SII POS for .NET Service Object
Application Programmer's Guide

Rev.01

[Products]

MP-B21L Series

MP-B30L Series

MP-B30 Series

Seiko Instruments Inc.

Rev.01 November 2024

Copyright© 2024 by Seiko Instruments Inc.
All rights reserved.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the U.S., Japan, and other countries.
Bluetooth® is a registered trademark of Bluetooth SIG, Inc.

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

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.

SII is a trademark of Seiko Instruments Inc.

Introduction

This manual describes "SII POS for .NET Service Object " (hereinafter referred to as the "software") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

Target Products

The products supported by this manual are listed below.

	Device Name	Description in This Manual
PosPrinter	MP-B21L MP-B30L MP-B30	Printer

See the following when using this software.

・ "UnifiedPOS Retail Peripheral Architecture Version 1.12"
・ "UnifiedPOS Retail Peripheral Architecture Version 1.14.1"
Hereafter, it is referred to as UPOS when it is a common term. When it is specific, the version is listed after UPOS.

・ "Microsoft Point of Service for .NET - POS for .NET v1.12 SDK Documentation" *1
・ "Microsoft Point of Service for .NET - POS for .NET v1.14.1 SDK Documentation"

*1: Not supported in MP-B21L.

Notation in This Manual

The notation in this manual is described.

Operation and Display

In principle, this manual is written on the basis of the following conditions:

- Screenshots and display layouts of Windows 10
- Operating instructions with a mouse and a keyboard

Terms

The terms used in this manual are defined as below.

Term	Description
UPOS	MP-B21L "UnifiedPOS Retail Peripheral Architecture Version 1.14.1" MP-B30L, MP-B30 "UnifiedPOS Retail Peripheral Architecture Version 1.12"
Configuration program	The program that executes addition and setting change of devices for PosPrinter provided by this software. When installing this software, it will be installed as [SIIPOS forNETSettingUtility] on the computer.
Default	The value immediately after satisfying the availability condition.
Line spacing	The height of each print line (total value of the printed line height and the whitespace between each pair of lines).
Technical Reference	Technical Reference shown as follows: ・ MP-B21L SERIES THERMAL PRINTER TECHNICAL REFERENCE ・ MP-B30L SERIES THERMAL PRINTER TECHNICAL REFERENCE ・ MP-B30 SERIES THERMAL PRINTER TECHNICAL REFERENCE
User's Guide	User's Guide shown as follows: ・ MP-B21L SERIES Thermal Printer USER'S GUIDE ・ MP-B30L SERIES Thermal Printer USER'S GUIDE ・ MP-B30 SERIES Thermal Printer USER'S GUIDE
Printer command	Command for controlling the printer described in "Technical Reference".

Table of Contents

Chapter 1 Overview 1-1

1.1 Configuration 1-1
1.1.1 Structural Diagram 1-1
1.2 Printer Settings 1-2
1.3 Limitations 1-4
1.3.1 General 1-4
1.3.2 PosPrinter 1-4

Chapter 2 Installation 2-1

Chapter 3 How to Operate Configuration Program 3-1

3.1 Screen Layout 3-2
3.1.1 Menu Bar 3-2
3.1.2 Device View 3-3
3.1.3 Setting View 3-3
3.2 Functions 3-10
3.2.1 Addition of Device 3-10
3.2.2 Changing Device Settings 3-20
3.2.3 Deletion of Device 3-21
3.2.4 Device Interactive Test 3-22
3.2.5 Memory Settings of Printer (PrinterUtility) 3-24

Chapter 4 Properties, Methods, and Events 4-1

4.1 PosPrinter 4-1
4.1.1 Summary 4-1
4.1.2 Data Characters and Escape Sequences 4-8
4.1.3 Common Properties 4-12
CapCompareFirmwareVersion Property 4-12
CapPowerReporting Property 4-12

CapStatisticsReporting Property 4-12
CapUpdateFirmware Property 4-13
CapUpdateStatistics Property 4-13
CheckHealthText Property 4-13
Claimed Property 4-14
DeviceDescription Property 4-14
DeviceEnabled Property R/W 4-14
DeviceName Property 4-15
FreezeEvents Property R/W 4-15
OutputId Property 4-16
PowerNotify Property R/W 4-16
PowerState Property 4-16
ServiceObjectDescription Property 4-17
ServiceObjectVersion Property 4-17
State Property 4-17
SynchronizingObject Property 4-18
4.1.4 Specific Properties 4-19
AsyncMode Property R/W 4-19
CapCharacterSet Property 4-19
CapCoverSensor Property 4-19
CapMapCharacterSet Property 4-20
CapRec2Color Property 4-20
CapRecBarCode Property 4-20
CapRecBitmap Property 4-20
CapRecBold Property 4-21
CapRecCartridgeSensor Property 4-21
CapRecColor Property 4-21
CapRecDHigh Property 4-21
CapRecDWide Property 4-22
CapRecDWideDHigh Property 4-22
CapRecEmptySensor Property 4-22
CapRecItalic Property 4-22
CapRecLeft90 Property 4-23
CapRecMarkFeed Property 4-23
CapRecNearEndSensor Property 4-23
CapRecPageMode Property 4-24
CapRecPaperCut Property 4-24
CapRecPresent Property 4-24
CapRecRight90 Property 4-24
CapRecRotate180 Property 4-25
CapRecStamp Property 4-25
CapRecUnderline Property 4-25
CapTransaction Property 4-25
CartridgeNotify Property R/W 4-26
CharacterSet Property R/W 4-26
CharacterSetList Property 4-27
CoverOpen Property 4-27
ErrorLevel Property 4-27
ErrorStation Property 4-28

ErrorString Property 4-28
FlagWhenIdle Property R/W 4-29
FontTypefaceList Property 4-29
MapCharacterSet Property R/W 4-29
MapMode Property R/W 4-30
PageModeArea Property 4-31
PageModeDescriptor Property 4-32
PageModeHorizontalPosition Property R/W 4-33
PageModePrintArea Property R/W 4-34
PageModePrintDirection Property R/W 4-34
PageModeStation Property R/W 4-36
PageModeVerticalPosition Property R/W 4-37
RecBarCodeRotationList Property 4-37
RecBitmapRotationList Property 4-38
RecCartridgeState Property 4-38
RecCurrentCartridge Property R/W 4-38
RecEmpty Property 4-39
RecLetterQuality Property R/W 4-39
RecLineChars Property R/W 4-39
RecLineCharsList Property 4-47
RecLineHeight Property R/W 4-48
RecLineSpacing Property R/W 4-49
RecLinesToPaperCut Property 4-49
RecLineWidth Property 4-50
RecNearEnd Property 4-51
RecSidewaysMaxChars Property 4-51
RecSidewaysMaxLines Property 4-52
RotateSpecial Property R/W 4-52
4.1.5 Common Methods 4-53
CheckHealth Method 4-53
Claim Method 4-53
ClearOutput Method 4-54
Close Method 4-54
CompareFirmwareVersion Method 4-54
DirectIO Method 4-54
Open Method 4-56
Release Method 4-56
ResetStatistic(string) Method 4-56
ResetStatistics() Method 4-56
ResetStatistics(StatisticCategories) Method 4-56
ResetStatistics(string[]) Method 4-57
RetrieveStatistic(string) Method 4-57
RetrieveStatistics() Method 4-57
RetrieveStatistics(StatisticCategories) Method 4-57
RetrieveStatistics(string[]) Method 4-57
UpdateFirmware Method 4-58
UpdateStatistic Method 4-58
UpdateStatistics(Statistic[]) Method 4-58
UpdateStatistics(StatisticCategories, Object) Method 4-58

4.1.6 Specific Methods 4-59
ClearPrintArea Method 4-59
MarkFeed Method 4-59
PageModePrint Method 4-62
PrintBarCode Method 4-64
PrintBitmap Method 4-76
PrintImmediate Method 4-78
PrintMemoryBitmap Method 4-78
PrintNormal Method 4-79
RotatePrint Method 4-79
SetBitmap Method 4-81
SetLogo Method 4-82
TransactionPrint Method 4-82
ValidateData Method 4-83
4.1.7 Events 4-84
DirectIOEvent Event 4-84
ErrorEvent Event 4-84
OutputCompleteEvent Event 4-84
StatusUpdateEvent Event 4-85

Appendix A Exceptions A-1

A.1 PosPrinter Exception Error List A-1

Appendix B Statistics B-1

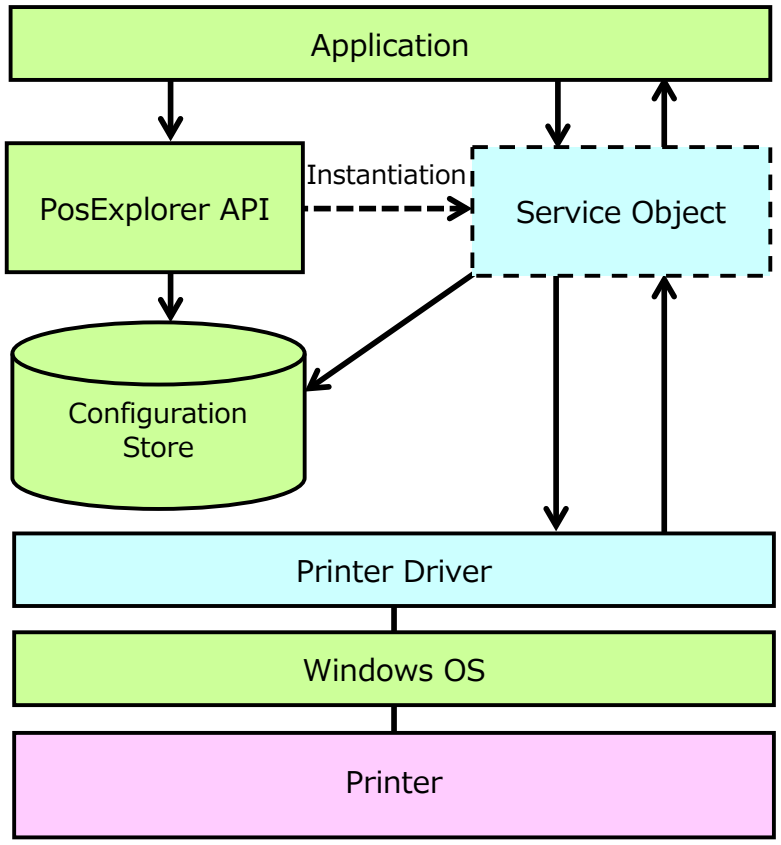
Chapter 1 Overview

This chapter describes the overview of this software.

1.1 Configuration

1.1.1 Structural Diagram

The structure of the software is as follows, and the scope of this manual is indicated by dashed lines.



1.2 Printer Settings

The memory switches of the printer are set to [Value] in the following table forcibly by executing **Claim** when using the software.
See "User's Guide" for details about the memory switches.

- MP-B21L, MP-B30L

MS	Function	Value	Note
1-2	Mark/Gap Mode Selection (Mark/Gap Mode)	0 : Enable 1 : Disable	Either of [Value] on the left can be set by [MarkMode] in the configuration program.
1-3 to 1-5	Command System Selection (Command System)	000B : ESC/POS	-
1-6	Data Discard Selection When Error Occurs (Error Through)	MP-B21L 0 : Enable	-
		MP-B30L 0 : Enable 1 : Disable	When adding a device in the configuration program, if PrinterDriver is specified, it will be set to "1: Disable". If anything else is specified, it will be set to "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	
2-3 to 2-4	Print Quality Selection (Print Quality)	01B : Quality 2 10B : Quality 1 11B : Standard	Any one of [Value] on the left can be set by [PrintSpeed] in the configuration program.
3-1 to 3-8	Print Width (Print Width)	MP-B21L 00010101B : 21mm/168dots 00010110B : 22mm/176dots 00011000B : 24mm/192dots 00011010B : 26mm/208dots 00011100B : 28mm/224dots 00011110B : 30mm/240dots 00100000B : 32mm/256dots 00100010B : 34mm/272dots 00100100B : 36mm/288dots 00100110B : 38mm/304dots 00101000B : 40mm/320dots 00101010B : 42mm/336dots 00101100B : 44mm/352dots 00101101B : 45mm/360dots 00101110B : 46mm/368dots 00110000B : 48mm/384dots	Any one of [Value] on the left can be set by [Number of Effective Dots(dots)] in the configuration program.

MS	Function	Value	Note
		MP-B30L 00101101B : 45mm/360dots 00101110B : 46mm/368dots 00110000B : 48mm/384dots 00110010B : 50mm/400dots 00110100B : 52mm/416dots 00110110B : 54mm/432dots 00111000B : 56mm/448dots 00111010B : 58mm/464dots 00111100B : 60mm/480dots 00111110B : 62mm/496dots 01000000B : 64mm/512dots 01000010B : 66mm/528dots 01000100B : 68mm/544dots 01000110B : 70mm/560dots 01001000B : 72mm/576dots	
9-1	Automatic Status Response Selection (Auto Status Back)	0 : Enable	-
9-2	Initialized Response Selection (Init. Response)	0 : Enable	

Reference

- For MP-B21L, set the memory switch MS 2-8 (Paper Width) of the printer correctly according to the paper to be used. See the "MP-B21L SERIES Thermal Printer USER'S GUIDE" for details of the memory switch.

- MP-B30

MS	Function	Value	Note
1-2	Mark Mode Selection (Mark Mode)	0 : Enable 1 : Disable	Either one of [Value] on the left can be set by [MarkMode] in the configuration program.
1-6	Data Discard Selection When Error Occurs (Error Through)	0 ; Enable 1 : Disable	When adding a device in the configuration program, if PrinterDriver is specified, it will be set to "1: Disable". If anything else is specified, it will be set to "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	

MS	Function	Value	Note
2-3 to 2-4	Print Quality Selection (Print Quality)	01B : Quality 2 10B : Quality 1 11B : Standard	Any one of [Value] on the left can be set by [PrintSpeed] in the configuration program.
9-1	Automatic Status Response Selection (Auto Status Back)	0 : Enable	-
9-2	Initialized Response Selection (Init. Response)	0 : Enable	

1.3 Limitations

The limitations of this software are described.

1.3.1 General

When using 1 printer simultaneously from multiple computers via TCP/IP connections, use **TransactionPrint** to prevent print data from other computers from interrupting.

1.3.2 PosPrinter

This software is based on UnifiedPOS Specification Version 1.14.1 (MP-B21L) or 1.12 (MP-B30L, MP-B30), and all interfaces of PosPrinter device are provided with the following limitations.

(a) The method and property settings related to journal and slip prints are not supported.

(b) The following functions are not supported.

- Paper cut
- Feed and Paper cut
- Feed, Paper cut and Stamp
- Stamp
- Feed reverse
- Print in-line ruled line
- Font typeface selection
- Italic
- Alternate color (Custom)
- Red color
- Shading
- RBG Color
- Sub Script
- Super Script
- Strike-through

(c) All the following methods always return *ErrorCode.Illegal* after they are enabled.

- **BeginInsertion**
- **BeginRemoval**
- **ChangePrintSide**
- **CutPaper**
- **EndInsertion**
- **EndRemoval**
- **PrintTwoNormal**

(d) **DirectIOEvent** (device-specific event) is not supported.

(e) When [ProcessCompletionTiming] is set as "Printed" in the configuration program, the printer command "Execution Response Request" is used inside this software for controlling the print operation. Therefore, an unexpected behavior may occur when sending "Execution Response Request" by the "Pass through embedded data" escape sequence (ESC|[*]#E).

(f) When an error occurs, the printer command "Hardware Reset" is sent from PosPrinter to cancel printing in the printer; however, printing may be performed a bit before PosPrinter stops printing in the printer.

(g) For Bluetooth connection, when the device becomes a recoverable error during printing, it may take some time to return to normal status from the error cancellation. The first processing after error cancelation should be done after about 10 seconds from the error cancellation.

Chapter 2 Installation

For the installation, see "SII Software Package for Windows Installation Guide".

Chapter 3 How to Operate Configuration Program

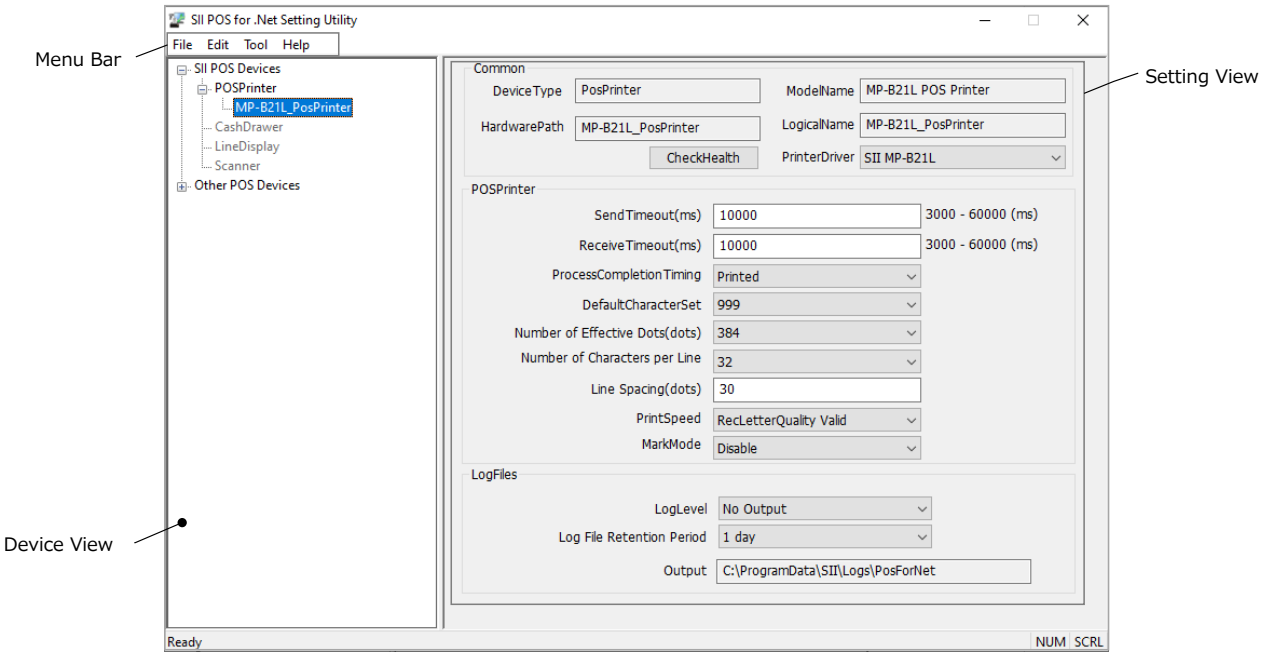
This chapter describes the configuration program provided by this software.

Caution

this software requires logon to the computer with administrator privileges.Using this software requires logon to the computer with administrator privileges.

3.1 Screen Layout

The screen layout of the configuration program is described.



Item	Description
Menu Bar	The menu bar of the configuration program. See "3.1.1 Menu Bar" for items in the menu bar.
Device View	The type, the name, and the logical name of the device registered in the system are displayed in a tree.
Setting View	Displays setting contents of the device selected in "Device View". See "3.1.3(1) POSPrinter" for items of each device.

3.1.1 Menu Bar

Item		Description
File	Exit	Ends the configuration program.
Edit	Add Device…	Adds a new device.
	Delete	Deletes the selected device.
Tool	CheckHealth	Executes an interactive test on the selected device.
Help(H)	About Utility…	Displays the version information of the configuration program.

3.1.2 Device View

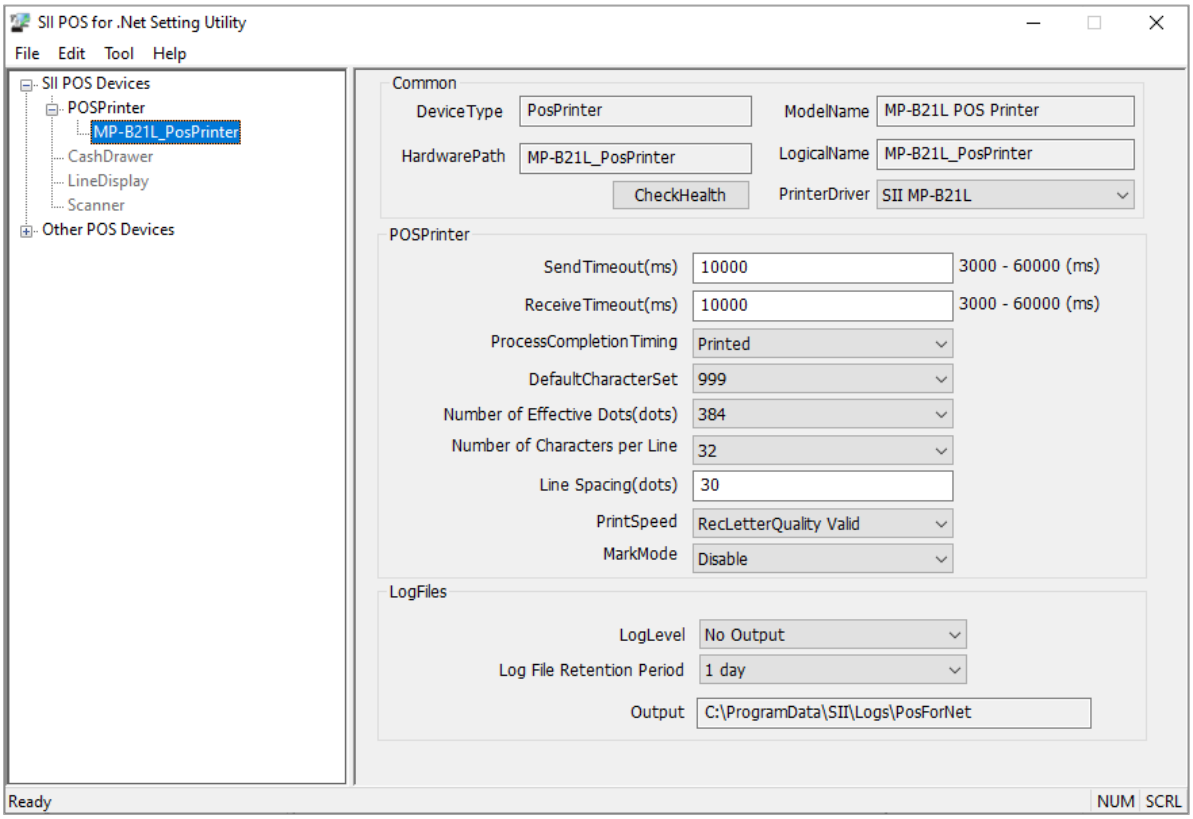
Name	Description
SII POS Devices	Displays SII devices. When the logical name is selected in "Device View", setting contents of the device can be changed or deleted.
Other POS Devices	Displays devices other than SII devices. Device settings cannot be changed or deleted.

3.1.3 Setting View

(1) POSPrinter

- Setting items when selecting the printer driver

The items and their setting contents of the "Setting View" displayed by selecting a printer driver when adding a device are described below.
The figure shows MP-B21L selected in the initial setting.



Item	Description	Setting Content (" " : Default)
Common		
DeviceType	Device type	[PosPrinter]
ModelName	Model name	MP-B21L: MP-B21L POS Printer MP-B30L: MP-B30L POS Printer MP-B30: MP-B30 POS Printer
HardwarePath	Set automatically.	-
LogicalName	Logical name	-
PrinterDriver	Printer driver name	-

Item	Description	Setting Content (" " : Default)
CheckHealth	Executes an interactive test for the selected device.	
PosPrinter		
Send Timeout(ms)	Send timeout value in communication with the printer (milliseconds)	3000 to 60000 (10000)
Receive Timeout(ms)	Receive timeout value in communication with the printer (milliseconds)	3000 to 60000 (10000)
ProcessCompletion Timing	Timing of method completion	Data Transmitted (Completion of data transmission) Printed (Completion of printing)
DefaultCharacterSet	Character set type **CharacterSet** is initialized with this value. See **CharacterSet** for details.	437 737 850 852 855 857 858 860 863 865 866 932*1 999*2 1250 1251 1252 1253 1254
Number of Effective Dots(dots)	Number of effective dots per line. **RecLineWidth** is initialized with this value.	MP-B21L 168 176 192 208 224 240 256 272 288 304 320 336 352 360 368 384

Item	Description	Setting Content (" " : Default)
Number of Effective Dots(dots)	Number of effective dots per line. **RecLineWidth** is initialized with this value.	MP-B30L 360 368 384 400 416 432 448 464 480 496 512 528 544 560 576
Number of Characters per Line	Number of 1-byte characters per line **RecLineChars** is initialized with this value.	MP-B21L 168 dots / 10,12,14,16,18,21 176 dots / 11,12,13,14,17,19,22 192 dots / 12,13,14,16,19,21,24 208 dots / 13,14,16,17,20,23,26 224 dots / 14,16,17,18,22,24,28 240 dots / 14,16,17,18,24,26,30 256 dots / 16,18,19,21,25,28,32 272 dots / 17,19,20,22,27,30,34 288 dots / 18,20,22,24,28,32,36 304 dots / 19,21,23,25,30,33,38 320 dots / 20,22,24,26,32,35,40 336 dots / 21,24,25,28,33,37,42 352 dots / 22,25,27,30,35,39,44 360 dots / 22,25,27,30,36,40,45 368 dots / 23,26,28,30,36,40,46 384 dots / 24,27,29,32,38,42,48 MP-B30L 360 dots / 22,25,27,30,36,40,45 368 dots / 23,26,28,30,36,40,46 384 dots / 24,27,29,32,38,42,48 400 dots / 25,28,30,33,40,44,50 416 dots / 26,29,32,34,41,46,52 432 dots / 27,30,33,36,43,48,54 448 dots / 28,32,34,37,44,49,56 464 dots / 29,33,35,38,46,51,58 480 dots / 30,34,36,40,48,53,60 496 dots / 31,35,38,41,49,55,62 512 dots / 32,36,39,42,51,56,64 528 dots / 33,37,40,44,52,58,66 544 dots / 34,38,41,45,54,60,68 560 dots / 35,40,43,46,56,62,70 576 dots / 36,41,44,48,57,64,72 MP-B30 576 dots / 36,41,44,48,57,64,72

Item	Description	Setting Content (" " : Default)
Line Spacing(dots)	Line spacing per line Settable range: The settable minimum value differs depending on the selected values of effective dots and characters per line. **RecLineSpacing** is initialized with this value.	MP-B21L Settable range: 24 to 255 (30) Number of valid dots / Number of Characters per Line 168 dots / 10,12,14 176 dots / 11,12,13,14 192 dots / 12,13,14,16 208 dots / 13,14,16,17 224 dots / 14,16,17,18 240 dots / 14,16,17,18 256 dots / 16,18,19,21 272 dots / 17,19,20,22 288 dots / 18,20,22,24 304 dots / 19,21,23,25 320 dots / 20,22,24,26 336 dots / 21,24,25,28 352 dots / 22,25,27,30 360 dots / 22,25,27,30 368 dots / 23,26,28,30 384 dots / 24,27,29,32 Settable range: 16 to 255 Number of valid dots / Number of Characters per Line 168 dots / 16,18,21 176 dots / 17,19,22 192 dots / 19,21,24 208 dots / 20,23,26 224 dots / 22,24,28 240 dots / 24,26,30 256 dots / 25,28,32 272 dots / 27,30,34 288 dots / 28,32,36 304 dots / 30,33,38 320 dots / 32,35,40 336 dots / 33,37,42 352 dots / 35,39,44 360 dots / 36,40,45 368 dots / 36,40,46 384 dots / 38,42,48

Item	Description	Setting Content (" " : Default)
Line Spacing(dots)	Line spacing per line Settable range: The settable minimum value differs depending on the selected values of effective dots and characters per line. **RecLineSpacing** is initialized with this value.	MP-B30L Settable range: 24 to 255 (30) Number of valid dots / Number of Characters per Line 360 dots / 22,25,27,30 368 dots / 23,26,28,30 384 dots / 24,27,29,32 400 dots / 25,28,30,33 416 dots / 26,29,32,34 432 dots / 27,30,33,36 448 dots / 28,32,34,37 464 dots / 29,33,35,38 480 dots / 30,34,36,40 496 dots / 31,35,38,41 512 dots / 32,36,39,42 528 dots / 33,37,40,44 544 dots / 34,38,41,45 560 dots / 35,40,43,46 576 dots / 36,41,44,48 Settable range: 16 to 255 Number of valid dots / Number of Characters per Line 360 dots / 36,40,45 368 dots / 36,40,46 384 dots / 38,42,48 400 dots / 40,44,50 416 dots / 41,46,52 432 dots / 43,48,54 448 dots / 44,49,56 464 dots / 46,51,58 480 dots / 48,53,60 496 dots / 49,55,62 512 dots / 51,56,64 528 dots / 52,58,66 544 dots / 54,60,68 560 dots / 56,62,70 576 dots / 57,64,72 MP-B30 Settable range: 24 to 255 (30) Number of valid dots / Number of Characters per Line 576 dots / 36,41,44,48 Settable range: 16 to 255 Number of valid dots / Number of Characters per Line 576 dots / 57,64,72
PrintSpeed	Print speed of the printer It is decided by **RecLetterQuality** when selecting RecLetterQuality Valid.	RecLetterQuality Valid Standard[*3] Quality 1[*3] Quality 2[*3]
MarkMode	Mark detection mode Selects whether to enable or disable mark detection.	Disable: Mark detection disable Enable: Mark detection enable

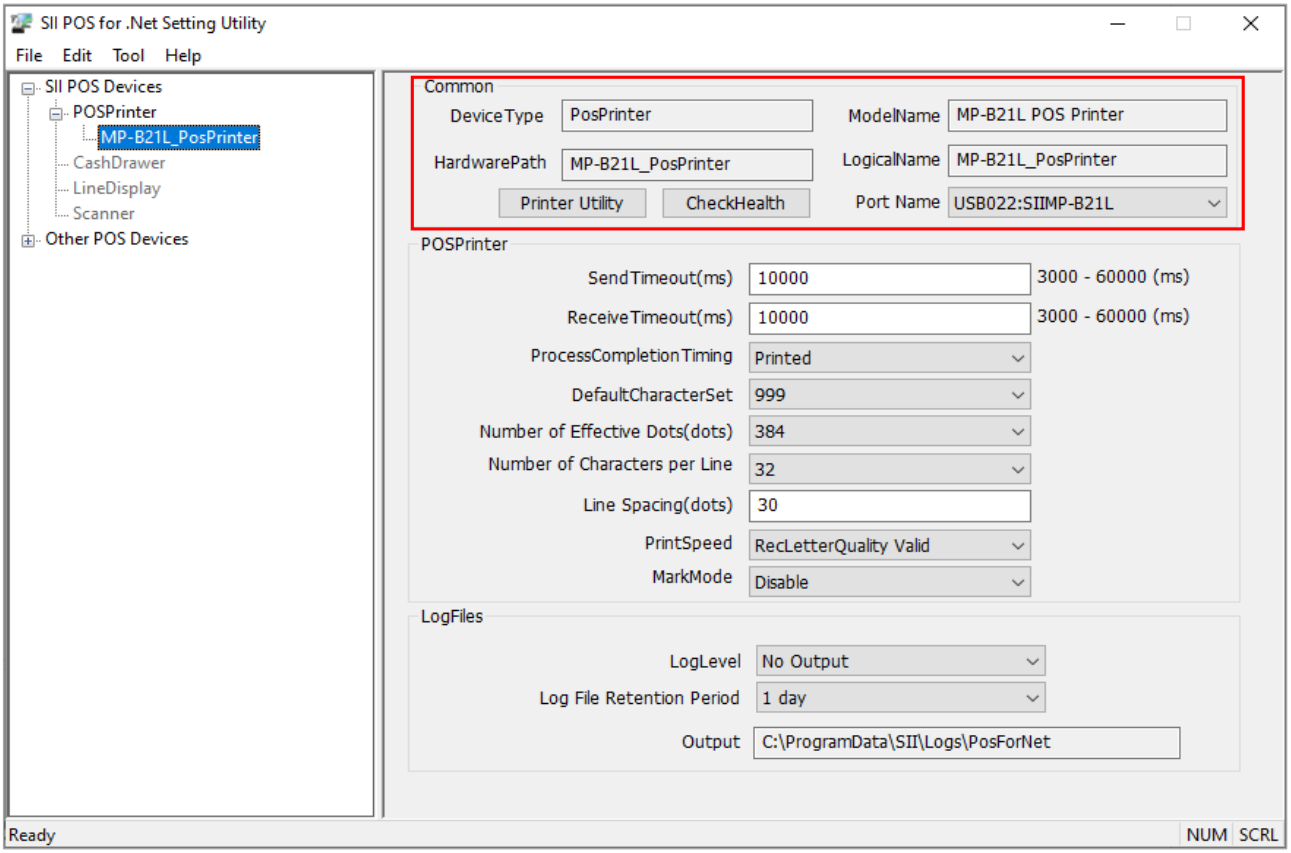
Item	Description	Setting Content (" " : Default)
LogFiles		
LogLevel	Log output level	No Output: No logs are output. Error: Error logs at execution are output. Info: Error logs at execution and highlighted event at execution are output. Debug: Error logs at execution, highlighted event at execution and more detailed information for debugging are output.
Log File Retention Period	Retention period for log files Log files past the retention period are deleted when logs are output. The actual retention period may be longer by 1 day at maximum.	1 day 3 days 10 days 30 days 90 days
Output	Log output directory (Unchangeable) The log output directory and the file name are as follows. Output Directory: <System Drive>:\ProgramData\SII\Logs\PosForNet The output directory cannot be changed. File Name: <yyyyMMdd>.log The maximum size of the log file is 32 MB. When the log file exceeds the maximum size, the file name is changed to <yyyyMMdd_hhmmssfff)>.log, and a new <yyyyMMdd>.log is created.*4	

*1: Default for Japanese
*2: Default for English
*3: See "User's Guide" for details of the print speed.
*4: The symbols used for the file name mean as follows. Each value comes from the Windows system clock.

yyyy :	Year
MM :	Month
dd :	Day
hh :	Hour
mm :	Minute
ss :	Second
fff :	Millisecond

- Setting items when selecting the port

The items and their setting contents of the "Setting View" displayed by selecting the port when adding a device are described below.



The difference between the settings in the "Settings View" when the port is selected and the setting items displayed when the printer driver is selected is only Common.
This section describes the items of Common and their settings.

Item	Description	Setting Content (" " : Default)
Common		
DeviceType	Device type	[PosPrinter]
ModelName	Model name	MP-B21L: MP-B21L POS Printer MP-B30L: MP-B30L POS Printer MP-B30: MP-B30 POS Printer
HardwarePath	Set automatically.	-
LogicalName	Logical name	-
Port Name	Port name for the selected device.	-
Printer Utility	Configuration program	-
CheckHealth	Executes an interactive test for the selected device.	

3.2 Functions

The functions of the configuration program are described.
The procedure is described using MP-B21L as an example.

3.2.1 Addition of Device

The procedure for adding a device is described.

When the configuration program is started up immediately after installing this software, a device needs to be added since no device has been added.

Maximum 8 printers can be added.

Caution

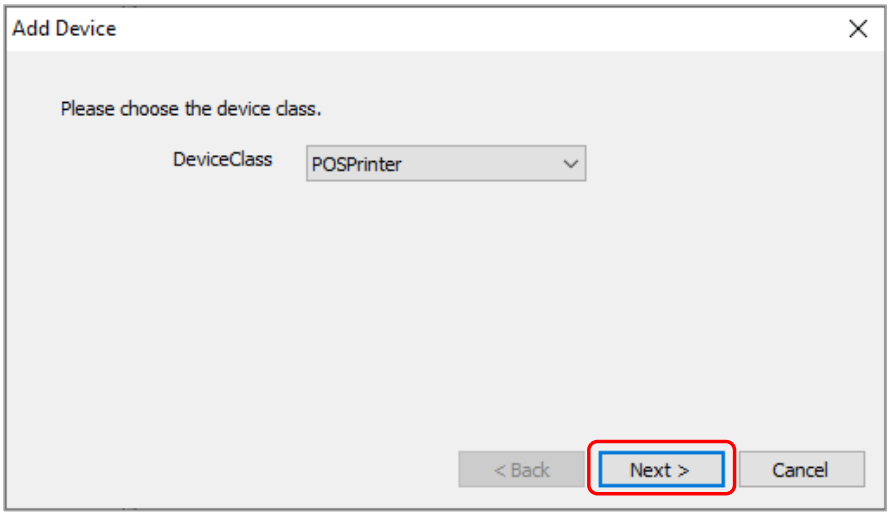
- When selecting USB for the interface, connect to the port in advance before starting up the configuration program.
- When selecting Bluetooth for the interface, pair the printer in advance before starting up the configuration program.
- When selecting LAN/WLAN for the interface, create the LAN port in advance before starting up the configuration program.
- The printer driver is required to be installed on the computer when selecting the PrinterDriver. See "SII Software Package for Windows Installation Guide" for installing the printer driver.

(1) Addition of Printer

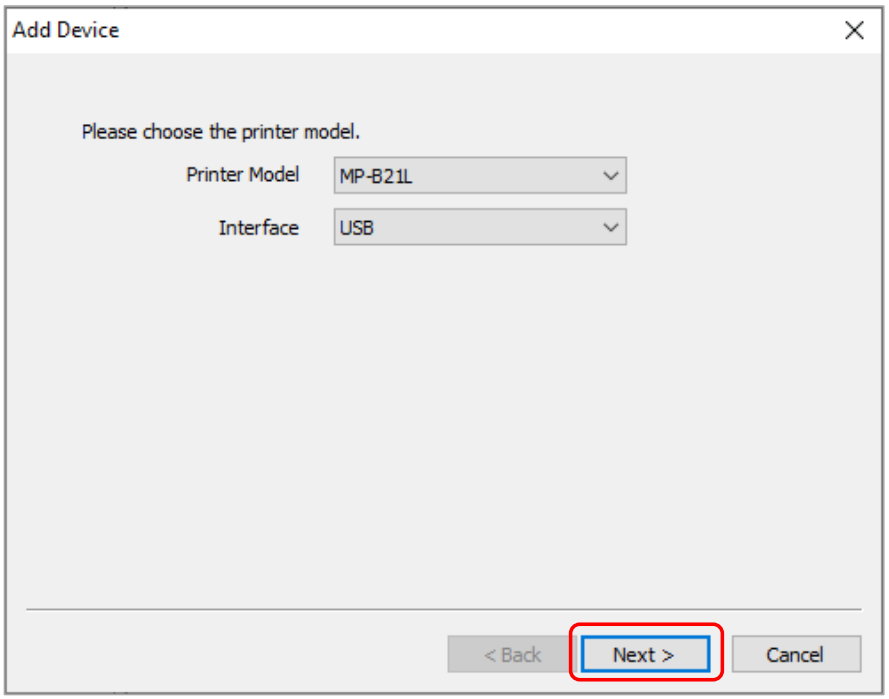
1) When the configuration program starts, the following window is displayed. Select the [Edit] - [Add Device…] button from "Menu Bar".

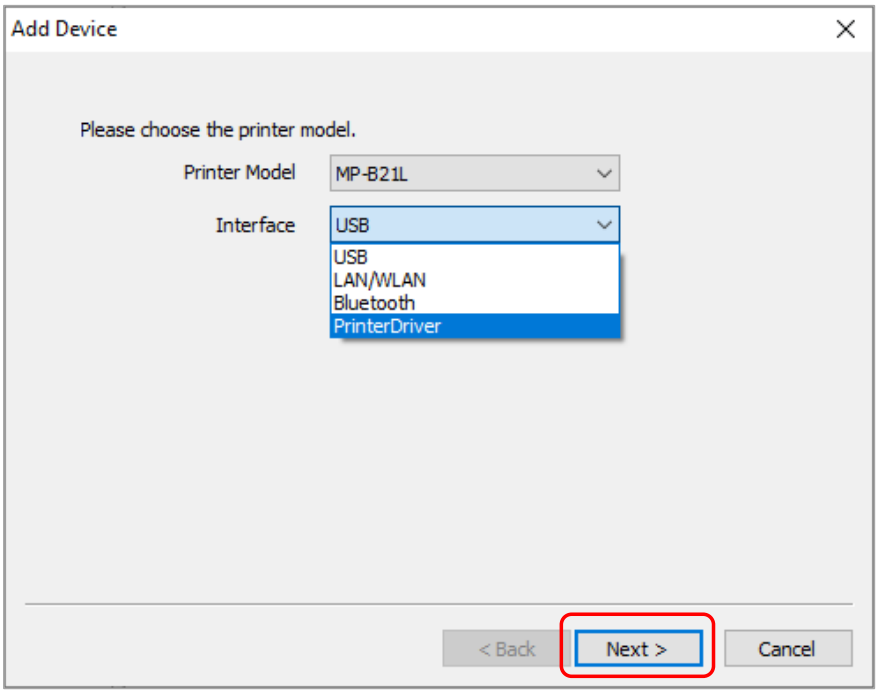


2) Select "POSPrinter" for [DeviceClass], and then click the [Next >] button.



3) Select the printer to be added from [Printer Model], and click the [Next >] button. Select the printer driver or interface from [Interface] and click the [Next >] button.



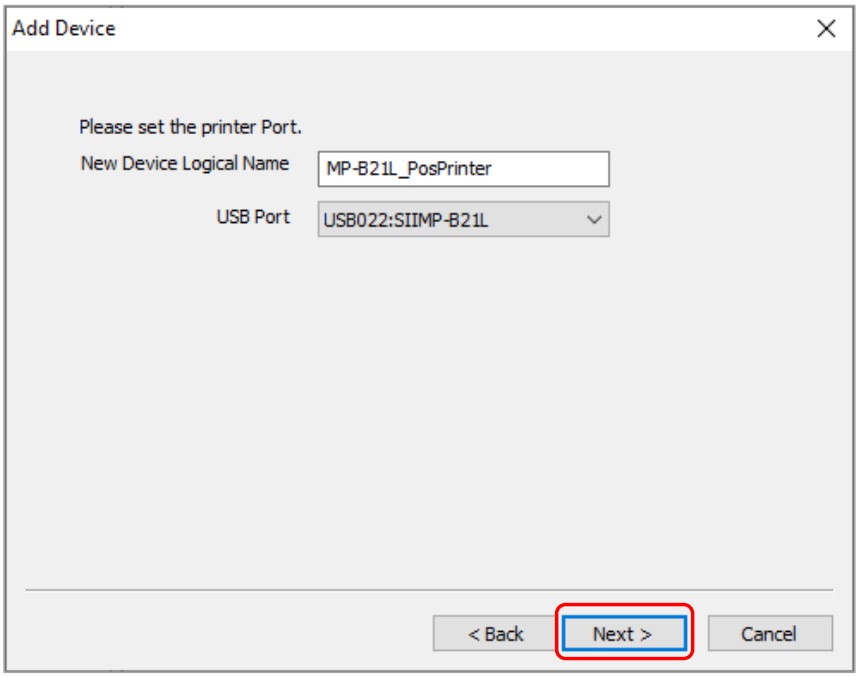


The following procedures vary depending on the selection of [Interface].

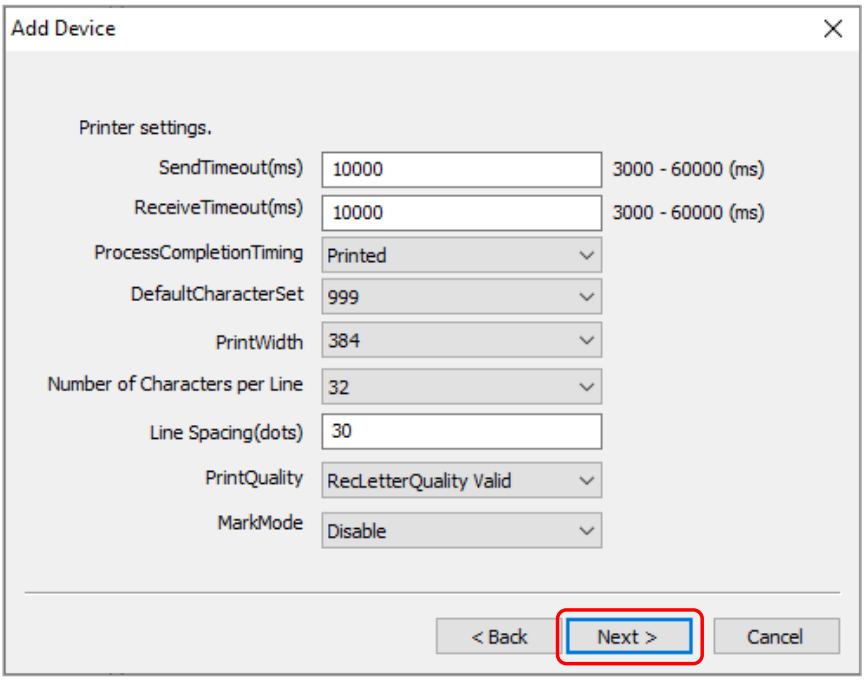
- Selecting USB: See "**When selecting the USB**".
- Selecting Bluetooth: See "**When selecting the Bluetooth**".
- Selecting LAN/WLAN: See "**When selecting the LAN/WLAN**".
- Selecting printer driver: See "**When selecting the printer driver**".

When selecting the USB

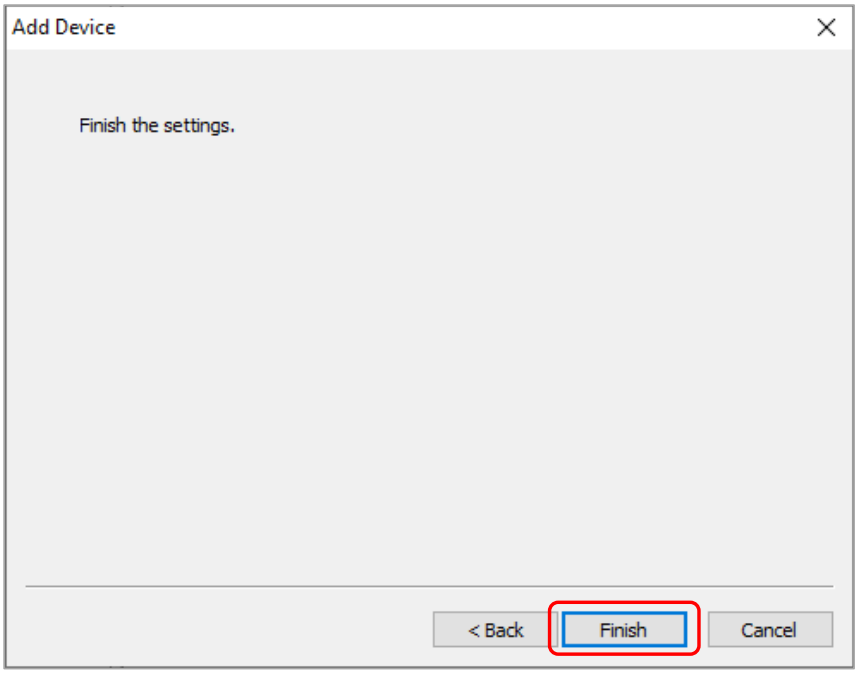
4) Select the port to be used from [USB Port], and then click the [Next >] button. [New Device Logical Name] is set automatically.



5) Enter or select the settings of the printer, and then click the [Next >] button.



6) Click the [Finish] button.

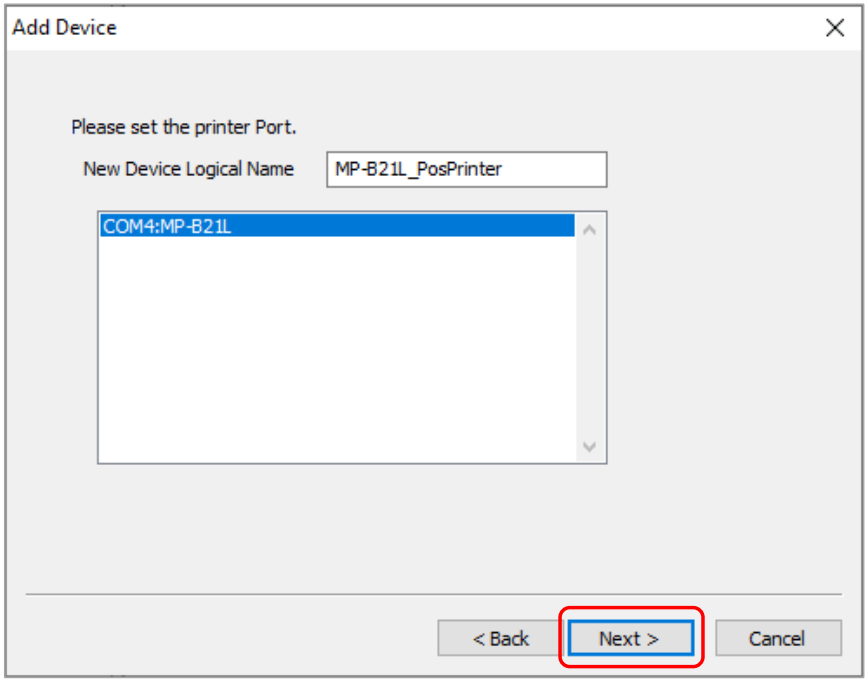


When selecting the Bluetooth

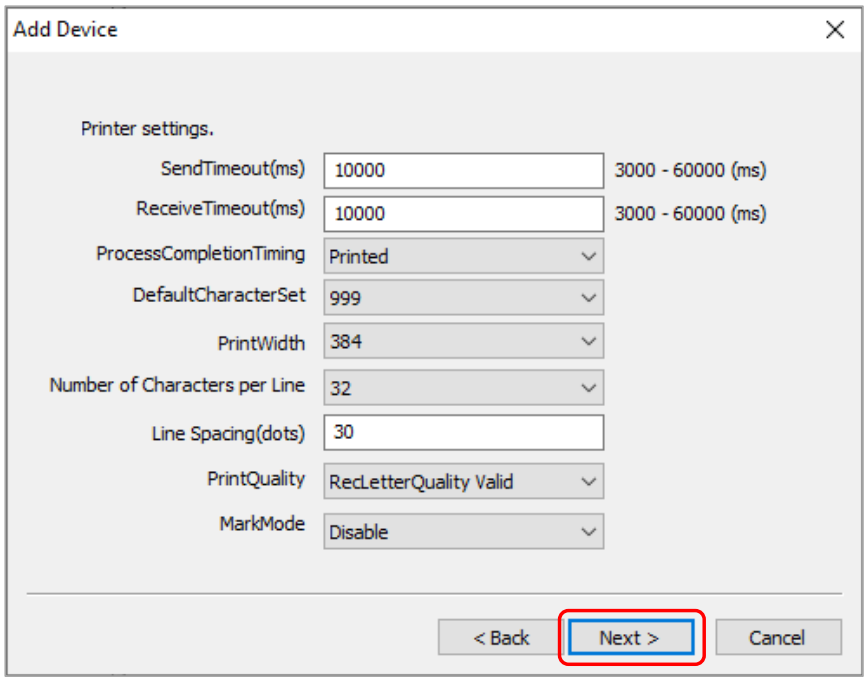
Caution

◆ Bluetooth connection cannot be used at the same time by multiple hosts.

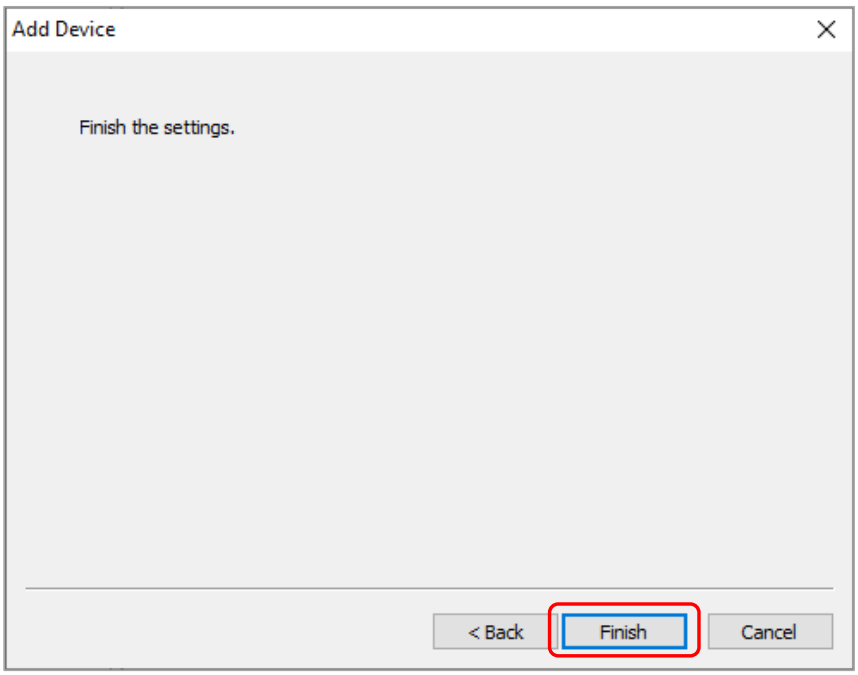
4) Printers that have been paired once will be shown, select a printer and click the [Next >] button. The [New Device Logical Name] is set automatically.



5) Enter or select the settings of the printer, and then click the [Next >] button.

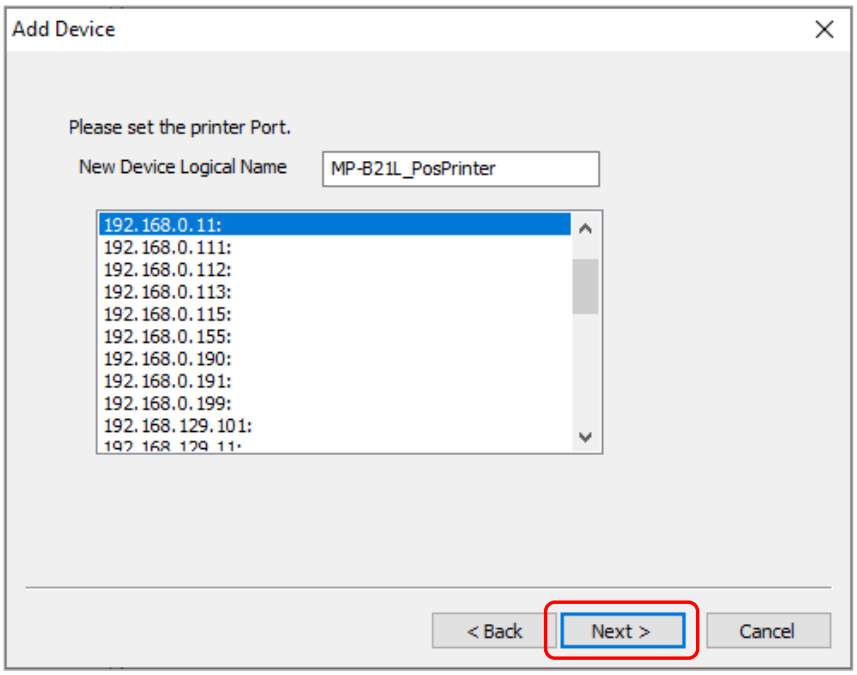


6) Click the [Finish] button.

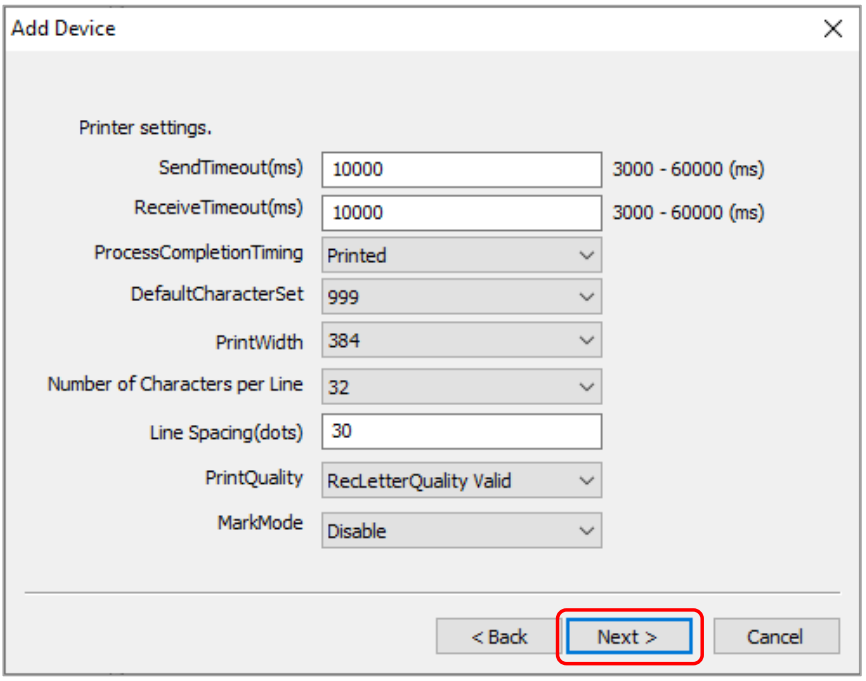


When selecting the LAN/WLAN

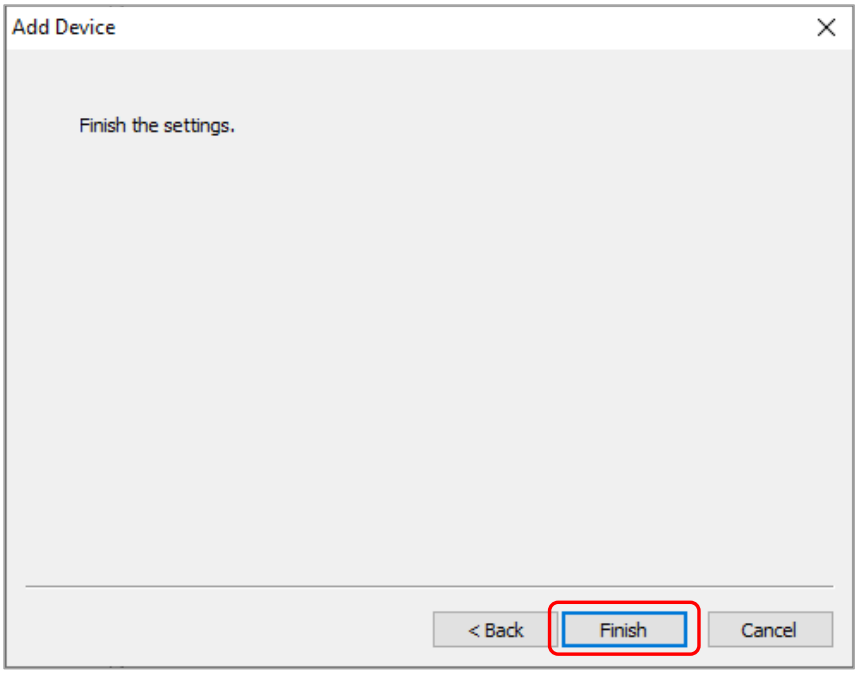
4) The printer will be shown, select the printer and click the [Next >] button. The [New Device Logical Name] is set automatically.



5) Enter or select the settings of the printer, and then click the [Next >] button.

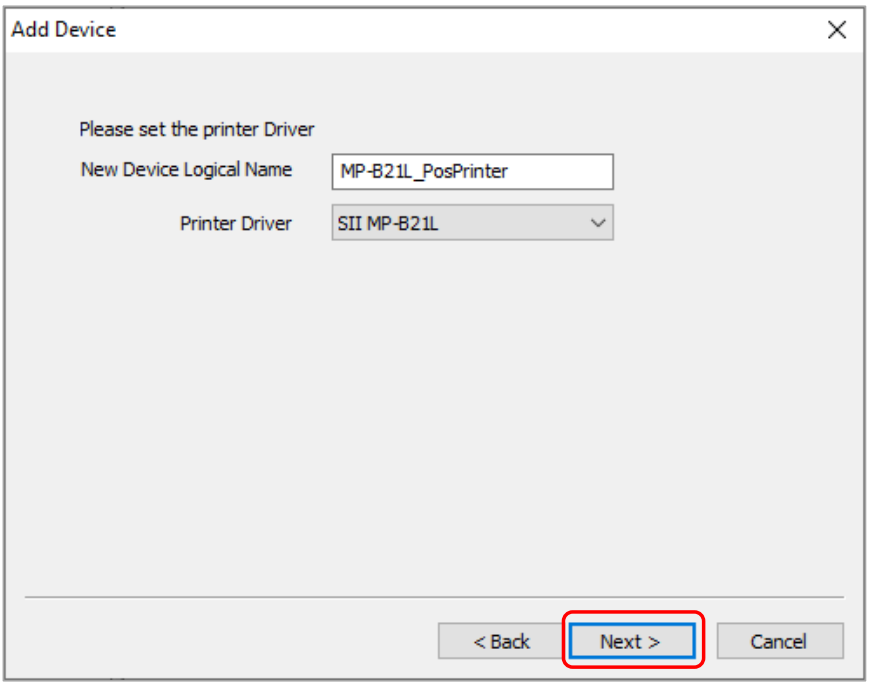


6) Click the [Finish] button.

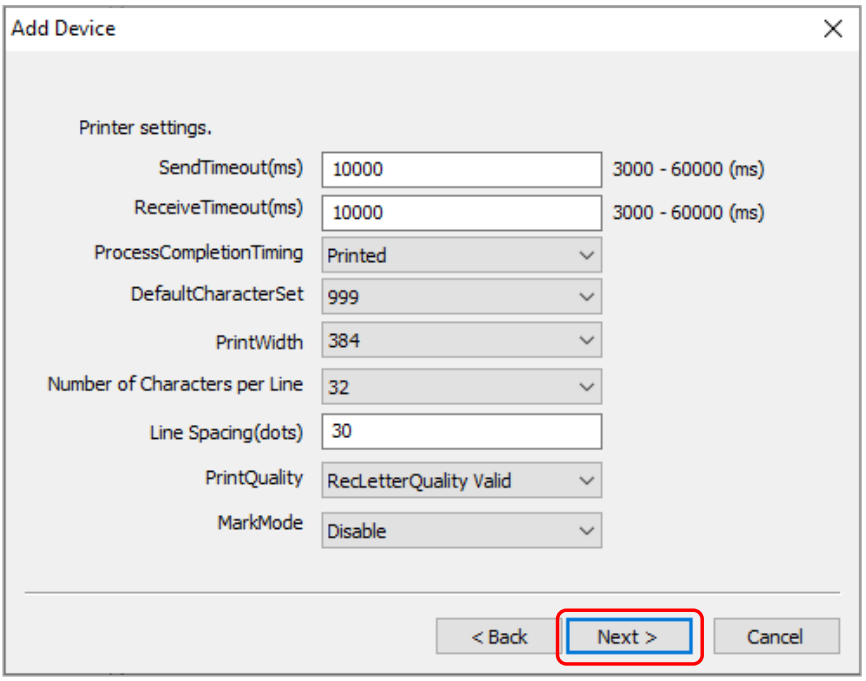


When selecting the printer driver

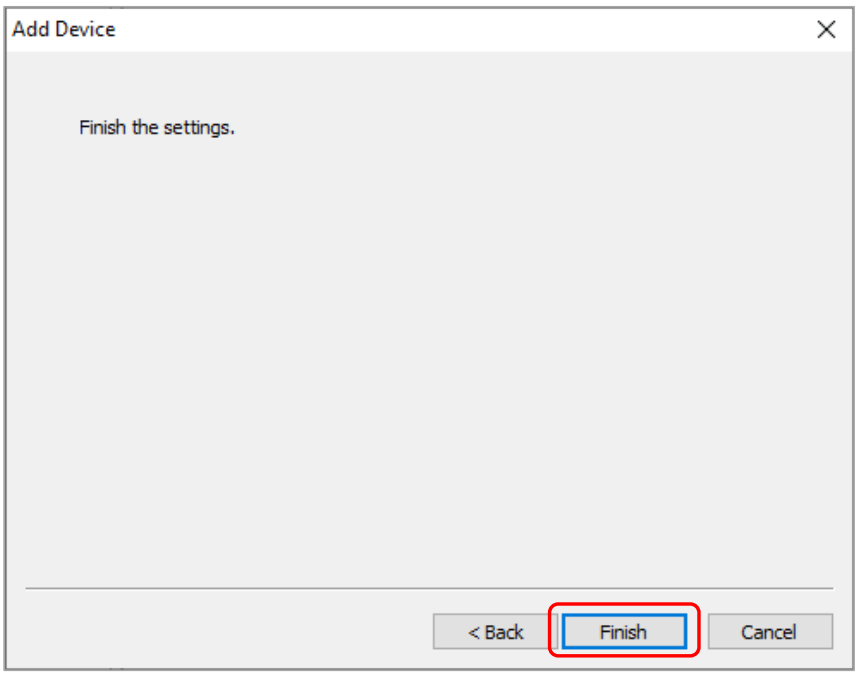
4) Select the printer driver from [Printer Driver], and then click the [Next >] button. [New Device Logical Name] is set automatically.



5) Enter or select the settings of the printer, and then click the [Next >] button.



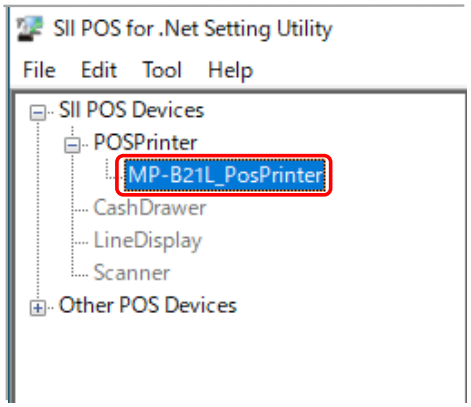
6) Click the [Finish] button.



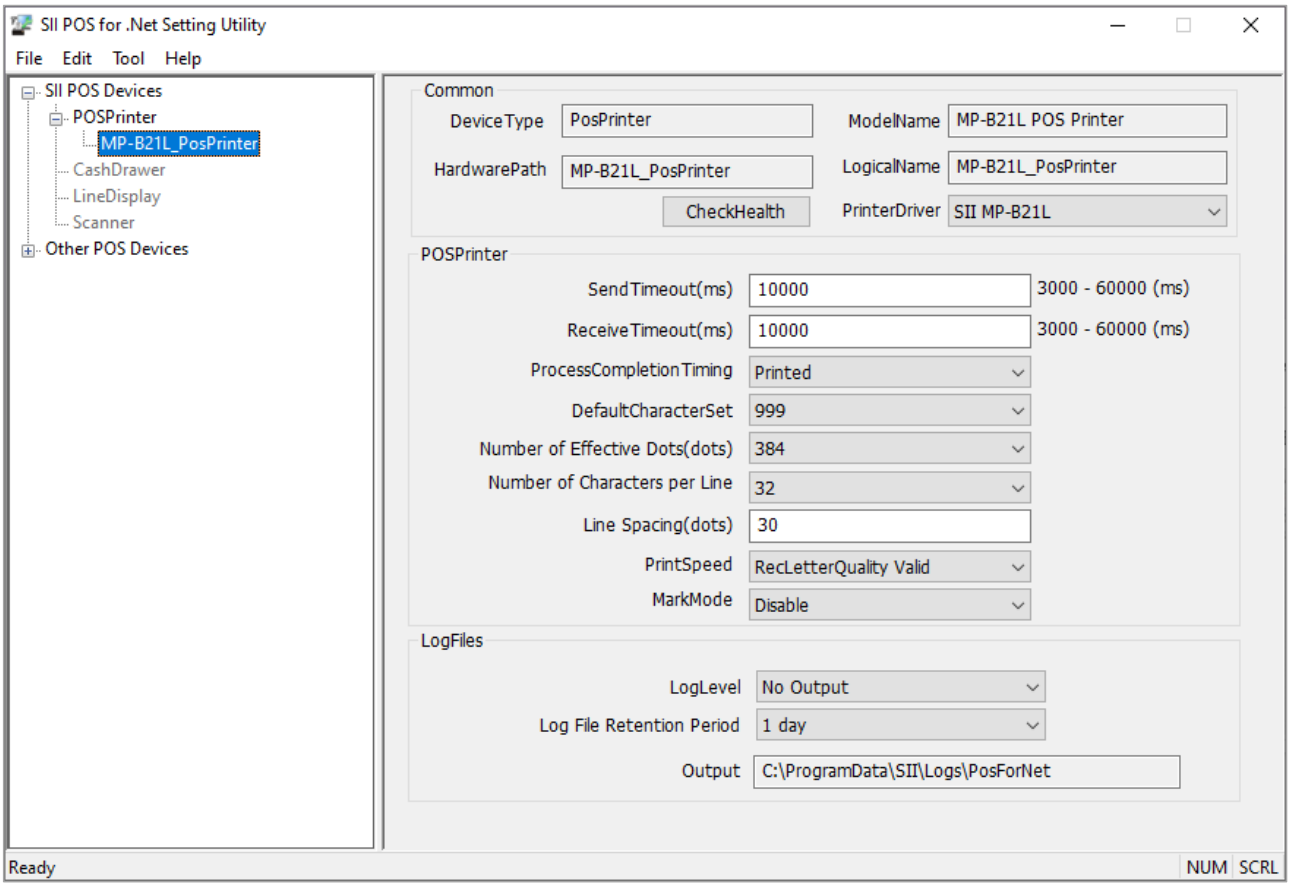
3.2.2 Changing Device Settings

The settings of the added device can be changed with the [Modify] button.

1) Select the logical name of PosPrinter to change from "Device View".



2) "Setting View" is displayed in editable state. Click the [OK] button after changing the contents.



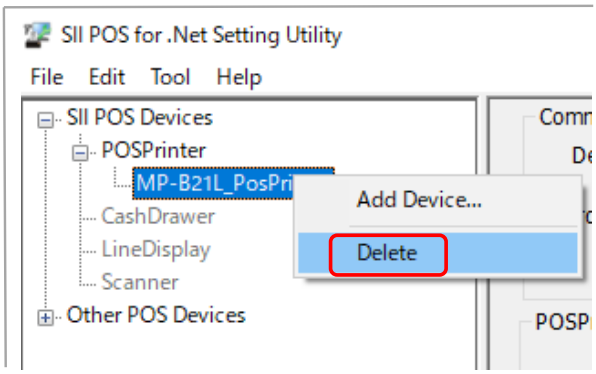
Reference

- When another device is selected or the configuration program is finished, the setting contents are saved.

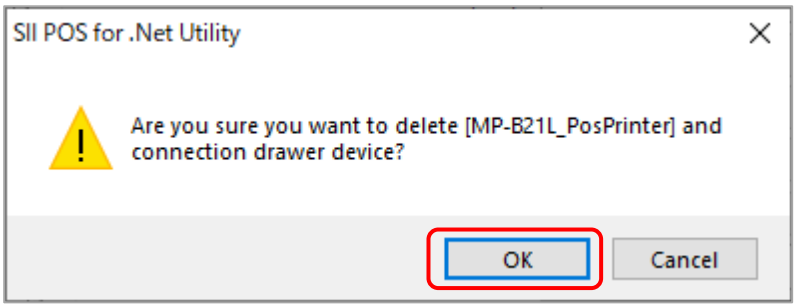
3.2.3 Deletion of Device

The procedure for deleting the added device is described below.

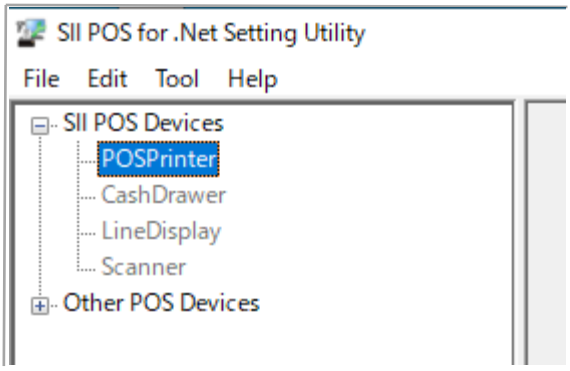
1) Select the device to be deleted from "Device View", and select [Delete] from the right-click menu.



2) Confirm the device name, and click the [OK] button.



3) Confirm that the selected device has been deleted from the "Device View".



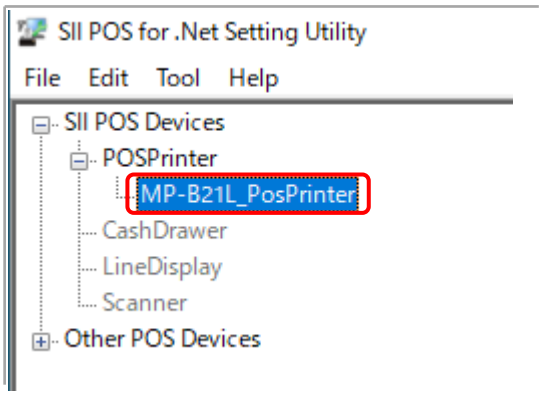
Caution

- When the POSPrinter device is selected, the selected POSPrinter device and each logical device name are all deleted.

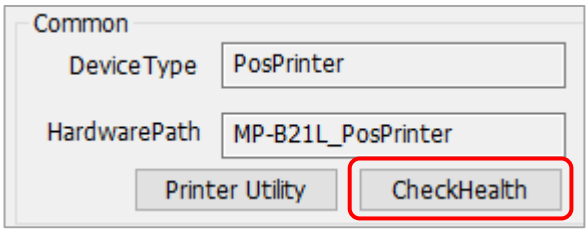
3.2.4 Device Interactive Test

In the configuration program, an interactive test can be performed on the device selected in "Device View". The procedure of the interactive test is described below.

1) Select the logical name of PosPrinter for interactive test from "Device View".



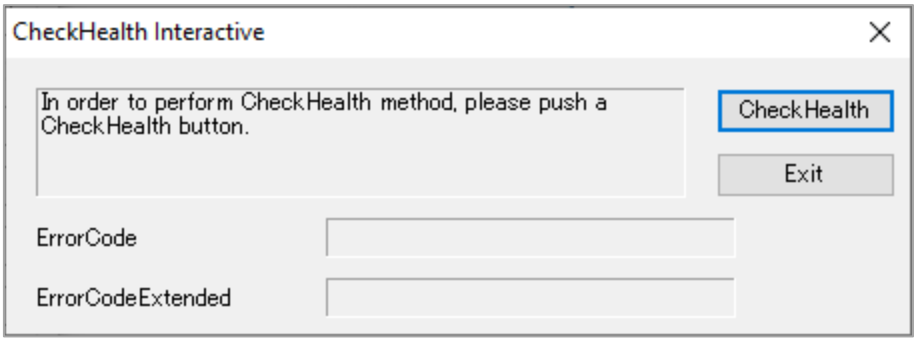
2) Click the [CheckHealth] button in "Setting View".



3) The preparation for the interactive test is started.

[When the preparation for the interactive test succeeded]

4) The dialogue to perform the interactive test is displayed.



To start the interactive test, click the [CheckHealth] button.
To exit the interactive test, click the [Exit] button.

[When the preparation for the interactive test failed]

4) The dialogue is displayed.



See "Appendix A Exceptions" for ErrorCode.
Click the [OK] button after confirming ErrorCode.

3.2.5 Memory Settings of Printer (PrinterUtility)

In the configuration program, the settings of the memory switches connected to the printer can be confirmed and changed.

(1) When the port is selected in the configuration program

1) Click the [Printer Utility] button in the "Setting View".



2) SII Printer Setting Utility starts up.

(2) When the printer driver is selected in the configuration program

When selecting the printer driver, it is necessary to install the printer driver.
See "SII Software Package for Windows Installation Guide" for installation of the printer driver.
Start up SII Printer Setting Utility following the procedures below after installing the printer driver.

- For Windows 11:
  Select [All apps] - [SII Printer Software] - [Printer Setting Utility] from the Start menu, and then the SII Printer Setting Utility starts up.

- For Windows 10:
  Select [SII Printer Software] - [Printer Setting Utility] from the Start menu, and then the SII Printer Setting Utility starts up.

Chapter 4 Properties, Methods, and Events

This chapter describes properties, methods, and events implemented in this software.

4.1 PosPrinter

4.1.1 Summary

(1) Common Properties

Property Name	Type	Access	Availability Condition	Default
CapCompareFirmwareVersion	**bool**	R	Open	*false*
CapPowerReporting	**PowerReporting**	R	Open	*Standard*
CapStatisticsReporting	**bool**	R	Open	*true*
CapUpdateFirmware	**bool**	R	Open	*false*
CapUpdateStatistics	**bool**	R	Open	*true*
CheckHealthText	**string**	R	Open	*""*
Claimed	**bool**	R	Open	*false*
DeviceDescription	**string**	R	Open	"SII MP-Bxx POS Printer"
DeviceEnabled	**bool**	R/W	Open & Claim	*false*
DeviceName	**string**	R	Open	"MP-Bxx POS Printer"
FreezeEvents	**bool**	R/W	Open & Claim	*false*
OutputId	**int**	R	Open	0
PowerNotify	**PowerNotification**	R/W	Open	*Disabled*
PowerState	**PowerState**	R	Open	*Unknown*
ServiceObjectDescription	**string**	R	Open	"SII MP-Bxx POS Printer Service Object, Copyright(C) 20xx Seiko Instruments Inc."
ServiceObjectVersion	**Version**	R	Open	MP-B21L 1.14.x.x MP-B30L, MP-B30 1.12.x.x
State	**ControlState**	R	--	*Idle*
SynchronizingObject	**System.ComponentModel. ISynchronizeInvoke**	R/W	Open	Depends on the application.

(2) Specific Properties
(MP-B21L: When **RecLineWidth**=384, **RecLineChars**=32, **RecLineSpacing**=30, **CharacterSet**=999)
(MP-B30L, MP-B30: When **RecLineWidth**=576, **RecLineChars**=48, **RecLineSpacing**=30, **CharacterSet**=999)

Property Name	Type	Access	Availability Condition	Default
AsyncMode	**bool**	R/W	Open	*false*
CapCharacterSet	**CharacterSetCapability**	R	Open	*Kanji*
CapCoverSensor	**bool**	R	Open	*true*
CapMapCharacterSet	**bool**	R	Open	*false*
CapRec2Color	**bool**	R	Open	*false*
CapRecBarCode	**bool**	R	Open	*true*
CapRecBitmap	**bool**	R	Open	*true*
CapRecBold	**bool**	R	Open	*true*
CapRecCartridgeSensor	**PrinterCartridgeSensors**	R	Open	*None*
CapRecColor	**PrinterColors**	R	Open	*Primary*
CapRecDHigh	**bool**	R	Open	*true*
CapRecDWide	**bool**	R	Open	*true*
CapRecDWideDHigh	**bool**	R	Open	*true*
CapRecEmptySensor	**bool**	R	Open	*true*
CapRecItalic	**bool**	R	Open	*false*
CapRecLeft90	**bool**	R	Open	*true*
CapRecMarkFeed	**PrinterMarkFeeds**	R	Open	*None*[*1]
CapRecNearEndSensor	**bool**	R	Open	*false*
CapRecPageMode	**bool**	R	Open	*true*
CapRecPaperCut	**bool**	R	Open	*false*
CapRecPresent	**bool**	R	Open	*true*
CapRecRight90	**bool**	R	Open	*true*
CapRecRotate180	**bool**	R	Open	*true*
CapRecStamp	**bool**	R	Open	*false*
CapRecUnderline	**bool**	R	Open	*true*
CapTransaction	**bool**	R	Open	*true*
CartridgeNotify	**PrinterCartridgeNotify**	R/W[*2]	Open	*Disabled*
CharacterSet	**int**	R/W	Open, Claim, & Enable	999[*1]
CharacterSetList	**int[]**	R	Open	{437, 737, 850, 852, 855, 857, 858, 860, 863, 865, 866, 932, 999, 1250, 1251, 1252, 1253, 1254}
CoverOpen	**bool**	R	Open, Claim, & Enable	Depends on printer status.
ErrorLevel	**PrinterErrorLevel**	R	Open	*None*
ErrorStation	**PrinterStation**	R	Open	*None*
ErrorString	**string**	R	Open	*""*
FlagWhenIdle	**bool**	R/W	Open	*false*

Property Name	Type	Access	Availability Condition	Default
FontTypefaceList	**string[]**	R	Open	[0]
MapCharacterSet	**bool**	R/W*2	Open	*false*
MapMode	**MapMode**	R/W	Open	*Dots*
PageModeArea	**System.Drawing.Point**	R	Open	{0, 0}
PageModeDescriptor	**PageModeDescriptors**	R	Open	*None*
PageModeHorizontalPosition	**int**	R/W	Open	0
PageModePrintArea	**System.Drawing.Rectangle**	R/W	Open	{0, 0, 0, 0}
PageModePrintDirection	**PageModePrintDirection**	R/W	Open	*None*
PageModeStation	**PrinterStation**	R/W	Open	*None*
PageModeVerticalPosition	**int**	R/W	Open	0
RecBarCodeRotationList	**Rotation[]**	R	Open	{*Normal, Left90, Right90, Rotate180*}
RecBitmapRotationList	**Rotation[]**	R	Open	{*Normal, Left90, Right90, Rotate180*}
RecCartridgeState	**PrinterCartridgeStates**	R	Open, Claim, & Enable	*Unknown*
RecCurrentCartridge	**PrinterColors**	R/W*2	Open, Claim, & Enable	*Primary*
RecEmpty	**bool**	R	Open, Claim, & Enable	Depends on printer status.
RecLetterQuality	**bool**	R/W*2	Open, Claim, & Enable	*false*
RecLineChars	**int**	R/W	Open, Claim, & Enable	MP-B21L 32*1 MP-B30L, MP-B30 48*1
RecLineCharsList	**int[]**	R	Open	MP-B21L {24,27,29,32,38,42, 48} MP-B30L, MP-B30 {36,41,44,48,57,64, 72}
RecLineHeight	**int**	R/W	Open, Claim, & Enable	24*3
RecLineSpacing	**int**	R/W	Open, Claim, & Enable	30*1
RecLinesToPaperCut	**int**	R	Open, Claim, & Enable	2*3
RecLineWidth	**int**	R	Open, Claim, & Enable	MP-B21L 384*1 MP-B30L 576*1 MP-B30 576
RecNearEnd	**bool**	R	Open, Claim, & Enable	*false*
RecSidewaysMaxChars	**int**	R	Open, Claim, & Enable	200*3

Property Name	Type	Access	Availability Condition	Default
RecSidewaysMaxLines	**int**	R	Open, Claim, & Enable	MP-B21L 13[*3] MP-B30L, MP-30 19[*3]
RotateSpecial	**Rotation**	R/W	Open	*Normal*

*1: Can be modified by the configuration program.
*2: Cannot be rewritten.
*3: Automatically modified by the configuration program.

The following specific properties are provided but the operation is not supported.

Property Name	Type	Access	Availability Condition	Default
CapConcurrentJrnRec	**bool**	R	Open	*false*
CapConcurrentJrnSlp	**bool**	R	Open	*false*
CapConcurrentPageMode	**bool**	R	Open	*false*
CapConcurrentRecSlp	**bool**	R	Open	*false*
CapJrn2Color	**bool**	R	Open	*false*
CapJrnBold	**bool**	R	Open	*false*
CapJrnCartridgeSensor	**PrinterCartridgeSensors**	R	Open	*None*
CapJrnColor	**PrinterColors**	R	Open	*None*
CapJrnDHigh	**bool**	R	Open	*false*
CapJrnDWide	**bool**	R	Open	*false*
CapJrnDWideDHigh	**bool**	R	Open	*false*
CapJrnEmptySensor	**bool**	R	Open	*false*
CapJrnItalic	**bool**	R	Open	*false*
CapJrnNearEndSensor	**bool**	R	Open	*false*
CapJrnPresent	**bool**	R	Open	*false*
CapJrnUnderline	**bool**	R	Open	*false*
CapSlp2Color	**bool**	R	Open	*false*
CapSlpBarCode	**bool**	R	Open	*false*
CapSlpBitmap	**bool**	R	Open	*false*
CapSlpBold	**bool**	R	Open	*false*
CapSlpBothSidesPrint	**bool**	R	Open	*false*
CapSlpCartridgeSensor	**PrinterCartridgeSensors**	R	Open	*None*
CapSlpColor	**PrinterColors**	R	Open	*None*
CapSlpDHigh	**bool**	R	Open	*false*
CapSlpDWide	**bool**	R	Open	*false*
CapSlpDWideDHigh	**bool**	R	Open	*false*
CapSlpEmptySensor	**bool**	R	Open	*false*
CapSlpFullSlip	**bool**	R	Open	*false*
CapSlpItalic	**bool**	R	Open	*false*

Property Name	Type	Access	Availability Condition	Default
CapSlpLeft90	**bool**	R	Open	*false*
CapSlpNearEndSensor	**bool**	R	Open	*false*
CapSlpPageMode	**bool**	R	Open	*false*
CapSlpPresent	**bool**	R	Open	*false*
CapSlpRight90	**bool**	R	Open	*false*
CapSlpRotate180	**bool**	R	Open	*false*
CapSlpUnderline	**bool**	R	Open	*false*
JrnCartridgeState	**PrinterCartridgeStates**	R	Open, Claim, & Enable	*Unknown*
JrnCurrentCartridge	**PrinterColors**	R/W	Open, Claim, & Enable	*None*
JrnEmpty	**bool**	R	Open, Claim, & Enable	*false*
JrnLetterQuality	**bool**	R/W	Open, Claim, & Enable	*false*
JrnLineChars	**int**	R/W	Open, Claim, & Enable	0
JrnLineCharsList	**int[]**	R	Open	[0]
JrnLineHeight	**int**	R/W	Open, Claim, & Enable	0
JrnLineSpacing	**int**	R/W	Open, Claim, & Enable	0
JrnLineWidth	**int**	R	Open, Claim, & Enable	0
JrnNearEnd	**bool**	R	Open, Claim, & Enable	*false*
SlpBarCodeRotationList	**Rotation[]**	R	Open	[0]
SlpBitmapRotationList	**Rotation[]**	R	Open	[0]
SlpCartridgeState	**PrinterCartridgeStates**	R	Open, Claim, & Enable	*Unknown*
SlpCurrentCartridge	**PrinterColors**	R/W	Open, Claim, & Enable	*None*
SlpEmpty	**bool**	R	Open, Claim, & Enable	*false*
SlpLetterQuality	**bool**	R/W	Open, Claim, & Enable	*false*
SlpLineChars	**int**	R/W	Open, Claim, & Enable	0
SlpLineCharsList	**int[]**	R	Open	[0]
SlpLineHeight	**int**	R/W	Open, Claim, & Enable	0
SlpLinesNearEndToEnd	**int**	R	Open, Claim, & Enable	0
SlpLineSpacing	**int**	R/W	Open, Claim, & Enable	0

Property Name	Type	Access	Availability Condition	Default
SlpLineWidth	**int**	R	Open, Claim, & Enable	0
SlpMaxLines	**int**	R	Open, Claim, & Enable	0
SlpNearEnd	**bool**	R	Open, Claim, & Enable	*false*
SlpPrintSide	**PrinterSide**	R	Open, Claim, & Enable	*Unknown*
SlpSidewaysMaxChars	**int**	R	Open, Claim, & Enable	0
SlpSidewaysMaxLines	**int**	R	Open, Claim, & Enable	0

(3) Common Methods

Method Name	Availability Condition
CheckHealth	Open, Claim, & Enable
Claim	Open
ClearOutput	Open & Claim
Close	Open
CompareFirmwareVersion	Open, Claim, & Enable
DirectIO	Open, Claim, & Enable
Open	-
Release	Open & Claim
ResetStatistic(string)	Open, Claim, & Enable
ResetStatistics()	Open, Claim, & Enable
ResetStatistics(StatisticCategories)	Open, Claim, & Enable
ResetStatistics(string[])	Open, Claim, & Enable
RetrieveStatistic(string)	Open, Claim, & Enable
RetrieveStatistics()	Open, Claim, & Enable
RetrieveStatistics(StatisticCategories)	Open, Claim, & Enable
RetrieveStatistics(string[])	Open, Claim, & Enable
UpdateFirmware	Open, Claim, & Enable
UpdateStatistic	Open, Claim, & Enable
UpdateStatistics(Statistic[])	Open, Claim, & Enable
UpdateStatistics(StatisticCategories, Object)	Open, Claim, & Enable

(4) Specific Methods

Method Name	Availability Condition
BeginInsertion	Open, Claim, & Enable
BeginRemoval	Open, Claim, & Enable
ChangePrintSide	Open, Claim, & Enable
ClearPrintArea	Open, Claim, & Enable
CutPaper	Open, Claim, & Enable
EndInsertion	Open, Claim, & Enable
EndRemoval	Open, Claim, & Enable
MarkFeed	Open, Claim, & Enable
PageModePrint	Open, Claim, & Enable
PrintBarCode	Open, Claim, & Enable
PrintBitmap	Open, Claim, & Enable
PrintImmediate	Open, Claim, & Enable
PrintMemoryBitmap	Open, Claim, & Enable
PrintNormal	Open, Claim, & Enable
PrintTwoNormal	Open, Claim, & Enable
RotatePrint	Open, Claim, & Enable
SetBitmap	Open, Claim, & Enable
SetLogo	Open, Claim, & Enable
TransactionPrint	Open, Claim, & Enable
ValidateData	Open, Claim, & Enable

(5) Events

Event Name	Availability Condition
DirectIOEvent	Open, Claim, & Enable*1
ErrorEvent	Open, Claim, & Enable
OutputCompleteEvent	Open, Claim, & Enable
StatusUpdateEvent	Open, Claim, & Enable

*1: The availability condition differs from that of UPOS V 1.14.1 or 1.12.

4.1.2 Data Characters and Escape Sequences

(1) Escape Sequence operated when specified

Name	Data	Remarks
Paper cut	ESC|[#]P	Not supported.
Feed and Paper cut	ESC|[#]fP	Not supported.
Feed, Paper cut, and Stamp	ESC|[#]sP	Not supported.
Print bitmap	ESC|#B	• Prints the pre-stored bitmap. The placeholder '#' is replaced by the bitmap number. A value from 1 to 20 can be specified for '#'. If values other than 1 to 20 are specified for '#', they are ignored. If the character '#' is omitted, the data is regarded as print data instead of an escape sequence.
Print top logo	ESC|tL	• Prints the pre-stored top logo.
Print bottom logo	ESC|bL	• Prints the pre-stored bottom logo.
Fire stamp	ESC|sL	Not supported.
Feed lines	ESC|[#]lF	• Feeds the paper forward by lines. The placeholder '#' is replaced by an ASCII decimal string indicating the number of lines to be fed. A value from 0 to 255 can be specified for '#'. If '#' exceeds this range, the maximum supported number of 255 lines are fed. If '#' is omitted, then one line is fed. • This is ignored during rotated 90° right/left mode by **RotatePrint** or during Page Mode by **PageModePrint**.
Feed units	ESC|[#]uF	• Feeds the paper forward by units in **MapMode**. If **MapMode** is set to *MapMode.Dots*, a value from 1 to 255 can be specified for '#'. The placeholder '#' is replaced by an ASCII decimal string indicating the number of units to be fed. If '#' is omitted, then one unit is fed. If '#' exceeds this range, the maximum supported number of 255 units is fed. • This is ignored during rotated 90° right/left mode by **RotatePrint** or during Page Mode by **PageModePrint**.
Feed reverse	ESC|[#]rF	Not supported.

Name	Data	Remarks
Pass through embedded data	ESC|[*]#E*1	• Sends the characters following "#E" through to the printer without modifying any of them. The placeholder '#' is replaced by an ASCII decimal string indicating the number of bytes following the escape sequence that should be passed through as-is to the printer. A value from 1 to 65535 can be specified for '#'. If '#' exceeds this range, transmission of embedded data is not executed. If the print data of the number of bytes specified by '#' is not set after the escape sequence is specified, only the transmittable print data is sent. (Example: If ESC|2Ea is specified, only "a" is sent since only one byte is set for the character string.) If '#' is omitted, the data is regarded as print data instead of an escape sequence. • During rotated 90° right/left mode by **RotatePrint**, the width cannot be calculated exactly because data string specified by transmission of embedded data is not counted as character string. Therefore, make an appropriate adjustment by inserting blanks.
Print in-line barcode	ESC|[*]#R*1	• Prints a barcode. The placeholder '#' is replaced by an ASCII decimal string indicating the number of characters of the string following R (definition of the barcode characteristics). If '#' is omitted, the data is regarded as print data instead of an escape sequence. • If the number of characters specified by '#' does not match the number of bytes following R, all the data within the range specified by '#' is discarded. • During rotated 90° right/left mode by **RotatePrint**, the width cannot be calculated exactly because data string specified by transmission of barcode printing is not counted as character string. Therefore, make an appropriate adjustment by inserting blanks.

*1: [*] is supported only by MP-B21L.

- In-Line Barcode Printing

  The application can print barcodes along with other print data by using the "Print in-line barcode" escape sequence (ESC|[*]#R). The placeholder '#' is replaced by the number of characters of the string (definition of the barcode characteristics) following R.
  The string following R specifies the barcode characteristics using lowercase alphabet letters and numbers. The available numbers are the constant values defined for **PrintBarCode**.

  The characters indicating the attributes are as follows:

  s: symbology (barcode type)
  h: height (barcode height)
  w: width (barcode width)
  a: alignment (position of barcode)
  t: text position (position of HRI string)
  d: start of data (start position of barcode data)
  e: end of data (end position of barcode data)

  Attributes must be written in the above order.
  Every attribute is mandatory. If one of these two conditions is not obeyed or a value outside of the range is specified for the number following each attribute, it may cause unpredictable print results.

Below is an example of printing UPC-A under the condition of center, HRI string printed below the barcode, 200 dots height, and 300 dots width.

ESC|33Rs101h200w300a-2t-13d123456789012e

For the barcode quiet zone, see the description of **PrintBarCode**.

(2) Escape Sequence valid until changed

Name	Data	Remarks
Font typeface	ESC|#fT	Not supported.

(3) Escape Sequence reset by end of print method or "Normal" escape sequence

Name	Data	Remarks
Bold	ESC|[!]bC	• Prints in bold. If '!' is specified, bold is disabled.
Underline	ESC|[!][#]uC	• Prints with underline. The placeholder '#' is replaced by an ASCII decimal string indicating the thickness of the underline in printer dot units. The available thickness is from 0 to 2. If '#' is 3 or larger, then a thickness of 2 is used for the underline. If '#' is omitted, then a thickness of 1 is used. If '!' is specified, underline is disabled.
Italic	ESC|[!]iC	Not supported.
Alternate color (Custom)	ESC|[#]rC	Not supported.
Red color	ESC|rC	Not supported.
Reverse video	ESC|[!]rvC	• Prints in a reverse video format. If '!' is specified, reverse video is disabled.
Shading	ESC|[#]sC	Not supported.
Single high and wide	ESC|1C	• Prints normal size.
Double wide	ESC|2C	• Prints double-wide characters.
Double high	ESC|3C	• Prints double-high characters.
Double high and wide	ESC|4C	• Prints double-high / double-wide characters.
Scale horizontally	ESC|#hC	• A supported value for the placeholder '#' is 1 to 8. If a value less than 1 is specified for '#', print in 1 scale. If a value greater than 8 is specified for '#', print in 8 scale. If '#' is omitted, the data is regarded as print data instead of an escape sequence.
Scale vertically	ESC|#vC	• A supported value for the placeholder '#' is 1 to 8. If a value less than 1 is specified for '#', print in 1 scale. If a value greater than 8 is specified for '#', print in 8 scale. If '#' is omitted, the data is regarded as print data instead of an escape sequence.
RGB Color	ESC|[#]fC	Not supported.

Name	Data	Remarks
Center	ESC|cA	• Aligns the text after ESC|cA in the center. This must be specified at the head of the line. If not, this is invalid. Also, if there is a linefeed on the print data, the center is valid after linefeed. • This specification is ignored during rotated 90° right/left mode by **RotatePrint** or during Page Mode by **PageModePrint**.
Right justify	ESC|rA	• Aligns the text after ESC|rA to the right. This must be specified at the head of the line. If not, this is invalid. Also, if there is a linefeed on the print data, the right justify is valid after linefeed. • This specification is ignored during rotated 90° right/left mode by **RotatePrint** or during Page Mode by **PageModePrint**.
Left justify	ESC|lA	• Aligns the text after ESC|lA to the left. This must be specified at the head of the line. If not, this is invalid. Also, if there is a linefeed on the print data, the left justify is valid after linefeed. • This specification is ignored during rotated 90° right/left mode by **RotatePrint** or during Page Mode by **PageModePrint**.
Normal	ESC|N	• Restores printer characteristics to normal condition.
SubScript	ESC|[!]tbC	Not supported.
SuperScript	ESC|[!]tpC	Not supported.
Strike-through	ESC|[!][#]stC	Not supported.

4.1.3 Common Properties

This section describes the details of the common properties for PosPrinter.
For details of the thrown exception errors, see "Appendix A Exceptions".

CapCompareFirmwareVersion Property

Type **bool**

Description Gets a Boolean value that indicates whether the Service Object/device supports comparing the firmware version in the physical device against that of a firmware file.
The following table shows the valid property value.

Value	Meaning
false	The function that compares firmware versions is not supported.

This property is initialized to *false* by **Open**.

CapPowerReporting Property

Type **PowerReporting**

Description Gets the power reporting capabilities of the device.
The following table shows the valid property value.

Value	Meaning
PowerReporting.Standard	The following 2 types of power states can be determined and reported. ・*PowerState.OffOffline* (power off or offline) ・*PowerState.Online*

This property is initialized to *PowerReporting.Standard* by **Open**.

CapStatisticsReporting Property

Type **bool**

Description Gets a Boolean value that indicates whether the device can accumulate and can provide various statistics regarding usage.
The following table shows the valid property value.

Value	Meaning
true	The device accumulates and can provide various statistics regarding usage. The information accumulated and reported is device specific, and is retrieved using **RetrieveStatistic(s)**.

This property is initialized to *true* by **Open**.

CapUpdateFirmware Property

Type **bool**

Description Gets a Boolean value that indicates whether the device's firmware can be updated through **UpdateFirmware**.
The following table shows the valid property value.

Value	Meaning
false	Firmware update is not supported.

This property is initialized to *false* by **Open**.

CapUpdateStatistics Property

Type **bool**

Description Gets a Boolean value that indicates whether some or all the device statistics can be reset to 0 using **ResetStatistic(s)**.
The following table shows the valid property value.

Value	Meaning
true	The device statistics, or some of the statistics, can be reset to 0 using **ResetStatistic(s)**.

This property is initialized to *true* by **Open**.

CheckHealthText Property

Type **string**

Description Gets a string that indicates the health of the device.
This property is updated by the Service Object when the application calls **CheckHealth**.
The following examples show the results of diagnosis.

Method Parameter	Method Result	CheckHealthText
HealthCheckLevel.External	Success	"External HCheck: Successful"
	Fail	"External HCheck: Failure"
HealthCheckLevel.Interactive[*1]	Success	"Interactive HCheck: Successful"
	Fail	"Interactive HCheck: Failure"
HealthCheckLevel.Internal	Success	"Internal HCheck: Successful"
	Fail	"Internal HCheck: Failure"

*1: In the case of *HealthCheckLevel.Interactive*, if the dialog box is closed without testing after execution, "Interactive HCheck: Canceled" is set.

This property is initialized to empty string by **Open**.

Claimed Property

Type **bool**

Description Gets a Boolean value that indicates whether the device is claimed for exclusive access.
The following table shows the valid property values.

Value	Meaning
false	The device is released for sharing with other applications.
true	The exclusive access to the device is obtained.

This property is initialized to *false* by **Open**.

DeviceDescription Property

Type **string**

Description Gets a string identifying the device and the company that manufactured it.
This property depends on **DeviceName**.
This property is initialized to the following value by **Open**.

DeviceName	Value
"MP-B21L POS Printer"	"SII MP-B21L POS Printer"
"MP-B30L POS Printer"	"SII MP-B30L POS Printer"
"MP-B30 POS Printer"	"SII MP-B30 POS Printer"

DeviceEnabled Property R/W

Type **bool**

Description Gets or sets a Boolean value that indicates whether the device has been placed in an operational state.
The following table shows the valid property values.

Value	Meaning
false	The device has been disabled. If changed to *false*, then the device is physically disabled when possible, any subsequent input will be discarded, and output operations are disallowed.
true	The device is in an operational state. If changed to *true*, then the device is brought to an operational state.

The application must set this property to *true* before using the device.
If **State** is other than *ControlState.Idle*, **DeviceEnabled** cannot be changed from *true* to *false*.

This property is initialized to *false* by **Open**.

DeviceName Property

Type **string**

Description Gets a short string identifying the device and any pertinent information about it.
This property is initialized to the following values by **Open**.

Printer	Value
MP-B21L	"MP-B21L POS Printer"
MP-B30L	"MP-B30L POS Printer"
MP-B30	"MP-B30 POS Printer"

FreezeEvents Property R/W

Type **bool**

Description Gets or sets a Boolean value that indicates whether the application has requested that the Service Object not deliver events.
The following table shows the valid property values.

Value	Meaning
false	The application allows events to be delivered. If some events have been held while events were frozen and all other conditions are correct for delivering the events, changing the **FreezeEvents** property to *false* allows these events to be delivered.
true	The application has requested that the Service Object not deliver events. Events will be queued by the Service Object but not delivered until the application changes the **FreezeEvents** property to *false*.

An application may choose to freeze events for a specific sequence of code where interruption by an event is not desirable.

If an error occurs while a print method such as **PrintNormal** is operated under **AsyncMode** is *true*, **ErrorEvent** is frozen and **State** turns to *ControlState.Busy*. In this case, discard the frozen event by **ClearOutput** or set **FreezeEvents** to *false* to cause **ErrorEvent**, and then execute **Close**, since the Service Object cannot be closed under this circumstance.

This property is initialized to *false* by **Open**.

OutputId Property

Type **int**

Description Holds the identifier of the most recently started asynchronous output (call to an asynchronous method when **AsyncMode** is set to *true*).

When a method successfully initiates an asynchronous output, the Service Object assigns an identifier to the request. When the output completes, the Service Object will fire an **OutputCompleteEvent** passing this output ID as a parameter.

OutputId is allocated automatically within the range of **int**.

This property is initialized to 0 by **Open**.

PowerNotify Property R/W

Type **PowerNotification**

Description Gets or sets the type of power notification selection made by the application.
The following table shows the valid property values.

Value	Meaning
PowerNotification.Disabled	The Service Object will not provide any power notifications to the application. No power notification **StatusUpdateEvents** will be fired, and **PowerState** may not be set.
PowerNotification.Enabled	When **DeviceEnabled** is set to *true*, the Service Object will fire the power notification **StatusUpdateEvents** and update **PowerState**. The level of functionality depends on the value of **CapPowerReporting**.

PowerNotify can be set only while the device is disabled; that is, while **DeviceEnabled** is *false*.

This property is initialized to *PowerNotification.Disabled* by **Open**.

PowerState Property

Type **PowerState**

Description Gets the current power condition of the device.
The following table shows the valid property values.

Value	Meaning
PowerState.OffOffline	The device is powered off or offline.
PowerState.Online	The device is powered on and ready for use.

Value	Meaning
PowerState.Unknown	Cannot determine the device's power state due to one of the following reasons. ・**PowerNotify** is *PowerNotification.Disabled*. ・**DeviceEnabled** is *false*.

This property is initialized to *PowerState.Unknown* by **Open**.

ServiceObjectDescription Property

Type **string**

Description Gets a string identifying the Service Object that supports the device and the company that produced it.
This property is initialized to the following values by **Open**.

DeviceName	Value
"MP-B21L POS Printer"	"SII MP-B21L POS Printer Service Object, Copyright (C) 20xx Seiko Instruments Inc."
"MP-B30L POS Printer"	"SII MP-B30L POS Printer Service Object, Copyright (C) 20xx Seiko Instruments Inc."
"MP-B30 POS Printer"	"SII MP-B30 POS Printer Service Object, Copyright (C) 20xx Seiko Instruments Inc."

ServiceObjectVersion Property

Type **Version**

Description Gets the Service Object version number.
Version numbers consist of four integers; Major, Minor, Build, and Revision.
The Major and Minor version numbers correspond to the UPOS version that the Service Object implements.
When Build version is A, Revision version is B, this property is initialized to 1.14.A.B (MP-B21L) or to 1.12.A.B (MP-B30L, MP-B30) by **Open**.

State Property

Type **ControlState**

Description Gets the current state of the device.
The following table shows the valid property values.

Value	Meaning
ControlState.Busy	The device is in a normal state and is busy executing output.
ControlState.Closed	The device is closed.
ControlState.Error	An error has been reported, and the application must recover the Control to a normal state before normal I/O can resume. This state is only possible inside the **ErrorEvent** event handler.

Value	Meaning
ControlState.Idle	The device is in a normal state and is not busy.

This property is always readable.
This property is initialized to *ControlState.Idle* by **Open**.

SynchronizingObject Property

Type **System.ComponentModel.ISynchronizeInvoke**

Description This property holds an object that implements the .NET Framework class **ISynchronizeInvoke** that is used to marshal events to a particular thread. If **SynchronizingObject** is set to *null*, events are raised on a system thread owned by the Service Object.
The application using a Windows form sets the *this* pointer of the **Form** class of the main form to **SynchronizationObject**, so that events are notified to the main application thread as required by the **Form** class.

4.1.4 Specific Properties

This section describes the details of the specific properties for PosPrinter.
For exception errors of specific properties that are not supported, see "Appendix A Exceptions".

AsyncMode Property R/W

Type **bool**

Description Gets or sets a Boolean value that indicates whether certain print methods will be performed asynchronously.
The following table shows the valid property values.

Value	Meaning
false	**PrintNormal**, **PrintBarCode**, **PrintBitmap**, **PrintMemoryBitmap**, **MarkFeed, RotatePrint**, **TransactionPrint**, and **PageModePrint** print methods are executed synchronously.
true	The methods are executed asynchronously.

This property is initialized to *false* by **Open**.

CapCharacterSet Property

Type **CharacterSetCapability**

Description Indicates the printable character setting.
The following table shows the valid property value.

Value	Meaning
CharacterSetCapability. Kanji	The character setting supports Code Page 932, including ASCII characters 0x20 through 0x7F and the one-byte katakana characters 0xA1 through 0xDF. It also includes the Shift-JIS code characters defined in JIS 1st and 2nd levels.

This property is initialized to *CharacterSetCapability.Kanji* by **Open**.

CapCoverSensor Property

Type **bool**

Description Gets a Boolean value that indicates whether the printer has a "cover open" sensor function for the receipt.
The following table shows the valid property value.

Value	Meaning
true	The printer has a "cover open" sensor.

This property is initialized to *true* by **Open**.

CapMapCharacterSet Property

Type **bool**

Description Gets a Boolean value that indicates that the Service Object is able to map the characters of the application to a character set defined by **CharacterSetList**.
The following table shows the valid property value.

Value	Meaning
false	The Service Object cannot exactly map the characters to the character sets defined in **CharacterSetList**.

This property is initialized to *false* by **Open**.

CapRec2Color Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can print dark plus an alternate color.
The following table shows the valid property value.

Value	Meaning
false	Two color printing of the receipt is not supported.

This property is initialized to *false* by **Open**.

CapRecBarCode Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt has barcode printing capability.
The following table shows the valid property value.

Value	Meaning
true	The receipt has barcode printing capability.

This property is initialized to *true* by **Open**.

CapRecBitmap Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can print bitmaps.
The following table shows the valid property value.

Value	Meaning
true	The receipt can print bitmaps.

This property is initialized to *true* by **Open**.

CapRecBold Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can print bold characters.
The following table shows the valid property value.

Value	Meaning
true	The receipt can print bold characters.

This property is initialized to *true* by **Open**.

CapRecCartridgeSensor Property

Type **PrinterCartridgeSensors**

Description Gets a value that indicates the presence of receipt cartridge monitoring sensors.
The following table shows the valid property value.

Value	Meaning
PrinterCartridgeSensors.None	Receipt cartridge monitoring sensors are not supported.

This property is initialized to *PrinterCartridgeSensors.None* by **Open**.

CapRecColor Property

Type **PrinterColors**

Description Gets a value that indicates available receipt color cartridges.
The following table shows the valid property value.

Value	Meaning
PrinterColors.Primary	Receipt supports primary color (Black).

This property is initialized to *PrinterColors.Primary* by **Open**.

CapRecDHigh Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can print double high characters.
The following table shows the valid property value.

Value	Meaning
true	The receipt can print double high characters.

This property is initialized to *true* by **Open**.

CapRecDWide Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can print double wide characters. The following table shows the valid property value.

Value	Meaning
true	The receipt can print double wide characters.

This property is initialized to *true* by **Open**.

CapRecDWideDHigh Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can print double high/double wide characters.
The following table shows the valid property value.

Value	Meaning
true	The receipt can print double high/double wide characters.

This property is initialized to *true* by **Open**.

CapRecEmptySensor Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt has an "out-of-paper" sensor. The following table shows the valid property value.

Value	Meaning
true	The receipt has an "out-of-paper" sensor.

This property is initialized to *true* by **Open**.

CapRecItalic Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can print italic characters. The following table shows the valid property value.

Value	Meaning
false	The receipt cannot print Italic characters.

This property is initialized to *false* by **Open**.

CapRecLeft90 Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can print in a rotated 90° left mode.
The following table shows the valid property value.

Value	Meaning
true	The receipt can print in a rotated 90° left mode.

This property is initialized to *true* by **Open**.

CapRecMarkFeed Property

Type **PrinterMarkFeeds**

Description Indicates the type of mark sensed paper handling available.
Either a logical OR of the following values is set in this property.

Value	Meaning
PrinterMarkFeeds.None	Control function for mark sensed paper is disabled.
PrinterMarkFeeds.Takeup	After detecting the mark or the gap, feeds the paper to the paper take-up position.
PrinterMarkFeeds.Cutter	After detecting the mark or the gap, feeds the paper to the cutting position. (Feeds the paper to the same position as *PrinterMarkFeeds.Takeup*.)
PrinterMarkFeeds.CurrentTof	Supported only by MP-B21L. Feeds backward the marked paper to the printing position.
PrinterMarkFeeds.NextTof	Supported only by MP-B21L and MP-B30L. After detecting the next mark or the gap, feeds the paper to the printing position.

The default of this property can be changed at the setting in the configuration program.
This property is initialized to *PrinterMarkFeeds.None* by **Open**.

CapRecNearEndSensor Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt has a low-paper sensor.
The following table shows the valid property value.

Value	Meaning
false	The printer does not have a "paper-near-end" sensor.

This property is initialized to *false* by **Open**.

CapRecPageMode Property

Type **bool**

Description Gets a Boolean value that indicates whether the printer can support Page Mode for the receipt station.
The following table shows the valid property value.

Value	Meaning
true	The printer can support Page Mode for the receipt station.

This property is initialized to *true* by **Open**.

CapRecPaperCut Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can perform paper cuts.
The following table shows the valid property value.

Value	Meaning
false	The receipt does not have a paper cut capability.

This property is initialized to *false* by **Open**.

CapRecPresent Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt print station is present.
The following table shows the valid property value.

Value	Meaning
true	The receipt print station is present.

This property is initialized to *true* by **Open**.

CapRecRight90 Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can print in a rotated 90° right mode.
The following table shows the valid property value.

Value	Meaning
true	The receipt can print in a rotated 90° right mode.

This property is initialized to *true* by **Open**.

CapRecRotate180 Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can print in a rotated upside down mode.
The following table shows the valid property value.

Value	Meaning
true	The receipt can print in a rotated upside down mode.

This property is initialized to *true* by **Open**.

CapRecStamp Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt has a stamp capability.
The following table shows the valid property value.

Value	Meaning
false	The receipt does not have a stamp capability.

This property is initialized to *false* by **Open**.

CapRecUnderline Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt can print underlined characters.
The following table shows the valid property value.

Value	Meaning
true	The receipt can print underlined characters.

This property is initialized to *true* by **Open**.

CapTransaction Property

Type **bool**

Description Gets a Boolean value that indicates whether receipt station supports printer transactions.
The following table shows the valid property value.

Value	Meaning
true	The receipt station supports printer transactions.

This property is initialized to *true* by **Open**.

CartridgeNotify Property R/W

Type **PrinterCartridgeNotify**

Description Gets or sets the type of printer cartridge state notification the application wants to receive.
The following table shows the valid property value.

Value	Meaning
PrinterCartridgeNotify.Disabled	Cartridge-state notifications are not available.

This property cannot be rewritten.
This property is initialized to *PrinterCartridgeNotify.Disabled* by **Open**.

CharacterSet Property R/W

Type **int**

Description Gets or sets the numeric value that indicates the character set that the application wants to use for printing characters.
The following table shows the valid property values.

Value	Meaning
437	Selects Code Page 437 character set.
737	Selects Code Page 737 character set.
850	Selects Code Page 850 character set.
852	Selects Code Page 852 character set.
855	Selects Code Page 855 character set.
857	Selects Code Page 857 character set.
858	Selects Code Page 858 character set.
860	Selects Code Page 860 character set.
863	Selects Code Page 863 character set.
865	Selects Code Page 865 character set.
866	Selects Code Page 866 character set.
932	Selects Katakana as Code Page 932 character set (Shift-JIS Code).
999	Selects Windows ANSI character set.*1
1250	Selects Code Page 1250 character set.
1251	Selects Code Page 1251 character set.
1252	Selects Code Page 1252 character set.*1
1253	Selects Code Page 1253 character set.
1254	Selects Code Page 1254 character set.

*1: Windows ANSI character set is equal to Code Page 1252 character set.

The default of this property can be changed at the setting in the configuration program.
This property is initialized to the value of character set which is set in [DefaultCharacterSet] of the configuration program by **Open**.

CharacterSetList Property

Type **int[]**

Description Gets the list of character set numbers supported for printing.
This property is initialized to {437, 737, 850, 852, 855, 857, 858, 860, 863, 865, 866, 932, 999, 1250, 1251, 1252, 1253, 1254} by **Open**.

CoverOpen Property

Type **bool**

Description Gets a Boolean value that indicates whether the printer's cover is open.
The following table shows the valid property values.

Value	Meaning
false	The printer's cover is closed.
true	The printer's cover is open.

This property is set and kept current by the Service Object while the device is enabled.

ErrorLevel Property

Type **PrinterErrorLevel**

Description Gets the severity of the most recent error condition.
The following table shows the valid property values.

Value	Meaning
PrinterErrorLevel.Fatal	A non-recoverable error has occurred.
PrinterErrorLevel.None	No error condition is present.
PrinterErrorLevel.Recoverable	A recoverable error has occurred.

This property is set by the Service Object only before delivering an **ErrorEvent** to the application. When the error is cleared, the Service Object changes **ErrorLevel** to *PrinterErrorLevel.None*.
This property is initialized to *PrinterErrorLevel.None* by **Open**.

ErrorStation Property

Type **PrinterStation**

Description Gets the station that was printing when an error was detected.
The following table shows the valid property values.

Value	Meaning
PrinterStation.None	The error was not detected.
PrinterStation.Receipt	The error is detected at the receipt station.

This property is set by the Service Object only before delivering an **ErrorEvent** to the application. When the error is cleared, the Service Object changes **ErrorStation** to *PrinterStation.None*.
This property is initialized to *PrinterStation.None* by **Open**

ErrorString Property

Type **string**

Description Holds a vendor-supplied description of the current error.
The following table shows the valid property values.

Setting Priority	ErrorCode	ErrorCodeExtended	String
1	*ErrorCode.NoHardware*		The power supply of the device is off.
2	*ErrorCode.Extended*	*ExtendedErrorFatal*(1010)	Unrecoverable error occurred.
3	*ErrorCode.Extended*	*ExtendedErrorBattery*(1013)	Battery error occurred.
4	*ErrorCode.Extended*	*ExtendedErrorHeadTemp*(1005)	Head temperature error.
5	*ErrorCode.Extended*	*ExtendedErrorVpPower*(1001)	Vp power error occurred.
6	*ErrorCode.Extended*	*ExtendedErrorCoverOpen*(201)	The cover is open.
7	*ErrorCode.Extended*	*ExtendedErrorReceiptEmpty*(203)	Out of receipt form.
8	*ErrorCode.Extended*	*ExtendedErrorMarkPaperJam*(1014)	Mark paper jam error occurred.
9	*ErrorCode.Failure*		Communication error occurred.
			Windows system error occurred.
			Time out.

The values in the above table are described in descending order of priority. When multiple errors occur simultaneously, the higher-priority value is set.
This property is set by the Service Object only before delivering an **ErrorEvent** to the application. When the error is cleared, the Service Object changes **ErrorString** to an empty string.
This property is initialized to empty string by **Open**.

FlagWhenIdle Property R/W

Type **bool**

Description Gets or sets a Boolean value that indicates whether or not to notify that **State** turns to *ControlState.Idle*.
The following table shows the valid property values.

Value	Meaning
false	**StatusUpdateEvent** is not notified.
true	**StatusUpdateEvent** will be sent when **State** is *ControlState.Idle*.

FlagWhenIdle is automatically reset to *false* when **StatusUpdateEvent** is notified after **FlagWhenIdle** is set to *true*.
By using **FlagWhenIdle** and **StatusUpdateEvent**, the application can know when all outstanding asynchronous outputs are finished. The event will be notified if the outputs are completed successfully or the outputs are deleted by **ClearOutput** or the event handler that receives **ErrorEvent**.
If **State** is already set to *ControlState.Idle* when **FlagWhenIdle** is set to *true*, then a **StatusUpdateEvent** is notified immediately. The application can therefore use this event without regard to the disagreement between the finish of asynchronous output and the setting of this flag.
This property is initialized to *false* by **Open**.

FontTypefaceList Property

Type **string[]**

Description Gets a string array that specifies the fonts and typefaces supported by the printer.
An empty array indicates that only the default font is supported.
This property is initialized to an empty string array by **Open**.

MapCharacterSet Property R/W

Type **bool**

Description Gets a Boolean value that indicates whether character mapping is supported or not.
The following table shows the valid property value.

Value	Meaning
false	No mapping is supported.

This property cannot be rewritten.
This property is initialized to *false* by **Open**.

MapMode Property R/W

Type **MapMode**

Description Holds the mapping mode of the printer.
The mapping mode defines the unit of measure used for other properties, such as line heights and line spacings.

The following mapping modes are supported.
The values in () indicate the values converted into dot.

Parameter	Meaning
MapMode.Dots	Printer's dot width 0.125 mm (1 dot)
MapMode.English	0.001 inch (0.203 dots)
MapMode.Metric	0.01 mm (0.08 dots)
MapMode.Twips	1/1440 of an inch (0.1411 dots)

For each mapping mode, the unit is converted using one of the following calculation formulae.

Parameter	Conversion
MapMode.Dots	No conversion
MapMode.English	k = 1/1000 ■*MapMode.Dots* to *MapMode.English* conversion english = dot / (dpi × k) ■*MapMode English* to *MapMode.Dots* conversion dot = english × dpi × k
MapMode.Metric	k = 1/100, mmpi = 25.4 ■*MapMode.Dots* to *MapMode.Metric* conversion metric = (mmpi × dot) / (dpi × k) ■*MapMode.Metric* to *MapMode.Dots* conversion dot = (metric × dpi × k) / mmpi
MapMode.Twips	k = 1/1440 ■*MapMode.Dots* to *MapMode.Twips* conversion twips = dot / (dpi × k) ■*MapMode.Twips* to *MapMode.Dots* conversion dot = twips × dpi × k

MapMode only changes the unit of each property for display, and all internal processings are executed in dots regardless of **MapMode**.
Therefore, the rounding errors of values do not accumulate.
When converting a dot value to a map mode value, the value is rounded up to an integer.
When converting from a map mode value to a dot value, the decimal part is truncated.

Setting this property may also change **RecLineSpacing**, **RecLineWidth**, **RecLineHeight**, **PageModeArea**, **PageModePrintArea**, **PageModeHorizontalPosition,** and **PageModeVerticalPosition**.

This property is initialized to *MapMode.Dots* when the device is first enabled following **Open**.

PageModeArea Property

Type **System.Drawing.Point**

Description Gets the page area for the selected **PageModeStation** expressed in the unit of measure given by **MapMode**.
Specify *PrinterStation.Receipt* for **PageModeStation** before accessing this property.
When *PrinterStation.Receipt* is specified for **PageModeStation**, the following values are set to this property.

MP-B21L

RecLineWidth	Value When MapMode = *MapMode.Dots*
168	*Point.X* = 168, *Point.Y* = 2400
176	*Point.X* = 176, *Point.Y* = 2400
192	*Point.X* = 192, *Point.Y* = 2400
208	*Point.X* = 208, *Point.Y* = 2400
224	*Point.X* = 224, *Point.Y* = 2400
240	*Point.X* = 240, *Point.Y* = 2400
256	*Point.X* = 256, *Point.Y* = 2400
272	*Point.X* = 272, *Point.Y* = 2400
288	*Point.X* = 288, *Point.Y* = 2400
304	*Point.X* = 304, *Point.Y* = 2400
320	*Point.X* = 320, *Point.Y* = 2400
336	*Point.X* = 336, *Point.Y* = 2400
352	*Point.X* = 352, *Point.Y* = 2400
360	*Point.X* = 360, *Point.Y* = 2400
368	*Point.X* = 368, *Point.Y* = 2400
384	*Point.X* = 384, *Point.Y* = 2400

MP-B30L

RecLineWidth	Value When MapMode = *MapMode.Dots*
360	*Point.X* = 360, *Point.Y* = 2400
368	*Point.X* = 368, *Point.Y* = 2400
384	*Point.X* = 384, *Point.Y* = 2400
400	*Point.X* = 400, *Point.Y* = 2400
416	*Point.X* = 416, *Point.Y* = 2400
432	*Point.X* = 432, *Point.Y* = 2400
448	*Point.X* = 448, *Point.Y* = 2400
464	*Point.X* = 464, *Point.Y* = 2400
480	*Point.X* = 480, *Point.Y* = 2400
496	*Point.X* = 496, *Point.Y* = 2400

RecLineWidth	Value When MapMode = *MapMode.Dots*
512	*Point.X* = 512, *Point.Y* = 2400
528	*Point.X* = 528, *Point.Y* = 2400
544	*Point.X* = 544, *Point.Y* = 2400
560	*Point.X* = 560, *Point.Y* = 2400
576	*Point.X* = 576, *Point.Y* = 2400

MP-B30

RecLineWidth	Value When MapMode = *MapMode.Dots*
576	*Point.X* = 576, *Point.Y* = 2400

This property is initialized to {*Point.X* = 0, *Point.Y* = 0} by **Open**.

PageModeDescriptor Property

Type **PageModeDescriptors**

Description Gets the basic Page Mode functionality of the printer for the selected **PageModeStation**. This property is indicated by OR of the following values.
Specify *PrinterStation.Receipt* for **PageModeStation** before accessing this property.
When *PrinterStation.Receipt* is specified for **PageModeStation**, OR of *PageModeDescriptors.Bitmap*, *PageModeDescriptors.BitmapRotate*, *PageModeDescriptors.Barcode*, and *PageModeDescriptors.BarcodeRotate* is set to this property.

Value	Meaning
PageModeDescriptors.Barcode	Printing of barcodes on the **PageModeStation** is supported
PageModeDescriptors.BarcodeRotate	Rotation of barcodes on the **PageModeStation** is supported
PageModeDescriptors.Bitmap	Printing of bitmaps on the **PageModeStation** is supported
PageModeDescriptors.BitmapRotate	Rotation of bitmaps on the **PageModeStation** is supported

This property is initialized to *PageModeDescriptors.None* by **Open**.

PageModeHorizontalPosition Property R/W

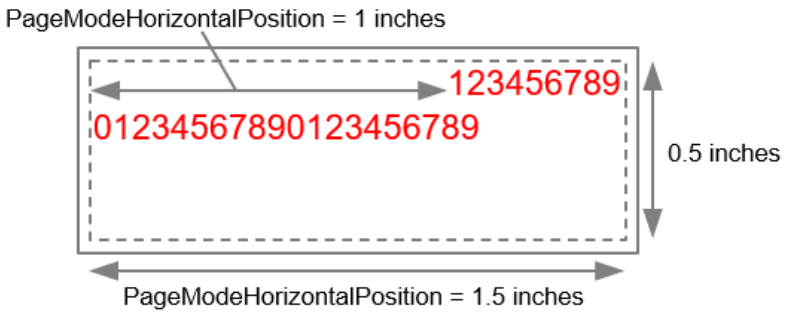
Type **int**

Description Gets or sets the horizontal start position offset within the print area for the print station specified by **PageModeStation**.
The property is expressed in the unit indicated by **MapMode**.
The horizontal direction is the same as the actual **PageModePrintDirection**.
A read/get on this property will return the horizontal position offset set by the last write/set and not the current position.
PageModeStation must be set to *PrinterStation.Receipt* before accessing this property, otherwise the value 0 is returned.

The following code sample shows the usage of **PageModeHorizontalPosition**.

```
myptr.MapMode=MapMode.English;
myptr.PageModeStation=PrinterStation.Receipt;
myptr.PageModePrint(PageModePrintControl.PageMode);
// Set print area to 1.5 inches by 0.5 inches
myptr.PageModePrintArea=new System.Drawing.Point(0,0,1500,500);
myptr.PageModePrintDirection=PageModePrintDirection.LeftToRight;
myptr.PageModeHorizontalPosition=1000;
myptr.PrintNormal(PrinterStation.Receipt,"12345678901234567890123456789O\n");
myptr.PageModePrint(PageModePrintControl.Normal);
```

The code sample above will generate the following receipt.



This property is initialized to 0 by **Open**.

PageModePrintArea Property R/W

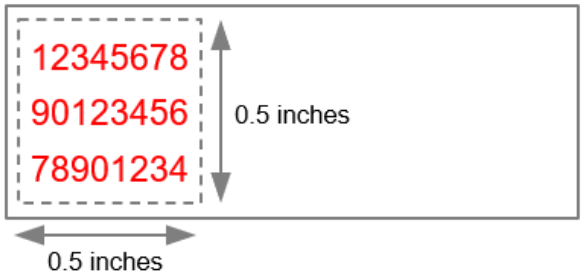
Type **System.Drawing.Rectangle**

Description Holds the Page Mode print area for the selected **PageModeStation** expressed in the unit specified for **MapMode**.
The maximum possible print area of **PageModePrintArea** is the page area of **PageModeArea**.
PageModeStation must be set to *PrinterStation.Receipt* before accessing this property, otherwise *Rectangle.Empty* is returned.

The following code sample shows the usage of **PageModePrintArea**.

```
myptr.MapMode=MapMode.English;
myptr.PageModeStation=PrinterStation.Receipt;
myptr.PageModePrint(PageModePrintControl.PageMode);
// Set print area to half inch square block
myptr.PageModePrintArea=new System.Drawing.Point(0,0,500,500);
myptr.PageModePrintDirection=PageModePrintDirection.LeftToRight;
myptr.PrintNormal(PrinterStation.Receipt,"12345678901234567890123 4\n");
myptr.PageModePrint(PageModePrintControl.Normal);
```

The code sample above will generate the following receipt.



This property is initialized to {*Rectangle.x* = 0, *Rectangle.y* = 0, *Rectangle.width* = 0, *Rectangle.height* = 0} by **Open**.

PageModePrintDirection Property R/W

Type **PageModePrintDirection**

Description Gets the print direction, as specified by the **PageModePrintDirection** enumeration.
PageModeStation must be set to *PrinterStation.Receipt* before accessing this property, otherwise *PageModePrintDirection.None* is returned.
The following table shows the valid property values.

Value	Meaning
PageModePrintDirection.BottomToTop	Rotated left 90° printing. Print from bottom to top, starting at the bottom left corner of the Page Mode print area.

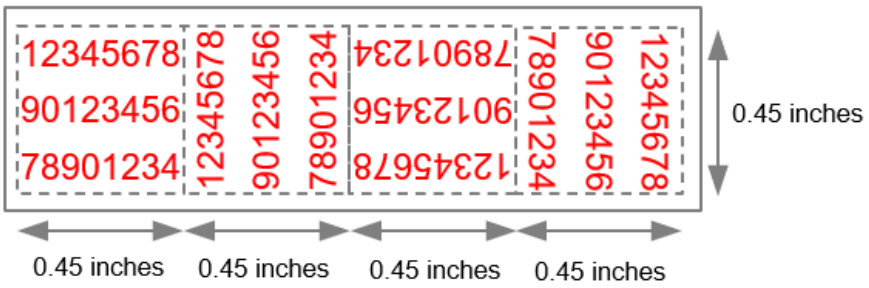
Value	Meaning
PageModePrintDirection.LeftToRight	Normal direction printing. Print from left to right, starting at the top left corner of the Page Mode print area.
PageModePrintDirection.None	The print direction is not specified.
PageModePrintDirection.RightToLeft	Upside down printing. Print from right to left, starting at the bottom right corner of the Page Mode print area.
PageModePrintDirection.TopToBottom	Rotated right 90° printing. Print from top to bottom, starting at the top right corner of the Page Mode print area.

Changing this property may also change the correction direction of the print start point indicated by **PageModeHorizontalPosition** and **PageModeVerticalPosition**. Changing this property is only effective for the current print area. By changing the print areas, it is possible to generate a receipt with text printed in multiple rotations.

The following code sample shows the usage of **PageModePrintDirection**.

```
myptr.MapMode=MapMode.English;
myptr.PageModeStation=PrinterStation.Receipt;
myptr.PageModePrint(PageModePrintControl.PageMode);
// Set print area to half inch square block
myptr.PageModePrintArea=new System.Drawing.Point(0,0,450,450);
myptr.PageModePrintDirection=PageModePrintDirection.LeftToRight;
myptr.PrintNormal(PrinterStation.Receipt," 12345678901234567890123456789012345678901234567890\n");
myptr.PageModePrintArea=new System.Drawing.Point(450,0,450,450);
myptr.PageModePrintDirection=PageModePrintDirection.BottomToTop;
myptr.PrintNormal(PrinterStation.Receipt," 12345678901234567890123456789012345678901234567890\n");
myptr.PageModePrintArea=new System.Drawing.Point(900,0,450,450);
myptr.PageModePrintDirection=PageModePrintDirection.RightToLeft;
myptr.PrintNormal(PrinterStation.Receipt," 12345678901234567890123456789012345678901234567890\n");
myptr.PageModePrintArea=new System.Drawing.Point(1350,0,450,450);
myptr.PageModePrintDirection=PageModePrintDirection.TopToBottom;
myptr.PrintNormal(PrinterStation.Receipt," 12345678901234567890123456789012345678901234567890\n");
myptr.PageModePrint(PageModePrintControl.Normal);
```

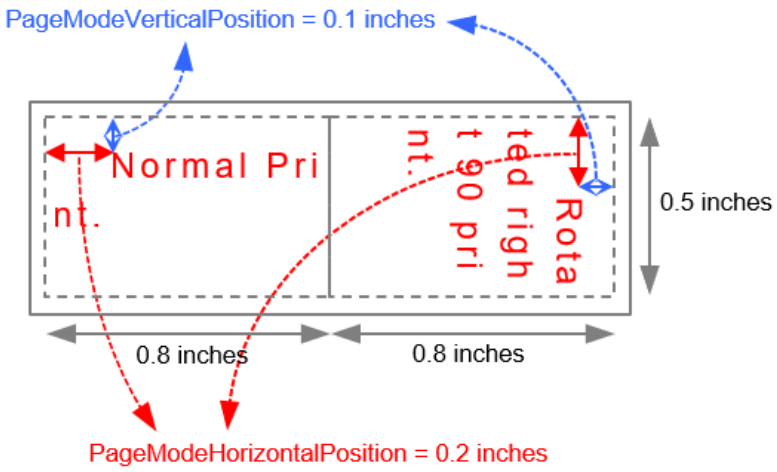
The code sample above will generate the following receipt.



It is also possible to generate rotated text.

```
myptr.MapMode=MapMode.English;
myptr.PageModeStation=PrinterStation.Receipt;
myptr.PageModePrint(PageModePrintControl.PageMode);
myptr.PageModeVerticalPosition=100;
myptr.PageModeHorizontalPosition=200;
myptr.PageModePrintArea=new System.Drawing.Point(0,0,800,500);
myptr.PageModePrintDirection=PageModePrintDirection.LeftToRight;
myptr.PrintNormal(PrinterStation.Receipt,"Normal print.\n");
myptr.PageModePrintArea=new System.Drawing.Point(800,0,800,500);
myptr.PageModePrintDirection=PageModePrintDirection.TopToBottom;
myptr.PrintNormal(PrinterStation.Receipt,"Rotated right 90 print.\n");
myptr.PageModePrint(PageModePrintControl.Normal);
```

The code sample above will generate the following receipt.



This property is initialized to *PageModePrintDirection.None* by **Open**.
And this property is set to *PageModePrintDirection.LeftToRight* when a valid station is specified.

PageModeStation Property R/W

Type **PrinterStation**

Description Gets or sets the printer station, as specified by the **PrinterStation** enumeration.
Be sure to specify *PrinterStation.Receipt* for this property before accessing the Page Mode properties or methods.

This property is initialized to *PrinterStation.None* by **Open**.

PageModeVerticalPosition Property R/W

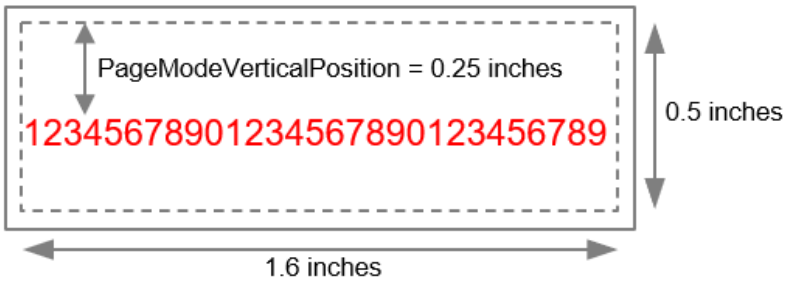
Type **int**

Description Gets or sets the vertical start position offset within the print area for the print station specified by **PageModeStation**.
The property is expressed in the unit indicated by **MapMode**.
The vertical direction is perpendicular to the direction specified in the actual **PageModePrintDirection**. A read/get on this property will return the vertical position offset set by the last write/set and not the current position.
Specify *PrinterStation.Receipt* for **PageModeStation** before accessing this property, otherwise the value 0 is returned.

The following code sample shows the usage of **PageModeVerticalPosition**.

```
myptr.MapMode=MapMode.English;
myptr.PageModeStation=PrinterStation.Receipt;
myptr.PageModePrint(PageModePrintControl.PageMode);
// Set print area to 2 inches by 0.5 inches
myptr.PageModePrintArea=new System.Drawing.Point(0,0,1600,500);
myptr.PageModePrintDirection=PageModePrintDirection.LeftToRight;
myptr.PageModeVerticalPosition=250;
myptr.PrintNormal(PrinterStation.Receipt,"12345678901234567890123456789\n");
myptr.PageModePrint(PageModePrintControl.Normal);
```

The code sample above will generate the following receipt.



This property is initialized to 0 by **Open**.

RecBarCodeRotationList Property

Type **Rotation[]**

Description Gets a list of the directions in which a receipt barcode can be rotated.
The following table shows the valid property values.

Value	Meaning
Rotation.Left90	Barcode may be printed in a rotated 90° to the left.
Rotation.Normal	Barcode may be printed in the normal orientation.

Value	Meaning
Rotation.Right90	Barcode may be printed in a rotated 90° to the right.
Rotation.Rotate180	Barcode may be rotated 180° upside down.

This property is initialized to {*Rotation.Normal*, *Rotation.Right90*, *Rotation.Left90*, *Rotation.Rotate180*} by **Open**.

RecBitmapRotationList Property

Type **Rotation[]**

Description Gets a list of the directions in which a receipt bitmap can be rotated.
The following table shows the valid property values.

Value	Meaning
Rotation.Left90	Bitmap may be printed in a rotated 90° to the left.
Rotation.Normal	Bitmap may be printed in the normal orientation.
Rotation.Right90	Bitmap may be printed in a rotated 90° to the right.
Rotation.Rotate180	Bitmap may be rotated 180° upside down.

This property is initialized to {*Rotation.Normal*, *Rotation.Right90*, *Rotation.Left90*, *Rotation.Rotate180*} by **Open**.

RecCartridgeState Property

Type **PrinterCartridgeStates**

Description Gets the status of the selected receipt cartridge.
The following table shows the valid property value.

Value	Meaning
PrinterCartridgeStates.Unknown	Device does not support cartridge state reporting.

This property is initialized to *PrinterCartridgeStates.Unknown* when the device is first enabled following the **Open** call.

RecCurrentCartridge Property R/W

Type **PrinterColors**

Description Specifies the currently selected receipt cartridge.
The following table shows the valid property value.

Value	Meaning
PrinterColors.Primary	Supports primary color.

This property cannot be rewritten.
This property is initialized to *PrinterColors.Primary* when the device is first enabled following the **Open** call.

RecEmpty Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt is out of paper.
The following table shows the valid property values.

Value	Meaning
false	The receipt paper is present.
true	The receipt is out of paper.

This property is initialized and kept current while the device is enabled.
When **CoverOpen** is *true*, **RecEmpty** is not updated.

RecLetterQuality Property R/W

Type **bool**

Description Gets a Boolean value that indicates whether the printer prints in high-quality mode.
The following table shows the valid property values.

Value	Meaning
false	Prints in high-speed mode.
true	Prints in high-quality mode.

If [PrintQuality] in the configuration program is [RecLetterQuality Valid], this property is enabled and determines the print speed of the printer according to print mode. For print speed settings in the configuration program other than that, see "User's Guide".

This property is initialized to *false* when the device is enabled.

RecLineChars Property R/W

Type **int**

Description Gets or sets the number of characters that the application wants to print on a receipt line.
This property is set to one of the values which **RecLineCharsList** has.
Depending on this property, the printer prints in the following font.

MP-B21L

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
168	10	Font A (24 dots × 12 dots)	4 dots	24
	12		2 dots	
	12		1 dot	
	14 (Default)		0 dots	
	16	Font B (16 dots × 8 dots)	2 dots	16
	18		1 dot	
	21		0 dots	
176	11	Font A (24 dots × 12 dots)	4 dots	24
	12		2 dots	
	13		1 dot	
	14 (Default)		0 dots	
	17	Font B (16 dots × 8 dots)	2 dots	16
	19		1 dot	
	22		0 dots	
192	12	Font A (24 dots × 12 dots)	4 dots	24
	13		2 dots	
	14		1 dot	
	16 (Default)		0 dots	
	19	Font B (16 dots × 8 dots)	2 dots	16
	21		1 dot	
	24		0 dots	
208	13	Font A (24 dots × 12 dots)	4 dots	24
	14		2 dots	
	16		1 dot	
	17 (Default)		0 dots	
	20	Font B (16 dots × 8 dots)	2 dots	16
	23		1 dot	
	26		0 dots	

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
224	14	Font A (24 dots × 12 dots)	4 dots	24
	16		2 dots	
	17		1 dot	
	18 (Default)		0 dots	
	22	Font B (16 dots × 8 dots)	2 dots	16
	24		1 dot	
	28		0 dots	
240	15	Font A (24 dots × 12 dots)	4 dots	24
	17		2 dots	
	18		1 dot	
	20 (Default)		0 dots	
	24	Font B (16 dots × 8 dots)	2 dots	16
	26		1 dot	
	30		0 dots	
256	16	Font A (24 dots × 12 dots)	4 dots	24
	18		2 dots	
	19		1 dot	
	21 (Default)		0 dots	
	25	Font B (16 dots × 8 dots)	2 dots	16
	28		1 dot	
	32		0 dots	
272	17	Font A (24 dots × 12 dots)	4 dots	24
	19		2 dots	
	20		1 dot	
	22 (Default)		0 dots	
	27	Font B (16 dots × 8 dots)	2 dots	16
	30		1 dot	
	34		0 dots	
288	18	Font A (24 dots × 12 dots)	4 dots	24
	20		2 dots	
	22		1 dot	
	24 (Default)		0 dots	
	28	Font B (16 dots × 8 dots)	2 dots	16
	32		1 dot	
	36		0 dots	

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
304	19	Font A (24 dots × 12 dots)	4 dots	24
	21		2 dots	
	23		1 dot	
	25 (Default)		0 dots	
	30	Font B (16 dots × 8 dots)	2 dots	16
	33		1 dot	
	38		0 dots	
320	20	Font A (24 dots × 12 dots)	4 dots	24
	22		2 dots	
	24		1 dot	
	26 (Default)		0 dots	
	32	Font B (16 dots × 8 dots)	2 dots	16
	35		1 dot	
	40		0 dots	
336	21	Font A (24 dots × 12 dots)	4 dots	24
	24		2 dots	
	25		1 dot	
	28 (Default)		0 dots	
	33	Font B (16 dots × 8 dots)	2 dots	16
	37		1 dot	
	42		0 dots	
352	22	Font A (24 dots × 12 dots)	4 dots	24
	25		2 dots	
	27		1 dot	
	29 (Default)		0 dots	
	35	Font B (16 dots × 8 dots)	2 dots	16
	39		1 dot	
	44		0 dots	
360	22	Font A (24 dots × 12 dots)	4 dots	24
	25		2 dots	
	27		1 dot	
	30 (Default)		0 dots	
	36	Font B (16 dots × 8 dots)	2 dots	16
	40		1 dot	
	45		0 dots	

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
368	23	Font A (24 dots × 12 dots)	4 dots	24
	26		2 dots	
	28		1 dot	
	30 (Default)		0 dots	
	36	Font B (16 dots × 8 dots)	2 dots	16
	40		1 dot	
	45		0 dots	
384	24	Font A (24 dots × 12 dots)	4 dots	24
	27		2 dots	
	29		1 dot	
	32 (Default)		0 dots	
	38	Font B (16 dots × 8 dots)	2 dots	16
	42		1 dot	
	48		0 dots	

MP-B30L

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
360	22	Font A (24 dots × 12 dots)	4 dots	24
	25		2 dots	
	27		1 dot	
	30 (Default)		0 dots	
	36	Font B (16 dots × 8 dots)	2 dots	16
	40		1 dot	
	45		0 dots	
368	23	Font A (24 dots × 12 dots)	4 dots	24
	26		2 dots	
	28		1 dot	
	30 (Default)		0 dots	
	36	Font B (16 dots × 8 dots)	2 dots	16
	40		1 dot	
	46		0 dots	

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
384	24	Font A (24 dots × 12 dots)	4 dots	24
	27		2 dots	
	29		1 dot	
	32 (Default)		0 dots	
	38	Font B (16 dots × 8 dots)	2 dots	16
	42		1 dot	
	48		0 dots	
400	25	Font A (24 dots × 12 dots)	4 dots	24
	28		2 dots	
	30		1 dot	
	33 (Default)		0 dots	
	40	Font B (16 dots × 8 dots)	2 dots	16
	44		1 dot	
	50		0 dots	
416	26	Font A (24 dots × 12 dots)	4 dots	24
	29		2 dots	
	32		1 dot	
	34 (Default)		0 dots	
	41	Font B (16 dots × 8 dots)	2 dots	16
	46		1 dot	
	52		0 dots	
432	27	Font A (24 dots × 12 dots)	4 dots	24
	30		2 dots	
	33		1 dot	
	36 (Default)		0 dots	
	43	Font B (16 dots × 8 dots)	2 dots	16
	48		1 dot	
	54		0 dots	
448	28	Font A (24 dots × 12 dots)	4 dots	24
	32		2 dots	
	34		1 dot	
	37 (Default)		0 dots	
	44	Font B (16 dots × 8 dots)	2 dots	16
	49		1 dot	
	56		0 dots	

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
464	29	Font A (24 dots × 12 dots)	4 dots	24
	33		2 dots	
	35		1 dot	
	38 (Default)		0 dots	
	46	Font B (16 dots × 8 dots)	2 dots	16
	51		1 dot	
	58		0 dots	
480	30	Font A (24 dots × 12 dots)	4 dots	24
	34		2 dots	
	36		1 dot	
	40 (Default)		0 dots	
	48	Font B (16 dots × 8 dots)	2 dots	16
	53		1 dot	
	60		0 dots	
496	31	Font A (24 dots × 12 dots)	4 dots	24
	35		2 dots	
	38		1 dot	
	41 (Default)		0 dots	
	49	Font B (16 dots × 8 dots)	2 dots	16
	55		1 dot	
	62		0 dots	
512	32	Font A (24 dots × 12 dots)	4 dots	24
	36		2 dots	
	39		1 dot	
	42 (Default)		0 dots	
	51	Font B (16 dots × 8 dots)	2 dots	16
	56		1 dot	
	64		0 dots	
528	33	Font A (24 dots × 12 dots)	4 dots	24
	37		2 dots	
	40		1 dot	
	44 (Default)		0 dots	
	52	Font B (16 dots × 8 dots)	2 dots	16
	58		1 dot	
	66		0 dots	

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
544	34	Font A (24 dots × 12 dots)	4 dots	24
	38		2 dots	
	41		1 dot	
	45 (Default)		0 dots	
	54	Font B (16 dots × 8 dots)	2 dots	16
	60		1 dot	
	68		0 dots	
560	35	Font A (24 dots × 12 dots)	4 dots	24
	40		2 dots	
	43		1 dot	
	46 (Default)		0 dots	
	56	Font B (16 dots × 8 dots)	2 dots	16
	62		1 dot	
	70		0 dots	
576	36	Font A (24 dots × 12 dots)	4 dots	24
	41		2 dots	
	44		1 dot	
	48 (Default)		0 dots	
	57	Font B (16 dots × 8 dots)	2 dots	16
	64		1 dot	
	72		0 dots	

MP-B30

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
576	36	Font A (24 dots × 12 dots)	4 dots	24
	41		2 dots	
	44		1 dot	
	48 (Default)		0 dots	
	57	Font B (16 dots × 8 dots)	2 dots	16
	64		1 dot	
	72		0 dots	

If changed to a line character width that is less than or equal to the maximum value allowed for the printer, then the width is set to the specified value. If the exact width cannot be supported, then subsequent lines will be printed with a character size that most closely supports the specified characters per line. (For example, if 40 is set with 48 mm / 384 dots, then the Service Object selects a **RecLineChars** = 42.)

If the character width is not supported, then an error is returned. (For example, if 60 is set with 48 mm / 384 dots, then an error occurs.)
Setting **RecLineChars** may also update **RecLineHeight**, and **RecLineSpacing**.
The default of this property can be changed at the setting in the configuration program. This property is initialized to the value set in [Number of Characters per Line] of the configuration program when the device is enabled.

RecLineCharsList Property

Type **int[]**

Description Gets a collection of the line widths (characters per line) supported by the receipt station.

MP-B21L

RecLineWidth	Value
168	10,12,14,16,18,21
176	11,12,13,14,17,19,22
192	12,13,14,16,19,21,24
208	13,14,16,17,20,23,26
224	14,16,17,18,22,24,28
240	15,17,18,20,24,26,30
256	16,18,19,21,25,28,32
272	17,19,20,22,27,30,34
288	18,20,22,24,28,32,36
304	19,21,23,25,30,33,38
320	20,22,24,26,32,35,40
336	21,24,25,28,33,37,42
352	22,25,27,29,35,39,44
360	22,25,27,30,36,40,45
368	23,26,28,30,36,40,46
384	24,27,29,32,38,42,48

MP-B30L

RecLineWidth	Value
360	22,25,27,30,36,40,45
368	23,26,28,30,36,40,46
384	24,27,29,32,38,42,48
400	25,28,30,33,40,44,50

RecLineWidth	Value
416	26,29,32,34,41,46,52
432	27,30,33,36,43,48,54
448	28,32,34,37,44,49,56
464	29,33,35,38,46,51,58
480	30,34,36,40,48,53,60
496	31,35,38,41,49,55,62
512	32,36,39,42,51,56,64
528	33,37,40,44,52,58,66
544	34,38,41,45,54,60,68
560	35,40,43,46,56,62,70
576	36,41,44,48,57,64,72

MP-B30

RecLineWidth	Value
576	36,41,44,48,57,64,72

This property is initialized by **Open**.

RecLineHeight Property R/W

Type **int**

Description Gets or sets the receipt print line height.
The property is expressed in the unit indicated by **MapMode**.
This property cannot be rewritten. The value is automatically set by **RecLineChars**.

When **RecLineChars** is changed, **RecLineHeight** is updated to the height of the print font that corresponds to the set **RecLineChars**. For the relation between **RecLineChars** and the print font, see **RecLineChars**.

The relation between **RecLineHeight** and the print font is as follows.
The values in the following table are at the time of **MapMode** is *MapMode.Dots*.

Print Font (H × W)	RecLineHeight
Font A (24 × 12 dots)	24
Font B (16 × 8 dots)	16

This property is initialized to one value in the above values by the values set in [Number of Characters per Line] of the configuration program when the device is enabled.

RecLineSpacing Property R/W

Type **int**

Description Gets or sets the spacing of each single-high print line. This includes both the printed line height and the whitespace between each pair of lines.
The property is expressed in the unit indicated by **MapMode**.
The values in the following table are at the time of **MapMode** is *MapMode.Dots*.

The relation between **RecLineHeight** and **RecLineCpacing** is as follows.
When a value out of the range is specified, *ErrorCode.Illegal* is thrown and the property is not set.

RecLineHeight	RecLineCpacing	Default
24	24 to 255	30
16	16 to 255	

The default of this property can be changed at the setting in the configuration program. This property is initialized to the value of line spacing set in [Number of Effective Dots(dots)] of the configuration program, when the device is enabled.

RecLinesToPaperCut Property

Type **int**

Description Gets the number of lines that must be advanced before cutting the receipt paper.
The value obtained by dividing the distance between the print head and the tear bar of the printer by line spacing indicated by **RecLineSpacing** is set. Therefore, this property changes when **RecLineSpacing** is changed.

[Calculation formula]
RecLinesToPaperCut = 43 / **RecLineSpacing**

Example:
When **RecLineSpacig** is 31 (**MapMode=***MapMode.Dots*)
RecLinesToPaperCut = 43 / 31 = 1.3… = 2
(The decimal part is rounded up.)

This property is initialized to the value based on the above calculation using [Number of Characters per Line] of the configuration program by **Open**.

RecLineWidth Property

Type **int**

Description Gets the width of a line for the number of characters indicated by **RecLineChars**.
The property is expressed in the unit indicated by **MapMode**. The values in the following table are the values when **MapMode** = *MapMode.Dots*.

MP-B21L

Number of Effective Dots (dots)
168
176
192
208
224
240
256
272
288
304
320
336
352
360
368
384

MP-B30L

Number of Effective Dots (dots)
360
368
384
400
416
432
448
464
480
496
512
528
544

Number of Effective Dots (dots)
560
576

MP-B30

Number of Effective Dots (dots)
576

In MP-B21L and MP-B30L, the default of this property can be changed at the setting in the configuration program.
This property is initialized to one value in the above values by the value set at the [PrintWidth] of the configuration program when the device is enabled.

RecNearEnd Property

Type **bool**

Description Gets a Boolean value that indicates whether the receipt paper is low.
This property always shows *false*.

RecSidewaysMaxChars Property

Type **int**

Description Gets the maximum number of one-byte characters that may be printed on each line in sideways mode (rotated 90° to the left or right).

[Calculation formula]
RecSidewaysMaxChars
= Maximum height of **PageModeArea** / (Print font width / 2 + Character space)

Example:
When **PageModeArea** is "384,2400", and **RecLineChars** is 48
RecSidewaysMaxChars = 2400 / ((24 / 2) + 0) = 200
(The decimal part is rounded down.)

This property is initialized to the value based on the above calculation depending on **PageModeArea** and the print font, when the device is first enabled following **Open**.

RecSidewaysMaxLines Property

Type **int**

Description Gets the maximum number of lines that can be printed in sideways mode (rotated 90° to the left or right).

[Calculation formula]
RecSidewaysMaxLines
= (**RecLineWidth - RecLineHeight) / RecLineSpacing** + 1

Example:
When **RecLineWidth** is 384, **RecLineHeight** is 24, and **RecLineSpacing** is 30
RecSidewaysMaxLines = (384 - 24) / 30 + 1 = 19
(The decimal part is rounded down.)

This property is initialized to the value based on the above calculation depending on **RecLineWidth**, **RecLineSpacing**, and **RecLineHeight**, when the device is first enabled following **Open**.

RotateSpecial Property R/W

Type **Rotation**

Description Gets or sets the rotation orientation for barcodes.
The following table shows the valid property values.
If *rotation* contains *PrintRotation.Barcode* in **RotatePrint**, the rotating direction of the *rotation* is selected.

Value	Meaning
Rotation.Left90	Specifies rotated 90° left (counter-clockwise).
Rotation.Normal	Prints the next barcode in the normal print direction.
Rotation.Right90	Specifies rotated 90° right (clockwise).
Rotation.Rotate180	Specifies rotated 180° printing (upside down printing).

This property is initialized to *Rotation.Normal* by **Open**.

4.1.5 Common Methods

This section describes the details of the common methods for PosPrinter.
For details of the thrown exception errors, see "Appendix A Exceptions".

CheckHealth Method

Syntax **string CheckHealth(HealthCheckLevel** *level*);

Parameter	Meaning
level	Specifies the type of health check to be executed on the device.

・Values of *level*

Value	Meaning
HealthCheckLevel.External	Executes a complete test using the device. ROM version ID of the printer, **ServiceObjectVersion**, and **DeviceName** are printed on the printer.
HealthCheckLevel.Interactive	Executes an interactive test of the device. Displays a modal dialog box to execute a complete test using the device and display results.
HealthCheckLevel.Internal	Execute a health check without using the device physically.

Description Tests the status of the device.
A text description of the results of this method is placed in **CheckHealthText**.
CheckHealth is always executed synchronously.

Claim Method

Syntax **void Claim(int** *timeout*);

Parameter	Meaning
timeout	Specifies the maximum waiting time (in milliseconds) for exclusive access. If it is 0, the method returns the result immediately even if exclusive access of the device cannot be obtained. If *WaitForever*(-1) is set, the method waits until exclusive access is obtained.

Description Requests exclusive access to the device.
The PosPrinter device cannot be used until the exclusive access is obtained.
When it is successful, **Claimed** is set to *true*.
When the power is OFF or the cable is not connected, **Claim** is not available.

ClearOutput Method

Syntax **void ClearOutput**();

Description Clears all the buffered device outputs.
Any output error events that were queued (usually waiting for **FreezeEvents** to be set to *false*) are also cleared.

Close Method

Syntax **void Close**();

Description Releases the device and its resources.
If **DeviceEnabled** is *true*, the device is first disabled.
If **Claimed** is *true*, exclusive access to the device is first released.
Do not execute this while the event is in progress (or in the event handler).

CompareFirmwareVersion Method

This method is not supported. For the thrown exception errors, see "Appendix A Exceptions".

Syntax **CompareFirmwareResult CompareFirmwareVersion(string** *firmwareFileName*);

DirectIO Method

Syntax **DirectIOData DirectIO(int** *command*, **int** *data*, **object** *obj*);

Parameter	Meaning
command	Command number. Specific values assigned by the Service Object.
data	Additional numeric data. Specific values vary by command number and Service Object.
obj	Additional data provided by the Service Object. The value varies depending on the command number and what the Service Object sends.

Description The following functions are supported:
·Remaining memory capacity response
·International character selection
·Status response
DirectIO is always executed synchronously.

- **Remaining memory capacity response**

  Issues the printer command "Send NV Graphics Memory Remaining Capacity" and returns its response as a numeric value.
  The response data is stored in *DirectIOData.data*.

Parameter	Description
command	3
data	*null*
obj	*null*

- **International character selection**

  Selects the international character.
  To change the international character, select the international character with this method after setting **CharacterSet**.
  When changing **CharacterSet** to 932 after changing the international character, the international character is set to Japan.
  When changing **CharacterSet** to other than 932 after changing the international character, the international character is set to USA.

Parameter	Description
command	201
data	International character number n $0 \leq n \leq 12$ Country names available for n are as follows: 0: USA 1: France 2: Germany 3: United Kingdom 4: Denmark I 5: Sweden 6: Italy 7: Spain I 8: Japan 9: Norway 10: Denmark II 11: Spain II 12: Latin America
obj	*null*

- **Status response**

  Returns the "out-of-paper" sensor status in a numeric value.

Parameter	Description
command	501
data	1
obj	*null*

The response data is stored in *DirectIOData.data*.

Value	State
0	Paper is ready
1	No paper

Open Method

Syntax **void Open**();

Description Opens the device.
When **Open** is successful, the common properties and other class-specific properties are initialized.

Release Method

Syntax **void Release**();

Description Releases exclusive access to the device.
If **DeviceEnabled** is *true*, and the device is an exclusive-use device, then the device is first disabled.
Do not execute this while the event is in progress (or in the event handler).

ResetStatistic(string) Method

Syntax **void ResetStatistic**(**string** *statistic*);

Description Resets the specified statistics to 0.
For the statistics that can be reset, see "Appendix B Statistics".
ResetStatistic is always executed synchronously.

ResetStatistics() Method

Syntax **void ResetStatistics**();

Description Resets all the statistics to 0.
For the statistics that can be reset, see "Appendix B Statistics".
ResetStatistics is always executed synchronously.

ResetStatistics(StatisticCategories) Method

Syntax **void ResetStatistics**(**StatisticCategories** *statistics*);

Description Resets all the statistics of the specified category to 0.
For the statistics that can be reset, see "Appendix B Statistics".
ResetStatistics is always executed synchronously.

ResetStatistics(string[]) Method

Syntax **void ResetStatistics(string[]** *statistics*);

Description Resets the specified statistics to 0.
For the statistics that can be reset, see "Appendix B Statistics".
ResetStatistics is always executed synchronously.

RetrieveStatistic(string) Method

Syntax **string RetrieveStatistic(string** *statistic*);

Description Retrieves the specified device statistics.
For *statistic*, specify the statistics to retrieve.
When it is successful, **RetrieveStatistic** returns the XML string of the statistics.
For the statistics that are retrieved, see "Appendix B Statistics".
RetrieveStatistic is always executed synchronously.

RetrieveStatistics() Method

Syntax **string RetrieveStatistics**();

Description Retrieves all the device statistics.
When it is successful, **RetrieveStatistics** returns the XML string of the statistics.
For the statistics that are retrieved, see "Appendix B Statistics".
RetrieveStatistics is always executed synchronously.

RetrieveStatistics(StatisticCategories) Method

Syntax **string RetrieveStatistics(StatisticCategories** *statistics*);

Description Retrieves the statistics of the specified category.
statistics stores the category of the statistics to be retrieved by the application.
When it is successful, **RetrieveStatistics** returns the XML string of the statistics.
For the statistics that are retrieved, see "Appendix B Statistics".
RetrieveStatistics is always executed synchronously.

RetrieveStatistics(string[]) Method

Syntax **string RetrieveStatistics(string[]** *statistics*);

Description Retrieves the specified device statistics.
For *statistic*, specify the statistics to retrieve.
When it is successful, **RetrieveStatistics** returns the XML string of the statistics.
For the statistics that are retrieved, see "Appendix B Statistics".
RetrieveStatistics is always executed synchronously.

UpdateFirmware Method

This method is not supported. For the thrown exception errors, see "Appendix A Exceptions".

Syntax **void UpdateFirmware**(**string** *firmwareFileName*);

UpdateStatistic Method

This method is not supported. For the thrown exception errors, see "Appendix A Exceptions".

Syntax **void UpdateStatistic**(**string** *name*, **object** *value*);

UpdateStatistics(Statistic[]) Method

This method is not supported. For the thrown exception errors, see "Appendix A Exceptions".

Syntax **void UpdateStatistics**(**Statistic[]** *statistics*);

UpdateStatistics(StatisticCategories, Object) Method

This method is not supported. For the thrown exception errors, see "Appendix A Exceptions".

Syntax **void UpdateStatistics**(**StatisticCategories** *statistics*, **object** *value*);

4.1.6 Specific Methods

This section describes the details of the specific methods for PosPrinter.
For exception errors of specific methods that are not supported, see "Appendix A Exceptions".

ClearPrintArea Method

Syntax **void ClearPrintArea**();

Description Clears the print data on the Page Mode print area defined by **PageModePrintArea**. To clear the entire Page Mode area, specify **PageModePrintArea** to the area equivalent to the one indicated by **PageModeArea**, and then call **ClearPrintArea**. Specify *PrinterStation.Receipt* for **PageModeStation** before calling this method.

MarkFeed Method

Syntax **MarkFeed**(**MarkFeedType** *type*);

Parameter	Meaning
type	Indicates the control type of paper indicated mark.

・Values of *type*

Value	Meaning
PrinterMarkFeeds.Takeup	After detecting the mark or the gap, feeds the paper to the paper take-up position. The paper feed length is the length of the memory switches MS 21 to 22 (Mark Detection Cut Position Correction) of the printer. The default of the paper feed is as follows: MP-B21L is 122 dots (15.2 mm). MP-B30L is 125 dots (15.6 mm). MP-B30 is 160 dots (20 mm).
PrinterMarkFeeds.Cutter	After detecting the mark or the gap, feeds the paper to the cutting position. (Feeds the paper to the same position as *PrinterMarkFeeds.Takeup*).
PrinterMarkFeeds.CurrentTof	Supported only by MP-B21L. Feeds the marked paper backward to the printing position. When feeding the marked paper to the cut position by *PrinterMarkFeeds.Cutter* and feeding backward by *PrinterMarkFeeds.CurrentTof* after cutting the marked paper, the top margin of the marked paper can be reduced for the next printing. The paper feed length is 23 dots (2.88 mm) backward from the current position.

Value	Meaning
PrinterMarkFeeds.NextTof	Supported only by MP-B21L and MP-B30L. After detecting the next mark or the gap, feeds the paper to the printing position. The paper feed length is the length of the memory switches MS 26 to 27 (Mark Detection Print Position Correction) of the printer. The default of the paper feed is as follows: MP-B21L is 122 dots (15.2 mm). MP-B30L is 125 dots (15.6 mm).

Description

This method is used to utilize the paper that is available for the mark or the gap detection.

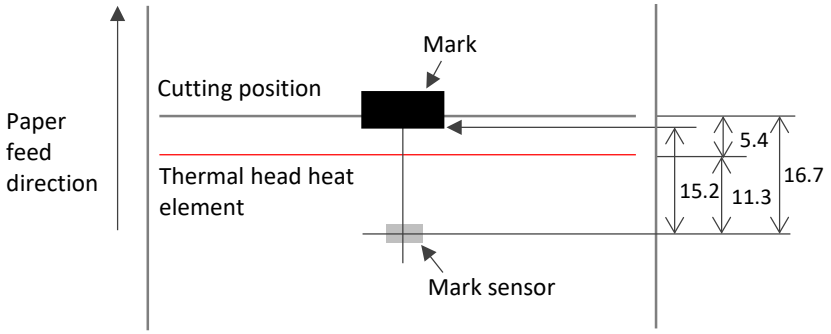
This method is executed synchronously if **AsyncMode** is *false*, and asynchronously if **AsyncMode** is *true*.

This method is available when the setting of [MarkMode] in the configuration program is set to Enable.

See "User's Guide" for the details of the memory switch of the printer.
The memory switches of the printer can be changed in the "SII Printer Setting Utility for Windows" that is to be expanded at the installation of "SII Printer Driver for Windows".

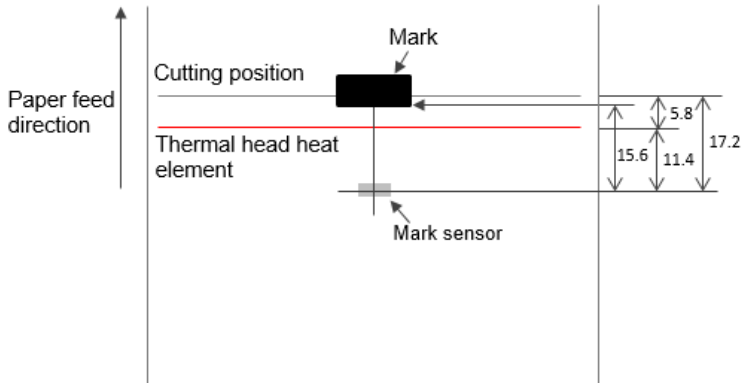
The relation between the sensor position and the defaults of memory switches MS 21 to 22 (Mark Detection Cut Position Correction) of the printer and memory switches MS 26 to 27 (Mark Detection Print Position Correction) of the printer are shown in the following figure.
When the memory switch of the printer is set to the default, the feed position of *PrinterMarkFeeds.Takeup*, the cutting position of *PrinterMarkFeeds.Cutter* and the next printing position of *PrinterMarkFeeds.NextTof* is the same paper feed length.

MP-B21L



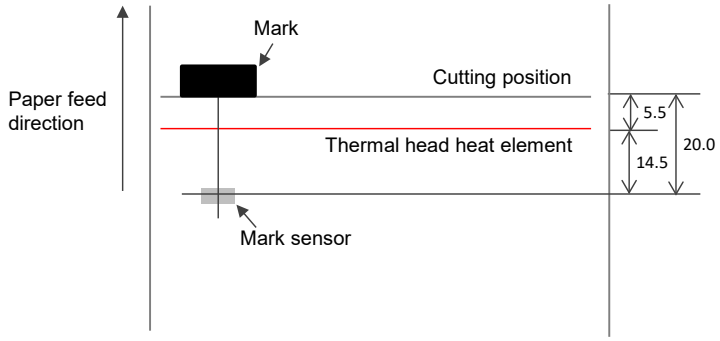
Unit : mm

MP-B30L



Unit : mm

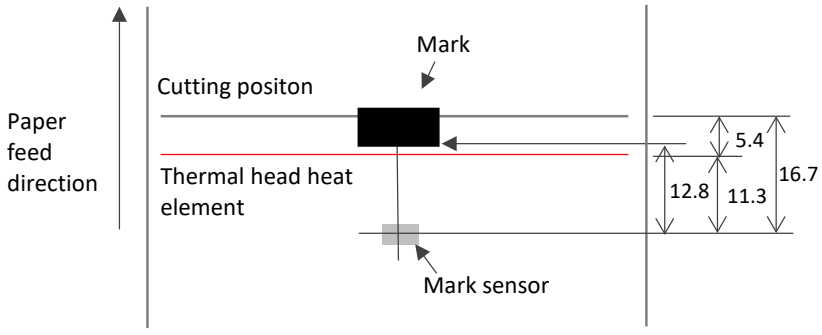
MP-B30



Unit : mm

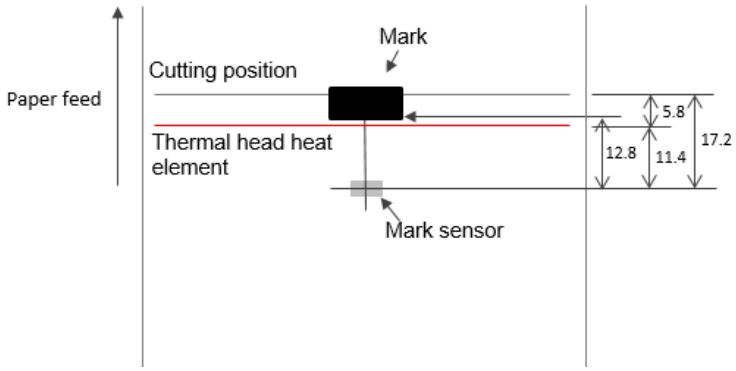
To set to shorter the next print position for the mark to save paper, set the values of the memory switches MS 26 to 27 (Mark Detection Print Position Correction) of the printer shorter.
As an example, the relation of the sensor position when the values of the memory switches MS 26 to 27 of the printer are set to 103 dots (12.8 mm) is shown in the following figure.

MP-B21L



Unit : mm

MP-B30L



Unit : mm

Notes The paper feed is not performed when this method is executed at the form feed position of the mark sensed paper or the label.

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

PageModePrint Method

Syntax **PageModePrint(PageModePrintControl** *control*);

Parameter	Meaning
control	Specifies the control type of Page Mode.

・Values of *control*

Value	Meaning
PageModePrintControl.Cancel	Clears the page and exits Page Mode without any printing of any print area.
PageModePrintControl.Normal	Prints the print area and destroys the canvas and exits Page Mode
PageModePrintControl.PageMode	Enters Page Mode.
PageModePrintControl.PrintSave	Prints the print data of the Page Mode print area and saves the data. This is used for repeated printings.

Description Enters or exits Page Mode for the station specified in **PageModeStation**.

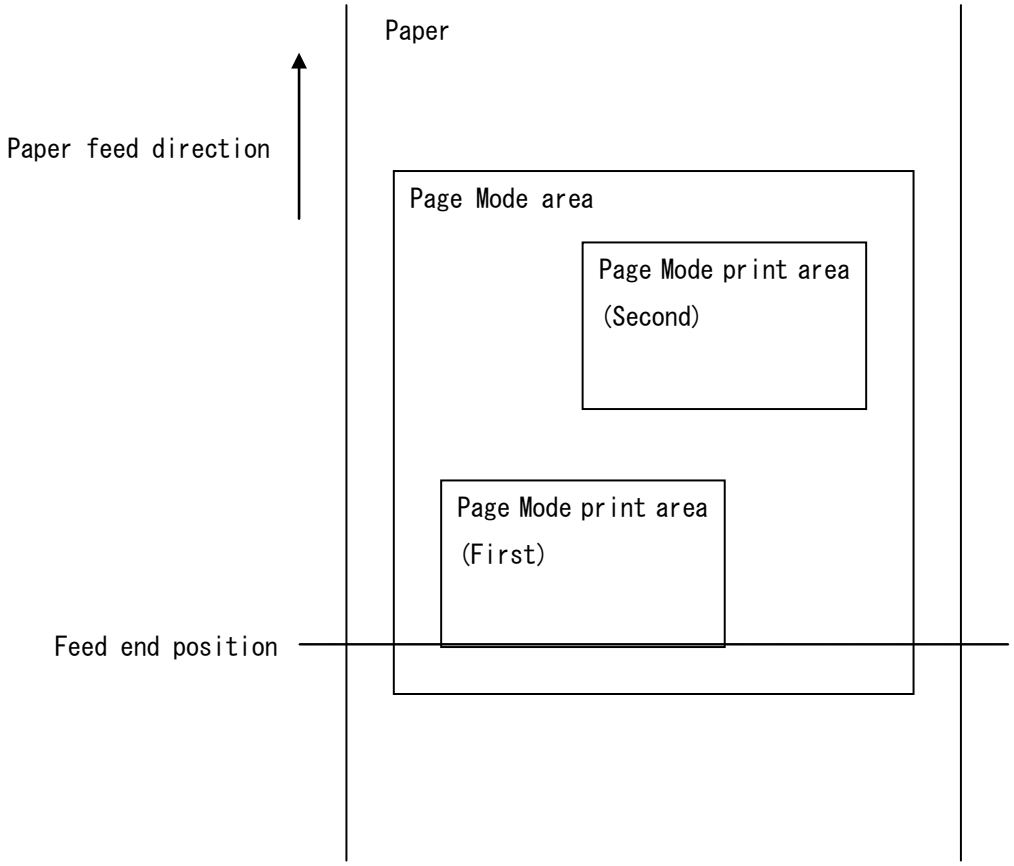
If *PageModePrintControl.PageMode* is specified for *control*, then Page Mode is started. Subsequently, the print data can be buffered with **PrintNormal**, **PrintBarCode**, **PrintBitmap**, or **PrintMemoryBitmap** until **PageModePrint** is called by specifying *PageModePrintControl.PrintSave*, *PageModePrintControl.Normal*, or *PageModePrintControl.Cancel*. (Methods called during this time only buffer the print data and they do not initiate printing. Also, the setting of **AsyncMode** does not affect the Page Mode function: No **OutputId** will be assigned, nor will **OutputCompleteEvent** be notified for each operation.)

If *PageModePrintControl.PrintSave* is specified for *control*, then Page Mode is continued. If some print data is buffered by one of **PrintNormal**, **PrintBarCode**, **PrintBitmap**, and **PrintMemoryBitmap**, then the data is saved and printed. This control is used to print the same page layout with additional print items inside of the page.

If *PageModePrintControl.Normal* is specified for *control*, then Page Mode is exited to return to the normal state. If some print data is buffered by one of **PrintNormal**, **PrintBarCode**, **PrintBitmap**, and **PrintMemoryBitmap**, then the data is printed. The buffered data will not be saved.

If *PageModePrintControl.Cancel* is specified for *control*, then Page Mode is exited to return to the normal state. If some print data is buffered by one of **PrintNormal**, **PrintBarCode**, **PrintBitmap**, and **PrintMemoryBitmap**, then the data is not printed or saved.

When **PageModePrint** is called while *PageModePrintControl.Normal* or *PageModePrintControl.PrintSave* is specified for *control*, all of the print data on the Page Mode print area defined by **PageModePrintArea** will be printed, and the paper is fed to the end of the area. If more than one Page Mode print area is defined, then after **PageModePrint** is called, all of the data that is to be printed in the respective Page Mode print area(s) will be printed, and the paper will be fed to the end of the Page Mode print area located the farthest "down" the sheet of paper. (See the figure below.)



This method is executed asynchronously if **AsyncMode** is *true*, or synchronously if **AsyncMode** is *false*.
Calling **ClearOutput** cancels Page Mode to return to the normal state. The buffered print data is also cleared. Page Mode can be used within a transaction print, but not within a rotate print.

Specify *PrinterStation.Receipt* for **PageModeStation** before calling this method.

PrintBarCode Method

Syntax **void PrintBarCode(PrinterStation** *station*,
string *data*,
BarCodeSymbology *symbology*,
int *height*,
int *width*,
int *alignment*,
BarCodeTextPosition *textPosition*);

Parameter	Meaning
station	Specifies the station to be used. Specify *PrinterStation.Receipt*.
data	Specifies the string of the barcode.
symbology	Specifies the barcode type to be used. See values below.
height	Specifies the height of the barcode. Expressed in the unit given by **MapMode**. When **MapMode** is *MapMode.Dots*, specify the height from 1 to 255. *height* of the following barcodes are ignored and set automatically by *width*, Specify from 1 to 255. ・QR Code ・Aztec Code[*1] ・GS1 Databar Omni-directional ・GS1 Databar Expanded ・GS1 Databar Expanded Stacked ・GS1 Databar Limited ・GS1 Databar Truncated *height* of the following barcodes are ignored and set with a fixed value. Specify from 1 to 255. ・GS1 Databar Stacked Omni-directional ・GS1 Databar Stacked During Page Mode by **PageModePrint**, specify the value within the range of print area specified by **PageModePrintArea** and **PageModeVerticalPosition**.

Parameter	Meaning
width	Specifies the width of the barcode. Expressed in the unit given by **MapMode**. The width of the barcode actually printed is the best fit within the width specified by *width*. When **MapMode** is *MapMode.Dots*, specify the width from 1 to **RecLineWidth** for upright position. When rotating the barcode 90° right/left by **RotateSpecial** or **RotatePrint**, specify in the range not exceeding the maximum value of the printer. (See **PageModeArea** for the maximum value of the printer.) During Page Mode by **PageModePrint**, specify the value within the range of print area specified by **PageModePrintArea** and **PageModeHorizontalPosition**.
alignment	Specifies the position of the barcode. See values below.
textPosition	Specifies the position of the text printed in the barcode. See values below.

*1: Not supported in MP-B30.

・Values of *symbology*

Value	Meaning
BarCodeSymbology.Codabar	Codabar (NW-7)
BarCodeSymbology.Code128	Code128
BarCodeSymbology.Code128Parsed	Code128 Parsed
BarCodeSymbology.Code39	Code39
BarCodeSymbology.Code93	Code93
BarCodeSymbology.Ean13S	EAN13 (JAN13) with supplemental barcode
BarCodeSymbology.EanJan13	EAN13 (JAN13)
BarCodeSymbology.EanJan8	EAN8 (JAN8)
BarCodeSymbology.Gs1DataBar	GS1 Databar Omni-directional
BarCodeSymbology.Gs1DataBarExpanded	GS1 Databar Expanded
BarCodeSymbology.Gs1DataBarStackedOmnidirectional or *BarCodeSymbology.Other* + 8	GS1 Databar Stacked Omni-directional
BarCodeSymbology.Gs1DataBarExpandedStacked or *BarCodeSymbology.Other* + 9	GS1 Databar Expanded Stacked
BarCodeSymbology.Itf	Interleaved 2 of 5
BarCodeSymbology.Other + 5	QR Code (Mixed mode)
BarCodeSymbology.Aztec	Aztec Code
BarCodeSymbology.Other + 6	GS1 Databar Limited
BarCodeSymbology.Other + 7	GS1 Databar Stacked
BarCodeSymbology.Other + 10	GS1 Databar Truncated
BarCodeSymbology.Pdf417	PDF417
BarCodeSymbology.Upca	UPC-A
BarCodeSymbology.Upce	UPC-E

・Values of *alignment*

Value	Meaning
PrinterBarCodeCenter	Printed with center.
PrinterBarCodeLeft	Printed with left justify.
PrinterBarCodeRight	Printed with right justify.
Other values	Printed with the left margin of the specified value. Expressed in the unit given by **MapMode**.

When rotation 90° right/left is specified by **RotateSpecial**, **RotatePrint**, and during Page Mode by **PageModePrint**, the setting of *alignment* is invalid and the data is always printed with left justify.

・Values of *textPosition*

Value	Meaning
BarCodeTextPosition.Above	Prints the text above the barcode.
BarCodeTextPosition.Below	Prints the text below the barcode.
BarCodeTextPosition.None	Does not print the text.

Call this method when printing the barcode on the specified printer.
This method is executed synchronously if **AsyncMode** is *false*, and asynchronously if **AsyncMode** is *true*.
If **RotateSpecial** indicates that the barcode is rotated, the barcode is printed in rotated mode. *height*, *width*, and *textPosition* are applied to the barcode before it is rotated.
For example, when specifying *Rotation.Left90* for **RotateSpecial** and calling this method specifying *BarCodeTextPosition.Below* for *textPosition* of this method, the text is placed below the barcode, and then the text and barcode are rotated 90° to the left and printed.

The barcode quiet zone is not secured. Verify that the barcode can be read with your actual device beforehand.
However, the following barcode quiet zones are unnecessary or secured:
・Aztec Code
・GS1 Databar Omni-directional
・GS1 Databar Expanded
・GS1 Databar Stacked Omni-directional
・GS1 Databar Expanded Stacked
・GS1 Databar Limited
・GS1 Databar Stacked
・GS1 Databar Truncated

The limitations for each barcode are described below.

[Codabar (NW-7)]

Parameter	Limitation
data	The head and end of line must be one of 'A' to 'D'. Other data must be at least one of '0' to '9', '$', '+', ':', '-', '.', and '/'.

Parameter	Limitation
width	When **MapMode** = *MapMode.Dots*: *width* = (((6 × X) + (2 × X × N)) × D) + ((X × N - X) × D') + (-1 × X) 20 × D + 2 × D' - 2 ≤ *width* ≤ 72 × D + 12 × D' - 6 D: the number of barcode characters D': the number of data characters (the number of 'A' to 'D', '+', '.', '/', ':' included in barcode data) X: fine element width 2 ≤ X ≤ 6 N: ratio of wide element width to fine element width (Set to 2, 2.5, or 3) X and N are automatically set according to *width*.

[Code128]

Parameter	Limitation
data	Specify any value consisting of decimal numbers from 0 to 105. Each numeric value is treated as the corresponding character shown in the table below. The first letter must be a decimal number 103, 104, or 105, and the barcode data of at least one letter must follow it.
width	When **MapMode** = *MapMode.Dots*: *width* = X × (((D + 2) × 11) + 2) 22 × D + 48 ≤ *width* ≤ 66 × D + 144 D: the number of barcode characters (including start code) X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *width*.

・Character set of Code128

Number	Character			Number	Character		
	Code A	Code B	Code C		Code A	Code B	Code C
0	SPACE[*1]	SPACE[*1]	00	53	U	U	53
1	!	!	01	54	V	V	54
2	"	"	02	55	W	W	55
3	#	#	03	56	X	X	56
4	$	$	04	57	Y	Y	57
5	%	%	05	58	Z	Z	58
6	&	&	06	59	[	[	59
7	'	'	07	60	\	\	60
8	(	(	08	61	]	]	61
9	)	)	09	62	^	^	62
10	*	*	10	63	_	_	63
11	+	+	11	64	NULL	`	64
12	,	,	12	65	SOH	a	65

Number	Character			Number	Character		
	Code A	Code B	Code C		Code A	Code B	Code C
13	-	-	13	66	STX	b	66
14	.	.	14	67	ETX	c	67
15	/	/	15	68	EOT	d	68
16	0	0	16	69	ENG	e	69
17	1	1	17	70	ACK	f	70
18	2	2	18	71	BEL	g	71
19	3	3	19	72	BS	h	72
20	4	4	20	73	HT	i	73
21	5	5	21	74	LF	j	74
22	6	6	22	75	VT	k	75
23	7	7	23	76	FF	l	76
24	8	8	24	77	CR	m	77
25	9	9	25	78	SO	n	78
26	:	:	26	79	SI	o	79
27	;	;	27	80	DLE	p	80
28	<	<	28	81	DC1	q	81
29	=	=	29	82	DC2	r	82
30	>	>	30	83	DC3	s	83
31	?	?	31	84	DC4	t	84
32	@	@	32	85	NAK	u	85
33	A	A	33	86	SYN	v	86
34	B	B	34	87	ETB	w	87
35	C	C	35	88	CAN	x	88
36	D	D	36	89	EM	y	89
37	E	E	37	90	SUB	z	90
38	F	F	38	91	ESC	{	91
39	G	G	39	92	FS	|	92
40	H	H	40	93	GS	}	93
41	I	I	41	94	RS	~	94
42	J	J	42	95	US	DEL	95
43	K	K	43	96	FNC3	FNC3	96
44	L	L	44	97	FNC2	FNC2	97
45	M	M	45	98	SHIFT	SHIFT	98
46	N	N	46	99	CODE C	CODE C	99
47	O	O	47	100	CODE B	FNC4	CODE B
48	P	P	48	101	FNC4	CODE A	CODE A
49	Q	Q	49	102	FNC1	FNC1	FNC1

Number	Character			Number	Character		
	Code A	Code B	Code C		Code A	Code B	Code C
50	R	R	50	103	START(CODE A)		
51	S	S	51	104	START(CODE B)		
52	T	T	52	105	START(CODE C)		

*1: Input a space.

[Code128 Parsed]

Parameter	Limitation
data	The head of line must be the special code (CODE A, CODE B, or CODE C) for the code set to use, and the barcode data of at least one letter must follow it. See "Code128 Special Code Table" for the special code. See "Input example of *data*" for input of *data*. The effective range of *data* differs by code set. ・Code A： 0x00 to 0x5f, FNC1, FNC2, FNC3, FNC4, SHIFT, CODE B, CODE C ・Code B： 0x20 to 0x7f, FNC1, FNC2, FNC3, FNC4, SHIFT, CODE A, CODE C ・Code C： 0x30 to 0x39, FNC1, CODE A, CODE B
width	When **MapMode** = *MapMode.Dots*: *width* = X × (((D + 2) × 11) + 2) $1^{*1} \leq$ *width* ≤ 66 × D + 144 D: the number of barcode characters (including start code) X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *width*.

*1: When 1 to 22 × D + 48 is set, *width* is set to 22 × D + 48.

・Code128 Special Code Table

data	Special Code
"{S"	SHIFT
"{A"	CODE A
"{B"	CODE B
"{C"	CODE C
"{1"	FNC1
"{2"	FNC2
"{3"	FNC3
"{4"	FNC4
"{{"	'{'

Input example of *data*

data is comprised of ASCII characters, which the service maps to the corresponding value for the selected code set. In Code A and Code B, this will be a one to one mapping. In Code C, each pair of digits is converted to a single Code C data character in the range 0x00 through 0x63. (If the Code C data contains an odd number of digits, then a leading 0 digit is added by the service before conversion.) A sentinel character, the left curly bracket “{”, followed by a certain value, is used to indicate a special character.

When creating a barcode of the barcode character "0123", the input of *data* is as follows according to the code set selected.

Selecting Code A : *data*="{A0123"
Selecting Code B : *data*="{B0123"
Selecting Code C : *data*="{C0123" or *data*="{C123"

[Code39]

Parameter	Limitation
data	At least one of '0' to '9', 'A' to 'Z', ' ', '$', '%', '+', '-', '.', '/' must be specified.
width	When **MapMode** = *MapMode.Dots*: *width* = (((X × 7)+(X × N × 3)) × (D + 2)) + (- 1 × X) 26 × D + 50 ≤ *width* ≤ 96 × D + 186 D: the number of barcode characters X: fine element width 2 ≤ X ≤ 6 N: ratio of wide element width to fine element width (Set to 2, 2.5, or 3) X and N are automatically set according to *width*.

[Code93]

Parameter	Limitation
data	Specify any value consisting of decimal numbers from 0 to 46. Each numeric value is treated as the corresponding character shown in the table below.
width	When **MapMode** = *MapMode.Dots*: *width* = X × ((D + 2 + 2) × 9) + 1) 18 × D + 74 ≤ *width* ≤ 54 × D + 222 D: the number of barcode characters X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *width*.

・Character set of Code93

Number	Character	Number	Character	Number	Character	Number	Character
0	0	12	C	24	O	36	-
1	1	13	D	25	P	37	.
2	2	14	E	26	Q	38	SPACE*1
3	3	15	F	27	R	39	$
4	4	16	G	28	S	40	/
5	5	17	H	29	T	41	+
6	6	18	I	30	U	42	%
7	7	19	J	31	V	43	($)
8	8	20	K	32	W	44	(%)
9	9	21	L	33	X	45	(/)
10	A	22	M	34	Y	46	(+)
11	B	23	N	35	Z		

*1: Input a space.

[EAN13 (JAN13) with supplemental barcode]

Parameter	Limitation
data	Specify 14, 15, 17, or 18 letters consisting of '0' to '9'. When 15 letters or 18 letters are entered, the 13th character does not affect the printing data.
width	When **MapMode** = *MapMode.Dots*: • When 14 or 15 letters are specified *width* = 122 × X 244 ≤ *width* ≤ 732 • When 17 or 18 letters are specified *width* = 149 × X 298 ≤ *width* ≤ 894 X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *width.*

[EAN13 (JAN13)]

Parameter	Limitation
data	Specify 12 or 13 letters consisting of '0' to '9'. The 13th letter does not affect the barcode printing data.
width	When **MapMode** = *MapMode.Dots*: *width* = 95 × X 190 ≤ *width* ≤ 570 X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *width.*

[EAN8 (JAN8)]

Parameter	Limitation
data	Specify 7 or 8 letters consisting of '0' to '9'. The 8th letter does not affect the barcode printing data.
width	When **MapMode** = *MapMode.Dots*: *width* = 67 × X 134 ≤ *width* ≤ 402 X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *width.*

[GS1 Databar Omni-directional]

Parameter	Limitation
data	Specify 13 letters consisting of '0' to '9'.
width	When **MapMode** = *MapMode.Dots*: *width* = 96 × X 1[*1] ≤ *width* ≤ 576 X: module width 2 ≤ X ≤ 6 X is automatically set according to *width.*

1*: When 1 to 287 is set, *width* is set to 192.

[GS1 Databar Expanded]

Parameter	Limitation
data	Specify '0' to '9', 'A' to 'Z', 'a' to 'z', space, '! ', '"', '%', '&', ''', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '>', '=', '?', '_' as 2 or more letters. Input "{1" for FNC1. Be sure to input the check digit since it is not automatically calculated by the printer.
width	Input the value other than 0.

[GS1 Databar Stacked Omni-directional]

Parameter	Limitation
data	Specify 13 letters consisting of '0' to '9'.
height	Input the value other than 0. The value is set as *height* = 138 regardless of the input value.
width	When **MapMode** = *MapMode.Dots*: Input the value other than 0. The value is set as *width* = 100 regardless of the input value.

Module width is fixed at 2.

[GS1 DataBar Expanded Stacked]

Parameter	Limitation
data	Two or more letters of '0' to '9', 'A' to 'Z', 'a' to 'z', space, '!', '"', '%', '&', '', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '>', '=', '?', '_' must be specified. Input "{1" for FNC1. Be sure to input the check digit since it is not automatically calculated by the printer.
width	Input the value other than 0.

Module width is fixed at 2.

[Interleaved 2 of 5]

Parameter	Limitation
data	Specify any value consisting of '0' to '9'. Note that the number of specified letters must be an even number except for 0.
width	When **MapMode** = *MapMode.Dots*: *width* = ((D × 2 + 1) × X × N) + ((D × 3 + 6) × X) 14 × D + 16 ≤ *width* ≤ 54 × D + 54 D: the number of barcode characters X: fine element width 2 ≤ X ≤ 6 N: ratio of wide element width to fine element width (Set to 2, 2.5, or 3) X and N are automatically set according to *width*.

[QR Code]

Parameter	Limitation
data	Specify characters of the following range: ASCII characters 8 bits Latin/Katakana characters based on JIS X 0201 Shift-JIS code based on JIS X 0208
width	*width* = (4V + 17) × M 42 ≤ *width* V: version of QR Code (1 to 40) M: module size (2 to 16) For version, the smallest value that input data can be converted to barcode is selected. For module size, the maximum size that does not exceed *width* is selected after the version is determined.

QR Code model is fixed at 2 and the error correction level is fixed at M. Printing size is based on *width*, and *height* is ignored since QR Code is a square.
If data other than the printable characters is specified, *ErrorCode.Illegal* is thrown.

[Aztec Code] *1

Parameter	Limitation
data	Specify characters of the following range: ASCII characters 8 bits Latin/Katakana characters based on JIS X 0201 Shift-JIS code based on JIS X 0208 For FNC1, input 1BH, 30H. For 1BH, input 1BH, 1BH.
width	*width* = V × X V: the number of module (15 to 151) X: module size (2 to 16) V and X are automatically set according to *data* and *width*.

*1: Not supported in MP-B30.

Aztec Code model is set automatically. The error correction level is fixed at 23%. Printing size is based on *width*, and *height* is ignored since Aztec Code is a square.

[GS1 Databar Limited]

Parameter	Limitation
data	Specify 13 letters consisting of '0' to '9'.
width	When **MapMode** = *MapMode.Dots*: The effective range of *height* depends on *width*. *width* = 79 × X 1*1 ≤ *width* ≤ 474 X: module width 2 ≤ X ≤ 6 X is automatically set according to *width.*

*1: When 1 to 236 is set, *width* is set to 158.

[GS1 Databar Stacked]

Parameter	Limitation
data	Specify 13 letters consisting of '0' to '9'.
height	Input the value other than 0. The value is set as *height* = 26 regardless of the input value.
width	When **MapMode** = *MapMode.Dots*: Input the value other than 0. The value is set as *width* = 100 regardless of the input value.

Module width is fixed at 2.

[GS1 Databar Truncated]

Parameter	Limitation
data	Specify 13 letters consisting of '0' to '9'.
width	When **MapMode** = *MapMode.Dots*: The effective range of *height* depends on *width*. *width* = 96 × X 1[*1] ≤ *width* ≤ 576 X: module width 2 ≤ X ≤ 6 X is automatically set according to *width.*

*1: When 1 to 287 is set, *width* is set to 192.

[PDF417]

Parameter	Limitation
data	It must be a character string in which 0x00 to 0x7F follow the ASCII code and 0x80 to 0xFF follow the extended character set of PC437 English list.
width *height*	*width* = (17 × C + 69) × X 180 ≤ *width* *height* = R × Y 14 ≤ *height* ≤ 255 X: module width (2 to 4) Y: module height (2 to 127) C: the number of vertical columns (1 to 30) R: the number of rows (3 to 90) For the number of rows and the number of vertical columns, the smallest value that input data can be converted to barcode is selected. For module width and module height, the maximum size that does not exceed *width* and *height* is selected after the number of rows and the number of vertical columns are determined.

Print mode is the normal mode and the error correction level is fixed to 4.

[UPC-A]

Parameter	Limitation
data	Specify 11 or 12 letters consisting of '0' to '9'. The 12th letter does not affect the barcode printing data.
width	When **MapMode** = *MapMode.Dots*: *width* = 95 × X 190 ≤ *width* ≤ 570 X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *width.*

[UPC-E]

Parameter	Limitation
data	Specify 11 or 12 letters consisting of '0' to '9'. The 12th letter does not affect the barcode printing data.
width	When **MapMode** = *MapMode.Dots*: *width* = 51 × X 102 ≤ *width* ≤ 306 X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *width.*

Additionally, the allowable character must follow the rules below.

1. The 1st letter is "0".
2. The UPC-A left code indicates the 2nd to the 6th characters, the UPC-A right code indicates the 7th to the 11th characters, and the code to be abbreviated is actually printed as UPC-E. If the specified UPC-A initial character is other than 0 or a character not included in the following list is specified, *ErrorCode.Illegal* is thrown.

Maker Code UPC-A Left Code					**Item Code UPC-A Right Code**					**Abbreviated Code**					
F1	F2	F3	F4	F5	A1	A2	A3	A4	A5	Z1	Z2	Z3	Z4	Z5	Z6
0-9	0-9	0	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	0
0-9	0-9	1	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	1
0-9	0-9	2	0	0	0	0	0-9	0-9	0-9	F1	F2	A3	A4	A5	2
0-9	0-9	3-9	0	0	0	0	0	0-9	0-9	F1	F2	F3	A4	A5	3
0-9	0-9	0-9	1-9	0	0	0	0	0	0-9	F1	F2	F3	F4	A5	4
0-9	0-9	0-9	0-9	1-9	0	0	0	0	5-9	F1	F2	F3	F4	F5	A5

PrintBitmap Method

Syntax **void PrintBitmap(PrinterStation** *station*, **string** *fileName*, **int** *width*, **int** *alignment*);

Parameter	Meaning
station	Specifies the station to be used. Specify *PrinterStation.Receipt*.
fileName	Specifies the name of bitmap file. For the supported image file, see below.
width	Specifies the print width of bitmap. See values below.
alignment	Specifies the print position of bitmap. See values below.

・Supported bitmap file

Item	Specifications
Extension	bmp
Format	Windows Bitmap
Color	1, 4, 8, 24, or 32 bits
Compression format	Uncompressed only

・Values of *width*

Value	Meaning
PrinterBitmapAsIs(-11)	Prints the bitmap with 1 pixel per printer dot.
Other values	Expresses the bitmap width in the unit defined by **MapMode**. If **MapMode** is *MapMode.Dots*, specify the value from 1 to **RecLineWidth**. When printing bitmap in a rotated 90° to right/left mode by **RotatePrint** is executed, specify the value from 1 to 2400. During Page Mode by **PageModePrint**, specify the value within the range of print area defined by **PageModePrintArea** and **PageModeHorizontalPosition**.

The value is rounded up to a multiple of 8 inside the Service Object.

・Values of *alignment*

Value	Meaning
PrinterBitmapCenter	Printed with center.
PrinterBitmapLeft	Printed with left justify.
PrinterBitmapRight	Printed with right justify.
Other values	Printed with the left margin of the specified value. Expressed in the unit given by **MapMode**.

When rotation 90° right/left is specified by **RotatePrint**, and during Page Mode by **PageModePrint**, the setting of *alignment* is invalid and the data is always printed with left justify.

Description

Prints a bitmap on the specified station.
The highest performance cannot be achieved since the bitmap data is transferred to the printer after **PrintBitmap** is called. It is recommended to print the bitmap data using **SetBitmap** and escape sequence.
If any character data is already sent but not yet printed, that character data is printed first, a linefeed is automatically added, and then the bitmap is printed on the next print line.
Any character data sent after **PrintBitmap** is printed on the print line following the bitmap.
This method is executed synchronously if **AsyncMode** is *false*, and asynchronously if **AsyncMode** is *true*.
width controls the transformation of bitmap data. If *width* is *PrinterBitmapAsIs*, then no transformation is executed. The bitmap is printed with 1 pixel per printer dot.
If *width* is not 0, then the bitmap will be transformed by stretching or compressing the bitmap such that its width is the specified width and the aspect ratio is unchanged.

PrintImmediate Method

Syntax **void PrintImmediate(PrinterStation** *station*, **string** *data*);

Parameter	Meaning
station	Specifies the station to be used. Specify *PrinterStation.Receipt*.
data	Specifies the characters to be printed. Consists of printable characters, escape sequences, carriage returns (CR), and line feeds (LF).

Description This method is used for printing immediately during asynchronous output.
When this method is specified during asynchronous output, processing of this method is executed before the next asynchronous output, after asynchronous output currently being processed.
PrintImmediate is intended for use in exception conditions when asynchronous output is not resolved, for example, in an error event handler.
The print data that exceeds the maximum number of characters per line is printed on the next print line.
If there is data remaining unprinted in the printer buffer, printing is executed after all the buffered data is printed.

The values and meanings of special characters within *data* are as follows.

Symbol	Operation
LF	Prints data in the buffer, and feeds to the next line.
CR	Replaceable with the same operation as line feed (LF).
LF & CR	Carriage return (CR) is replaceable with the same operation as line feed (LF). Therefore, operation of line feed (LF) is executed twice.
CR & LF	Carriage return (CR) is ignored. Operation of line feed (LF) is executed once.

PrintMemoryBitmap Method

Syntax **void PrintMemoryBitmap(PrinterStation** *station*,
Bitmap *data*,
int *width*,
int *alignment*);

Parameter	Meaning
station	Specifies the station to be used. Specify *PrinterStation.Receipt*.
data	Specifies the byte array holding the bitmap data. For the supported image file, see **PrintBitmap**.
width	Specifies the print width of bitmap. See **PrintBitmap** for values.
alignment	Specifies the print position of bitmap. See **PrintBitmap** for values.

Description Prints the bitmap on the specified station.
For the operation specifications, see **PrintBitmap**.
This method is executed synchronously if **AsyncMode** is *false*, and asynchronously if **AsyncMode** is *true*.

PrintNormal Method

Syntax **void PrintNormal(PrinterStation** *station*, **string** *data*);

Parameter	Meaning
station	Specifies the station to be used. Specify *PrinterStation.Receipt*.
data	Specifies the characters to be printed. Consists of printable characters, escape sequences, carriage returns (CR), and line feeds (LF).

Description Prints *data* on the printer.
The print data that exceeds the maximum number of characters per line is printed on the next print line.
If there is data remaining unprinted in the printer buffer, printing is executed after all the buffered data is printed.
This method is executed synchronously if **AsyncMode** is *false*, and asynchronously if **AsyncMode** is *true*.

The values and meanings of special characters within *data* are as follows.

Symbol	Operation
LF	Prints data in the buffer, and feed to the next line.
CR	Replaceable with the same operation as line feed (LF).
LF & CR	Carriage return (CR) is replaceable with the same operation as line feed (LF). Therefore, operation of line feed (LF) is executed twice.
CR & LF	Carriage return (CR) is ignored. Operation of line feed (LF) is executed once.

RotatePrint Method

Syntax **void RotatePrint(PrinterStation** *station*, **PrintRotation** *rotation*);

Parameter	Meaning
station	Specifies the station to be used. Specify *PrinterStation.Receipt*.
rotation	Specifies the rotation direction. See values below.

・Values of *rotation*

Value	Meaning
PrintRotation.Right90	Starts rotated 90° right (clockwise) printing.
PrintRotation.Left90	Starts rotated 90° left (counterclockwise) printing.
PrintRotation.Rotate180	Starts rotated 180° printing (upside down printing).
PrintRotation.Barcode	Starts rotated barcode printing. This value is ORed with one of the above start rotated print values.
PrintRotation.Bitmap	Starts rotated bitmap printing. This value is ORed with one of the above start rotated print values.
PrintRotation.Normal	Ends rotated printing.

Description

Executes the rotated printing.
This method is executed synchronously if **AsyncMode** is *false*, and asynchronously if **AsyncMode** is *true*.

When *rotation* contains *PrintRotation.Rotate180*, the upside down print mode is started. Subsequent calls to **PrintNormal** or **PrintImmediate** will print the data upside down until **RotatePrint** is called with *rotation* set to *PrintRotation.Normal*. Lines are printed in the order that they are sent to the Service Object, with the start of each line justified at the right margin of the printer. When *rotation* does not contain *PrintRotation.Barcode* or *PrintRotation.Bitmap*, only the print methods **PrintNormal** and **PrintImmediate** are used during the upside down print mode.

When *rotation* contains *PrintRotation.Right90* or *PrintRotation.Left90*, the sideways print mode is started. Until **RotatePrint** is called with *rotation* set to *PrintRotation.Normal*, the data called by **PrintNormal** is buffered. The value of **AsyncMode** does not affect the operation. In other words, no **OutputId** will be assigned to the request, nor will **OutputCompleteEvent** be notified.
Each print line is rotated by 90°. If all lines do not have the same length, the start positions of the lines are aligned. When *rotation* does not contain *PrintRotation.Barcode* or *PrintRotation.Bitmap*, only **PrintNormal** is used in the sideways print mode.

When *rotation* contains *PrintRotation.Normal*, the rotated print mode is exited. If some data is buffered by **PrintNormal** while the sideways rotated print mode is in effect, the buffered data is printed. The whole block of rotated lines is treated as one message.

When *rotation* contains *PrintRotation.Barcode* or *PrintRotation.Bitmap*, all of barcodes (printed by **PrintBarCode** or the "Print in-line barcode" escape sequence (ESC|[*]#R)) and bitmaps (printed by **PrintBitmap** or the "Print bitmap" escape sequence (ESC|#B)) can be printed in a rotated mode by **RotatePrint**. The rotation direction of barcodes and bitmaps are limited by **RecBarCodeRotationList** and **RecBitmapRotationList** respectively. When *rotation* contains *PrintRotation.Barcode*, the contents of **RotateSpecial** are ignored.
Calling **ClearOutput** cancels the rotated print mode.
Any buffered lines of sideways rotated print are also cleared.

The Service Object calculates so that the width in the sideways print mode becomes best size. The maximum width is 2400 dots in the sideways print mode. If the print data per line exceeds this range, non-printed data is printed by feeding to the next print line.
If the bitmap print or barcode print by the "Print in-line barcode" escape sequence (ESC|[*]#R) or the "Print bitmap" escape sequence (ESC|#B) is specified on **PrintNormal** during the rotation mode, the print data rotates regardless of whether or not *PrintRoatation.Bitmap* and *PrintRotation.Barcode* is specified in *rotation* with OR.

SetBitmap Method

Syntax

void SetBitmap(int *bitmapNumber*,
PrinterStation *station*,
string *fileName*,
int *width*,
int *alignment*);

Parameter	Meaning
bitmapNumber	Specifies the number to be assigned to this bitmap. Valid values are 1 to 20.
station	Specifies the station to be used. Specify *PrinterStation.Receipt*.
fileName	Specifies the name of bitmap file. If an empty string is set, the bitmap setting is canceled. For the supported image file, see **PrintBitmap**.
width	Specifies the print width of bitmap. See **PrintBitmap** for values.
alignment	Specifies the print position of bitmap. See **PrintBitmap** for values.

Description

Saves information of a bitmap to be printed.
The bitmap is printed by calling **PrintNormal** or **PrintImmediate** with the "Print Bitmap" escape sequence (ESC|#B) in the print data. The "Print Bitmap" escape sequence (ESC|#B) usually contains the character strings for printing the start and end process headers.

If any character data was sent before the "Print Bitmap" escape sequence (ESC|#B) and has not been printed, that character data is printed first, a linefeed is automatically placed, and then the bitmap is printed. Any character data sent after the "Print Bitmap" escape sequence (ESC|#B) is printed on the line next to the bitmap.
The Service Object prepares printing with downloading bitmap data in the NV graphics area of the printer. When bitmap print is specified by escape sequence, only command which conducts printing is transmitted to provide better performance.

SetLogo Method

Syntax **void SetLogo(PrinterLogoLocation** *location*, **string** *data*);

Parameter	Meaning
location	Specifies the logo to be set.
data	Specifies the characters that produce the logo. Consists of printable characters, escape sequences, carriage returns (CR), and line feeds (LF).

・Values of *location*

Value	Meaning
PrinterLogoLocation.Bottom	Produces a bottom logo.
PrinterLogoLocation.Top	Produces a top logo.

Description Saves a data string as the top or bottom logo.
The logo can be printed by calling **PrintNormal** or **PrintImmediate** with the "Print Top Logo" escape sequence (ESC|tL) or "Print Bottom Logo" escape sequence (ESC|bL) in the print data.

TransactionPrint Method

Syntax **void TransactionPrint(PrinterStation** *station*, **PrinterTransactionControl** *control*);

Parameter	Meaning
station	Specifies the station to be used. Specify *PrinterStation.Receipt*.
control	Specifies the type of the transaction. See below for values.

・Values of *control*

Value	Meaning
PrinterTransactionControl.Normal	Ends a transaction by printing the buffered data.
PrinterTransactionControl.Transaction	Starts a transaction.

Description Calls this method to enter or exit transaction mode.
If *control* is *PrinterTransactionControl.Transaction*, then transaction mode is entered. Subsequent calls to **PrintNormal**, **RotatePrint**, **PrintBarCode**, **PrintBitmap**, or **PageModePrint** will buffer the print data until **TransactionPrint** is called with *control* set to *PrinterTransactionControl.Normal*.
The value of **AsyncMode** does not affect the operation. In other words, no **OutputId** will be assigned to the request, nor will **OutputCompleteEvent** be notified.

If *control* is *PrinterTransactionControl.Normal*, then transaction mode is exited.
If some data was buffered by calls to **PrintNormal**, **RotatePrint**, **PrintBarCode**, **PrintBitmap**, or **PageModePrint**, then the buffered data is printed. The entire transaction is treated as one message.

This method is executed synchronously if **AsyncMode** is *false*, and asynchronously if **AsyncMode** is *true*.

Calling **ClearOutput** cancels transaction mode. Any buffered print lines are also cleared.

ValidateData Method

Syntax **void ValidateData(PrinterStation** *station*, **string** *data*);

Parameter	Meaning
station	Specifies the station to be used. Specify *PrinterStation.Receipt*.
data	Specifies the data to be validated. Consists of printable characters, escape sequences, carriage returns (CR), and line feeds (LF).

Description Before calling **PrintNormal** or **PrintImmediate**, this method is called to determine whether a data sequence, possibly including one or more escape sequences, is valid for the specified station.
This method does not cause any printing but is used to determine the capability of the station.
When not valid, the exception error is thrown. For details about the thrown exception errors, see "Appendix A Exceptions".

4.1.7 Events

This section describes the details of PosPrinter events.

DirectIOEvent Event

This event is not supported.

Syntax **DirectIOEventHandler DirectIOEvent;**

ErrorEvent Event

Syntax **DeviceErrorEventHandler ErrorEvent;**

Description This event is notified when an error occurs and **State** of the Service Object enters the error state.
When *DeviceErrorEventArgs.ErrorCode* is *ErrorCode.Extended*, *DeviceErrorEventArgs.ErrorCodeExtended* is set to one of the following values:

Value	Meaning
ExtendedErrorCoverOpen(201)	The printer cover is open.
ExtendedErrorRecEmpty(203)	The receipt is out of paper.
ExtendedErrorVpPower(1001)	Vp voltage error has occurred.
ExtendedErrorHeadTemp(1005)	Head-temperature error has occurred.
ExtendedErrorFatal(1010)	Unrecoverable error has occurred.
ExtendedErrorBattery(1013)	Battery error has occurred.
ExtendedErrorMarkPaperJam(1014)	Paper jam error has occurred at the mark or the gap detection.

The *DeviceErrorEventArgs.ErrorResponse* can be set to either of the following values by the application. The default is *ErrorResponse.Retry*.

Value	Meaning
ErrorResponse.Clear	Exits the error state and clears the asynchronous output.
ErrorResponse.Retry	Exits the error state and retries the asynchronous output.

OutputCompleteEvent Event

Syntax **OutputCompleteEventHandler OutputCompleteEvent;**

Description This event is notified when the previously started asynchronous output request is completed successfully.
OutputId indicates the ID number of the completed asynchronous output request.

StatusUpdateEvent Event

Syntax **StatusUpdateEventHandler StatusUpdateEvent;**

Description The status of the device is notified when the device encounters an important status change.
The Service Object notifies the first **StatusUpdateEvent** when the device is enabled.

StatusUpdateEventArgs.Status is set to one of the following values:

Value	Meaning
StatusCoverOpen(11)	The printer cover is open.
StatusCoverOK(12)	The printer cover is closed.
StatusReceiptEmpty(24)	The receipt is out of paper.
StatusReceiptPaperOK(26)	The receipt is ready.
StatusIdle(1001)	All the asynchronous outputs finished either successfully or by being cleared. **State** is now *ControlState.Idle*. **FlagWhenIdle** must be *true* for this event to be notified. And the Service Object automatically resets the property to *false* before the event is notified.
StatusPowerOnline(2001)*1	The device is powered on and ready.
StatusPowerOffOffline(2004)*1	The device is powered off or offline.

*1: Notified when **PowerNotify** = *PowerNotification.Enabled*.

Appendix A Exceptions

A.1 PosPrinter Exception Error List

(1) Property

ErrorCode	ErrorCode Extended	Meaning
Disabled	0	Not enabled. Call this after setting **DeviceEnabled** to *true*.
Illegal	0	This property is not supported. Parameter has an error.
NotClaimed	0	Exclusive access is not available. Call **Claim** to gain exclusive access.

(2) Method

Method	ErrorCode	ErrorCode Extended	Meaning
BeginInsertion **BeginRemoval** **ChangePrintSide** **CutPaper** **CompareFirmwareVersion** **EndInsertion** **EndRemoval** **PrintTwoNormal** **UpdateFirmware** **UpdateStatistic(s)**	Illegal	0	This method is not supported.
CheckHealth	Busy	0	Cannot perform while output is in progress or an error occurs.
	Disabled	0	Not enabled. Call this after setting **DeviceEnabled** to *true*.
	Illegal	0	Parameter has an error.
	NotClaimed	0	Exclusive access is not available. Call **Claim** to gain exclusive access.

Method	ErrorCode	ErrorCode Extended	Meaning
Claim	Claimed	0	Attempt was made to access a device that is exclusively accessed by another process.
	Failure	0	Communication with the printer failed.
	Illegal	0	Parameter has an error.
	NoHardware	0	The printer is powered off or the cable is not connected.
	Timeout	0	Another application has exclusive access to the device and the *timeout* (in milliseconds) has elapsed before the device is released. Or the device did not become available even though the *timeout* (in milliseconds) has elapsed.
ClearOutput	NotClaimed	0	Exclusive access is not available. Call **Claim** to gain exclusive access.
ClearPrintArea	Disabled	0	Not enabled. Call this after setting **DeviceEnabled** to *true*.
	NotClaimed	0	Exclusive access is not available. Call **Claim** to gain exclusive access.
Close	Busy	0	**State** is set to *ControlState.Busy*. This means that the device is busy and cannot be stopped.
	Closed	0	The device is already closed.
Open	Illegal	0	The device is already open.
DirectIO **MarkFeed** **PageModePrint** **PrintBarCode** **PrintBitmap** **PrintMemoryBitmap** **PrintNormal** **PrintImmediate** **RotatePrint** **SetBitmap** **TransactionPrint**	Busy	0	Cannot perform while output is in progress or an error occurs.
	Disabled	0	Not enabled. Call this after setting **DeviceEnabled** to *true*.
	Extended	201	The printer cover is open.
	Extended	203	The receipt is out of paper.
	Extended	1001	A Vp voltage error occurred.
	Extended	1005	A head temperature error occurred.
	Extended	1010	An unrecoverable error occurred.
	Extended	1013	A battery error occurred.
	Extended	1014	Paper jam error has occurred at the mark or the gap detection.
	Failure	0	Communication with the printer failed.
	Illegal	0	Parameter has an error. **PageModeStation** is not specified. (**PageModePrint**)
	NoHardware	0	The printer is powered off or the cable is not connected.
	NotClaimed	0	Exclusive access is not available. Call **Claim** to gain exclusive access.
	Timeout	0	Data transmit timeout or data receive timeout has occurred.
Release	NotClaimed	0	The device is not exclusive.

Method	ErrorCode	ErrorCode Extended	Meaning
ValidateData	Disabled	0	Not enabled. Call this after setting **DeviceEnabled** to *true*.
	Failure	0	At least one of the escape sequences is not supported. No alternatives can be selected.
	Illegal	0	At least one of the escape sequences is out of the range. However, the Service Object can select valid alternatives. Also, this value is placed when the escape sequence is not supported by the Page Mode or rotated 90° left or right print mode.
	Illegal	0	Parameter has an error.
	NotClaimed	0	Exclusive access is not available. Call **Claim** to gain exclusive access.

ErrorCode.Illegal is thrown for **ValidateData** in the following cases:

Escape Sequence	Condition
Print bitmap	No printable bitmap exists.
Feed lines	It is in one of the following states. ・ The number of lines '#' is not correct. ・ Not supported during rotated 90° right/left mode by **RotatePrint**. ・ Not supported during Page Mode by **PageModePrint**.
Feed units	It is in one of the following states. ・ The amount of feed '#' is not precisely supported due to occurrence of rounding error of one dot depending on the setting of **MapMode**. ・ The amount of feed '#' is not correct. ・ Not supported during rotated 90° right/left mode by **RotatePrint**. ・ Not supported during Page Mode by **PageModePrint**.
Pass through embedded data	The number of bytes of embedded data '#' is not correct.
Print in-line barcode	The character string following ESC|[*]#R is not correct.
Underline	The thickness '#' is not correct.
Scale vertically	The scale factor '#' is not correct.
Scale horizontally	The scale factor '#' is not correct.
Left justify	It is in one of the following states. ・ Not supported during rotated 90° right/left mode by **RotatePrint**. ・ Not supported during Page Mode by **PageModePrint**.
Center	It is in one of the following states. ・ Not supported during rotated 90° right/left mode by **RotatePrint**. ・ Not supported during Page Mode by **PageModePrint**.
Right justify	It is in one of the following states. ・ Not supported during rotated 90° right/left mode by **RotatePrint**. ・ Not supported during Page Mode by **PageModePrint**.

ErrorCode.Failure is thrown for **ValidateData** in the following cases:

Escape Sequence	Condition
Paper cut	Not supported.
Feed and Paper cut	Not supported.
Feed, Paper cut, and Stamp	Not supported.
Print bitmap	The bitmap number '#' is not correct.
Print stamp	Not supported.
Feed reverse	Not supported.
Font typeface	Not supported.
Italic	Not supported.
Custom color	Not supported.
Red color	Not supported.
Shading character	Not supported.
Color option	Not supported.
SubScript	Not supported.
SuperScript	Not supported.
Strike-through	Not supported.

Appendix B Statistics

(1) StatisticCategories.Upos

XML Definition Name	Response	Can Be Reset
JournalCoverOpenCount	0	-
ReceiptLineFeedCount	Number of receipt line feeds performed (unit: 100 dot-line)	✓
PrintSideChangeCount	0	-
ReceiptCharacterPrintedCount	0	-
ReceiptCoverOpenCount	0	-
ManufactureDate	Unknown	-
PaperCutCount	0	-
UnifiedPOSVersion	MP-B21L: 1.14.1 MP-B30L, MP-B30: 1.12	-
SlipCoverOpenCount	0	-
HoursPoweredCount	Number of hours powered on (unit: hour)	✓
FirmwareRevision	Firmware version	-
SerialNumber	Unknown	-
ReceiptLinePrintedCount	0	-
InstallationDate	Unknown	-
MechanicalRevision	MP-B21L: 2A MP-B30L, MP-B30: 24	-
FailedPaperCutCount	0	-
StampFiredCount	0	-
FailedPrintSideChangeCount	0	-
JournalCharacterPrintedCount	0	-
SlipCharacterPrintedCount	0	-
ManufacturerName	Seiko Instruments Inc.	-
PrinterFaultCount	0	-

XML Definition Name	Response	Can Be Reset
MaximumTempReachedCount	0	-
ModelName	MP-B21L MP-B30L MP-B30	-
CommunicationErrorCount	0	-
JournalLinePrintedCount	0	-
SlipLineFeedCount	0	-
HomeErrorCount	0	-
FormInsertionCount	0	-
Interface	Unknown	-
DeviceCategory	POSPrinter	-
BarcodePrintedCount	0	-
NVRAMWriteCount	0	-
SlipLinePrintedCount	0	-

(2) StatisticCategories.Manufacturer

XML Definition Name	Response	Can Be Reset
HoursPoweredCount_Accumulated	Number of hours powered on (unit: hour) (accumulated)	-
ReceiptLineFeedCount_Accumulated	Number of receipt line feeds performed (unit: 100 dot-line) (accumulated)	-