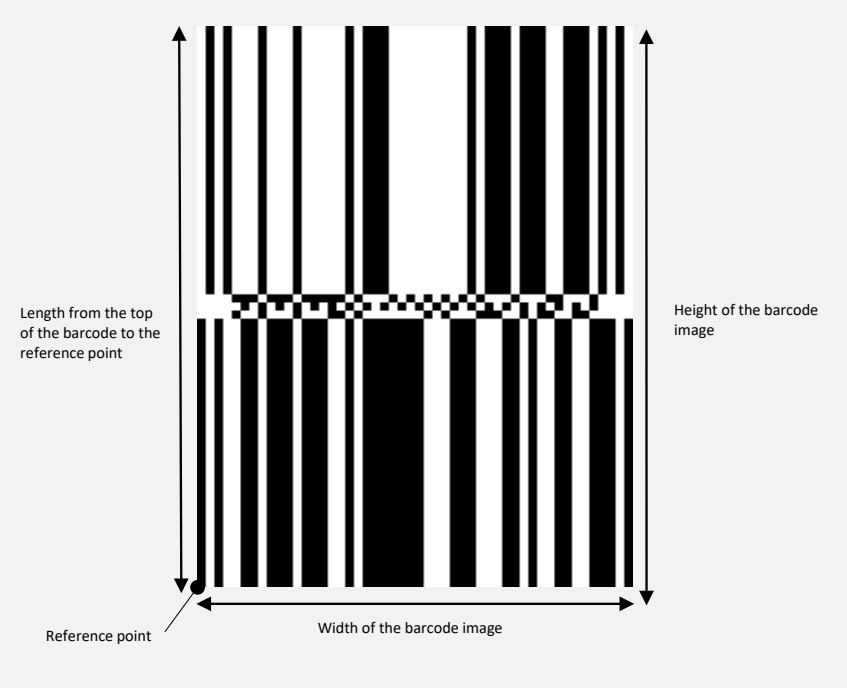
*1: Height of the barcode image = Length from the top of the barcode to the reference point

Width of the barcode image = 50 × module size value

Module Size Value

moduleSize	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[*1] = (moduleHeight × 2 + 3) × module size value

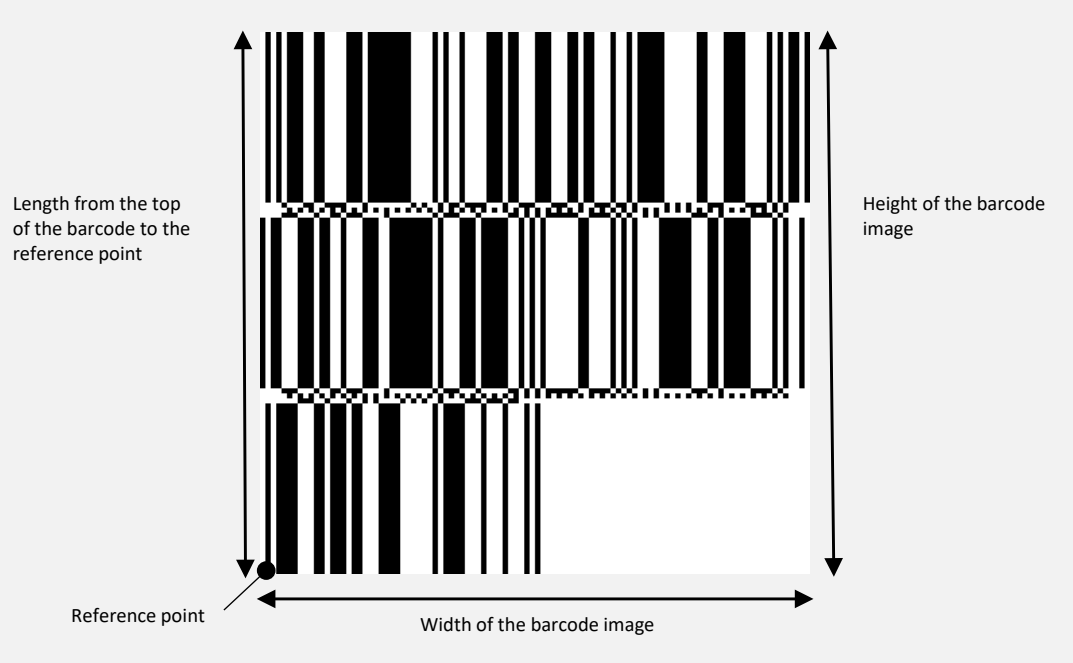
*1: Height of the barcode image = Length from the top of the barcode to the reference point

Width of the barcode image = 50 × module size value

Module Size Value

moduleSize	**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[*1] = ((34 + 3) × number of row[*2] + 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

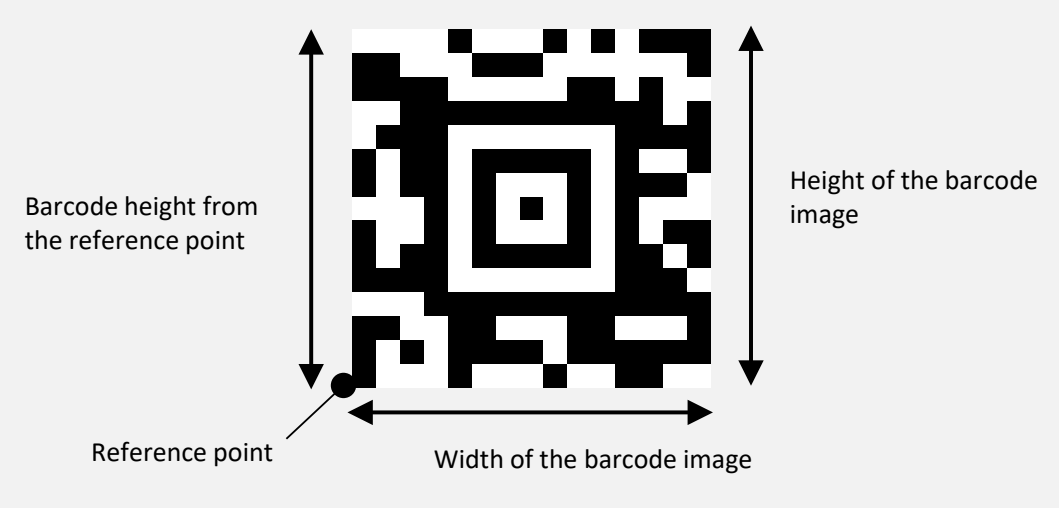
Module Size Value

moduleSize	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.2 Barcode Size List (MP-B30L, MP-B21L)

For MP-B30L and MP-B21L, Aztec Code can be used in addition to the barcode of "B.1 Barcode Size List (SLP720RT, SLP721RT, RP-F10, RP-G10, MP-B30, MP-B30L, MP-B21L)".

B.2.1 `PrintAztecCode, PrintPageModeAztecCode`



(1) Height and width of the barcode image

Height[*1] and width of the barcode image = number of modules + module size value

*1: Height of the barcode image = barcode height from the reference point

Sample: When `aztecSymbol` is **`AZTECCODE_COMPACT`**, `layer` is 1, and `moduleSize` is **`AZTECCODE_MODULE_SIZE_6`**

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

Number of Module

`aztecSymbol`	`layer`	**Number of Module**
`AZTECCODE_FULLRANGE`	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

aztecSymbol	layer	Number of Module
AZTECCODE_FULLRANGE	19	95
	20	101
	21	105
	22	109
	23	113
	24	117
	25	121
	26	125
	27	131
	28	135
	29	139
	30	143
	31	147
	32	151
AZTECCODE_COMPACT	1	15
	2	19
	3	23
	4	27

Module Size Value

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

B.3 Barcode Size List (RP-E10, RP-D10, MP-B20, MP-A40)

B.3.1 `PrintBarcode, PrintPageModeBarcode`



(1) Height of the barcode image

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

(2) Width of the barcode image

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

barcodeSymbol	moduleSize	**Width of Barcode Image**
BARCODE_CODE128	**BARCODE_MODULE_WIDTH_2**	22 × number of barcode data + 66
	BARCODE_MODULE_WIDTH_3	33 × number of barcode data + 99
	BARCODE_MODULE_WIDTH_4	44 × number of barcode data + 132
	BARCODE_MODULE_WIDTH_5	55 × number of barcode data + 165
	BARCODE_MODULE_WIDTH_6	66 × number of barcode data + 198
BARCODE_GS1_OMNI_DIRECTIONAL[*1]	**BARCODE_MODULE_WIDTH_2**	192
	BARCODE_MODULE_WIDTH_3	288
	BARCODE_MODULE_WIDTH_4	384
	BARCODE_MODULE_WIDTH_5	480
	BARCODE_MODULE_WIDTH_6	576
BARCODE_GS1_TRUNCATED[*1]	**BARCODE_MODULE_WIDTH_2**	192
	BARCODE_MODULE_WIDTH_3	288
	BARCODE_MODULE_WIDTH_4	384
	BARCODE_MODULE_WIDTH_5	480
	BARCODE_MODULE_WIDTH_6	576
BARCODE_GS1_LIMITED[*1]	**BARCODE_MODULE_WIDTH_2**	158
	BARCODE_MODULE_WIDTH_3	237
	BARCODE_MODULE_WIDTH_4	316
	BARCODE_MODULE_WIDTH_5	395
	BARCODE_MODULE_WIDTH_6	474
BARCODE_GS1_EXPANDED[*1*2]	**BARCODE_MODULE_WIDTH_2**	number of barcode module × 2
	BARCODE_MODULE_WIDTH_3	number of barcode module × 3
	BARCODE_MODULE_WIDTH_4	number of barcode module × 4
	BARCODE_MODULE_WIDTH_5	number of barcode module × 5
	BARCODE_MODULE_WIDTH_6	number of barcode module × 6

*1: Supported only by MP-B20 and MP-A40.
*2: The number of barcode module is determined by the barcode data to be specified.

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

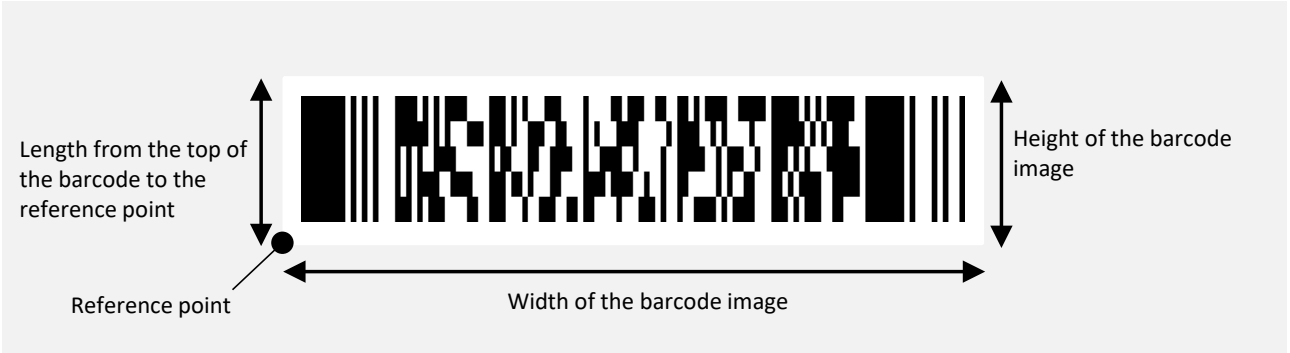
barcodeSymbol	nwRatio	moduleSize	**Width of Barcode Image**
BARCODE_CODE39	**NWRATIO_1TO3**	**BARCODE_MODULE_WIDTH_2**	32 × number of barcode data + 102
		BARCODE_MODULE_WIDTH_3	48 × number of barcode data + 153
		BARCODE_MODULE_WIDTH_4	64 × number of barcode data + 204
		BARCODE_MODULE_WIDTH_5	80 × number of barcode data + 255
		BARCODE_MODULE_WIDTH_6	96 × number of barcode data + 306
BARCODE_ITF	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	14 × number of barcode data + 56
		BARCODE_MODULE_WIDTH_3	21 × number of barcode data + 84
		BARCODE_MODULE_WIDTH_4	28 × number of barcode data + 112
		BARCODE_MODULE_WIDTH_5	35 × number of barcode data + 140
		BARCODE_MODULE_WIDTH_6	42 × number of barcode data + 168
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	16 × number of barcode data + 57
		BARCODE_MODULE_WIDTH_3	25 × number of barcode data + 86
		BARCODE_MODULE_WIDTH_4	32 × number of barcode data + 114
		BARCODE_MODULE_WIDTH_5	41 × number of barcode data + 143
		BARCODE_MODULE_WIDTH_6	48 × number of barcode data + 171
	NWRATIO_1TO3	**BARCODE_MODULE_WIDTH_2**	18 × number of barcode data + 58
		BARCODE_MODULE_WIDTH_3	27 × number of barcode data + 87
		BARCODE_MODULE_WIDTH_4	36 × number of barcode data + 116
		BARCODE_MODULE_WIDTH_5	45 × number of barcode data + 145
		BARCODE_MODULE_WIDTH_6	54 × number of barcode data + 174
BARCODE_CODABAR*1	**NWRATIO_1TO2**	**BARCODE_MODULE_WIDTH_2**	20 × number of data + 2 × (2 + number of wide data) + 38
		BARCODE_MODULE_WIDTH_3	30 × number of data + 3 × (2 + number of wide data) + 57
		BARCODE_MODULE_WIDTH_4	40 × number of data + 4 × (2 + number of wide data) + 76
		BARCODE_MODULE_WIDTH_5	50 × number of data + 5 × (2 + number of wide data) + 95
		BARCODE_MODULE_WIDTH_6	60 × number of data + 6 × (2 + number of wide data) + 114
	NWRATIO_1TO2_5	**BARCODE_MODULE_WIDTH_2**	22 × number of data + 3 × (2 + number of wide data) + 38
		BARCODE_MODULE_WIDTH_3	34 × number of data + 5 × (2 + number of wide data) + 57
		BARCODE_MODULE_WIDTH_4	44 × number of data + 6 × (2 + number of wide data) + 76
		BARCODE_MODULE_WIDTH_5	56 × number of data + 8 × (2 + number of wide data) + 95
		BARCODE_MODULE_WIDTH_6	66 × number of data + 9 × (2 + number of wide data) + 114

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

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

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

B.3.2 PrintPDF417, PrintPageModePDF417



(1) Height of the barcode image

moduleSize	**Height of Barcode Image**[*1]
PDF417_MODULE_WIDTH_2	moduleHeight × row[*2] + 8
PDF417_MODULE_WIDTH_3	moduleHeight × row[*2] + 12
PDF417_MODULE_WIDTH_4	moduleHeight × row[*2] + 16
PDF417_MODULE_WIDTH_5[*3]	moduleHeight × row[*2] + 20
PDF417_MODULE_WIDTH_6[*3]	moduleHeight × row[*2] + 24
PDF417_MODULE_WIDTH_7[*3]	moduleHeight × row[*2] + 28
PDF417_MODULE_WIDTH_8[*3]	moduleHeight × row[*2] + 32

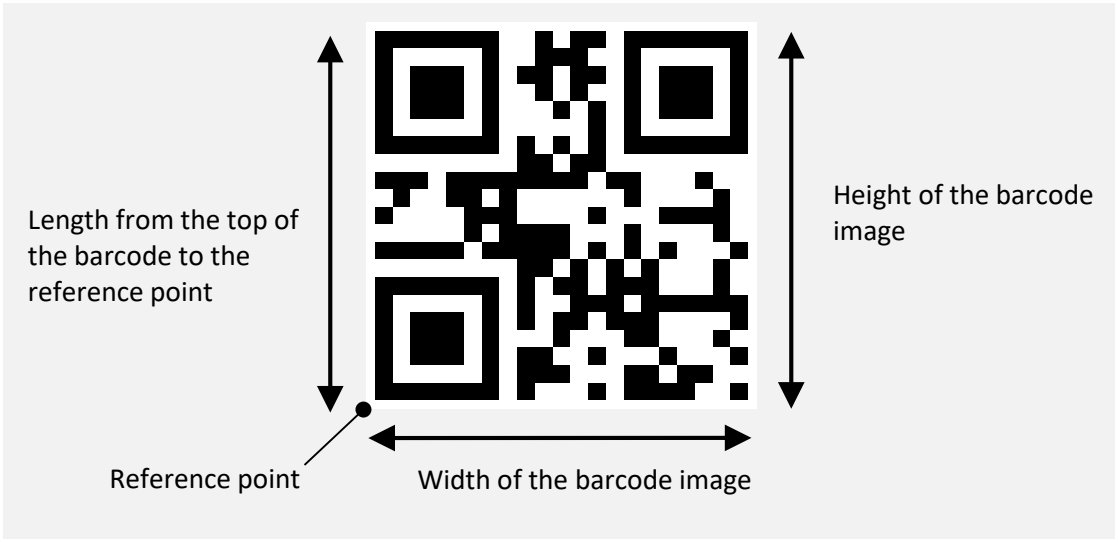
*1: Height of the barcode image = Length from the top of the barcode to the reference point
*2: Parameter row≠0
*3: Supported only by MP-B20 and MP-A40.

(2) Width of the barcode image

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

*1: Parameter column≠0
*2: Supported only by MP-B20 and MP-A40.

B.3.3 PrintQRCode, PrintPageModeQRCode



(1) Height and width of the barcode image

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

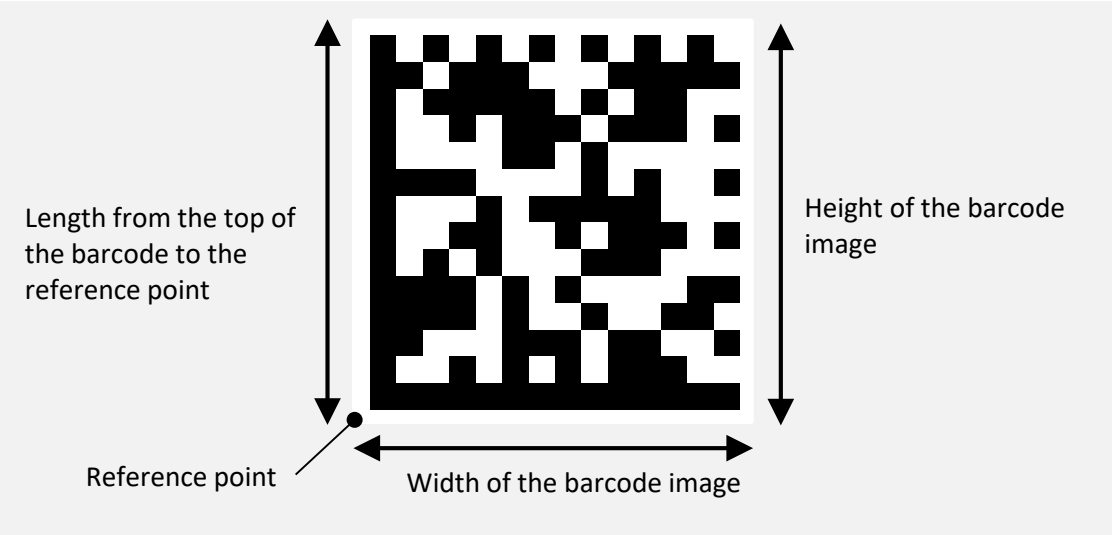
*1: Height of the barcode image = Length from the top of the barcode to the reference point
*2: The version is determined by the content of the barcode data and the error correction level.

Module Size Value

moduleSize	Module Size Value
QR_MODULE_SIZE_2	2
QR_MODULE_SIZE_3	3
QR_MODULE_SIZE_4	4
QR_MODULE_SIZE_5	5
QR_MODULE_SIZE_6	6
QR_MODULE_SIZE_7	7
QR_MODULE_SIZE_8	8
QR_MODULE_SIZE_9	9
QR_MODULE_SIZE_10	10
QR_MODULE_SIZE_11	11
QR_MODULE_SIZE_12[*1]	12
QR_MODULE_SIZE_13[*1]	13
QR_MODULE_SIZE_14[*1]	14
QR_MODULE_SIZE_15[*1]	15
QR_MODULE_SIZE_16[*1]	16

*1: Supported only by MP-B20 and MP-A40.

B.3.4 PrintDataMatrix, PrintPageModeDataMatrix



(1) Height and width of the barcode image

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

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

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

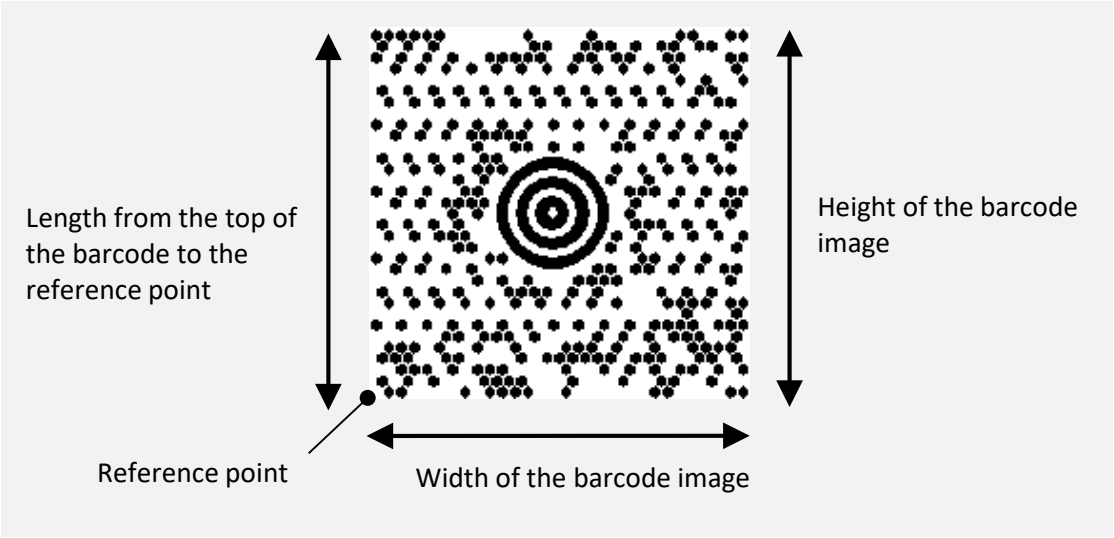
dataMatrixModule	Number of Vertical Module	Number of Horizontal Module
DATA_MATRIX_120_120	120	120
DATA_MATRIX_132_132	132	132
DATA_MATRIX_144_144	144	144
DATA_MATRIX_8_18	8	18
DATA_MATRIX_8_32	8	32
DATA_MATRIX_12_26	12	26
DATA_MATRIX_12_36	12	36
DATA_MATRIX_16_36	16	36
DATA_MATRIX_16_48	16	48

Module Size Value

moduleSize	Module Size Value
DATAMATRIX_MODULE_SIZE_2	2
DATAMATRIX_MODULE_SIZE_3	3
DATAMATRIX_MODULE_SIZE_4	4
DATAMATRIX_MODULE_SIZE_5	5
DATAMATRIX_MODULE_SIZE_6	6
DATAMATRIX_MODULE_SIZE_7	7
DATAMATRIX_MODULE_SIZE_8	8
DATAMATRIX_MODULE_SIZE_9	9
DATAMATRIX_MODULE_SIZE_10	10
DATAMATRIX_MODULE_SIZE_11	11
DATAMATRIX_MODULE_SIZE_12[*1]	12
DATAMATRIX_MODULE_SIZE_13[*1]	13
DATAMATRIX_MODULE_SIZE_14[*1]	14
DATAMATRIX_MODULE_SIZE_15[*1]	15
DATAMATRIX_MODULE_SIZE_16[*1]	16

*1: Supported only by MP-B20 and MP-A40.

B.3.5 PrintMaxicode, PrintPageModeMaxicode



(1) Height of the barcode image

Height of the barcode image*1 = 215

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

(2) Width of the barcode image

Width of the barcode image = 225

B.3.6 `PrintGS1DataBarStacked, PrintPageModeGS1DataBarStacked`

Supported only by MP-B20 and MP-A40.



(1) Height and width of the barcode image

Height of the barcode image[*1] = 13 × module size value

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

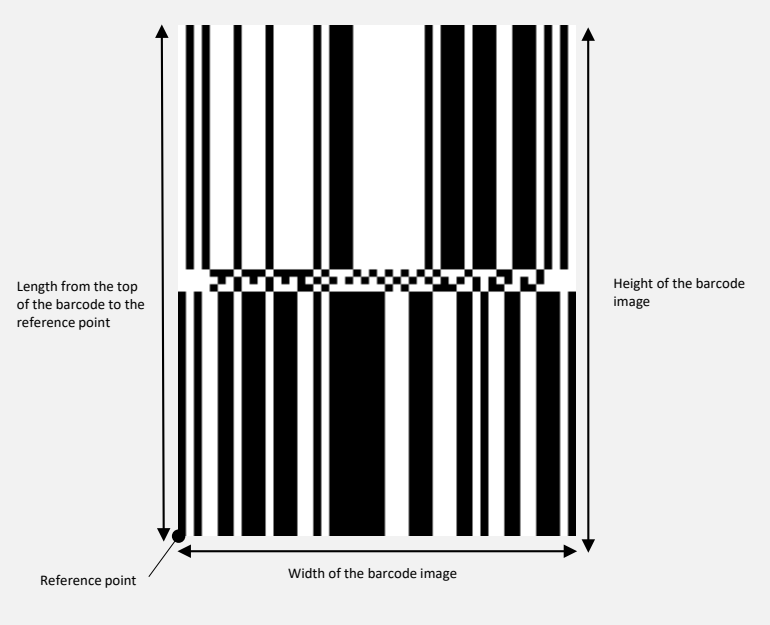
Width of the barcode image = 50 × module size value

Module Size Value

moduleSize	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.3.7 `PrintGS1DataBarStackedOmnidirectional`, `PrintPageModeGS1DataBarStackedOmnidirectional`

Supported only by MP-B20 and MP-A40.



(1) Height and width of the barcode image

Height of the barcode image[*1] = (`moduleHeight` × 2 + 3) × module size value

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

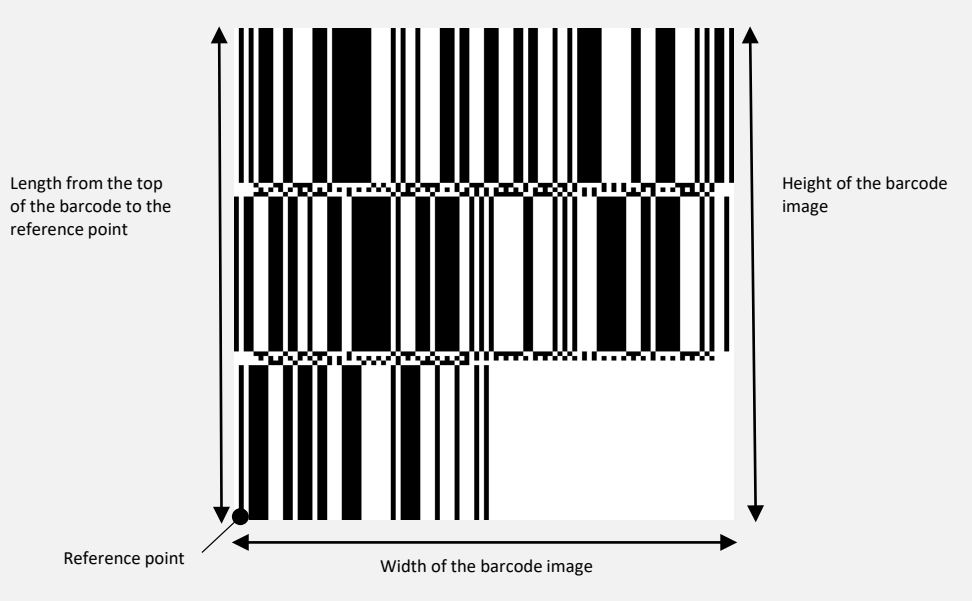
Width of the barcode image = 50 × module size value

Module Size Value

`moduleSize`	**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.3.8 PrintGS1DataBarExpandedStacked, PrintPageModeGS1DataBarExpandedStacked

Supported only by MP-B20 and MP-A40.



(1) Height and width of the barcode image

Height of the barcode image[*1] = ((34 + 3) × number of row[*2] + 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

moduleSize	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

Appendix C
Open Source Software License

This chapter describes the License of open source software used in the library.

C.1 MIT License

- **SSZipArchive**

  Copyright (c) 2010-2012 Sam Soffes

  Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

  The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

C.2 Apache License 2.0

- **zxingify-objc**

  Copyright 2012 ZXing authors

  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

  https://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and limitations under the License.

- **zxing-android-embedded**

  Copyright (C) 2012-2022 ZXing authors, Journey Mobile

  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

  https://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and limitations under the License.