SII OPOS
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 OPOS" (hereinafter referred to as the "software") provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

This software is OpenPOS control based on UnifiedPOS Retail Peripheral Architecture Version 1.14.1 for controlling printers.

Target Products

This section lists the products supported by the software.

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

See "UnifiedPOS Retail Peripheral Architecture Version 1.14.1" (hereinafter referred to as "UPOS V 1.14.1") before using this software.

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.

Definition	Description
UPOS V1.14.1	"UnifiedPOS Retail Peripheral Architecture Version 1.14.1"
OPOS specification	Specification of the POS device interface defined in UPOS V1.14.1.
OPOS Control	The device driver which conforms to the ActiveX Control and provides the function to easily integrate the POS device (POS peripheral equipment) into the POS system built on Windows. The OPOS Control described in this manual supports the API defined in the OPOS specification.
Device class	A category of POS devices that share a set of properties, methods, and events defined in the OPOS specification. In this manual, the device class of POS printer is used.
Device name	Name for identifying the POS device targeted for control which is specified when the OPOS Control is used. It is impossible to create the duplicated device name in the same device class.
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 [SIIOPOSSettingUtility] on the computer.
Logical device name	Alias of device name. Arbitrary name which is set in the configuration program.
Control Object	Provides a set of properties, methods, and events to an application for each device class.
Service Object	This is called by the Control Object to execute the OPOS prescribed function for each device.
POS Printer Control	OPOS Control of the POS Printer device class. Controlled device is the printer.
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 Configuratioin 1-2
1.1.1 Structural Diagram 1-2
1.2 Printer Settings 1-3
1.3 Limitations 1-5
1.3.1 General 1-5
1.3.2 POS Printer Control 1-5

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 3-8
3.2.1 Addition of Device 3-8
3.2.2 Changing Device Settings 3-18
3.2.3 Deletion of Device 3-19
3.2.4 Adding and Deleting Logical Device Name 3-20
3.2.5 Device Interactive Test 3-24
3.2.6 Memory settings of printer (PrinterUtility) 3-26

Chapter 4 OPOS Control 4-1

4.1 POS Printer Control 4-2

5.1 Summary 1
5.1.1 Common Properties 1
5.1.2 Specific Properties 2
5.1.3 Common Methods 6
5.1.4 Specific Methods 6
5.1.5 Events 7
5.2 Data Characters and Escape Sequences 8
5.2.1 Escape Sequence Operated when Specified 8
5.2.2 Escape Sequence Valid Until Changed 11
5.2.3 Escape Sequence Reset by End of Print Method or "Normal" Escape Sequence 11
5.3 Common Properties 13
BinaryConversion Property R/W 13
CapCompareFirmwareVersion Property 14
CapPowerReporting Property 14
CapStatisticsReporting Property 14
CapUpdateFirmware Property 15
CapUpdateStatistics Property 15
CheckHealthText Property 15
Claimed Property 16
ControlObjectDescription Property 16
ControlObjectVersion Property 16
DeviceDescription Property 16
DeviceEnabled Property R/W 17
DeviceName Property 17
FreezeEvents Property R/W 18
OpenResult Property 18
OutputID Property 19
PowerNotify Property R/W 19
PowerState Property 20
ResultCode Property 20
ResultCodeExtended Property 21
ServiceObjectDescription Property 22
ServiceObjectVersion Property 22
State Property 22
5.4 Specific Properties 23
AsyncMode Property R/W 23
CapCharacterSet Property 23
CapCoverSensor Property 23
CapMapCharacterSet Property 24
CapRec2Color Property 24
CapRecBarCode Property 24
CapRecBitmap Property 24
CapRecBold Property 25
CapRecCartridgeSensor Property 25

CapRecColor Property 25
CapRecDhigh Property 25
CapRecDwide Property 26
CapRecDwideDhigh Property 26
CapRecEmptySensor Property 26
CapRecItalic Property 26
CapRecLeft90 Property 27
CapRecMarkFeed Property 27
CapRecNearEndSensor Property 27
CapRecPageMode Property 28
CapRecPapercut Property 28
CapRecPresent Property 28
CapRecRight90 Property 28
CapRecRotate180 Property 29
CapRecRuledLine Property 29
CapRecStamp Property 29
CapRecUnderline Property 29
CapTransaction Property 30
CartridgeNotify Property R/W 30
CharacterSet Property R/W 30
CharacterSetList Property 31
CoverOpen Property 31
ErrorLevel Property 32
ErrorStation Property 32
ErrorString Property 32
FlagWhenIdle Property R/W 33
FontTypefaceList Property 33
MapCharacterSet Property R/W 34
MapMode Property R/W 34
PageModeArea Property 35
PageModeDescriptor Property 36
PageModeHorizontalPosition Property R/W 37
PageModePrintArea Property R/W 37
PageModePrintDirection Property R/W 38
PageModeStation Property R/W 40
PageModeVerticalPosition Property R/W 40
RecBarCodeRotationList Property 41
RecBitmapRotationList Property 42
RecCartridgeState Property 42
RecCurrentCartridge Property R/W 42
RecEmpty Property 42
RecLetterQuality Property R/W 43
RecLineChars Property R/W 43
RecLineCharsList Property 50
RecLineHeight Property R/W 51
RecLineSpacing Property R/W 52

RecLinesToPaperCut Property 53
RecLineWidth Property 53
RecNearEnd Property 54
RecSidewaysMaxChars Property 55
RecSidewaysMaxLines Property 55
RotateSpecial Property R/W 55
5.5 Common Methods 56
CheckHealth Method 56
ClaimDevice Method 57
ClearOutput Method 57
Close Method 58
CompareFirmwareVersion Method 58
DirectIO Method 58
Open Method 60
ReleaseDevice Method 61
ResetStatistics Method 61
RetrieveStatistics Method 62
UpdateFirmware Method 65
UpdateStatistics Method 65
5.6 Specific Methods 66
BeginInsertion Method 66
BeginRemoval Method 66
ChangePrintSide Method 66
ClearPrintArea Method 66
CutPaper Method 67
DrawRuledLine Method 67
EndInsertion Method 68
EndRemoval Method 69
MarkFeed Method 69
PageModePrint Method 72
PrintBarCode Method 74
PrintBitmap Method 86
PrintImmediate Method 88
PrintMemoryBitmap Method 89
PrintNormal Method 91
PrintTwoNormal Method 92
RotatePrint Method 92
SetBitmap Method 94
SetLogo Method 95
TransactionPrint Method 96
ValidateData Method 97
5.7 Events 100
DirectIOEvent Event 100
ErrorEvent Event 100
OutputCompleteEvent Event 101
StatusUpdateEvent Event 101

Chapter 6 Registry Used by This Software 6-1

6.1 POS Printer Control 6-1

Chapter 7 Header File 7-1

7.1 POS Printer Header File..... 7-1

Chapter 1 Overview

This chapter describes the overview of the software.

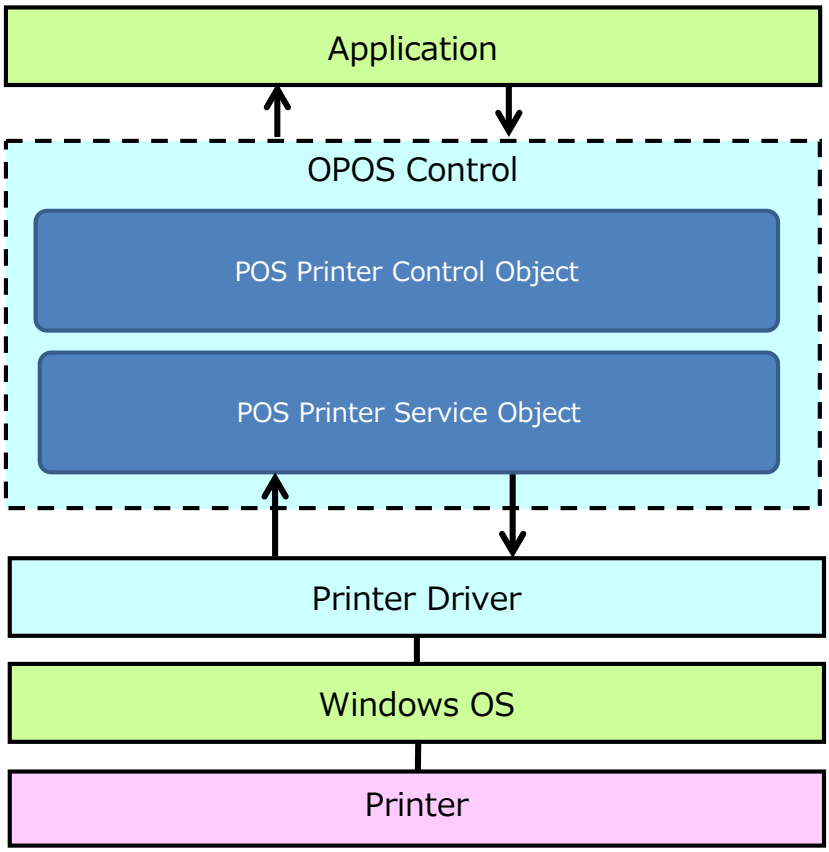
This software is based on ActiveX control specifications. This software provides the properties, methods and events to the applications. This software is not seen on UI at an application execution; only the application using those executes processing requests through the methods and the properties. The application receives processing result through the return values of methods, parameters, properties and events. This software is implemented as in-process server.

For the contents of the header file used by the control, see the "OPOS Internal Header Files" of "Appendix A OLE FOR RETAIL POS - OPOS IMPLEMENTATION REFERENCE" in UPOS V1.14.1.
In addition, the additional specific values of the OPOS Control are defined. For these values, refer to "Chapter 7 Header File" in this manual.

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 broken lines.



1.2 Printer Settings

The memory switches of the printer are set to [Value] in the following table 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 one [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	The execution of **ClaimDevice** sets [Value] on the left forcibly.
1-6	Data Discard Selection When Error Occurs (Error Through)	MP-B21L 0 : Enable	The execution of **ClaimDevice** sets [Value] on the left forcibly.
		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".
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 [PrintWidth] in the configuration program.

MS	Function	Value	Note
3-1 to 3-8	Print Width (Print Width)	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	Any one of [Value] on the left can be set by [PrintWidth] in the configuration program.
9-1	Automatic Status Response Selection (Auto Status Back)	0 : Enable	The execution of **ClaimDevice** sets [Value] on the left forcibly.
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	Any 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".
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	The execution of **ClaimDevice** sets [Value] on the left forcibly.
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 POS Printer Control

All interfaces of POS Printer Device Class defined by OPOS specifications are provided, with the following limitations.

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

(b) The following functions are not supported.

- Paper cut
- Feed and Paper cut
- Feed, Paper cut and Stamp
- Stamp
- Feed reverse
- Font typeface selection
- Italic
- Alternate color (Custom)
- Shading
- RGB Color
- Sub Script
- Super Script
- Strike-through

(c) All the following methods always return OPOS_E_ILLEGAL(106) after they are enabled.

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

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

(e) When the printer setting item [ProcessCompletionTiming] is set to "Printed" in the configuration program, the printer command "Execution Response Request" is used to control the print operation. Therefore, unexpected behavior may occur when the printer command "Execution Response Request" is transmitted by the "Pass through embedded data" escape sequence (ESC|#E).

(f) When an error occurs, the printer command "Hardware Reset" is transmitted by the POS Printer Control to cancel printing in the printer. However, printing may be performed a bit before POS Printer 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

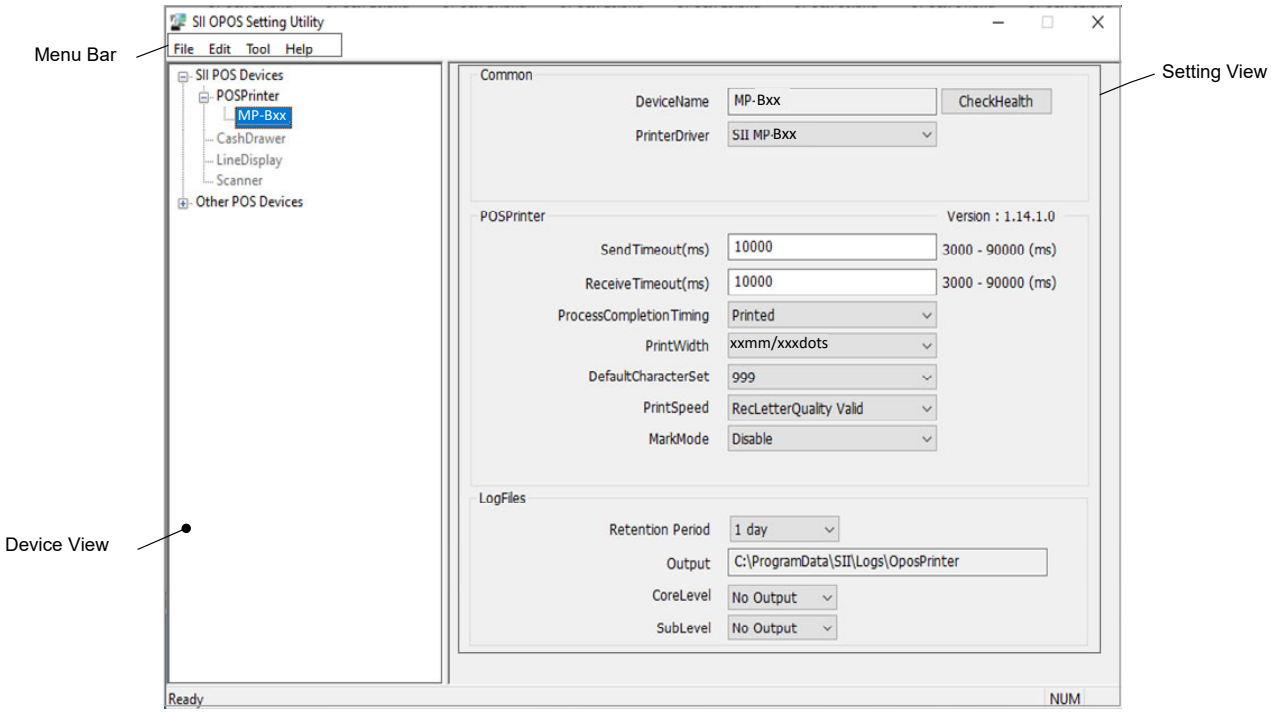
This chapter describes the configuration program provided by this software.

Caution

◆ 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 the device.

3.1.1 Menu Bar

Item		Description
File	Exit	Ends the configuration program.
Edit	Add Device…	Adds a new device.
	Add Logical Device Name…	Adds a new logical device name.
	Delete	Deletes the selected device.
Tool	CheckHealth	Executes an interactive test for the selected device.
Help	About Utility…	Displays the version information of the configuration program.

3.1.2 Device View

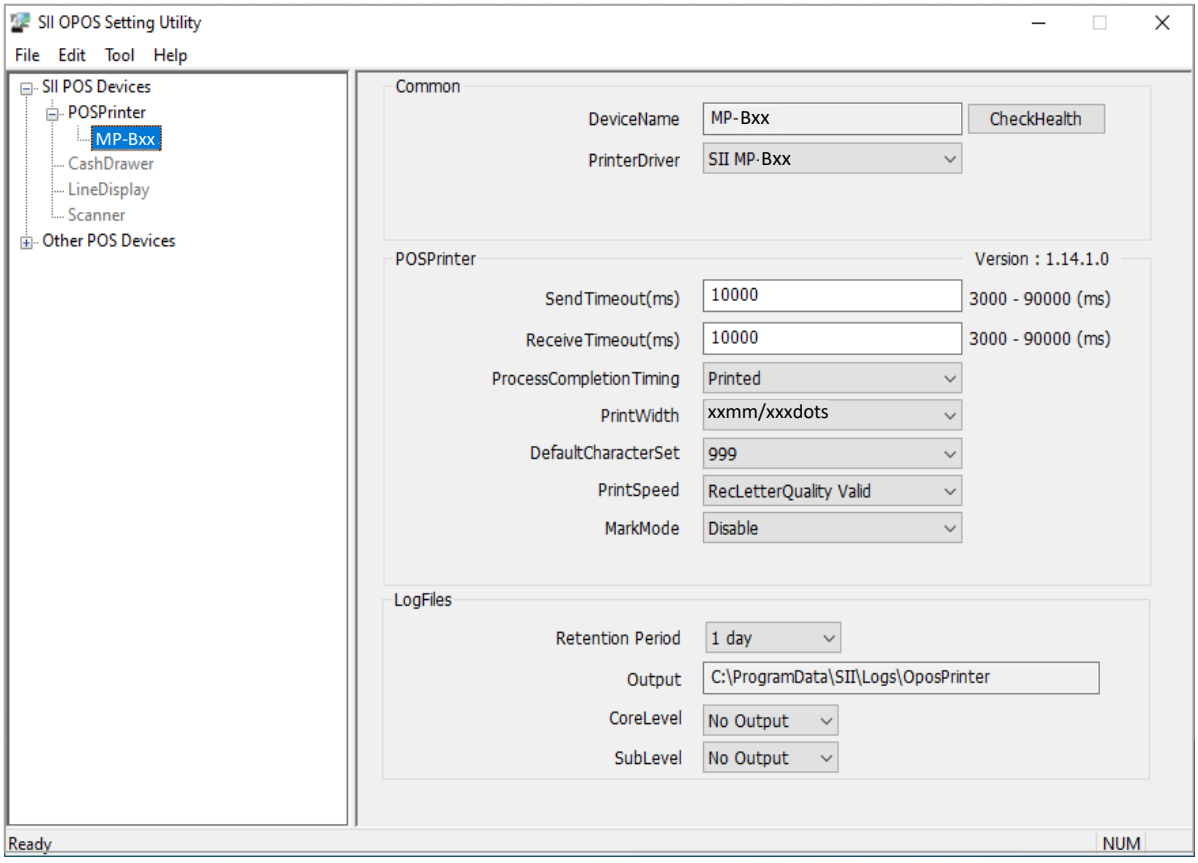
Name	Description
SII POS Devices	Displays SII devices. When a device name is selected in "Device View", the device can be deleted and the logical device name can be added.
Other POS Devices	Displays devices other than SII devices. Device settings cannot be changed or deleted.

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.



Reference

- The items supported by the printer are displayed.

Item	Description	Setting Content (" " : Default)
Common		
DeviceName	Device name	Device name of POSPrinter selected in "Device View".
CheckHealth	Executes an interactive test for the selected device.	-
PrinterDriver	Printer driver used for communication with the printer	-
POSPrinter		
SendTimeout(ms)	Send timeout value in communication with the printer (milliseconds)	3000 to 90000 (10000)
ReceiveTimeout (ms)	Receive timeout value in communication with the printer (milliseconds)	3000 to 90000 (10000)
ProcessCompletion Timing	Timing of method completion	Data transmitted (Completion of data transmission) Printed (Completion of printing)
PrintWidth	Print width / Number of effective dots	MP-B21L 21mm/168dots 22mm/176dots 24mm/192dots 26mm/208dots 28mm/224dots 30mm/240dots 32mm/256dots 34mm/272dots 36mm/288dots 38mm/304dots 40mm/320dots 42mm/336dots 44mm/352dots 45mm/360dots 46mm/368dots 48mm/384dots MP-B30L 45mm/360dots 46mm/368dots 48mm/384dots 50mm/400dots 52mm/416dots 54mm/432dots 56mm/448dots 58mm/464dots 60mm/480dots 62mm/496dots 64mm/512dots 66mm/528dots 68mm/544dots 70mm/560dots 72mm/576dots
PaperSize	Paper width / Number of effective dots	MP-B30 80mm/576dots

Item		Description	Setting Content (" " : Default)
	DefalutCharacterSet	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
	PrintQuality	Print quality of the printer It is decided by **RecLetterQuality** when selecting RecLetterQuality Valid.	RecLetterQuality Valid: Enabled **RecLetterQuality** Standard[*3] Quality 1[*3] Quality 2[*3]
	MarkMode	Mark detection mode Selects whether to enable or disable mark detection.	Disable Enable
LogFiles			
	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\OposPrinter 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]	
	CoreLevel	Log output level of POS Printer service object	No Output: No logs are output. Error: Error logs at execution are output. Trace: Detailed operation history and dump logs are output.
	SubLevel	Log output level of dedicated POS Printer service object for MP-Bxx	No Output: No logs are output. Error: Error logs at execution are output. Trace: Detailed operation history and dump logs are output.

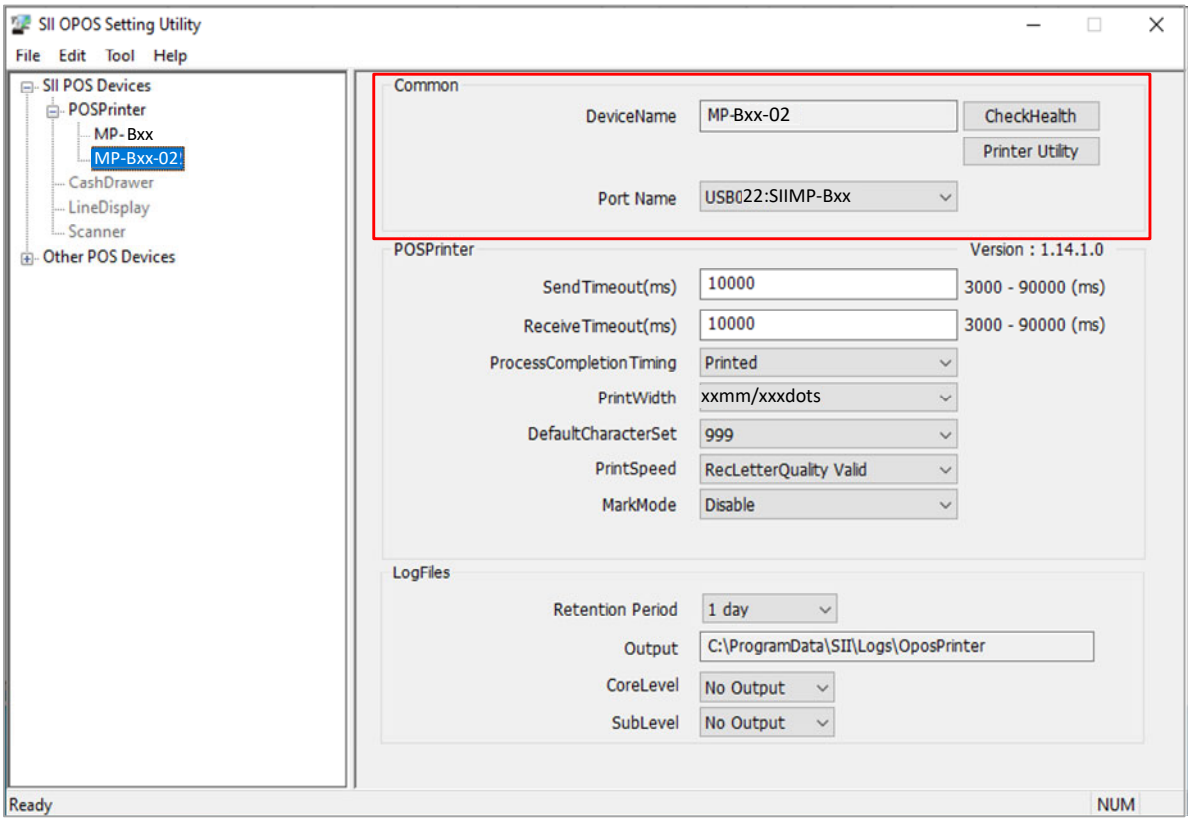
*1: Default for Japanese

*2: Default for English
*3: See "User's Guide" for details of the setting.
*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		
DeviceName	Device name	Device name of POSPrinter selected in "Device View".
CheckHealth	Executes an interactive test for the selected device.	-
Printer Utility	Starts up Printer Setting Utility that can set the memory switches.	-
PortName	Port name for the selected device.	-

3.2 Functions

The functions of the configuration program are described.

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.

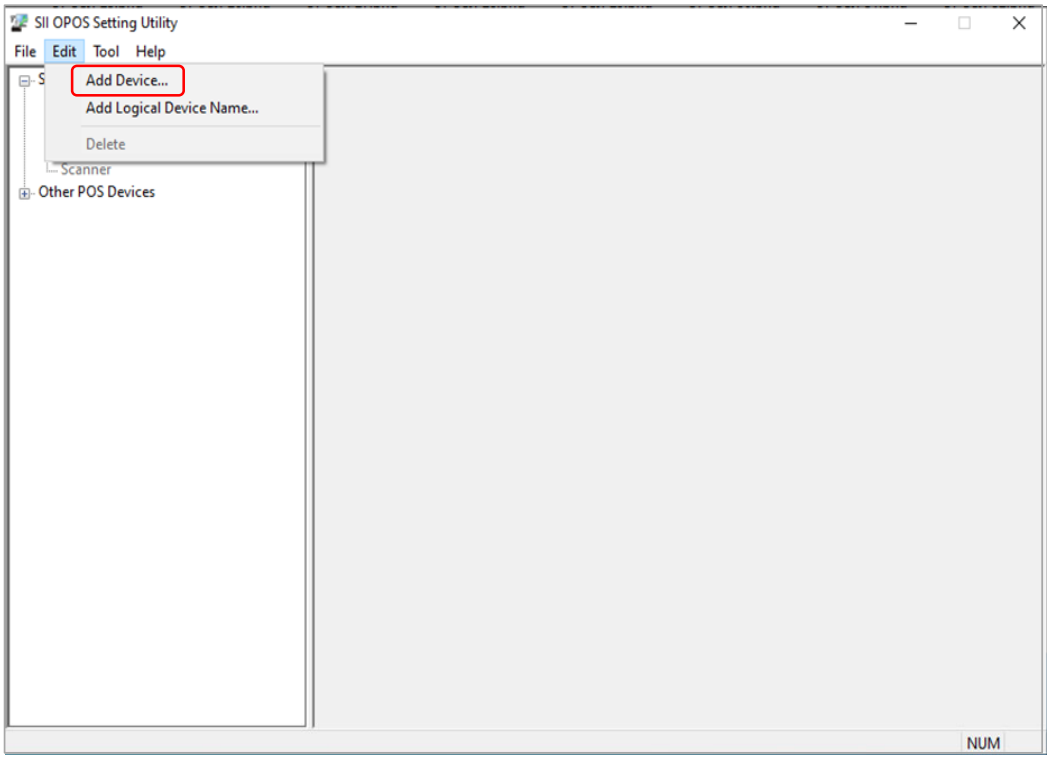
Maximum 8 printers can be added.

Caution

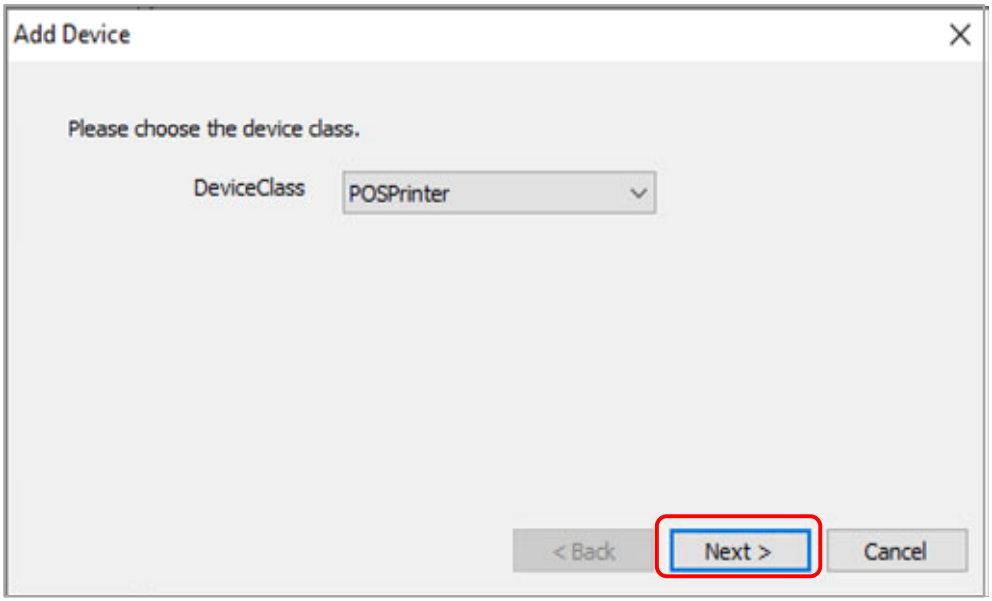
- When selecting USB for the interface, connect to the port in advance before starting up the configuration program.
- When selecting Bluetooth for the interface, pair the printer in advance before starting up the configuration program.
- When selecting LAN/WLAN for the interface, create the LAN port in advance before starting up the configuration program.
- When selecting COM for the interface, connect to the 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

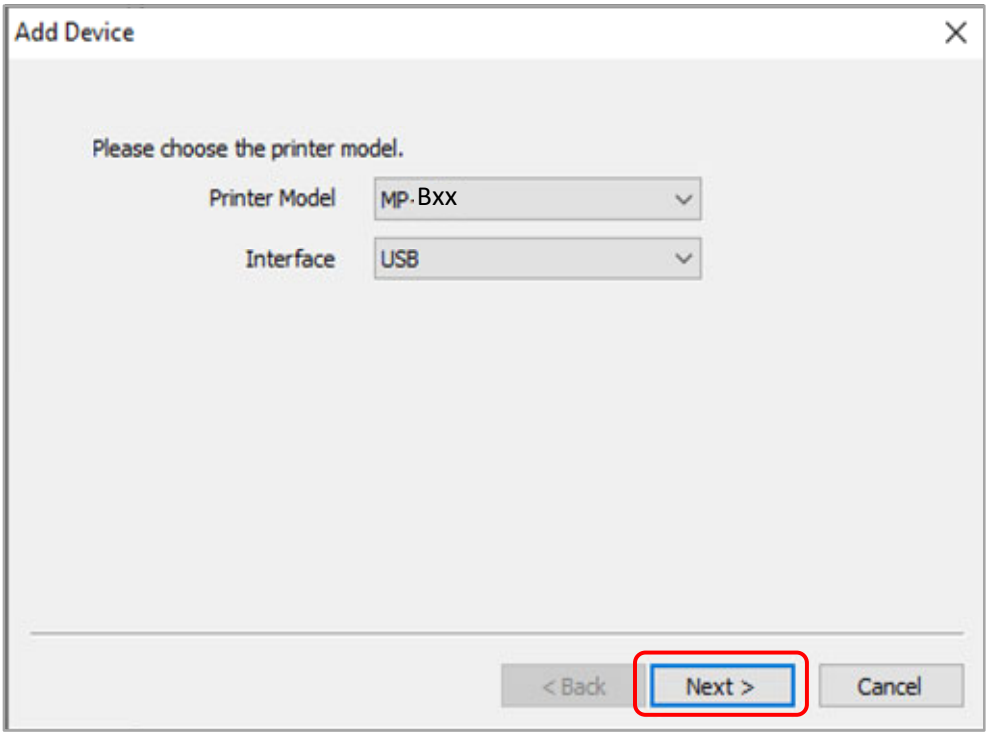
1) When the configuration program starts, the following window is displayed. Select [Edit] – [Add Device…] from "Menu Bar".

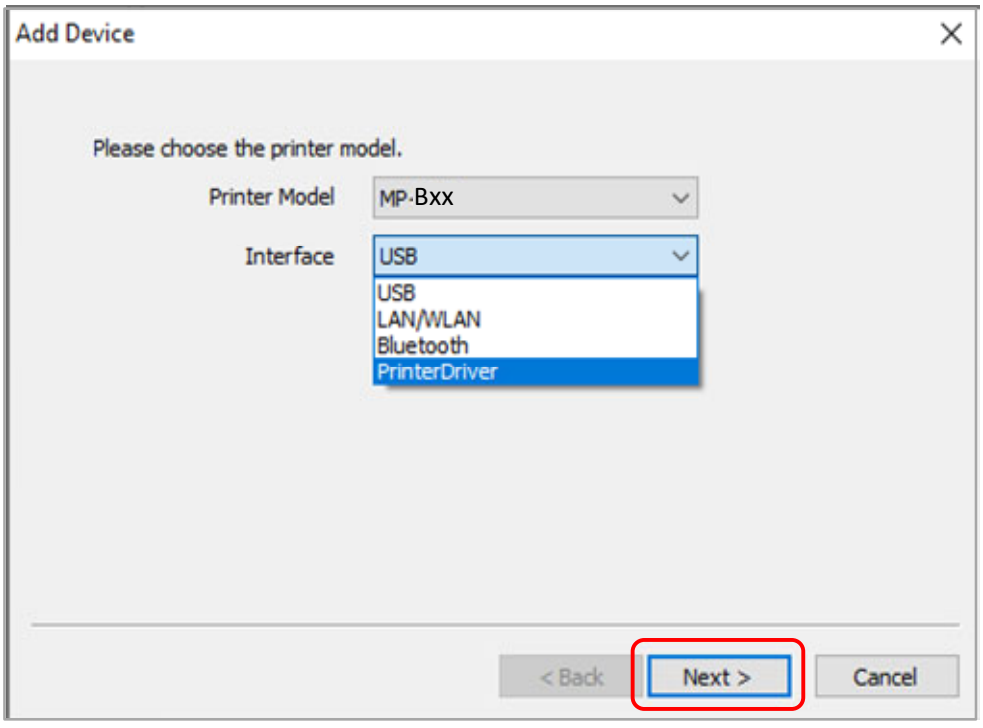


(2) Select "POSPrinter" for [DeviceClass], then click the [Next >] button.



(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.



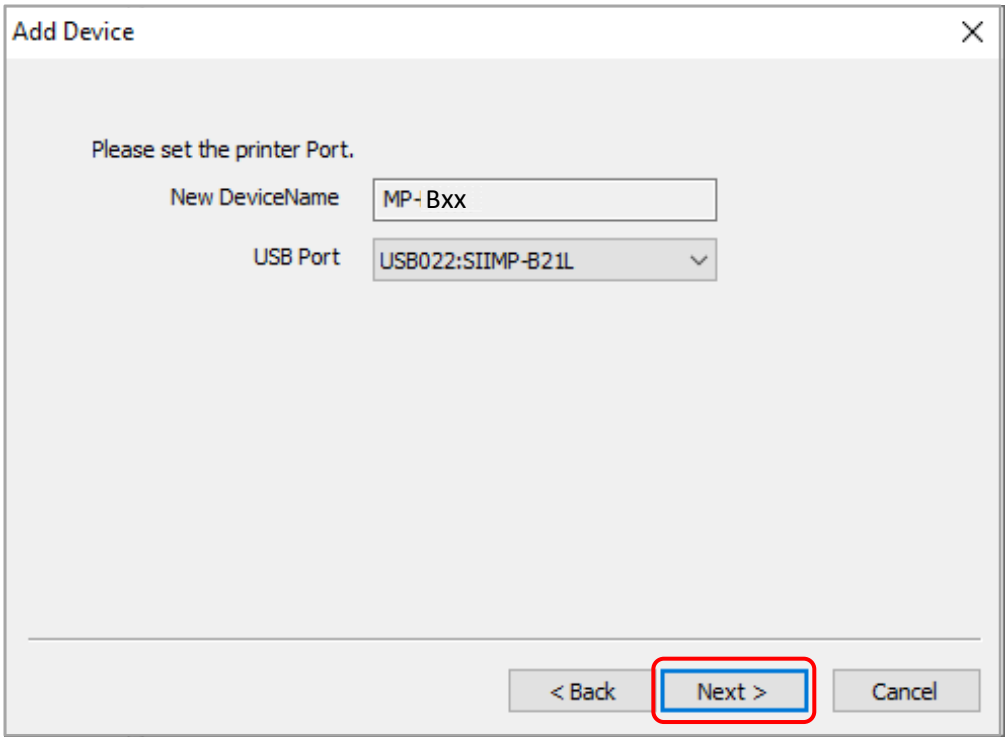


The following procedures vary depending on the selection of [Interface].

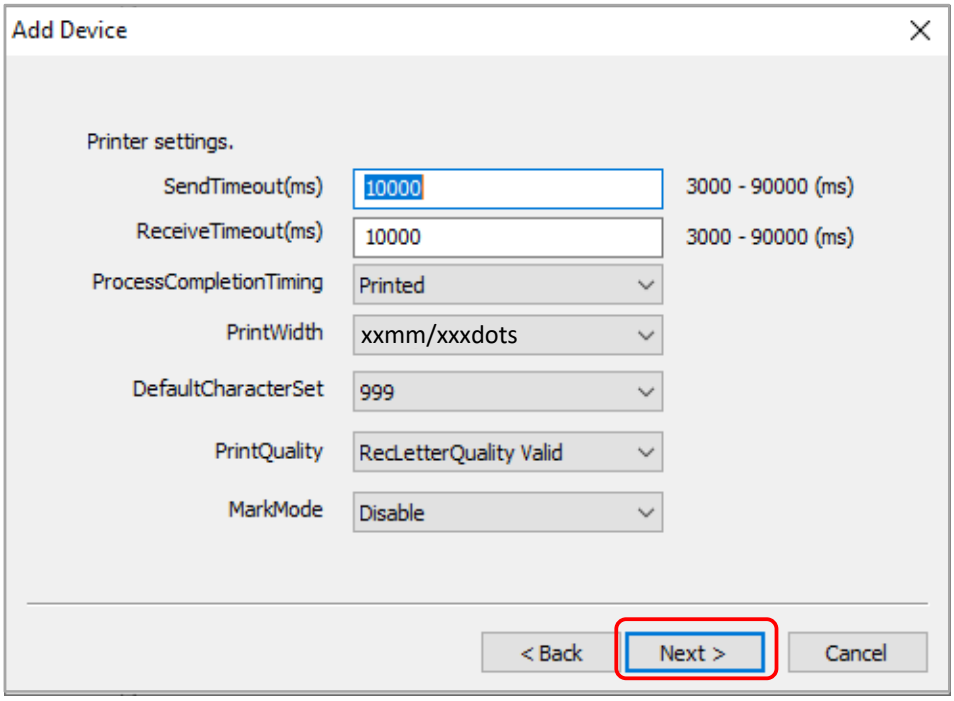
- Selecting USB: See "**When selecting the USB** ".
- Selecting Bluetooth: See "**When selecting the Bluetooth**".
- Selecting LAN/WLAN: See "**When selecting the LAN/WLAN**".
- Selecting COM: See "**When selecting the COM**".
- Selecting printer driver: See "**When selecting the printer driver**".

When selecting the USB

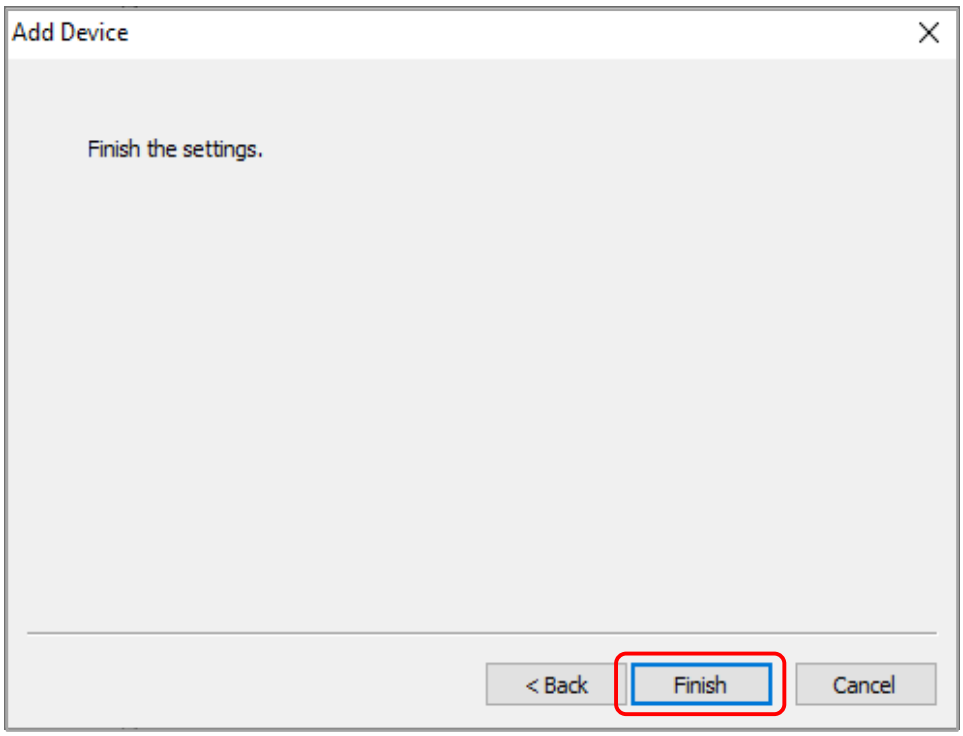
4) Select the port to be used from [USB Port], and then click the [Next >] button. [New DeviceName] is set automatically.



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



6) Click the [Finish] button.

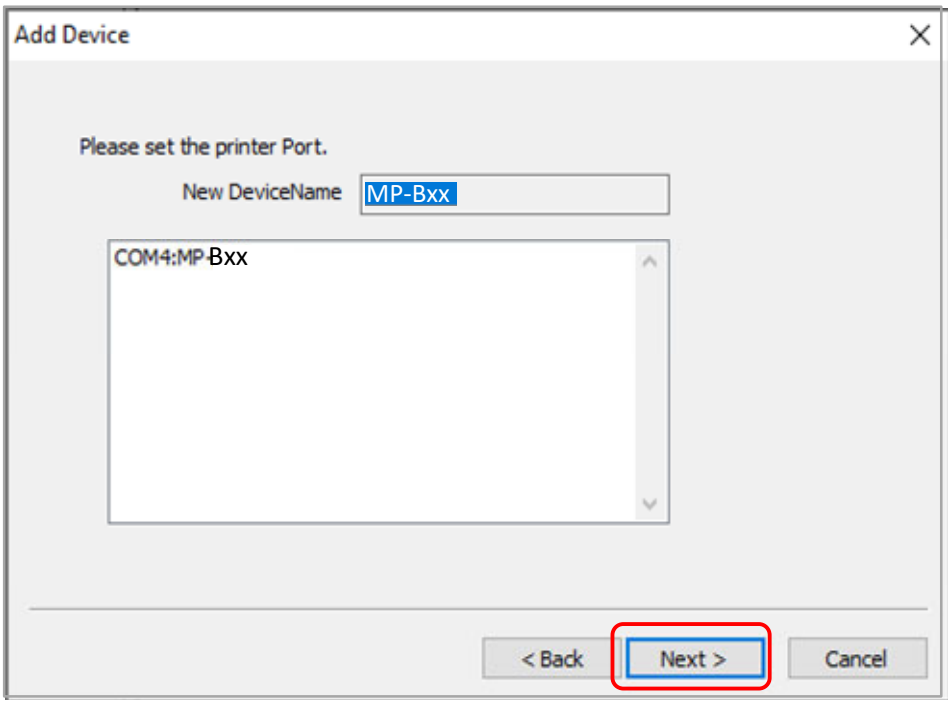


When selecting the Bluetooth

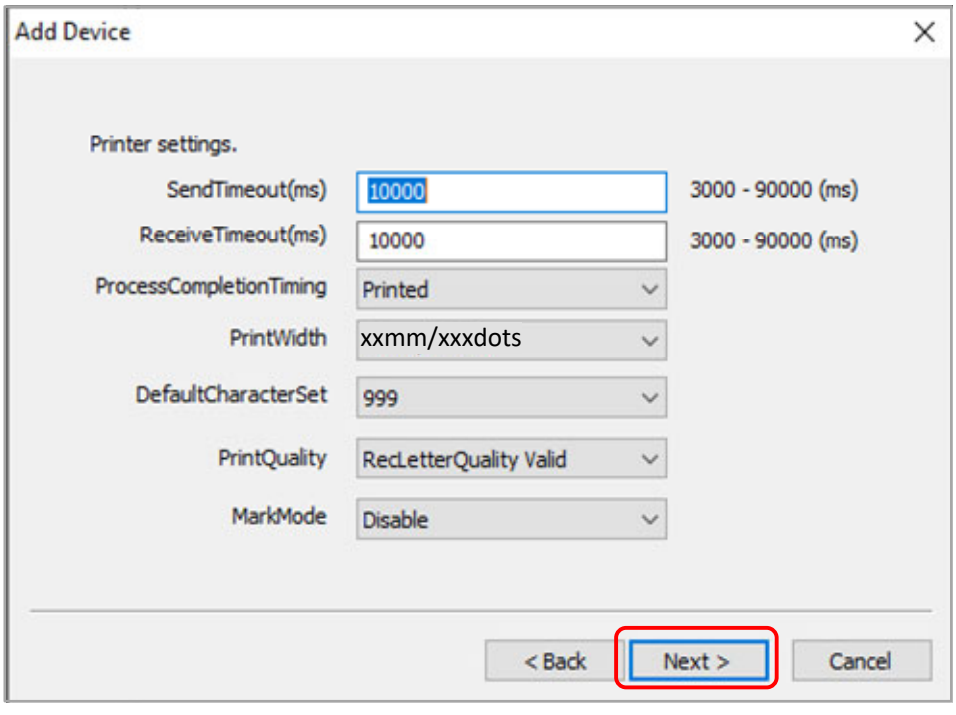
Caution

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

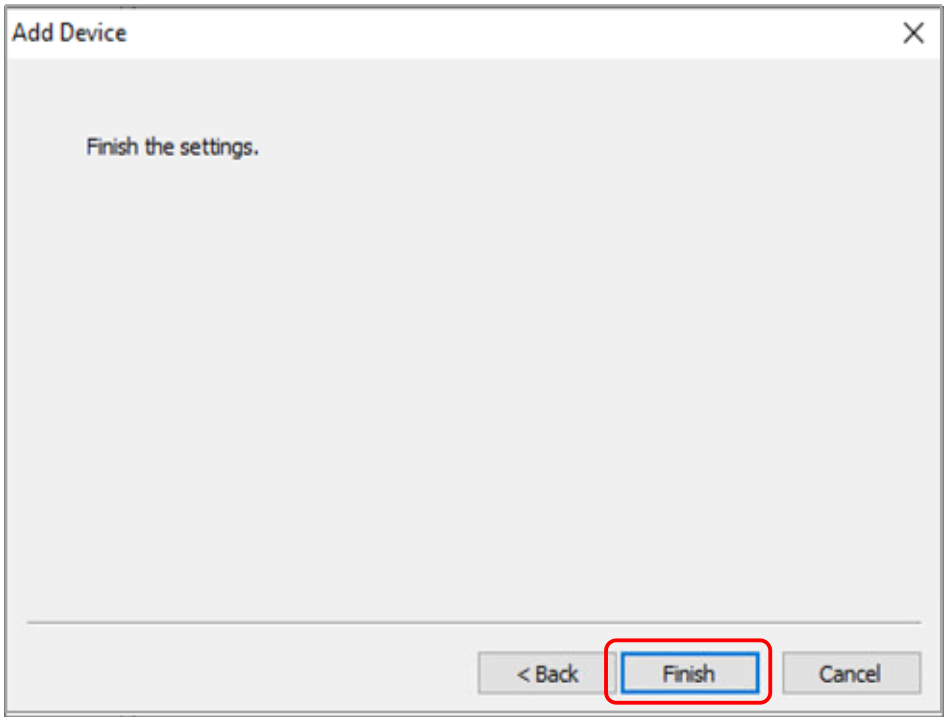
4) Printers that have been paired once will be shown, select a printer and click the [Next >] button. The [New DeviceName] is set automatically.



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

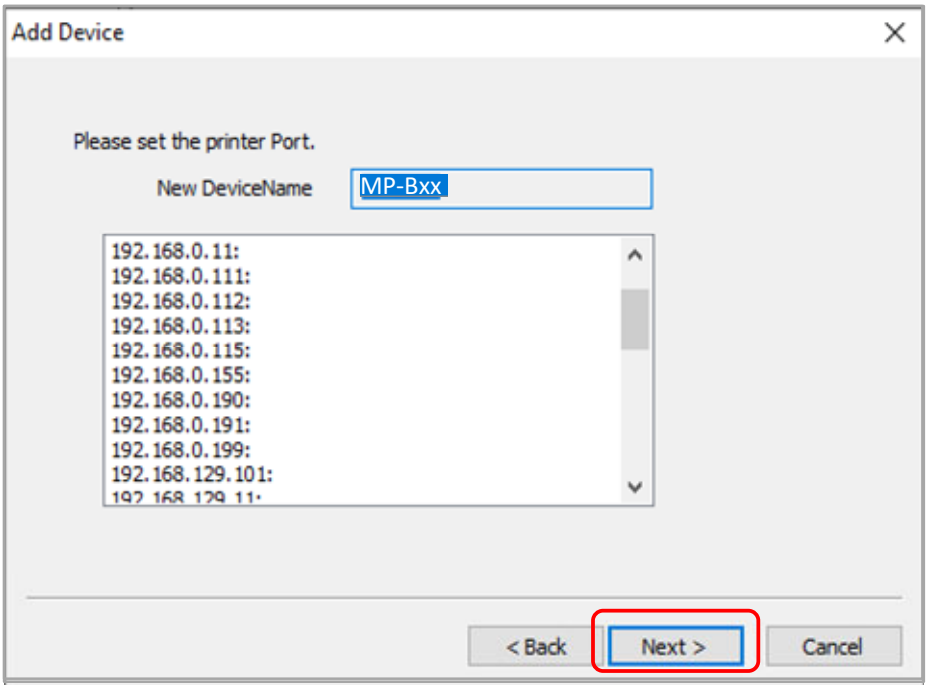


6) Click the [Finish] button.

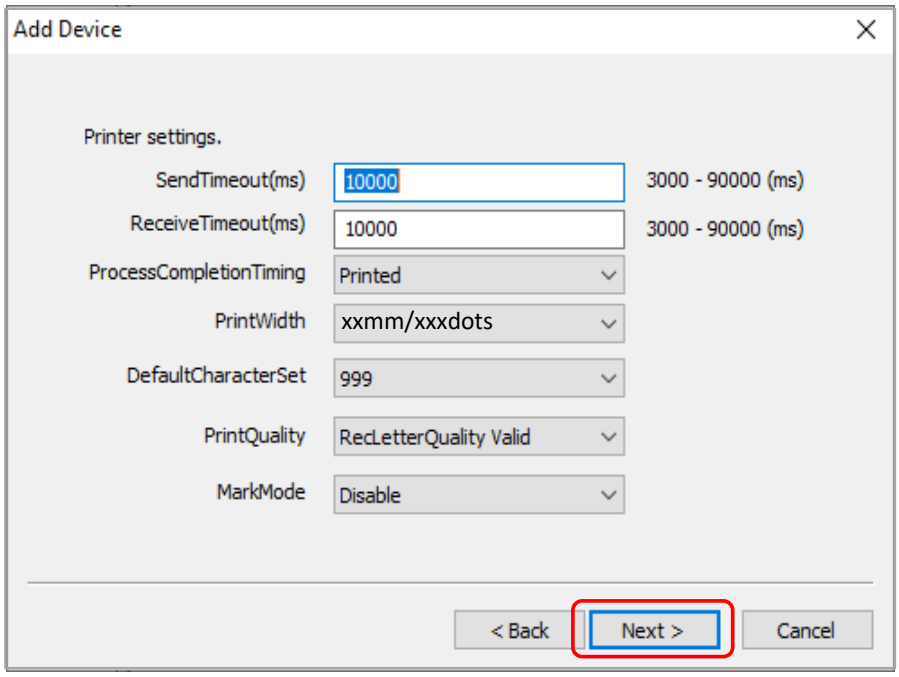


When selecting the LAN/WLAN

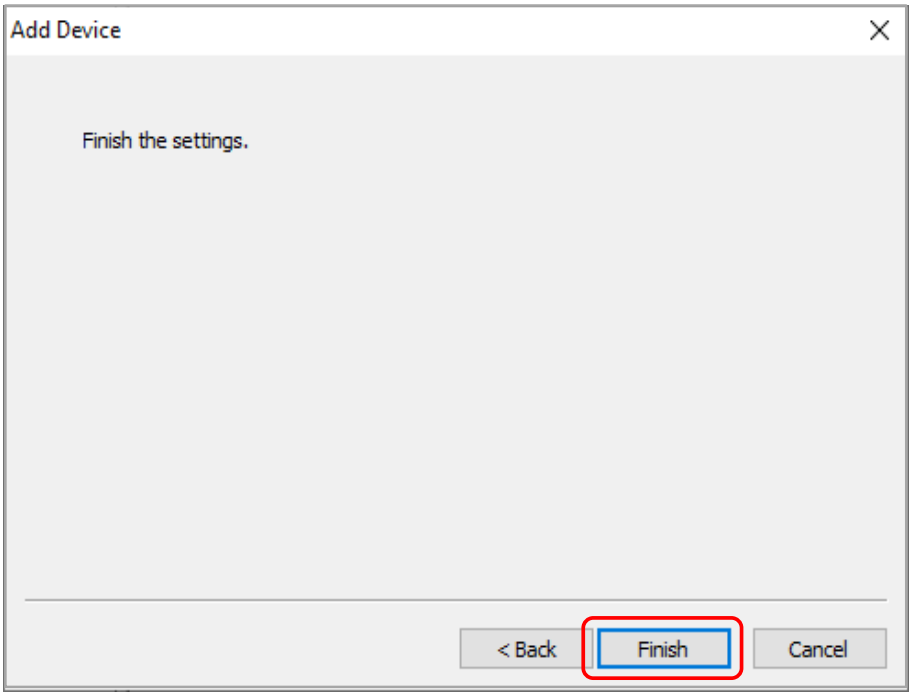
4) The printer will be shown, select the printer and click the [Next >] button. The [New DeviceName] is set automatically.



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

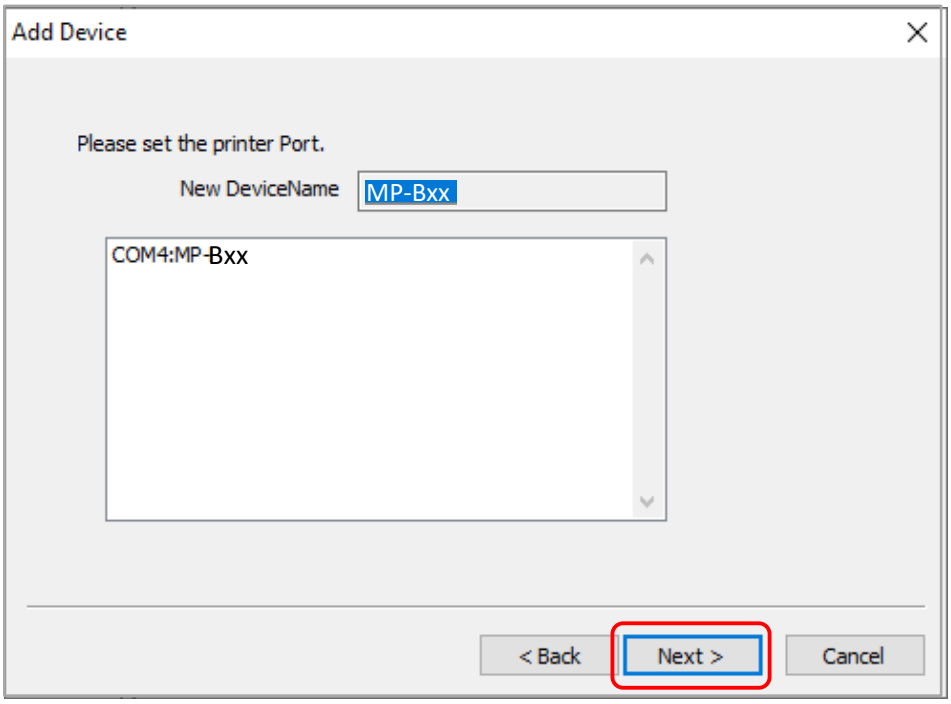


6) Click the [Finish] button.

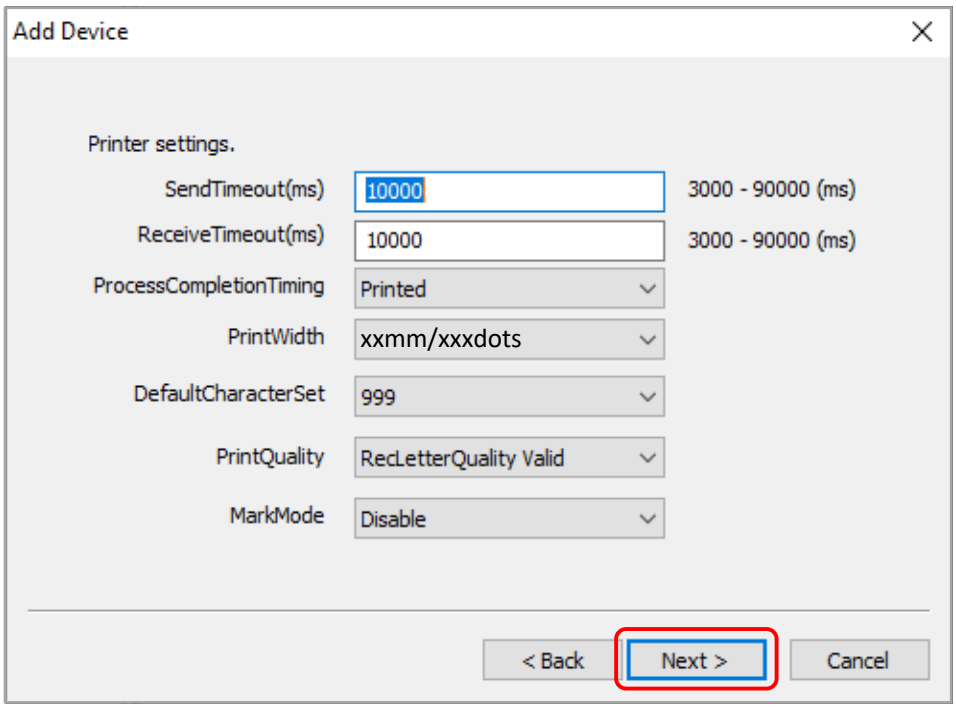


When selecting the COM

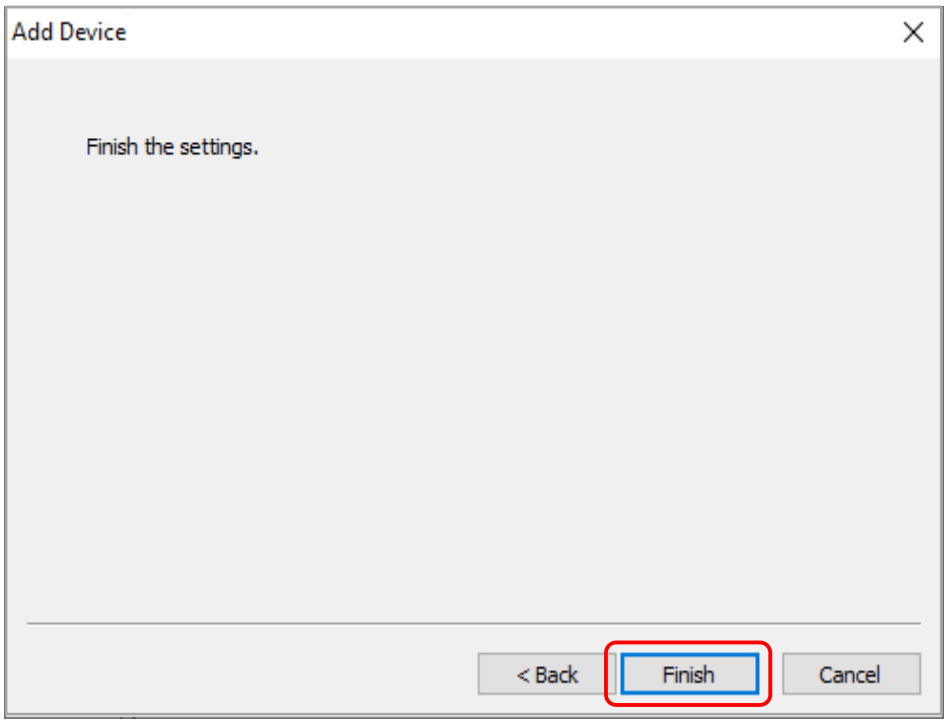
4) When the serial port is displayed, select the port to be used, and then click the [Next >] button. [New DeviceName] is set automatically.



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

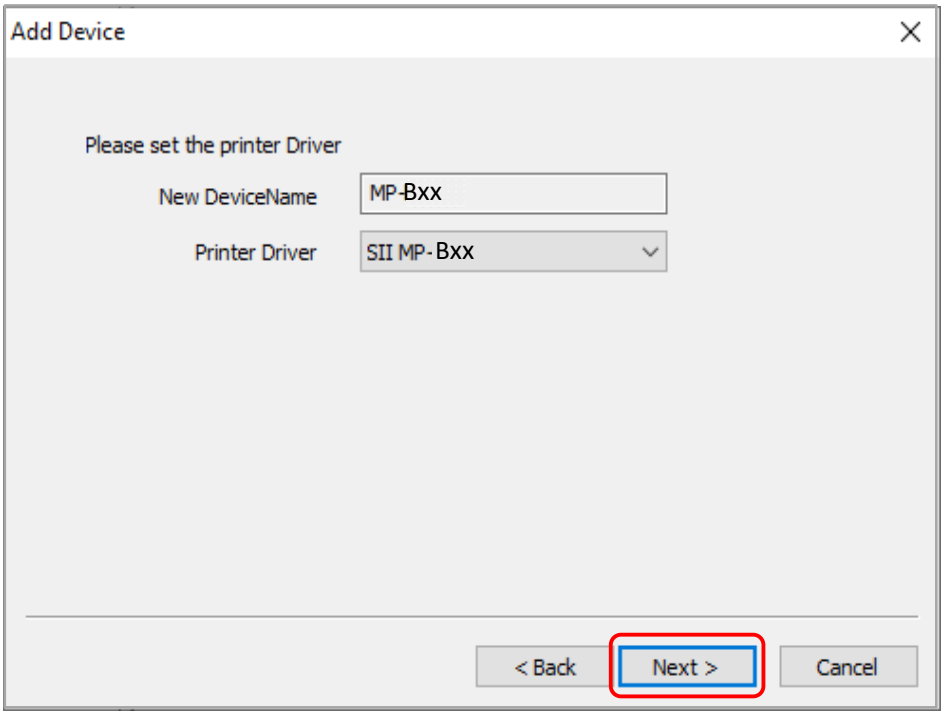


6) Click the [Finish] button.

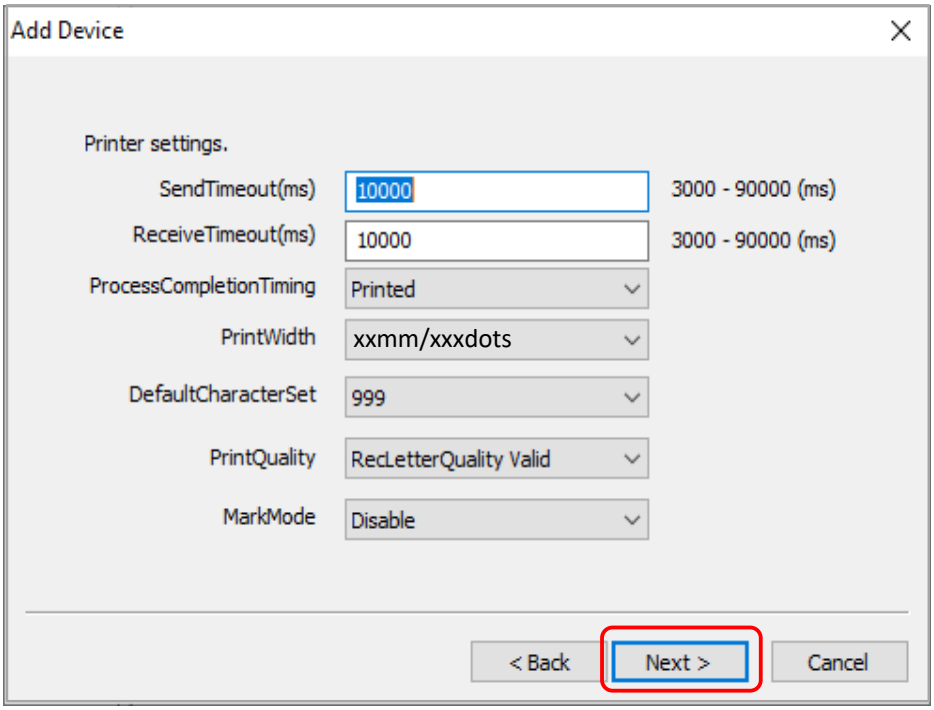


When selecting the printer driver

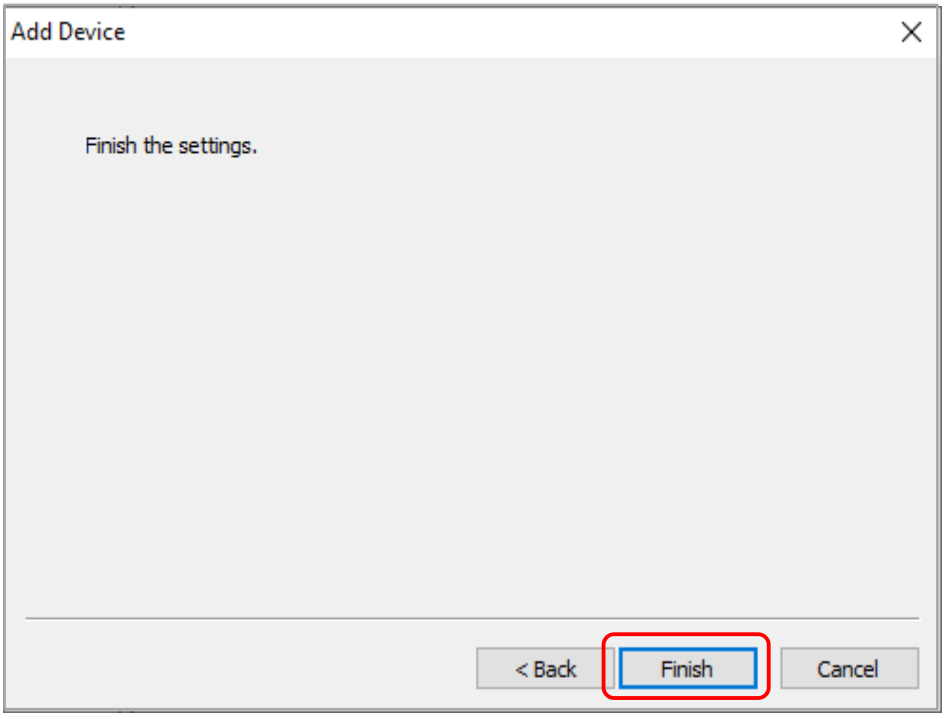
4) Select the printer driver from [Printer Driver], and then click the [Next >] button. [New DeviceName] is set automatically.



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



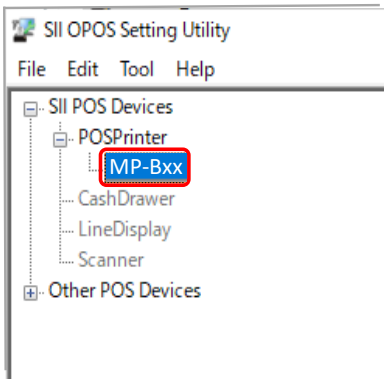
6) Click the [Finish] button.



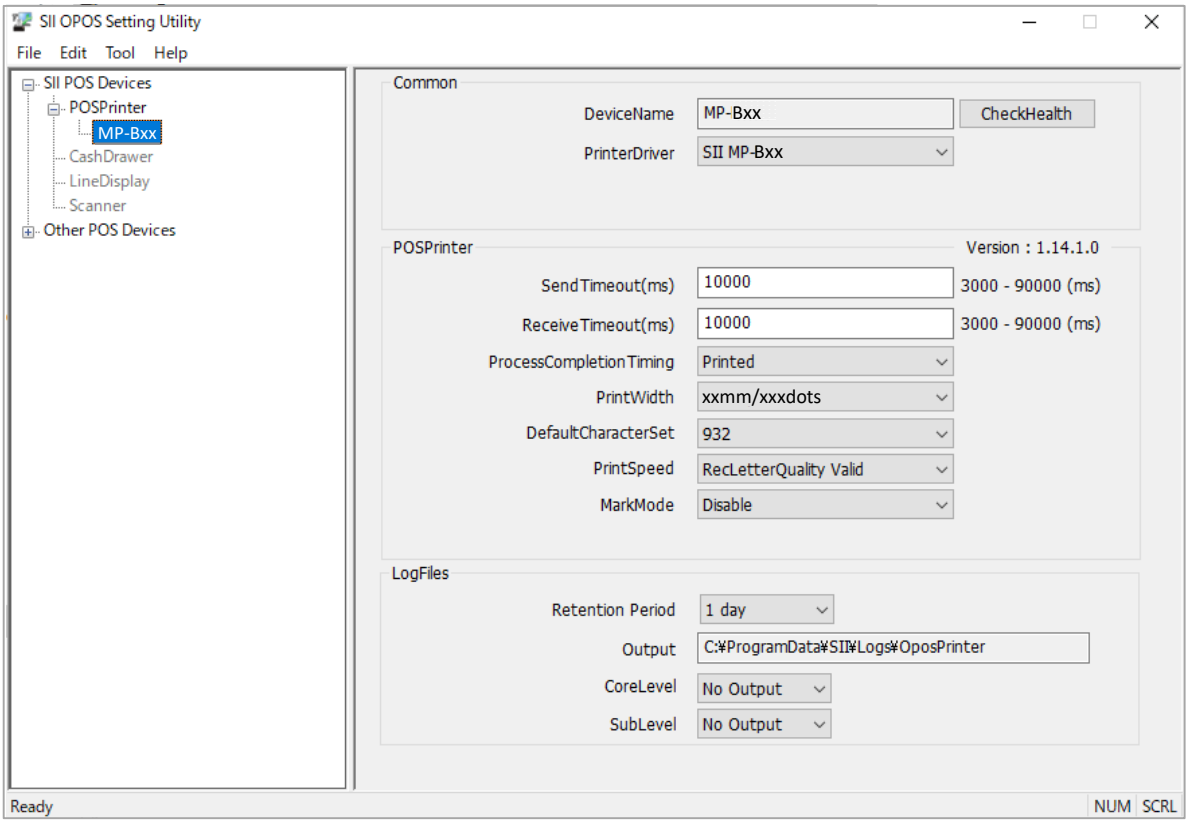
3.2.2 Changing Device Settings

The procedure for changing settings of the added device is described.

1) Select the device name to be changed from "Device View".



2) "Setting View" is displayed in editable state. Change the contents.



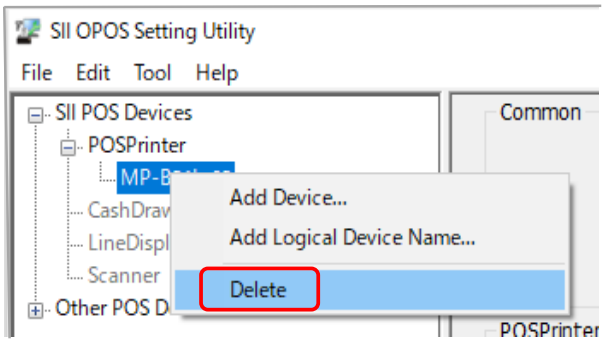
Reference

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

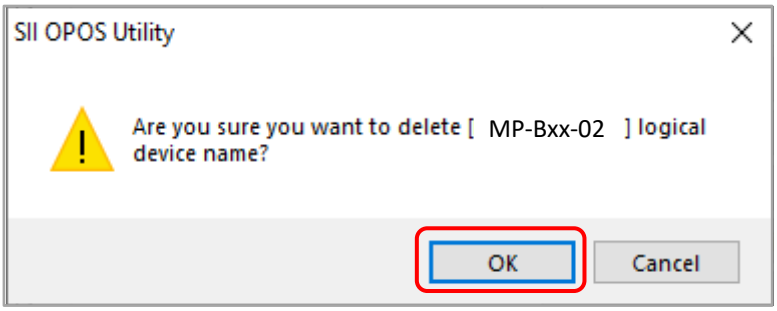
3.2.3 Deletion of Device

The procedure for deleting the added device is described below.

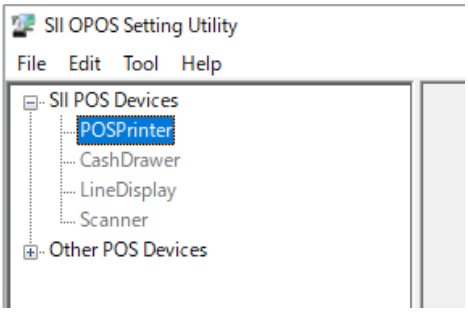
1) Select the device to be deleted from "Device View", and select [Delete] from the right-click menu.



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



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



Caution

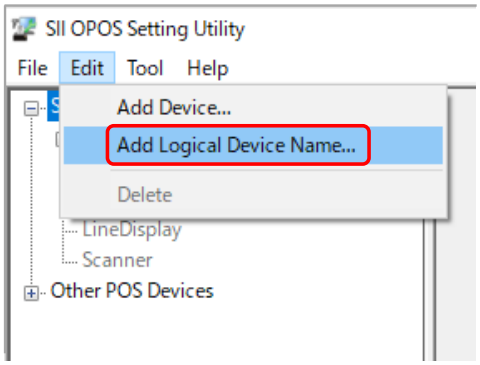
◆ When the POSPrinter device is selected, the selected POSPrinter device and each logical device name are all deleted.

3.2.4 Adding and Deleting Logical Device Name

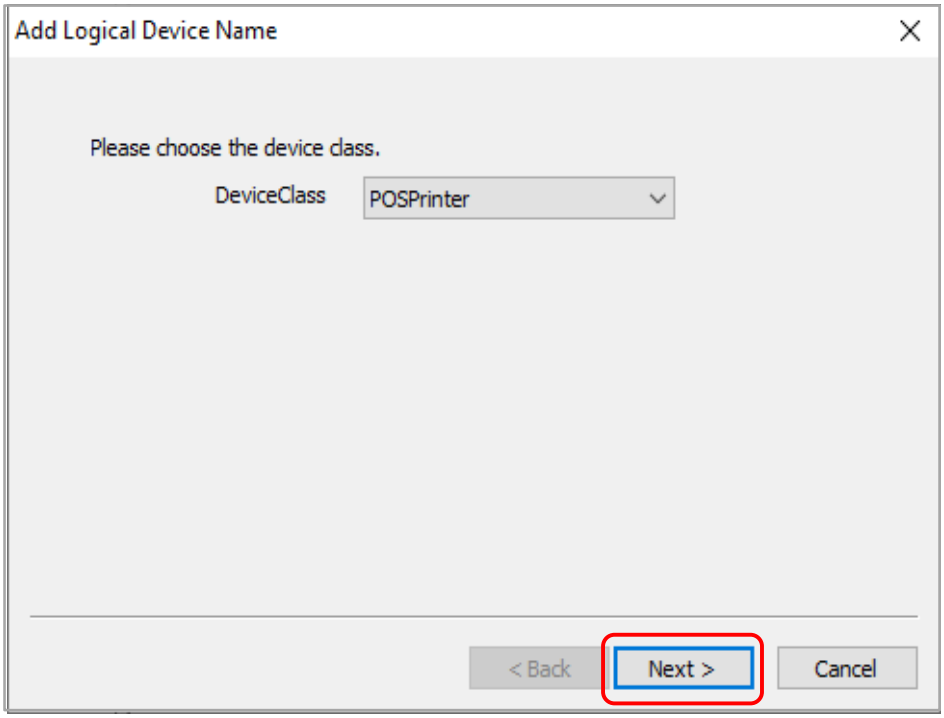
The procedures for adding and deleting the logical device name are described below.

(1) Addition of logical device name

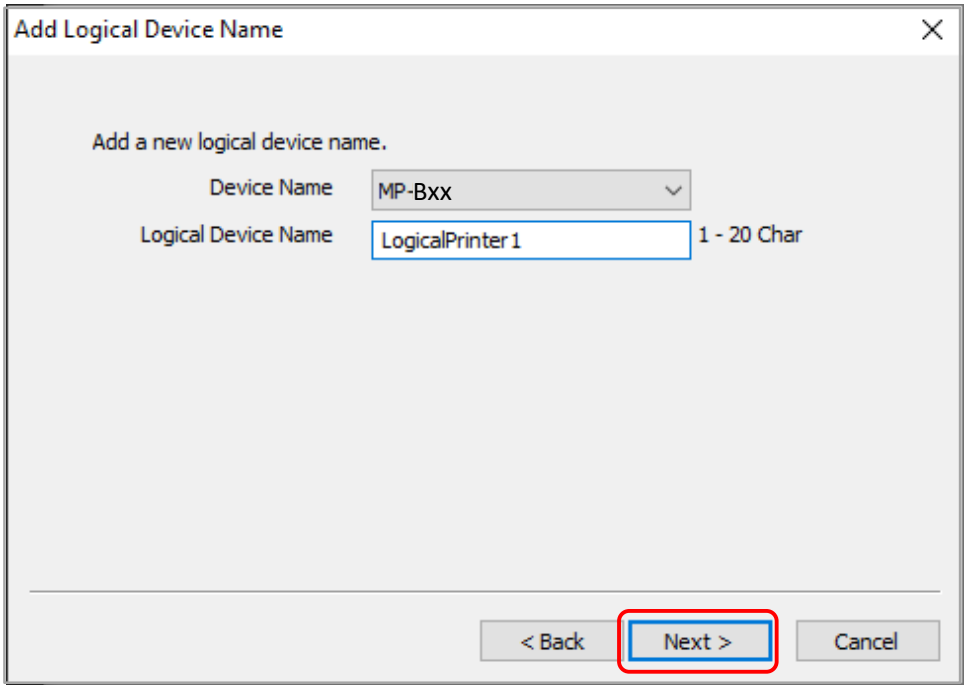
1) Start the configuration program, and select [Edit] – [Add Logical Device Name…] from "Menu Bar".



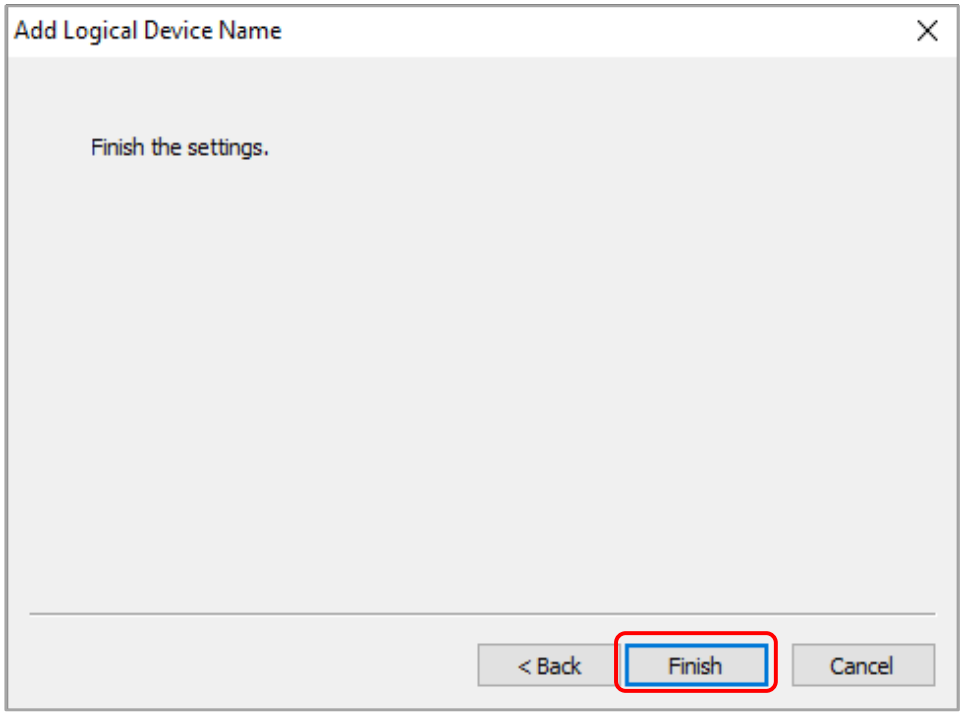
2) Select the target device in [DeviceClass], and click the [Next >] button.



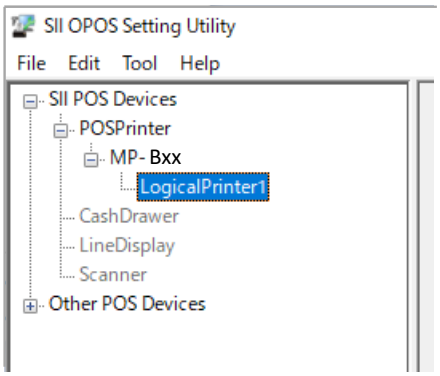
3) Select the target device name in [Device Name], and enter the [Logical Device Name]. Click the [Next >] button.



4) Click the [Finish] button.

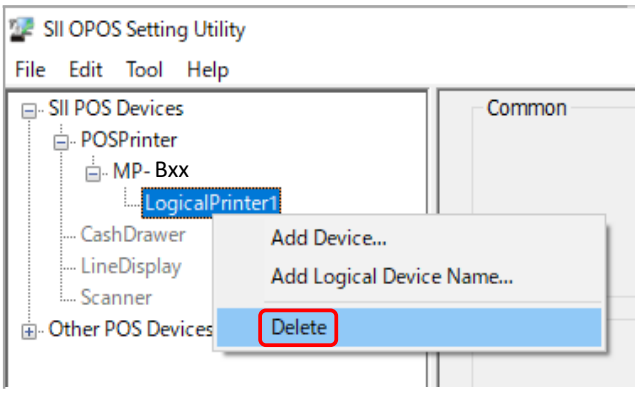


5) Confirm the contents of "Device View".

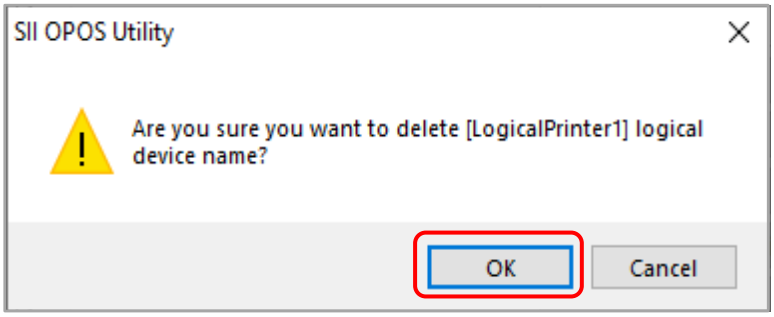


(2) Deletion of Logical Device Name

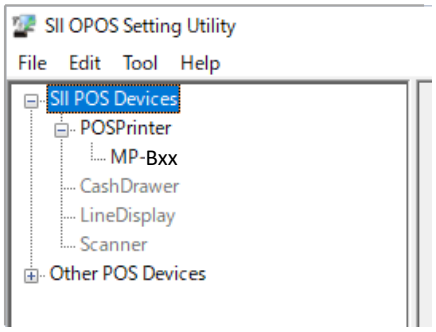
1) Select the logical device name to be deleted from "Device View", and select [Delete] from the right-click menu.



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



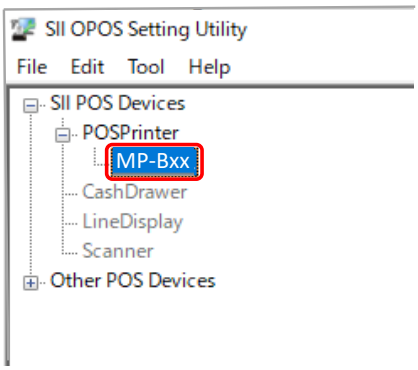
3) Confirm that the selected logical device name has been deleted from the "Device View".



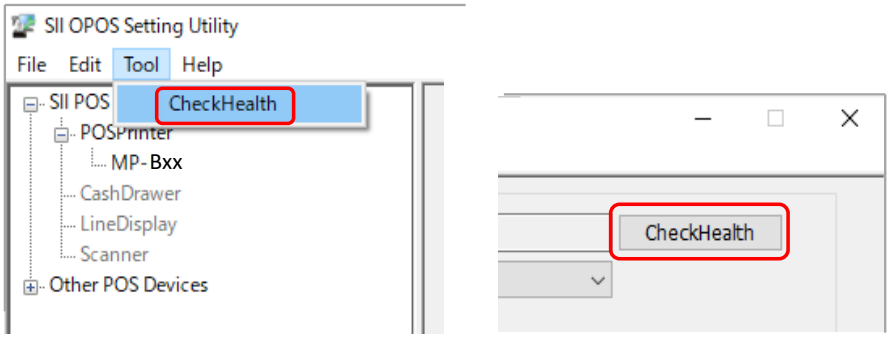
3.2.5 Device Interactive Test

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

1) Select the device name or the logical device name for which the interactive test is performed from "Device View".



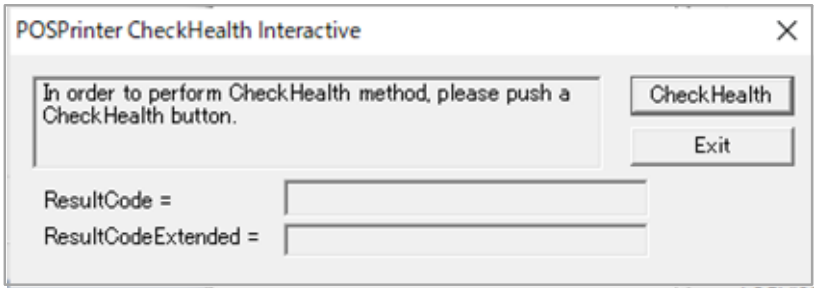
2) Select [Tool] menu - [CheckHealth] from "Menu bar", or click [CheckHealth] button in "Setting View".



3) The preparation for the interactive test is started.

[When the preparation for the interactive test succeeded]

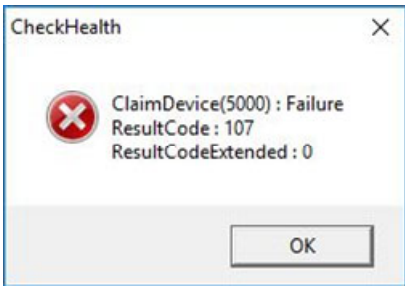
4) The dialogue to perform the interactive test is displayed.



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 error message is displayed.



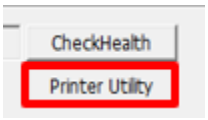
See **ResultCode** for ResultCode.
Click the [OK] button after confirming ResultCode.

3.2.6 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 [AII 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 OPOS Control

This chapter describes the use procedure of the OPOS control.

Use the software in the following procedure for the application.

(1) **Open**: Link the Control Object to the Service Project.

(2) **ClaimDevice**: Call to gain exclusive access to the device.

(3) **DeviceEnabled**: Set to TRUE to make the device operational.

(4) Use the device (Properties, methods, and events).

(5) **DeviceEnabled**: Set to FALSE to disable the device.

(6) **ReleaseDevice**: Release exclusive access to the device.

(7) **Close**: Release the Service Object from the Control Object.

4.1 POS Printer Control

POS Printer Control supports only "Receipt".
The interfaces for the methods and properties related to "Journal" and "Slip" are provided but the operation is not supported.

POS Printer Control supports synchronous/asynchronous output following the general output model.
In addition, the printer controlled by the POS Printer Control is the device used exclusively.

Chapter 5 POS Printer Control Interface Specifications

This chapter describes the POS Printer Control properties, methods and events implemented in this software.

5.1 Summary

5.1.1 Common Properties

Property Name	Type	Access	Availability Condition	Default
BinaryConversion	Long	R/W	Open	OPOS_BC_NONE(0)
CapCompareFirmwareVersion	Boolean	R	Open	FALSE
CapPowerReporting	Long	R	Open	OPOS_PR_STANDARD(1)
CapStatisticsReporting	Boolean	R	Open	TRUE
CapUpdateFirmware	Boolean	R	Open	FALSE
CapUpdateStatistics	Boolean	R	Open	TRUE
CheckHealthText	String	R	Open	""
Claimed	Boolean	R	Open	FALSE
ControlObjectDescription	String	R	-	"OPOS POS Printer Control 1.14.xxx [Public, by CRM/RCS-Dayton]"
ControlObjectVersion	Long	R	-	1014xxx
DeviceDescription	String	R	Open	"SII MP-Bxx POS Printer"
DeviceEnabled	Boolean	R/W	Open & Claim	FALSE
DeviceName	String	R	Open	"MP-Bxx POS Printer"
FreezeEvents	Boolean	R/W	Open	FALSE
OpenResult	Long	R	-	OPOS_SUCCESS(0)
OutputID	Long	R	Open	0
PowerNotify	Long	R/W	Open	OPOS_PN_DISABLED(0)
PowerState	Long	R	Open	OPOS_PS_UNKNOWN(2000)
ResultCode	Long	R	-	OPOS_SUCCESS(0)
ResultCodeExtended	Long	R	Open	0
ServiceObjectDescription	String	R	Open	"SII MP-Bxx POS Printer Service Object, Copyright(C) 20xx Seiko Instruments Inc."
ServiceObjectVersion	Long	R	Open	1014xxx
State	Long	R	-	OPOS_S_CLOSED(1)

5.1.2 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	Boolean	R/W	Open	FALSE
CapCharacterSet	Long	R	Open	PTR_CCS_KANJI(11)
CapCoverSensor	Boolean	R	Open	TRUE
CapMapCharacterSet	Boolean	R	Open	FALSE
CapRec2Color	Boolean	R	Open	FALSE
CapRecBarCode	Boolean	R	Open	TRUE
CapRecBitmap	Boolean	R	Open	TRUE
CapRecBold	Boolean	R	Open	TRUE
CapRecCartridgeSensor	Long	R	Open	0
CapRecColor	Long	R	Open	PTR_COLOR_PRIMARY (0x00000001)
CapRecDhigh	Boolean	R	Open	TRUE
CapRecDwide	Boolean	R	Open	TRUE
CapRecDwideDhigh	Boolean	R	Open	TRUE
CapRecEmptySensor	Boolean	R	Open	TRUE
CapRecItalic	Boolean	R	Open	FALSE
CapRecLeft90	Boolean	R	Open	TRUE
CapRecMarkFeed	Long	R	Open	0[*1]
CapRecNearEndSensor	Boolean	R	Open	FALSE
CapRecPageMode	Boolean	R	Open	TRUE
CapRecPapercut	Boolean	R	Open	FALSE
CapRecPresent	Boolean	R	Open	TRUE
CapRecRight90	Boolean	R	Open	TRUE
CapRecRotate180	Boolean	R	Open	TRUE
CapRecRuledLine	Long	R	Open	PTR_RL_HORIZONTAL(1) + PTR_RL_VERTICAL(2)
CapRecStamp	Boolean	R	Open	FALSE
CapRecUnderline	Boolean	R	Open	TRUE
CapTransaction	Boolean	R	Open	TRUE
CartridgeNotify	Long	R/W	Open	PTR_CN_DISABLED (0)
CharacterSet	Long	R/W	Open, Claim, & Enable	999[*1]
CharacterSetList	String	R	Open	"437,737,850,852,855,857, 858,860,863,865,866,932,999, 1250,1251,1252,1253,1254"
CoverOpen	Boolean	R	Open, Claim, & Enable	Depends on printer status
ErrorLevel	Long	R	Open	PTR_EL_NONE(1)
ErrorStation	Long	R	Open	0
ErrorString	String	R	Open	""
FlagWhenIdle	Boolean	R/W	Open	FALSE

Property Name	Type	Access	Availability Condition	Default
FontTypefaceList	String	R	Open	""
MapCharacterSet	Boolean	R/W	Open	FALSE
MapMode	Long	R/W	Open	PTR_MM_DOTS(1)
PageModeArea	String	R	Open	""
PageModeDescriptor	Long	R	Open	0
PageModeHorizontalPosition	Long	R/W	Open	0
PageModePrintArea	String	R/W	Open	""
PageModePrintDirection	Long	R/W	Open	0
PageModeStation	Long	R/W	Open	0
PageModeVerticalPosition	Long	R/W	Open	0
RecBarCodeRotationList	String	R	Open	"0,R90,L90,180"
RecBitmapRotationList	String	R	Open	"0,R90,L90,180"
RecCartridgeState	Long	R	Open, Claim, & Enable	PTR_CART_UNKNOWN (0x10000000)
RecCurrentCartridge	Long	R/W	Open, Claim, & Enable	PTR_COLOR_PRIMARY (0x00000001)
RecEmpty	Boolean	R	Open, Claim, & Enable	Depends on printer status
RecLetterQuality	Boolean	R/W	Open, Claim, & Enable	FALSE
RecLineChars	Long	R/W	Open, Claim, & Enable	MP-B21L 32 MP-B30L, MP-B30 48
RecLineCharsList	String	R	Open	MP-B21L "24,27,29,32,38,42,48" MP-B30L, MP-B30 "36,41,44,48,57,64,72"
RecLineHeight	Long	R/W	Open, Claim, & Enable	24
RecLineSpacing	Long	R/W	Open, Claim, & Enable	30
RecLinesToPaperCut	Long	Open, Claim, & Enable	Open, Claim, & Enable	2
RecLineWidth	Long	R	Open, Claim, & Enable	MP-B21L 384*1 MP-B30L 576*1 MP-B30 576
RecNearEnd	Boolean	R	Open, Claim, & Enable	FALSE
RecSidewaysMaxChars	Long	R	Open, Claim, & Enable	200
RecSidewaysMaxLines	Long	R	Open, Claim, & Enable	MP-B21L 13 MP-B30L, MP-B30 19
RotateSpecial	Long	R/W	Open	PTR_RP_NORMAL(1)

*1: Can be 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	Boolean	R	Open	FALSE
CapConcurrentJrnSlp	Boolean	R	Open	FALSE
CapConcurrentPageMode	Boolean	R	Open	FALSE
CapConcurrentRecSlp	Boolean	R	Open	FALSE
CapJrn2Color	Boolean	R	Open	FALSE
CapJrnBold	Boolean	R	Open	FALSE
CapJrnCartridgeSensor	Long	R	Open	0
CapJrnColor	Long	R	Open	0
CapJrnDhigh	Boolean	R	Open	FALSE
CapJrnDwide	Boolean	R	Open	FALSE
CapJrnDwideDhigh	Boolean	R	Open	FALSE
CapJrnEmptySensor	Boolean	R	Open	FALSE
CapJrnItalic	Boolean	R	Open	FALSE
CapJrnNearEndSensor	Boolean	R	Open	FALSE
CapJrnPresent	Boolean	R	Open	FALSE
CapJrnUnderline	Boolean	R	Open	FALSE
CapSlp2Color	Boolean	R	Open	FALSE
CapSlpBarCode	Boolean	R	Open	FALSE
CapSlpBitmap	Boolean	R	Open	FALSE
CapSlpBold	Boolean	R	Open	FALSE
CapSlpBothSidesPrint	Boolean	R	Open	FALSE
CapSlpCartridgeSensor	Long	R	Open	0
CapSlpColor	Long	R	Open	0
CapSlpDhigh	Boolean	R	Open	FALSE
CapSlpDwide	Boolean	R	Open	FALSE
CapSlpDwideDhigh	Boolean	R	Open	FALSE
CapSlpEmptySensor	Boolean	R	Open	FALSE
CapSlpFullslip	Boolean	R	Open	FALSE
CapSlpItalic	Boolean	R	Open	FALSE
CapSlpLeft90	Boolean	R	Open	FALSE
CapSlpNearEndSensor	Boolean	R	Open	FALSE
CapSlpPageMode	Boolean	R	Open	FALSE
CapSlpPresent	Boolean	R	Open	FALSE
CapSlpRight90	Boolean	R	Open	FALSE
CapSlpRotate180	Boolean	R	Open	FALSE
CapSlpRuledLine	Long	R	Open	0
CapSlpUnderline	Boolean	R	Open	FALSE
JrnCartridgeState	Long	R	Open, Claim, & Enable	PTR_CART_UNKNOWN (0x10000000)

Property Name	Type	Access	Availability Condition	Default
JrnCurrentCartridge	Long	R/W	Open, Claim, & Enable	PTR_COLOR_PRIMARY (0x00000001)
JrnEmpty	Boolean	R	Open, Claim, & Enable	FALSE
JrnLetterQuality	Boolean	R/W	Open, Claim, & Enable	FALSE
JrnLineChars	Long	R/W	Open, Claim, & Enable	0
JrnLineCharsList	String	R	Open	""
JrnLineHeight	Long	R/W	Open, Claim, & Enable	0
JrnLineSpacing	Long	R/W	Open, Claim, & Enable	0
JrnLineWidth	Long	R	Open, Claim, & Enable	0
JrnNearEnd	Boolean	R	Open, Claim, & Enable	FALSE
SlpBarCodeRotationList	String	R	Open	""
SlpBitmapRotationList	String	R	Open	""
SlpCartridgeState	Long	R	Open, Claim, & Enable	PTR_CART_UNKNOWN (0x10000000)
SlpCurrentCartridge	Long	R/W	Open, Claim, & Enable	PTR_COLOR_PRIMARY (0x00000001)
SlpEmpty	Boolean	R	Open, Claim, & Enable	FALSE
SlpLetterQuality	Boolean	R/W	Open, Claim, & Enable	FALSE
SlpLineChars	Long	R/W	Open, Claim, & Enable	0
SlpLineCharsList	String	R	Open	""
SlpLineHeight	Long	R/W	Open, Claim, & Enable	0
SlpLinesNearEndToEnd	Long	R	Open, Claim, & Enable	0
SlpLineSpacing	Long	R/W	Open, Claim, & Enable	0
SlpLineWidth	Long	R	Open, Claim, & Enable	0
SlpMaxLines	Long	R	Open, Claim, & Enable	0
SlpNearEnd	Boolean	R	Open, Claim, & Enable	FALSE
SlpPrintSide	Long	R	Open, Claim, & Enable	0
SlpSidewaysMaxChars	Long	R	Open, Claim, & Enable	0
SlpSidewaysMaxLines	Long	R	Open, Claim, & Enable	0

5.1.3 Common Methods

Method Name	Availability Condition
CheckHealth	Open, Claim, & Enable
ClaimDevice	Open
ClearOutput	Open, Claim, & Enable*1
Close	Open
CompareFirmwareVersion	Open, Claim, & Enable
DirectIO	Open, Claim, & Enable*1
Open	-
ReleaseDevice	Open & Claim
ResetStatistics	Open, Claim, & Enable
RetrieveStatistics	Open, Claim, & Enable
UpdateFirmware	Open, Claim, & Enable
UpdateStatistics	Open, Claim, & Enable

*1: The availability condition differs from that of UPOS V 1.14.

5.1.4 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
DrawRuledLine	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

5.1.5 Events

Event Name	Occurrence 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.

5.2 Data Characters and Escape Sequences

5.2.1 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 and OPOS_SUCCESS(0) is returned when the **ValidateData** is used.
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 is 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**. 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 **MapMode** is set to PTR_MM_DOTS(1), a value from 0 to 255 can be specified for '#'. 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	• 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 0 to 65535 can be specified for '#'. If the '#' exceeds this range, transmission of embedded data is not executed. The **ValidateData** returns OPOS_E_ILLEGAL(106). If '#' is omitted, 0 is assumed to be specified, and OPOS_SUCCESS(0) is returned. If the print data for the number of bytes specified by '#' is not set after the escape sequence is specified, only the available print data to send is sent. (Example: IF ESC|2E"a" is specified, only "a" is sent since only one byte is set for the character string.) • 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 (See "In-Line Barcode Printing" below)	ESC|[*]#R	• 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. The **ValidateData** returns OPOS_E_ILLEGAL(106). • During rotated 90° right/left mode by the **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.
Print in-line ruled line (See "Ruled Line Drawing Printing" below)	ESC|*#dL	• Prints ruled lines. The placeholder '#' is replaced by an ASCII decimal string indicating the number of characters of the string following dL (definition of the ruled line 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 dL, or if the string following dL is improper, all the data within the range specified by '#' is discarded. The **ValidateData** returns OPOS_E_ILLEGAL(106). • This is ignored during rotated 90° right/left mode by **RotatePrint** or during Page Mode by **PageModePrint**.

- **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)

  The attributes must be written in the above order. Every attribute is mandatory. If either of the two conditions, parameter range violation or unsupported parameter value, is satisfied, the barcode is not printed, and the data following R for the number of characters specified by '#' is discarded. For the range of supported parameter values, see **PrintBarCode**.

  The following example prints UPC-A with the conditions 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**.

- **Ruled Line Drawing Printing**

  The application can print ruled lines along with other print data by using the "Print in-line ruled line" escape sequence (ESC|*#dL). The placeholder '#' is replaced by the number of characters of the string (definition of the ruled line characteristics) following dL.

  The string following dL specifies the ruled line characteristics using lowercase alphabet letters and numbers. The available numbers are the constant values defined for **DrawRuledLine**.

  The characters indicating the attributes are as follows:

  p: position (position of ruled line)

  d: direction (direction of ruled line)

  w: width (ruled line width)

  s: style (ruled line style)

  c: color (ruled line color)

  The attributes must be written in the above order. Every attribute is mandatory. If either of the two conditions, parameter range violation and unsupported parameter value, is satisfied, the ruled line is not printed, and the data following dL for the number of characters specified by '#' is discarded. For the range of supported parameter values, see **DrawRuledLine**.

The following example prints a horizontal ruled line with the conditions of solid line, 300 dots long starting at the 0th dot, 1 dot width, and black (Primary color).

ESC|*14dLp0,300d1w1s1c1

5.2.2 Escape Sequence Valid Until Changed

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

5.2.3 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 omitted, then a thickness of 1 is used for the underline. If '#' is 3 or larger, then a thickness of 2 is used. • If '!' is specified, the underline mode is cleared.
Italic	ESC|[!]iC	Not supported.
Alternate color (Custom)	ESC|[#]rC	Not supported.
Red	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 '#' is 8 or larger, then the scale factor 8 is used. If '#' is omitted, the data is regarded as print data instead of escape sequence and **ValidateData** returns OPOS_E_FAILURE(111).

Name	Data	Remarks
Scale vertically	ESC|#vC	• A supported value for the placeholder '#' is 1 to 8. If '#' is 8 or larger, then the scale factor 8 is used. If '#' is omitted, the data is regarded as print data instead of escape sequence and **ValidateData** returns OPOS_E_FAILURE(111).
RGB Color	ESC|[#]fC	Not supported.
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 line feed 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 line feed on the print data, the left justify valid after line feed. • 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.

5.3 Common Properties

BinaryConversion Property R/W

Syntax **LONG BinaryConversion;**

Remarks OPOS passes multi-character input and output using BStrings. BStrings may be safely used for text data. As the BStrings are passed between the application and the OPOS Control, OLE may execute language-specific translations to or from Unicode. When BStrings are used to pass binary data, then these translations may alter the data such that the data byte in a BString character at the application does not match the corresponding byte at the Control. This mismatch is more likely when BString pointers are used, since the Unicode characters are presented to the application and/or OPOS Control, and a language difference between them may cause misinterpretation.

Characters between 0x00 and 0x7F may be sent without fear of language-specific translation. Only characters between 0x80 and 0xFF sometimes cause incorrect translations.

The values of **BinaryConversion** are as follows.

Value	Meaning
OPOS_BC_NONE(0)	Data is placed one byte per BString character, with no conversion.
OPOS_BC_NIBBLE(1)	Each byte is converted into two characters (This option provides for the fastest conversion between binary and ASCII characters). First character = 0x30 + bits 7 to 4 of the data byte. Second character = 0x30 + bits 3 to 0 of the data byte. Example: Byte value 154 = 0x9A is converted to the characters 0x39 0x3A (= string "9:"). Note that this conversion is not the more common hexadecimal ASCII, which would have converted 154 to 0x39 0x41 (= string "9A").
OPOS_BC_DECIMAL(2)	Each byte is converted into three characters. VAL (string) may be used on each 3 characters to convert from ASCII to binary. RIGHT ("^^"+STR(byte),3) may be used to produce 3 ASCII characters from each byte, where '^' represents the space character. Example 1: Byte value 154 = 0x9A becomes the characters 0x31 0x35 0x34 (= string "154"). Example 2: Byte value 8 becomes the characters 0x30 0x30 0x38 (= string "008").

When **BinaryConversion** is on (that is, not OPOS_BC_NONE(0)) and the property or method parameter description specifies that **BinaryConversion** applies, before setting the property or passing the method parameter, convert the string data into the format specified by the **BinaryConversion** value.

This property is initialized to OPOS_BC_NONE(0) by **Open**.

Return When this property is set, one of the following values is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An illegal value was specified.

CapCompareFirmwareVersion Property

Syntax **BOOL CapCompareFirmwareVersion;**

Remarks Indicates the function that compares firmware version.
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

Syntax **LONG CapPowerReporting;**

Remarks Identifies the reporting capabilities of the device.
The following table shows the valid property value.

Value	Meaning
OPOS_PR_STANDARD(1)	The following 2 types of power state can be determined and reported. • OFF_OFFLINE (power off or offline) • ONLINE

This property is initialized to OPOS_PR_STANDARD(1) by **Open**.

CapStatisticsReporting Property

Syntax **BOOL CapStatisticsReporting;**

Remarks Identifies the reporting capabilities of the device.
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 **RetrieveStatistics**.

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

CapUpdateFirmware Property

Syntax **BOOL CapUpdateFirmware;**

Remarks Indicates whether the device supports firmware updating.
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

Syntax **BOOL CapUpdateStatistics;**

Remarks Indicates the function that some or all device statistics can be reset.
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 **ResetStatistics**.

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

CheckHealthText Property

Syntax **BSTR CheckHealthText;**

Remarks Holds the results of the immediately preceding call to the **CheckHealth**.
The following examples show the results of diagnosis.

Value	Meaning
"Internal HCheck: Successful"	Confirmed that the printer is in the printable state.
"Internal HCheck: Failure"	Unable to confirm that the printer is in the printable state.
"External HCheck: Successful"	Succeeded in the communication confirmation of the printer and printing test.
"External HCheck: Failure"	Failed in the communication confirmation of the printer and printing test using the device.
"Interactive HCheck: Successful"	Succeeded in the interactive test of the device.
"Interactive HCheck: Failure"	Failed in the interactive test of the device.
"Interactive HCheck: Canceled"	For the interactive test of the device, the dialog is closed without testing.

This value is initialized to an empty string before the first call to **CheckHealth**.

Claimed Property

Syntax **BOOL Claimed;**

Remarks Indicates whether the device is claimed for exclusive access.
The following table shows the valid property values.

Value	Meaning
TRUE	The device is claimed for exclusive access
FALSE	The device is released for sharing with other applications.

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

ControlObjectDescription Property

Syntax **BSTR ControlObjectDescription;**

Remarks Identifies the Control Object by this property.
"OPOS POSPrinter Control 1.14.xxx [Public, by CRM/RCS-Dayton]" is set.

This property is always readable.

ControlObjectVersion Property

Syntax **LONG ControlObjectVersion;**

Remarks This property holds the Control Object version number.

This property is always readable.

DeviceDescription Property

Syntax **BSTR DeviceDescription;**

Remarks This property identifies devices and related information.
The value differs depending on the printer.

Printer	Default
MP-B21L	"SII MP-B21L POS Printer"
MP-B30L	"SII MP-B30L POS Printer"
MP-B30	"SII MP-B30 POS Printer"

This property is initialized by **Open**.

DeviceEnabled Property R/W

Syntax **BOOL DeviceEnabled;**

Remarks Indicates whether the device has been placed in an operational state.
The following table shows the valid property values.

Value	Meaning
TRUE	The device has been placed in an operetional state. If changed to TRUE, then the device is brought to an operational state.
FALSE	The device has been disabled. If changed to FALSE, then the device is physically disabled.

The application must set this property to TRUE before using output devices.

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

Return When this property is set, one of the following values is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_NOTCLAIMED(103)	To enable an exclusive use device, its exclusive access must be obtained previously.
OPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
OPOS_E_FAILURE(111)	The device cannot be enabled. Setting information and the information from device may be different.
OPOS_E_TIMEOUT(112)	The Service Object timed out waiting for a response from the device, or the data was unable to be transmitted to the device within the timeout.
OPOS_E_BUSY(113)	Setting of property was failed due to processing. Set the property after process is completed.

DeviceName Property

Syntax **BSTR DeviceName;**

Remarks Identifies the device and any pertinent information about it.
The value differs depending on the printer.

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

This property is initializedby **Open**.

FreezeEvents Property R/W

Syntax **BOOL FreezeEvents;**

Remarks Selects whether to notify events.
The following table shows the valid property values.

Value	Meaning
TRUE	The application has requested that the Control not deliver events. Events will be held by the Control until events are unfrozen.
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, then changing **FreezeEvents** to FALSE will cause these events to be delivered.

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 the print method such as the **PrintNormal** is operated under the **AsyncMode** is TRUE, **ErrorEvent** is frozen and the **State** turns to OPOS_S_BUSY(3). In this case, clear the frozen event by the **ClearOutput**, or conduct the **Close** after **ErrorEvent** is occurred by setting FALSE since the control cannot be closed under this circumstance.

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

Return When this property is set, the following value is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.

OpenResult Property

Syntax **LONG OpenResult;**

Remarks Holds additional details about the most recent **Open**. The values of **OpenResult** are as follows.

Value	Meaning
OPOS_SUCCESS(0)	Successful open.
OPOS_OR_ALREADYOPEN(301)	Already open.
OPOS_OR_REGBADNAME(302)	The registry does not contain a key for the specified device name. Or, the device name is not specified.
OPOS_OR_REGPROGID(303)	Could not read the device name key's default, or could not convert the Programmatic ID it holds into a valid Class ID.
OPOS_OR_CREATE(304)	Could not create a Service Object instance, or could not get its IDispatch interface.
OPOS_OR_BADIF(305)	The Service Object does not support one or more of the methods required by its release. The setting of device name may be different from the Service Object.

This property is initialized to OPOS_SUCCESS(0) by **Open**.

OutputID Property

Syntax **LONG OutputID;**

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

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

Output ID is numbered from 1 to 2147483646 cyclically.

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

PowerNotify Property R/W

Syntax **LONG PowerNotify;**

Remarks Contains the type of power notification selection made by the application.
The following table shows the valid property values.

Value	Meaning
OPOS_PN_DISABLED(0)	The Control will not provide any power notifications to the application. No power notification **StatusUpdateEvents** will be fired, and **PowerState** is not set.
OPOS_PN_ENABLED(1)	The Control will fire power notification **StatusUpdateEvents** and update **PowerState**, beginning when **DeviceEnabled** is set to TRUE. The function level depends on **CapPowerReporting**.

PowerNotify may only be set while the device is disabled; that is, while **DeviceEnabled** is FALSE.

This property is initialized to OPOS_PN_DISABLE (0) by **Open**.

Return When this property is set, one of the following values is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	One of the following reasons: • **DeviceEnabled** = TRUE • The setting range of property is improper.

PowerState Property

Syntax **LONG PowerState;**

Remarks Contains the current power condition of the device.
The following table shows the valid property values.

Value	Meaning
OPOS_PS_ONLINE(2001)	The device is powered on and ready.
OPOS_PS_OFF_OFFLINE(2004)	The device is powered off or offline.
OPOS_PS_UNKNOWN(2000)	Cannot determine the device's power state, for one of the following reasons: • **PowerNotify** = OPOS_PN_DISABLED(0) • **DeviceEnabled** = FALSE

This property is initialized to OPOS_PS_UNKNOWN(2000) by **Open**.

ResultCode Property

Syntax **LONG ResultCode;**

Remarks This property is set by each method. It is also set when a writable property is set.

This property is always readable. Before **Open** is called, it returns the value OPOS_E_CLOSED(101).

The values of **ResultCode** are as follows.

Value	Meaning
OPOS_SUCCESS(0)	Successful operation.
OPOS_E_CLOSED(101)	Attempt was made to access a closed device. This error is not mentioned in the description of property and method.
OPOS_E_NOTCLAIMED(103)*1	Attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
OPOS_E_NOSERVICE(104)	The Control cannot communicate with the Service Object. The software must be re-installed the software. See "SII Software Package for Windows Installation Guide" for the installation and uninstallation procedure.
OPOS_E_DISABLED(105)*1	Cannot execute operation while device is disabled.
OPOS_E_ILLEGAL(106)	Attempt was made to execute an illegal or unsupported operation with the device, or an invalid parameter value was used.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_NOEXIST(109)	The file name (or other specified value) does not exist.
OPOS_E_FAILURE(111)	The device cannot execute the requested procedure, even though the device is connected to the system, powered on, and on-line.

Value	Meaning
OPOS_E_TIMEOUT(112)	The Service Object timed out waiting for a response from the device, or the data was unable to be transmitted to the device within the timeout.
OPOS_E_BUSY(113)	The Service Object does not allow this request currently. For example, if asynchronous output is in progress, certain methods may not be allowed.
OPOS_E_EXTENDED(114)	A class-specific error condition occurred. The error condition code is available in **ResultCodeExtended**.

*1: When multiple values of OPOS_E_NOTCLAIMED(103) and OPOS_E_DISABLED(105) are valid, the value of OPOS_E_NOTCLAIMED(103) has the high priority.

ResultCodeExtended Property

Syntax **LONG ResultCodeExtended;**

Remarks When **ResultCode** is set to OPOS_E_EXTENDED(114), the following POS Printer Device class-specific error information value and this software-specific error information value are set. When **ResultCode** is set to other value, a value of 0 is set.

This software takes the following values.

Value	Meaning
OPOS_EPTR_COVER_OPEN (201)	The printer cover is open.
OPOS_EPTR_REC_EMPTY (203)	The receipt is out of paper.
OPOS_EPTR_TOOBIG(206)	The bitmap file is too wide to print without conversion or too big to convert. The maximum print width must be up to the value of **RecLineWidth** during normal and inversion mode and within the maximum value of the printer during the bitmap is rotated 90° to the right/left by the **RotatePrint**. For the height, the range not exceeding the maximum value of the printer is supported during normal and inversion mode and **RecLineWidth** dots is supported during rotated 90° right/left mode.
OPOS_EPTR_BADFORMAT(207)	The specified file is not a bitmap file or not a supported format.
OPOS_ESTATS_ERROR(280)	One or more of the specified statistics could not be reset.
OPOS_EPTR_VPPOWER(1001)	Vp voltage error has occurred. (Specific error)
OPOS_EPTR_HEAD_TEMP(1005)	Head-temperature error has occurred. (Specific error)
OPOS_EPTR_IMAGEAREA_FULL (1009)	No memory is available to register the image to the user area of the printer. (Specific error)
OPOS_EPTR_UNRECOVERABLE (1010)	A non-recoverable error (hardware error) has occurred. (Specific error)
OPOS_EPTR_BATTERY(1013)	A battery error has occurred. (Specific error)
OPOS_EPTR_MARK_JAM(1014)	Paper jam error has occurred at the mark or the gap detection. (Specific error)

ServiceObjectDescription Property

Syntax **BSTR ServiceObjectDescription;**

Remarks Contains a string for identifying the service object to this property.
The value differs depending on the printer.

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

This property is initialized by **Open**.

ServiceObjectVersion Property

Syntax **LONG ServiceObjectVersion;**

Remarks This property holds the Service Object version number.

This property is initialized by **Open**.

State Property

Syntax **LONG State;**

Remarks Contains the current state of the Control.
The following table shows the valid property values.

Value	Meaning
OPOS_S_CLOSED(1)	The Control is closed.
OPOS_S_IDLE(2)	The Control is in a good state and is not busy.
OPOS_S_BUSY(3)	The Control is in a good state and is busy executing output.
OPOS_S_ERROR(4)	An error has been reported, and the application must recover the Control to a good state before normal I/O can resume. This state is only possible inside the **ErrorEvent** event handler.

This property is always readable.
This property returns OPOS_S_CLOSED(1) until **Open** is successfully completed.

5.4 Specific Properties

AsyncMode Property R/W

Syntax **BOOL AsyncMode;**

Remarks Indicates whether certain print methods will be performed asynchronously.
The following table shows the valid property values.

Value	Meaning
TRUE	**PrintNormal**, **PageModePrint**, **PrintBarCode**, **PrintBitmap**, **PrintMemoryBitmap**, **DrawRuledLine**, **MarkFeed**, **RotatePrint**, and **TransactionPrint** print methods are executed asynchronously.
FALSE	The above methods are executed synchronously.

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

Return When this property is set, one of the following value is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.

CapCharacterSet Property

Syntax **LONG CapCharacterSet;**

Remarks Holds the printable character setting of the printer.
This property has the following value.

Value	Meaning
PTR_CCS_KANJI(11)	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 PTR_CCS_KANJI(11) by **Open**.

CapCoverSensor Property

Syntax **BOOL CapCoverSensor;**

Remarks Indicates whether the printer has a "cover open" sensor.
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

Syntax **BOOL CapMapCharacterSet;**

Remarks Indicates that the Service Object is able to map the characters of the application to a character set.
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

Syntax **BOOL CapRec2Color;**

Remarks Indicates whether the receipt can print dark plus an alternate color.
The following table shows the valid property value.

Value	Meaning
FALSE	The receipt cannot support 2 color printing.

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

CapRecBarCode Property

Syntax **BOOL CapRecBarCode;**

Remarks 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

Syntax **BOOL CapRecBitmap;**

Remarks 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

Syntax **BOOL CapRecBold;**

Remarks 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

Syntax **LONG CapRecCartridgeSensor;**

Remarks Indicates the presence of receipt cartridge monitoring sensors.
The following table shows the valid property value.

Value	Meaning
0	Receipt cartridge monitoring sensors are not supported.

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

CapRecColor Property

Syntax **LONG CapRecColor;**

Remarks Indicates the availability of receipt color cartridges.
The following table shows the valid property value.

Value	Meaning
PTR_COLOR_PRIMARY(0x00000001)	Receipt color cartridges are not supported.

This property is initialized to PTR_COLOR_PRIMARY(0x00000001) by **Open**.

CapRecDhigh Property

Syntax **BOOL CapRecDhigh;**

Remarks 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

Syntax **BOOL CapRecDwide;**

Remarks 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

Syntax **BOOL CapRecDwideDhigh;**

Remarks 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

Syntax **BOOL CapRecEmptySensor;**

Remarks 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

Syntax **BOOL CapRecItalic;**

Remarks Indicates whether the receipt can print italic characters.
The following table shows the valid property value.

Value	Meaning
FALSE	The receipt cannot support Italic characters.

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

CapRecLeft90 Property

Syntax **BOOL CapRecLeft90;**

Remarks 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

Syntax **LONG CapRecMarkFeed;**

Remarks Indicates the type of mark sensed paper handling available.
Either a logical OR of the following values is set in this property.
Specifying one of the following values in **MarkFeed**, its function can be executed.

Value	Meaning
0	The function of mark detection is Disabled.
PTR_MF_TO_TAKEUP(1)	After detecting the mar or the gap, feeds the paper to the paper take-up position.
PTR_MF_TO_CUTTER(2)	After detecting the mar or the gap, feeds the paper to the cutting position. (Feeds the paper to the same position as PTR_MF_TO_TAKEUP(1)).
PTR_MF_TO_CUTTERTOF(4)	Supported only by MP-B21L. Feeds backward the marked paper to the printing position.
PTR_MF_TO_NEXT_TOF(8)	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 0 by **Open**.

CapRecNearEndSensor Property

Syntax **BOOL CapRecNearEndSensor;**

Remarks Indicates whether the receipt has a low paper sensor.
The following table shows the valid property value.

Value	Meaning
FALSE	A low paper sensor is not supported.

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

CapRecPageMode Property

Syntax **BOOL CapRecPageMode;**

Remarks 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

Syntax **BOOL CapRecPapercut;**

Remarks 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

Syntax **BOOL CapRecPresent;**

Remarks Indicates the presence of the receipt station.
The following table shows the valid property value.

Value	Meaning
TRUE	The receipt station is present.

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

CapRecRight90 Property

Syntax **BOOL CapRecRight90;**

Remarks 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

Syntax **BOOL CapRecRotate180;**

Remarks 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**.

CapRecRuledLine Property

Syntax **LONG CapRecRuledLine;**

Remarks Indicates the printer has the ability to support the use of ruled lines in the receipt.
A logical OR of the following values is set.

Value	Meaning
PTR_RL_HORIZONTAL(1)	Able to print horizontal ruled lines.
PTR_RL_VERTICAL(2)	Able to print vertical ruled lines.

This property is initialized to a logical OR of the above values(3) by **Open**.

CapRecStamp Property

Syntax **BOOL CapRecStamp;**

Remarks Indicates whether the receipt has a stamp capability.
The following table shows the valid property value.

Value	Meaning
FALSE	The receipt cannot support stamp printing.

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

CapRecUnderline Property

Syntax **BOOL CapRecUnderline;**

Remarks 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

Syntax **BOOL CapTransaction;**

Remarks Indicates whether receipt station supports printer transactions.
The following table shows the valid property value.

Value	Meaning
TRUE	The printer transactions are supported by each station.

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

CartridgeNotify Property R/W

Syntax **LONG CartridgeNotify;**

Remarks Contains the type of cartridge state notification selected by the application.
The following table shows the valid property value.

Value	Meaning
PTR_CN_DISABLED(0)	Cartridge state notification is not provided.

This property is initialized to PTR_CN_DISABLE(0) by **Open**.

Return When this property is set, the following values are stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully. (Only when the default is set.)
OPOS_E_ILLEGAL(106)	Property setting is not supported.

CharacterSet Property R/W

Syntax **LONG CharacterSet;**

Remarks Holds the character set for printing characters.
One of the following values is set to this property:

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.

Value	Meaning
866	Selects Code Page 866 character set.
932	Selects Katakana as the Code Page 932 character set (Shift-JIS Code).
PTR_CS_ANSI (999) / PTR_CS_WINDOWS (999)	Sets the Windows ANSI characters.*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**.

Return When this property is set, one of the following values is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	An illegal value was specified.

CharacterSetList Property

Syntax **BSTR CharacterSetList;**

Remarks Holds the character set numbers.
"437,737,850,852,855,857,858,860,863,865,866,932,999,1250,1251,1252,1253,1254" is set.

This property is initialized to the above value by **Open**.

CoverOpen Property

Syntax **BOOL CoverOpen;**

Remarks Indicates whether the printer cover is open.
The following table shows the valid property values.

Value	Meaning
TRUE	The printer cover is open.
FALSE	The printer cover is closed.

This property is initialized while the device is enabled and keeps the current state.

ErrorLevel Property

Syntax **LONG ErrorLevel;**

Remarks Holds the severity of the error condition.
One of the following values is set to this property:

Value	Meaning
PTR_EL_NONE(1)	No error condition is present.
PTR_EL_RECOVERABLE(2)	A recoverable error has occurred.
PTR_EL_FATAL(3)	A non-recoverable error has occurred. This error is set when **ResultCode** is OPOS_E_EXTENDED(114) and **ResultCodeExtended** is OPOS_EPTR_UNRECOVERABLE(1010).

This property is set by the Control just before the notification of **ErrorEvent**.
When the error is cleared, then the property is changed to PTR_EL_NONE(1).

ErrorStation Property

Syntax **LONG ErrorStation;**

Remarks Holds the station that was printing when an error was detected.
The following value is set to this property.

Value	Meaning
PTR_S_RECEIPT(2)	The error is detected at the receipt station.

This property is set before **ErrorEvent** is notified.

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

ErrorString Property

Syntax **BSTR ErrorString;**

Remarks Holds a vendor-supplied description of the current error.

Value	Meaning
"Off/Offline error"	OPOS_E_NOHARDWARE(107)
"Unrecoverable error"	OPOS_EPTR_UNRECOVERABLE(1010)
"Battery error"	OPOS_EPTR_BATTERY(1013)
"Head temperature error"	OPOS_EPTR_HEAD_TEMP(1005)
"Vp voltage error"	OPOS_EPTR_VPPOWER(1001)
"Cover open error"	OPOS_EPTR_COVER_OPEN(201)
"Out-of-paper error"	OPOS_EPTR_REC_EMPTY(203)
"Mark paper jam error"	OPOS_EPTR_MARK_JAM(1014)
"Failure"	OPOS_E_FAILURE(111)

Value	Meaning
"Time out"	OPOS_E_TIMEOUT(112)

The values in the above table are described in descending order of priority. When multiple errors occur simultaneously, the higher priority item is set to **ErrorString**.

This property is set by the Control only before the notification of **ErrorEvent** to the application. When none of the above values are available, the property is set to an empty string. When the error is cleared, then the property is changed to an empty string.

FlagWhenIdle Property R/W

Syntax **BOOL FlagWhenIdle;**

Remarks Indicates whether or not to notify that **StatusUpdateEvent**.
The following table shows the valid property values.

Value	Meaning
TRUE	A **StatusUpdateEvent** will be notified when the POS Printer Control is in the idle state.
FALSE	This event is not notified.

FlagWhenIdle is automatically reset to FALSE when this status event is delivered.
The main use of idle status event that is controlled by this property is to give the application control when all outstanding asynchronous outputs have been processed. The event will be notified if the outputs were completed successfully or if they were cleared by the event handler that receives **ErrorEvent**.
If **State** is already set to OPOS_S_IDLE(2) when **FlagWhenIdle** is set to TRUE, then **StatusUpdateEvent** is notified immediately. The application can therefore depend on the event, with no race condition between the starting of its last asynchronous output and the setting of this flag.

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

When this property is set, the following value is stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.

FontTypefaceList Property

Syntax **BSTR FontTypefaceList;**

Remarks Holds the fonts and/or typefaces that are supported by the printer.
This property is initialized to an empty string by **Open**.

MapCharacterSet Property R/W

Syntax **BOOL MapCharacterSet;**

Remarks This property is initialized to FALSE by **Open**.

Return When this property is set, the following values are stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully. (Only when the default is set.)
OPOS_E_ILLEGAL(106)	Property setting is not supported.

MapMode Property R/W

Syntax **LONG MapMode;**

Remarks Holds the mapping mode of the POS Printer Control. 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 the brackets indicate the value converted into dot.

Value	Meaning
PTR_MM_DOTS(1)	Printer's dot width, 0.125 mm (1 dot)
PTR_MM_TWIPS(2)	1/1440 of an inch (0.1411 dots)
PTR_MM_ENGLISH(3)	0.001 inch (0.203 dots)
PTR_MM_METRIC(4)	0.01 mm (0.08 dots)

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

MapMode	Conversion
PTR_MM_DOTS(1) Printer dot width (dot value)	No conversion
PTR_MM_TWIPS(2) 1 / 1440 inch	k = 1 / 1440 ■DOT -> TWIPS conversion twips = dot / (dpi × k) ■TWIPS -> DOT conversion dot = twips × dpi × k
PTR_MM_ENGLISH(3) 0.001 inch	k = 1 / 1000 ■DOT -> ENGLISH conversion english = dot / (dpi × k) ■ENGLISH -> DOT conversion dot = english × dpi × k
PTR_MM_METRIC(4) 0.01 mm	k = 1 / 100, mmpi = 25.4 ■DOT -> METRIC conversion Metric = (mmpi × dot) / (dpi × k) ■METRIC -> DOT conversion dot = (metric × dpi × k) / mmpi

(1 inch = 25.4 mm)

MapMode only changes the unit of each property for display, and all internal processings are executed in dot 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 **MapMode** may also change **RecLineSpacing**, **RecLineWidth**, **RecLineHeight**, **PageModeArea**, **PageModePrintArea**, **PageModeHorizontalPosition**, and **PageModeVerticalPosition**.

This property is initialized to PTR_MM_DOTS(1) by **Open**.

Return When this property is set, the following values are stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	Improper mapping mode was specified.

PageModeArea Property

Syntax **BSTR PageModeArea;**

Remarks Holds the page area for the selected **PageModeStation** expressed in the unit of the measure given by **MapMode**.

This page area can be different than the print area and is determined by the hardware capability of the printer. The string consists of two ASCII numbers separated by a comma, in the following order: horizontal size, vertical size.
Specify PTR_S_RECEIPT(2) for **PageModeStation** before accessing this property.
Otherwise an empty string is returned until PTR_S_RECEIPT(2) is specified.

One of the following values is set to this property. (When **MapMode** = PTR_MM_DOTS(1))

MP-B21L

Value	Meaning
"168,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 168.
"176,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 176.
"192,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 192.
"208,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 208.
"224,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 224.
"240,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 240.
"256,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 256.
"272,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 272.
"288,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 288.
"304,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 304.
"320,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 320.
"336,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 336.
"352,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 352.
"356,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 356.
"368,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 368.
"384,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 384.

MP-B30L

Value	Meaning
"360,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 360.
"368,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 368.
"384,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 384.
"400,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 400.
"416,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 416.
"432,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 432.
"448,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 448.
"464,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 464.
"480,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 480.
"496,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 496.
"512,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 512.
"528,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 528.
"544,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 544.
"560,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 560.
"576,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 576.

MP-B30

Value	Meaning
"576,2400"	The **PageModeArea** value when the **RecLineWidth** is set to 576.

PageModeDescriptor Property

Syntax **LONG PageModeDescriptor;**

Remarks The Page Mode functionality available on the station specified for **PageModeStation** is indicated by OR of the following values.
Specify PTR_S_RECEIPT(2) for **PageModeStation** before accessing this property.
Otherwise 0 is returned until PTR_S_RECEIPT(2) is specified.

Value	Meaning
PTR_PM_BITMAP(1)	Printing of bitmaps on the **PageModeStation** is supported.
PTR_PM_BARCODE(2)	Printing of barcodes on the **PageModeStation** is supported.
PTR_PM_BM_ROTATE(4)	Rotation of bitmaps on the **PageModeStation** is supported.
PTR_PM_BC_ROTATE(8)	Rotation of barcodes on the **PageModeStation** is supported.

This property is initialized to OR of the above values after PTR_S_RECEIPT(2) is set.

PageModeHorizontalPosition Property R/W

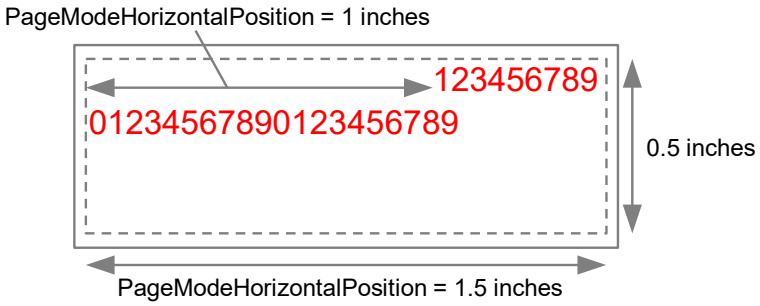
Syntax **LONG PageModeHorizontalPosition;**

Remarks Holds the offset value for correcting the horizontal start position within the Page Mode print area for the selected **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. Specify PTR_S_RECEIPT(2) for the **PageModeStation** before accessing this property.
Specify PTR_S_RECEIPT(2) for **PageModeStation** before accessing this property. Otherwise 0 is returned until PTR_S_RECEIPT(2) is specified.

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

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to 1.5 inches by 0.5 inches
myptr.setPageModePrintArea("0,0,1500,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.setPageModeHorizontalPosition(1000);
myptr.printNormal(PTR_S_RECEIPT, "12345678901234567890123456789O\n");
```

The code sample above will generate the following receipt.



Return When this property is set, the following values are stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An illegal value was specified.

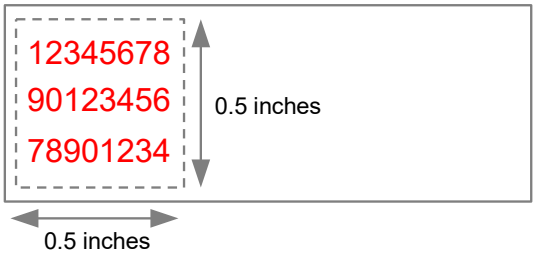
PageModePrintArea Property R/W

Syntax **BSTR PageModePrintArea;**

Remarks Holds the Page Mode print area for the selected **PageModeStation** expressed in the unit specified by **MapMode**.
The maximum print area of **PageModePrintArea** is the page area of **PageModeArea**.
The property consists of four ASCII numbers separated by commas, in the following order: horizontal start, vertical start, horizontal size, vertical size. For example, if the string is "50,100,200,300," then the station print area is a rectangle beginning at the top left position (50,100), and the bottom right position (199,299).
The text beyond the right edge of the Page Mode print area will be printed to the next line. Any text or image beyond the bottom of the print area will not be printed. For example:

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to half inch square block
myptr.setPageModePrintArea("0,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789O\n");
```

The code sample above will generate the following receipt.



Specify PTR_S_RECEIPT(2) for **PageModeStation** before accessing this property. Otherwise an empty string is returned until PTR_S_RECEIPT(2) is specified. It is initialized to "0,0,0,0" when PTR_S_RECEIPT(2) is specified.

Return When this property is set, the following values are stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An illegal value was specified.

PageModePrintDirection Property R/W

Syntax **LONG PageModePrintDirection;**

Remarks Holds the print direction. The print direction shall be as follows:

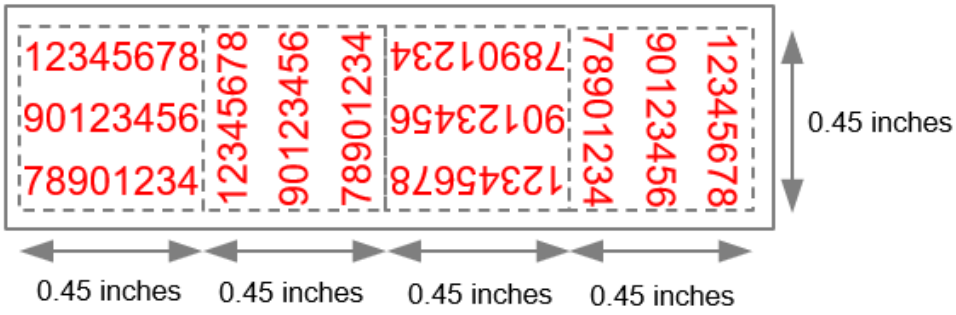
Value	Meaning
PTR_PD_LEFT_TO_RIGHT(1)	Prints from left to right, starting at the top left corner of the Page Mode print area. Normal direction printing.
PTR_PD_BOTTOM_TO_TOP(2)	Prints from bottom to top, starting at the bottom left corner of the Page Mode print area. Rotated left 90° printing.
PTR_PD_RIGHT_TO_LEFT(3)	Prints from right to left, starting at the bottom right corner of the Page Mode print area. Upside down printing.
PTR_PD_TOP_TO_BOTTOM(4)	Prints from top to bottom, starting at the top right corner of the Page Mode print area. Rotated right 90° printing.

Changing this property may also changes 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.

For example:

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to half inch square block
myptr.setPageModePrintArea("0,0,450,450");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789012345678901234567890\n");
myptr.setPageModePrintArea("450,0,450,450");
myptr.setPageModePrintDirection(PTR_PD_BOTTOM_TO_TOP);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789012345678901234567890\n");
myptr.setPageModePrintArea("900,0,450,450");
myptr.setPageModePrintDirection(PTR_PD_RIGHT_TO_LEFT);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789012345678901234567890\n");
myptr.setPageModePrintArea("1350,0,450,450");
myptr.setPageModePrintDirection(PTR_PD_TOP_TO_BOTTOM);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789012345678901234567890\n");
```

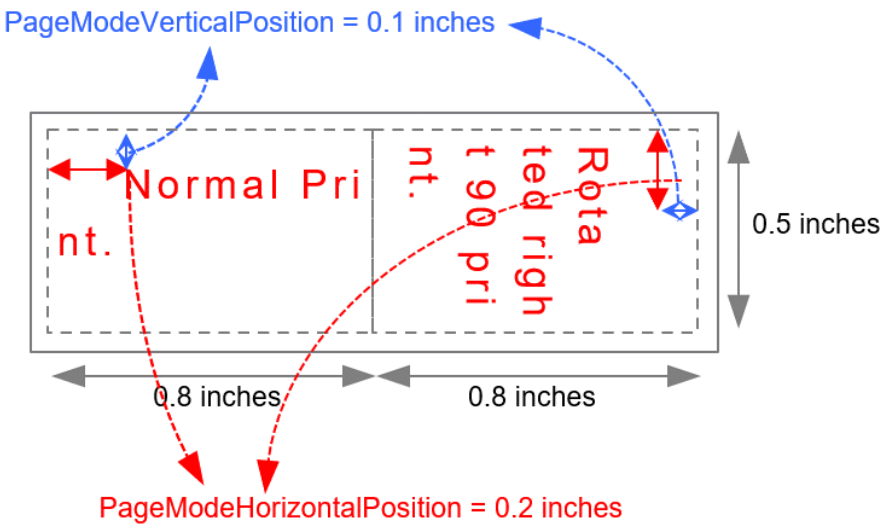
The code sample above will generate the following receipt.



It is also possible to generate rotated text.

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
myptr.setPageModePrintArea("0,0,800,500");
myptr.pageModeVerticalPosition(100);
myptr.pageModeHorizontalPosition(200);
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.printNormal(PTR_S_RECEIPT, "Normal print.\n");
myptr.setPageModePrintArea("800,0,800,500");
myptr.setPageModePrintDirection(PTR_PD_TOP_TO_BOTTOM);
myptr.printNormal(PTR_S_RECEIPT, "Rotated right 90 print.\n");
myptr.setPageModePrint(PTR_PM_NORMAL);
```

The code sample above will generate the following receipt.



Specify PTR_S_RECEIPT(2) for **PageModeStation** before accessing this property. Otherwise 0 is returned until PTR_S_RECEIPT(2) is specified. It is initialized to PTR_PD_LEFT_TO_RIGHT(1) when PTR_S_RECEIPT(2) is specified.

Return When this property is set, the following values are stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An illegal value was specified.

PageModeStation Property R/W

Syntax **LONG PageModeStation;**

Remarks Sets the print station for subsequent Page Mode properties.
Be sure to specify PTR_S_RECEIPT(2) for this property before accessing the property or method of the Page Mode function.

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

Return When this property is set, the following values are stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An illegal value was specified.

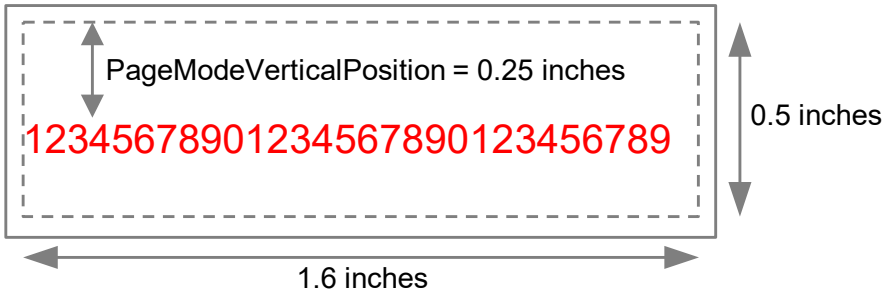
PageModeVerticalPosition Property R/W

Syntax **LONG PageModeVerticalPosition;**

Remarks Holds the offset value for correcting the vertical start position within the Page Mode print area for the selected **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.
The following code sample shows usage of **PageModeVerticalPosition**.

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to 1.6 inches by 0.5 inches
myptr.setPageModePrintArea("0,0,1600,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.setPageModeVerticalPosition(250);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789\n");
```

The code example above will generate the following receipt.



Specify PTR_S_RECEIPT(2) for **PageModeStation** before accessing this property. Otherwise 0 is returned until PTR_S_RECEIPT(2) is specified.

Return When this property is set, the following values are stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	An illegal value was specified.

RecBarCodeRotationList Property

Syntax **BSTR RecBarCodeRotationList;**

Remarks Holds the directions in which a receipt barcode may be rotated. This property is initialized to "0,R90,L90,180" by **Open**. The string consists of rotation strings separated by commas. The legal rotation strings are as follows:

Value	Meaning
0	Barcode may be printed in the normal orientation.
R90	Barcode may be printed in a rotated 90° to the right.
L90	Barcode may be printed in a rotated 90° to the left.
180	Barcode may be rotated 180° upside down.

RecBitmapRotationList Property

Syntax **BSTR RecBitmapRotationList;**

Remarks Holds the directions in which a receipt bitmap may be rotated.
This property is initialized to "0,R90,L90,180" by **Open**. The string consists of rotation strings separated by commas. The legal rotation strings are as follows:

Value	Meaning
0	Bitmap may be printed in the normal orientation.
R90	Bitmap may be printed in a rotated 90° to the right.
L90	Bitmap may be printed in a rotated 90° to the left.
180	Bitmap may be rotated 180° upside down.

RecCartridgeState Property

Syntax **LONG RecCartridgeState;**

Remarks This property contains the status of the currently selected receipt cartridge (ink, ribbon or toner). It contains the following value.

Value	Meaning
PTR_CART_UNKNOWN(0x10000000)	Device does not support cartridge state reporting.

This property is initialized to PTR_CART_UNKNOWN(0x10000000) when the device is first enabled by **Open**.

RecCurrentCartridge Property R/W

Syntax **LONG RecCurrentCartridge;**

Remarks Receipt Cartridge selection is not supported.

This property is initialized to PTR_COLOR_PRIMARY(0x00000001) when the device is first enabled by **Open**.

Return When this property is set, the following values are stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully. (Only when the default is set.)
OPOS_E_ILLEGAL(106)	An invalid property value was specified.

RecEmpty Property

Syntax **BOOL RecEmpty;**

Remarks Indicates whether the receipt is out of paper.
The following table shows the valid property values.

Value	Meaning
TRUE	The receipt is out of paper.

Value	Meaning
FALSE	The receipt is present.

This property is initialized while the device is enabled and keeps the current state.

RecLetterQuality Property R/W

Syntax **BOOL RecLetterQuality;**

Remarks Indicates whether the printer prints in high quality mode.
The following table shows the valid property values.

Value	Meaning
TRUE	Prints in the high quality mode.
FALSE	Prints in the high speed mode.

This property is initialized to FALSE when the device is first enabled by **Open**.

In the configuration program, if [PrintQuality] is other than the value of "RecLetterQuality Valid", this property is disabled and print is conducted in fixed speed printing mode as [PrintQuality] setting.

Return When this property is set, the following values are stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.

RecLineChars Property R/W

Syntax **LONG RecLineChars;**

Remarks Holds the number of characters that may be printed on a receipt line.
Depending on the specified number of characters, the printer prints in the following print font.

MP-B21L

<table>
<tr><th>RecLineWidth</th><th>RecLineChars</th><th>Print Font
(H × W)</th><th>Character Space</th><th>RecLineHeight</th></tr>
<tr><td rowspan="8">168</td><td>10</td><td rowspan="4">Font A
(24 dots × 12 dots)</td><td>4 dots</td><td rowspan="4">24</td></tr>
<tr><td>12</td><td>2 dots</td></tr>
<tr><td>12</td><td>1 dot</td></tr>
<tr><td>14 (Default)</td><td>0 dots</td></tr>
<tr><td>16</td><td rowspan="3">Font B
(16 dots × 8 dots)</td><td>2 dots</td><td rowspan="3">16</td></tr>
<tr><td>18</td><td>1 dot</td></tr>
<tr><td>21</td><td>0 dots</td></tr>
</table>

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
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	
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	

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
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	
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	

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
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	
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	

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	
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	

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
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	
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	

RecLineWidth	RecLineChars	Print Font (H × W)	Character Space	RecLineHeight
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	
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 value of **RecLineChars** is initialized to the default when the device is first enabled by **Open**.

Return When this property is set, one of the following values is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	Improper character width was specified.

RecLineCharsList Property

Syntax **BSTR RecLineCharsList;**

Remarks Holds the line character widths supported.
This property is initialized to the following values by **Open**.

MP-B21L

Value	Meaning
"10,12,14,16,18,21"	The **RecLineCharsList** value when the **RecLineWidth** is set to 168.
"11,12,13,14,17,19,22"	The **RecLineCharsList** value when the **RecLineWidth** is set to 176.
"12,13,14,16,19,21,24"	The **RecLineCharsList** value when the **RecLineWidth** is set to 192.
"13,14,16,17,20,23,26"	The **RecLineCharsList** value when the **RecLineWidth** is set to 208.
"14,16,17,18,22,24,28"	The **RecLineCharsList** value when the **RecLineWidth** is set to 224.
"15,17,18,20,24,26,30"	The **RecLineCharsList** value when the **RecLineWidth** is set to 240.
"16,18,19,21,25,28,32"	The **RecLineCharsList** value when the **RecLineWidth** is set to 256.

Value	Meaning
"17,19,20,22,27,30,34"	The **RecLineCharsList** value when the **RecLineWidth** is set to 272.
"18,20,22,24,28,32,36"	The **RecLineCharsList** value when the **RecLineWidth** is set to 288.
"19,21,23,25,30,33,38"	The **RecLineCharsList** value when the **RecLineWidth** is set to 304.
"20,22,24,26,32,35,40"	The **RecLineCharsList** value when the **RecLineWidth** is set to 320.
"21,24,25,28,33,37,42"	The **RecLineCharsList** value when the **RecLineWidth** is set to 336.
"22,25,27,29,35,39,44"	The **RecLineCharsList** value when the **RecLineWidth** is set to 352.
"22,25,27,30,36,40,45"	The **RecLineCharsList** value when the **RecLineWidth** is set to 360.
"23,26,28,30,36,40,46"	The **RecLineCharsList** value when the **RecLineWidth** is set to 368.
"24,27,29,32,38,42,48"	The **RecLineCharsList** value when the **RecLineWidth** is set to 384.

MP-B30L

Value	Meaning
"22,25,27,30,36,40,45"	The **RecLineCharsList** value when the **RecLineWidth** is set to 360.
"23,26,28,30,36,40,46"	The **RecLineCharsList** value when the **RecLineWidth** is set to 368.
"24,27,29,32,38,42,48"	The **RecLineCharsList** value when the **RecLineWidth** is set to 384.
"25,28,30,33,40,44,50"	The **RecLineCharsList** value when the **RecLineWidth** is set to 400.
"26,29,32,34,41,46,52"	The **RecLineCharsList** value when the **RecLineWidth** is set to 416.
"27,30,33,36,43,48,54"	The **RecLineCharsList** value when the **RecLineWidth** is set to 432.
"28,32,34,37,44,49,56"	The **RecLineCharsList** value when the **RecLineWidth** is set to 448.
"29,33,35,38,46,51,58"	The **RecLineCharsList** value when the **RecLineWidth** is set to 464.
"30,34,36,40,48,53,60"	The **RecLineCharsList** value when the **RecLineWidth** is set to 480.
"31,35,38,41,49,55,62"	The **RecLineCharsList** value when the **RecLineWidth** is set to 496.
"32,36,39,42,51,56,64"	The **RecLineCharsList** value when the **RecLineWidth** is set to 512.
"33,37,40,44,52,58,66"	The **RecLineCharsList** value when the **RecLineWidth** is set to 528.
"34,38,41,45,54,60,68"	The **RecLineCharsList** value when the **RecLineWidth** is set to 544.
"35,40,43,46,56,62,70"	The **RecLineCharsList** value when the **RecLineWidth** is set to 560.
"36,41,44,48,57,64,72"	The **RecLineCharsList** value when the **RecLineWidth** is set to 576.

MP-B30

Value	Meaning
"36,41,44,48,57,64,72"	The **RecLineCharsList** value when the **RecLineWidth** is set to 576.

RecLineHeight Property R/W

Syntax **LONG RecLineHeight;**

Remarks Holds the print line height.
The property is expressed in the unit indicated by **MapMode**.
This property is automatically updated by the change of **RecLineChars**,
OPOS_E_ILLEGAL(106) is returned when setting a value to this property.

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

The relation between **RecLineHeight** and the print font is as follows.
(When **MapMode** is PTR_MM_DOTS(1))

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

The value of **RecLineHeight** is initialized to 24 when the device is first enabled by **Open**.

RecLineSpacing Property R/W

Syntax **LONG RecLineSpacing;**

Remarks Holds the line spacing (the spacing of each single-high print line).
The line spacing includes both the printed line height plus the whitespace between each pair of lines. The property is expressed in the unit indicated by **MapMode**.
A value smaller than **RecLineHeight** cannot be specified. If a smaller value is specified, OPOS_E_ILLEGAL(106) is notified and the property is not changed.

The relation between **RecLineHeight** and **RecLineSpacing** is as follows. (When **MapMode** is PTR_MM_DOTS(1))

RecLineHeight	RecLineSpacing	Default
24	24 to 255	30
16	16 to 255	

If the value other than the configurable range is specified, OPOS_E_ILLEGAL(106) is notified and the property is ignored.
When **RecLineChars** is changed or new **RecLineHeight** is bigger than the value specified for **RecLineSpacing,** the value of **RecLineHeight** is set. In the same way, if the value of **RecLineSpacing** exceeds the configurable range, the maximum value of configurable range is automatically set.

The value of **RecLineSpacing** is initialized to 30 (if **MapMode** is PTR_MM_DOTS(1)) when the device is first enabled by **Open**.

Return When this property is set, the following values ares stored in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	The setting range of property is improper.

RecLinesToPaperCut Property

Syntax **LONG RecLinesToPaperCut;**

Remarks Holds the number of lines that must be advanced before cutting the receipt.
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.

Printer	Default
MP-B21L	2
MP-B30L	
MP-B30	

This property is initialized to the default depended on the printer when the device is first enabled by **Open**.

RecLineWidth Property

Syntax **LONG RecLineWidth;**

Remarks Holds the width of a line of **RecLineChars**.
The property is expressed in the unit indicated by **MapMode**.
This property is initialized to one of the following values.

MP-B21L

PrintWidth	RecLineWidth
21mm/168dots	168
22mm/176dots	176
24mm/192dots	192
26mm/208dots	208
28mm/224dots	224
30mm/240dots	240
32mm/256dots	256
34mm/272dots	272
36mm/288dots	288
38mm/304dots	304
40mm/320dots	320
42mm/336dots	336
44mm/352dots	352
45mm/360dots	360
46mm/368dots	368
48mm/384dots	384

MP-B30L

PrintWidth	RecLineWidth
45mm/360dots	360
46mm/368dots	368
48mm/384dots	384
50mm/400dots	400
52mm/416dots	416
54mm/432dots	432
56mm/448dots	448
58mm/464dots	464
60mm/480dots	480
62mm/496dots	496
64mm/512dots	512
66mm/528dots	528
68mm/544dots	544
70mm/560dots	560
72mm/576dots	576

MP-B30

PaperSize	RecLineWidth
80mm/576dots	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. (When **MapMode** is PTR_MM_DOTS(1))

RecNearEnd Property

Syntax **BOOL RecNearEnd;**

Remarks Indicates whether the receipt is low.
The following table shows the valid property value.

Value	Meaning
FALSE	A low paper sensor is not supported.

This property always indicates FALSE.

RecSidewaysMaxChars Property

Syntax **LONG RecSidewaysMaxChars;**

Remarks Holds the maximum number of one-byte characters that may be printed on each line in sideways mode (rotated 90° to the left or right).
This property is determined by the following calculation based on the **PageModeArea** and print font when the device is first enabled by **Open**.

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

Example: When the **PageModeArea** is "384,2400", and the **RecLineChars** is "32"

RecSidewaysMaxChars
= 2400 / (24 / 2) + 0 = 200 (The decimal part is round down.)

RecSidewaysMaxLines Property

Syntax **LONG RecSidewaysMaxLines;**

Remarks Holds the maximum number of lines that may be printed in the sideways mode (rotated 90° to the left or right).

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

This property is initialized when the device is first enabled by **Open**.

RotateSpecial Property R/W

Syntax **LONG RotateSpecial;**

Remarks Holds the rotation orientation for barcodes.
One of the following values is set to this property:

Value	Meaning
PTR_RP_NORMAL(1)	Prints subsequent barcodes in normal orientation.
PTR_RP_RIGHT90(257)	Rotates printing 90° to the right (clockwise).
PTR_RP_LEFT90(258)	Rotates printing 90° to the left (counter-clockwise).
PTR_RP_ROTATE180(259)	Rotates printing 180°, that is, print upside down.

This property is initialized to PTR_RP_NORMAL(1) by **Open**.

Return When this property is set, one of the following values is placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	Improper value of the property was specified.

5.5 Common Methods

CheckHealth Method

Syntax **LONG CheckHealth (LONG** *Level***);**

The *Level* indicates the type of health check to be executed on the device. The following values may be specified:

Value	Meaning
OPOS_CH_INTERNAL(1)	Executes an internal test without changing the physical state of the device. Confirm if the printer is in printable condition.
OPOS_CH_EXTERNAL(2)	Executes the test print after confirming the communication with the printer. The printer ROM version ID, **ServiceObjectVersion** and **DeviceName** of the POS Printer Control are printed.
OPOS_CH_INTERACTIVE(3)	Executes an interactive test of the device. The Service Object displays the modal dialog and prints the printer ROM version ID, **ServiceObjectVersion** and **DeviceName** of the POS Printer Control.

Remarks Calls to test the state of a device. A text description of the results of this method is stored in **CheckHealthText**. **CheckHealth** is always synchronous.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Indicates that the health checking procedure was initiated properly and, when possible to determine, indicates that the device is healthy. However, the health of many devices can only be determined by a visual inspection of the test results.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	Unsupported *Level* is specified.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_FAILURE(111)	A communication error has occurred.
OPOS_E_TIMEOUT(112)	Data transmission timeout or data receive timeout has occurred.
OPOS_E_BUSY(113)	Cannot execute while output is in progress or an error occurs.
OPOS_E_EXTENDED(114)	Except for OPOS_EPTR_TOOBIG(206) and OPOS_EPTR_BADFORMAT(207), the error defined by **ResultCodeExtended** is notified.

ClaimDevice Method

Syntax **LONG ClaimDevice (LONG** *Timeout***);**

The *Timeout* gives the maximum number of milliseconds to wait for exclusive access to be satisfied.
If the parameter is 0, the method returns the appropriate status immediately even though it cannot gain exclusive access to the device.
If OPOS_FOREVER(-1) is set, the method waits until exclusive access is satisfied.

Remarks Call this method to request exclusive access to the device.
The POS Printer device cannot be used until exclusive access is obtained.
When successful, **Claimed** is changed to TRUE.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Exclusive access has been granted. **Claimed** is now TRUE. It is also returned if this application has already gained the exclusive access to the device.
OPOS_E_ILLEGAL(106)	An invalid *Timeout* is specified.
OPOS_E_TIMEOUT(112)	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 a processable state before the *Timeout* (in milliseconds) has elapsed.

ClearOutput Method

Syntax **LONG ClearOutput ();**

Remarks Call to clear all buffered output data. Any output error events that were queued - usually waiting for that **FreezeEvents** to be set to FALSE - are also cleared.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Output has been cleared.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_TIMEOUT(112)	Data transmission timeout or data receive timeout has occurred.
OPOS_E_EXTENDED(114)	Except for OPOS_EPTR_TOOBIG(206) and OPOS_EPTR_BADFORMAT(207), the error defined by **ResultCodeExtended** is notified.

Close Method

Syntax **LONG Close ();**

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

Return Either of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Device has been disabled and closed.
OPOS_E_BUSY(113)	Asynchronous output is in progress.

CompareFirmwareVersion Method

Syntax **LONG CompareFirmwareVersion (BSTR** *FirmwareFileName***, Long** *result***);**

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

DirectIO Method

Syntax **LONG DirectIO (LONG** *Command***, LONG*** *pData***, BSTR*** *pString***);**

Parameter	Description
Command	Command number. Specific values assigned by the Service Object.
pData	Pointer to additional numeric data. Specific values vary by *Command* and Service Object.
pString	Pointer to additional string data. Specific values vary by *Command* and Service Object. The format of this data depends on the value of **BinaryConversion**. See **BinaryConversion** for details.

Remarks The following functions are supported.
- Remaining memory capacity response
- International character selection
- Status response

- **Remaining memory capacity response**

  Issues the printer command "Send NV Graphics Memory Remaining Capacity", and returns its response as a numeric value. Asynchronous processing cannot be executed.
  When **AsyncMode** is TRUE and **State** is OPOS_S_IDLE(2), this method is executed synchronously.

Parameter	Description
Command	PTR_DI_GET_REMAINING_MEMORY (3)
pData	Not used
pString	OUT Remaining memory

- **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.
  When **AsyncMode** is TRUE and **State** is OPOS_S_IDLE(2), this method is executed synchronously.

Parameter	Description
Command	PTR_DI_SET_INTERNATIONAL_CHARACTER(201)
pData	IN International character number n 0 ≤ n ≤ 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
pString	Not used

- **Status response**

  Returns the paper sensor status.
  When **AsyncMode** is TRUE and **State** is OPOS_S_IDLE(2), this method is executed synchronously.

Parameter	Description
Command	PTR_DI_GET_STATUS_DATA(501)
pData	IN Status type 1: Status response of paper sensor
pString	OUT Status '0': Paper is ready '1': No paper

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	**DirectIO** is successfully completed.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	Parameter has an error or an invalid command number is specified.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_FAILURE(111)	A communication error has occurred.
OPOS_E_TIMEOUT(112)	Data transmission timeout or data response timeout has occurred.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress or an error occurs.
OPOS_E_EXTENDED(114)	Except for OPOS_EPTR_TOOBIG(206) and OPOS_EPTR_BADFORMAT(207), the error defined by **ResultCodeExtended** is notified.

Open Method

Syntax **LONG Open (BSTR** *DeviceName***);**

DeviceName specifies the device name to open. Specify the registered device name (such as "MP-B21L") or the logical device name of the printer to execute this method.

Remarks Call this method to open the device.
When **Open** is successful, the common properties and other class-specific properties are initialized.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Successful open.
OPOS_E_NOSERVICE(104)	Could not establish a connection to the corresponding Service Object.
OPOS_E_ILLEGAL(106)	The Control is already open.
OPOS_E_NOEXIST(109)	The specified *DeviceName* is not found.
OPOS_E_FAILURE(111)	Initialization of the OPOS Driver is failed.

Note The value of **ResultCode** after calling **Open** may not be the same as **Open** return value for the following two cases.

When the OPOS Control is closed and **Open** fails:
- **ResultCode** will continue to return OPOS_E_CLOSED(101).

When the OPOS Control is already open:
- **Open** will return OPOS_E_ILLEGAL(106), but **ResultCode** may continue to return the value held before **Open**.

ReleaseDevice Method

Syntax **LONG ReleaseDevice ();**

Remarks Call this method to release 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 method while the event is in progress (or in the event handler).

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Exclusive access has been released. **Claimed** is now FALSE.
OPOS_E_ILLEGAL(106)	The application does not have exclusive access to the device.
OPOS_E_BUSY(113)	Asynchronous output is in progress.

ResetStatistics Method

Syntax **LONG ResetStatistics (BSTR** *StatisticsBuffer*);

Parameter	Description
StatisticsBuffer	This is the data buffer defining the statistics to be reset.

This is a comma-separated list of name(s), where an empty string means ALL resettable statistics are to be reset, "U_" means all UnifiedPOS defined resettable statistics are to be reset, "M_" means all SII defined resettable statistics are to be reset, and "actual_name1, actual_name2" (from the XML file definitions) means that the specifically defined resettable statistics are to be reset.

Remarks Resets the resettable statistics defined in a device.

Resettable statistics are as follows.

XML Definition Name	Description
ReceiptLineFeedCount	Number of receipt line feeds performed (unit: 100 dot-line)
HoursPoweredCount	Number of hours powered on (unit: hour)

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Resetting of all the specified statistics is successful.
OPOS_E_ILLEGAL(106)	The number of the statistics specified for the parameter in the CSV format exceeds the number of the resettable statistics.
OPOS_E_EXTENDED(114)	**ResultCodeExtended**=OPOS_ESTATS_ERROR (280) One or more of the specified statistics could not be reset.

RetrieveStatistics Method

Syntax **LONG RetrieveStatistics (BSTR** ***pStatisticsBuffer*****);**

Parameter	Description
pStatisticsBuffer	This is the data buffer defining the statistics to be retrieved and where the statistics to be notified are placed.

This is a comma-separated list of name(s), where an empty string means ALL statistics are to be retrieved, "U_" means all UnifiedPOS defined statistics are to be retrieved, "M_" means all SII defined statistics are to be retrieved, and "actual_name1, actual_name2" (from the XML file definitions) means that the specifically defined statistics are to be retrieved.

Remarks Retrieves the requested statistics from a device. **CapStatisticsReporting** must be TRUE in order to successfully use this method. This method is always executed synchronously.

All calls to **RetrieveStatistics** will return the following XML as a minimum.

```
<?xml version='1.0' ?>
<UPOSStat version="1.14.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.nrf-arts.org/IXRetail/namespace/"
xsi:schemaLocation="http://www.nrf-arts.org/IXRetail/namespace/UPOSStat.xsd">
  <Event>
    <Parameter>
      <Name>RequestedStatistic</Name>
      <Value>StatisticValue</Value>
    </Parameter>
  </Event>
  <Equipment>
    <UnifiedPOSVersion>Version</UnifiedPOSVersion>
    <DeviceCategory UPOS="DeviceCategoryName"/>
    <ManufacturerName>Device Manufacturer</ManufacturerName>
    <ModelName>Device Model Name</ModelName>
  </Equipment>
</UPOSStat>
```

If the application requests a statistic name which is not supported by the device, the <Parameter> entry will be returned with an empty <Value>. For example,

```
    <Parameter>
      <Name>RequestedStatistic</Name>
      <Value></Value>
    </Parameter>
```

All SII-specific statistics (not defined in the schema) which are collected by the device will be returned in a <ManufacturerSpecific> tag instead of a <Parameter> tag.

```
    <ManufacturerSpecific>
      <Name>TheAnswer</Name>
      <Value>42</Value>
    </ManufacturerSpecific>
```

When an application requests all statistics from the device, the device will return a <Parameter> entry for every defined statistic for the device category as defined by the XML schema version specified by the version attribute in the <UPOSStat> tag. If the device does not record any of the statistics, the corresponding <Value> tag will be empty.

The available statistics are as follows.

XML Definition Name	Description
UnifiedPOSVersion	Version of the UnifiedPOS specification supported
DeviceCategory	Device category (e.g., POSPrinter)
ManufacturerName	Device manufacturer's name
ModelName	Device model name
SerialNumber	Device serial number
ManufactureDate	Device manufacture date
MechanicalRevision	Device hardware revision
FirmwareRevision	Device firmware revision
Interface	Device hardware interface (e.g., serial, USB)
InstallationDate	Device installation date
HoursPoweredCount	Number of hours powered on (unit: hour)

XML Definition Name	Description
CommunicationErrorCount	Number of communication errors
BarcodePrintedCount	Number of barcodes printed
FormInsertionCount	Number of forms inserted into the document/slip station
HomeErrorCount	Number of home errors
JournalCharacterPrintedCount	Number of journal characters printed
JournalLinePrintedCount	Number of journal lines printed (unit: line)
MaximumTempReachedCount	Number of times maximum temperature reached
NVRAMWriteCount	Number of times NVRAM is written to
PrinterFaultCount	Number of printer faults
PrintSideChangeCount	Number of print side changes (check flips) performed
FailedPrintSideChangeCount	Number of print side changes (check flips) failures
ReceiptCharacterPrintedCount	Number of receipt characters printed
ReceiptLinePrintedCount	Number of receipt lines printed (unit: line)
ReceiptLineFeedCount	Number of receipt line feeds performed (unit: 100 dot-line)
ReceiptCoverOpenCount	Number of times the receipt cover was opened
SlipCharacterPrintedCount	Number of document/slip characters printed
SlipLinePrintedCount	Number of document/slip lines printed (unit: line)
SlipLineFeedCount	Number of document/slip line feeds performed (unit: line)
SlipCoverOpenCount	Number of times the document/slip station cover opened
StampFiredCount	Number of stamps fired
HoursPoweredCount_Accumulated	Number of hours powered on (unit: hour) (accumulated)
ReceiptLineFeedCount_Accumulated	Number of receipt line feeds performed (unit: 100 dot-line) (accumulated)

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	Specified statistics are successfully gained.
OPOS_E_ILLEGAL(106)	All the statistics of the specified name are undefined.
OPOS_E_EXTENDED(114)	**ResultCodeExtended**=OPOS_ESTATS_ERROR (280) One or more of the specified statistics could not be obtained.
Other values	Refer to the description of **ResultCode**.

UpdateFirmware Method

Syntax **LONG UpdateFirmware (BSTR** *FirmwareFilaName*);

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

UpdateStatistics Method

Syntax **LONG UpdateStatistics (BSTR** *StatisticsBuffer*);

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

5.6 Specific Methods

BeginInsertion Method

Syntax **LONG BeginInsertion (LONG** *Timeout*);

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

BeginRemoval Method

Syntax **LONG BeginRemoval (LONG** *Timeout*);

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

ChangePrintSide Method

Syntax **LONG ChangePrintSide (LONG** *Side*);

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

ClearPrintArea Method

Syntax **LONG ClearPrintArea ();**

Remarks Clears the print data on the Page Mode print area defined by **PageModePrintArea**.
The entire page may be cleared by setting **PageModePrintArea** to be the same as **PageModeArea** and then by using **ClearPrintArea**.
Specify PTR_S_RECEIPT(2) for **PageModeStation** before calling this method.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS (0)	The method was successful.
Other values	Refer to the description of **ResultCode**.

CutPaper Method

Syntax **LONG CutPaper (LONG** *Percentage*);

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

DrawRuledLine Method

Syntax **LONG DrawRuledLine (LONG** *Station*, **BSTR** *PositionList*, **LONG** *LineDirection*, **LONG** *LineWidth*, **LONG** *LineStyle*, **LONG** *LineColor*);

Parameter	Description
Station	Station to be used. Specify PTR_S_RECEIPT(2).
PositionList	Specifies the position of ruled line.
LineDirection	Specifies the direction of ruled line. Available values are PTR_RL_HORIZONTAL(1) (horizontal ruled line) or PTR_RL_VERTICAL(2) (vertical ruled line).
LineWidth	Specifies the width of ruled line. Specifies a value in the valid range (1 to 255) in dots.
LineStyle	Specifies the type of ruled line. Only solid line PTR_LS_SINGLE_SOLID_LINE(1) is supported for ruled line type.
LineColor	Specifies the color of ruled line. Only black PTR_COLOR_PRIMARY(0x00000001) is supported for ruled line color.

Remarks Draws ruled lines.
This method is executed synchronously if **AsyncMode** is FALSE, and asynchronously if **AsyncMode** is TRUE.
How to specify *PositionList* string differs depending on whether *LineDirection* is set to horizontal ruled line or vertical ruled line.

For horizontal ruled line:
Specify a horizontal ruled line with a pair of ASCII numbers separated by a comma, indicating the drawing start position and length. You can specify multiple lines with multiple pairs separated by ASCII semicolon (";"). The drawing start position and length can be specified in the unit defined by **MapMode**.

These are examples of the *PositionList*.

- "0,300"
  This setting draws a horizontal ruled line 300 dots long starting at the 0th dot from the left.
- "0,200;300,50"
  This setting draws a horizontal ruled line 200 dots long starting at the 0th dot from the left.
  In addition, it draws a horizontal ruled line 50 dots long starting at the 300th dot from the left.

For vertical ruled line:
Specify ASCII numbers separated by a comma, indicating the drawing positions of vertical ruled lines. The drawing position can be specified in the unit defined by **MapMode**.
These are examples of the *PositionList* parameter.

- "0,100,200,300"
  This setting draws vertical ruled lines at the 0th, 100th, 200th, and 300th dot from the left.
  The valid range of the drawing position of vertical ruled line is 0 to (**RecLineWidth** value - *LineWidth* value). If the specified value exceeds this valid range, an error is returned. Vertical ruled lines are drawn at the subsequent printing operation.
  To end the drawing of vertical ruled lines, specify an empty string ("") for *PositionList*.

The drawing position is relative to the top left.

DrawRuledLine supports the batch print mode, but **RotatePrint** and **PageModePrint** do not.

When **ClearOutput** is executed, the specification of vertical ruled line ends.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	One of the following parameter errors occurred. - The specified parameter is improper. - The print mode is improper.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_TIMEOUT(112)	Data transmission timeout has occurred.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress or an error occurs. (**State** is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if **AsyncMode** is FALSE, and to OPOS_S_ERROR(4) if **AsyncMode** is TRUE.)
OPOS_E_EXTENDED(114)	Except for OPOS_EPTR_TOOBIG(206) and OPOS_EPTR_BADFORMAT(207), the error defined by **ResultCodeExtended** is notified.

EndInsertion Method

Syntax **LONG EndInsertion ();**

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

EndRemoval Method

Syntax **LONG EndRemoval ();**

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

MarkFeed Method

Syntax **LONG MarkFeed (LONG** *Type***);**

The values of *Type* are as follows.

Value	Meaning
PTR_MF_TO_TAKEUP(1)	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).
PTR_MF_TO_CUTTER(2)	After detecting the mark or the gap, feeds the paper to the cutting position. (Feeds the paper to the same position as PTR_MF_TO_TAKEUP(1)).
PTR_MF_TO_CUTTERTOF(4)	Supported only by MP-B21L. Feeds the marked paper backward to the printing position. When feeding the marked paper to the cut position by PTR_MF_TO_CUTTER(2) and feeding backward by PTR_MF_TO_CUTTERTOF(4) 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.
PTR_MF_TO_NEXT_TOF(8)	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 Printing 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).

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

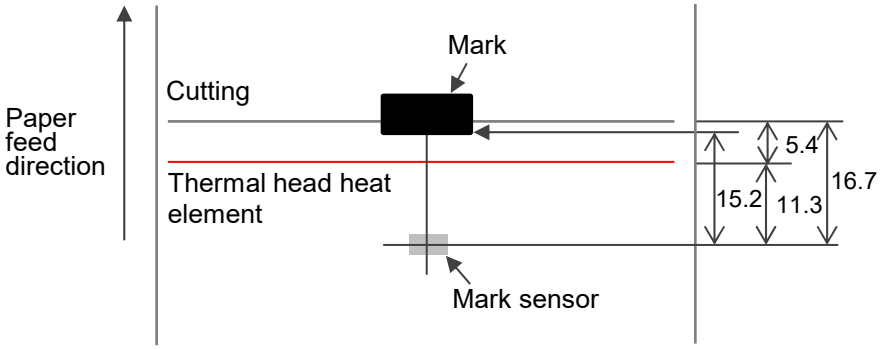
This method is executed synchronously if **AsyncMode** is FALSE, and asynchronously if **AsyncMode** is TRUE.

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 Mobile Printer Software for Windows".

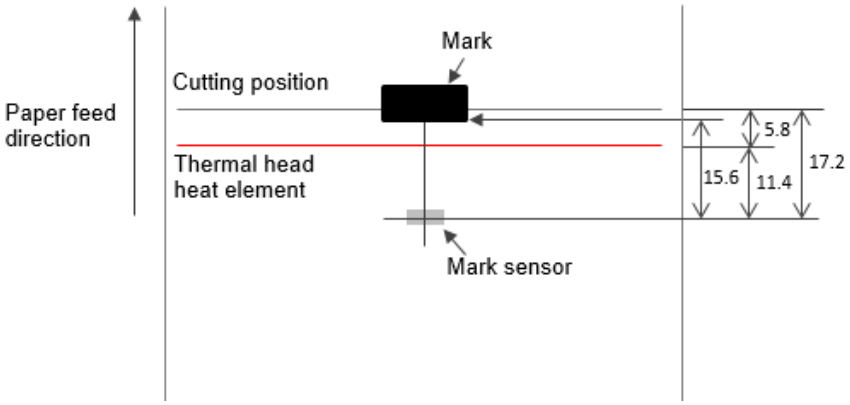
The relation between the sensor position and the defaults of memory switches MS 21 to 22 (Mark Detection Cut Position Correction) of the printer and memory switches MS 26 to 27 (Mark Detection Print Position Correction) of the printer are shown in the following figure.
When the memory switch of the printer is set to the default, the feed position of PTR_MF_TO_TAKEUP(1), the cutting position of PTR_MF_TO_CUTTER(2) and the next printing position of PTR_MF_TO_NEXT_TOF(8) is the same paper feed length.

MP-B21L



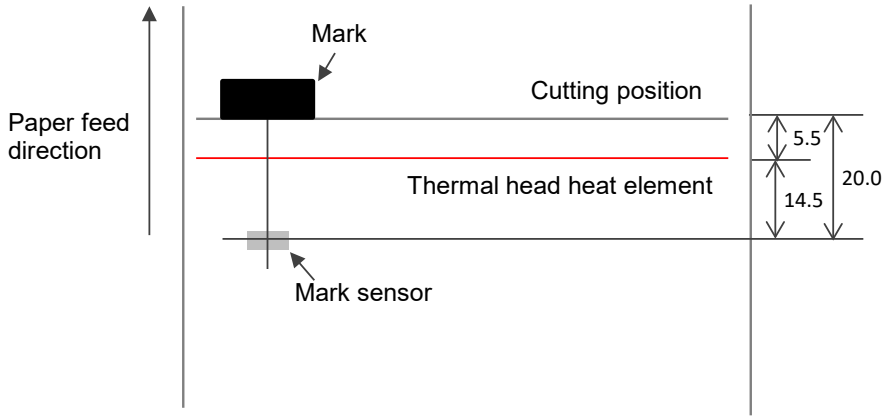
Unit : mm

MP-B30L



Unit : mm

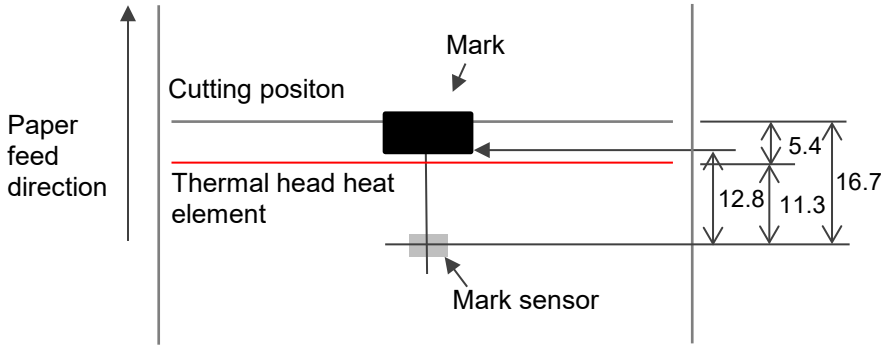
MP-B30



Unit : mm

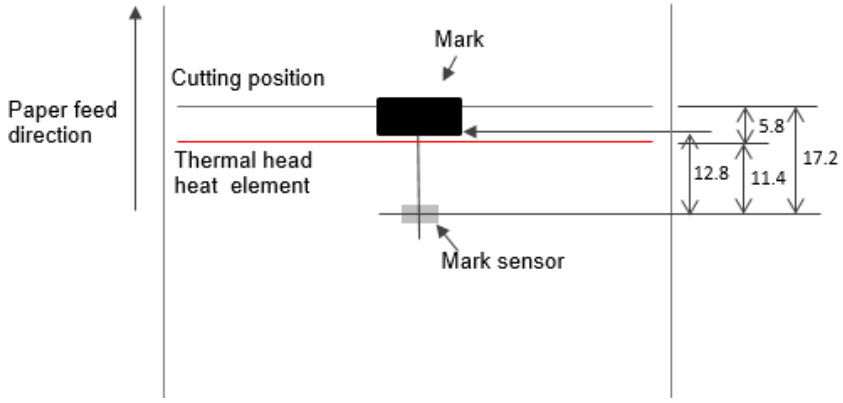
To set to shorter the next print position for the mark to save paper, set the values of the memory switches MS 26 to 27 (Mark Detection Print Position Correction) of the printer shorter.
As an example, the relation of the sensor position when the values of the memory switches MS 26 to 27 of the printer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

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_ILLEGAL(106)	Station is not exist or the value of **CapRecMarkFeed** is 0.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress or an error occurs. (**State** is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if **AsyncMode** is FALSE, and to OPOS_S_ERROR(4) if **AsyncMode** is TRUE.)
OPOS_E_EXTENDED(114)	The error defining in **ResultCodeExtended** excepting OPOS_EPTR_TOOBIG(206) and OPOS_EPTR_BADFORMAT(207) is notified.
Other Values	See **ResultCode**.

PageModePrint Method

Syntax **LONG PageModePrint (LONG** *Control***);**

The values of the *Control* are as follows.

Value	Meaning
PTR_PM_PAGE_MODE(1)	Enters Page Mode.
PTR_PM_PRINT_SAVE(2)	Prints the print data of the Page Mode print area and saves the data. This is used for repeated printings.
PTR_PM_NORMAL(3)	Prints the print area and destroys the canvas and exits Page Mode.
PTR_PM_CANCEL(4)	Clears the page and exits Page Mode without any printing of any print area.

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

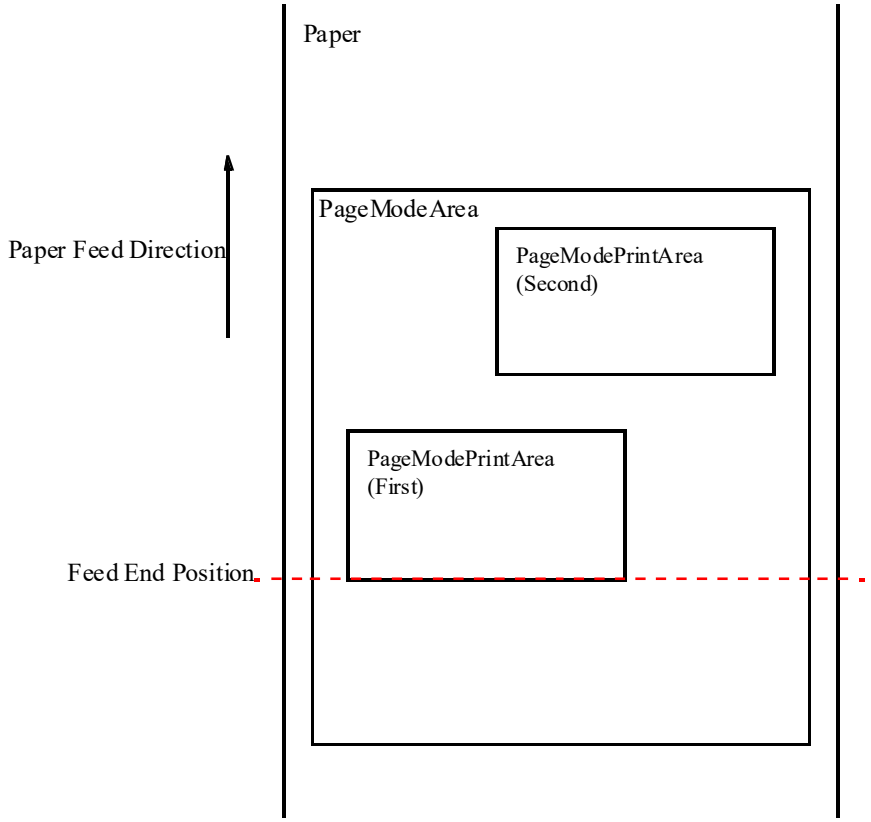
If PTR_PM_PAGE_MODE(1) is specified for *Control*, then Page Mode is started. Subsequently, the print data can be buffered with **PrintNormal**, **PrintBarCode**, **PrintBitmap**, or **PrintMemoryBitmap** until **PageModePrint** is called by specifying PTR_PM_PRINT_SAVE(2), PTR_PM_NORMAL(3), or PTR_PM_CANCEL(4). Methods called during this time only buffer the print data and they do not start printing. Also, the value of **AsyncMode** does not affect the Page Mode function. No **OutputID** will be assigned and no **OutputCompleteEvent** will be notified for each operation.

If PTR_PM_PRINT_SAVE(2) 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 PTR_PM_NORMAL(3) 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 PTR_PM_CANCEL(4) 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 after specifying TR_PM_PRINT_SAVE(2) or PTR_PM_NORMAL(3) 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) is printed, and the paper is fed to the end of page of the Page Mode print area located the farthest "down" the sheet of paper. (See figure below.)



The entire Page Mode area is processed in a batch as one transaction. 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.
The Page Mode function can be used within a transaction print, but not within a rotate print. Specify PTR_S_RECEIPT(2) for **PageModeStation** before calling this method.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Description
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	The station specified by **PageModeStation** does not exist, or **CapRecPageMode** is FALSE. Or, while the station specified by **PageModeStation** is in the state before the transition to the Page Mode, PTR_PM_NORMAL(3), PTR_PM_PRINT_SAVE(4), or PTR_PM_CANCEL(4) is specified for *Control*.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.

Value	Description
OPOS_E_BUSY(113)	Cannot be executed while output is in progress or an error occurs. (It is notified only when **AsyncMode** is FALSE, and PTR_PM_NORMAL(3), PTR_PM_PRINT_SAVE(2), or PTR_PM_CANCEL(4) is specified to *Control*.)
OPOS_E_EXTENDED(114)	Except for OPOS_EPTR_TOOBIG(206) and OPOS_EPTR_BADFORMAT(207), the error defined by **ResultCodeExtended** is notified.
Other values	Refer to the description of **ResultCode**.

PrintBarCode Method

Syntax **LONG PrintBarCode (LONG** *Station*, **BSTR** *Data*, **LONG** *Symbology*, **LONG** *Height*, **LONG** *Width*, **LONG** *Alignment*, **LONG** *TextPosition*);

Parameter	Description
Station	Station to be used. Specify PTR_S_RECEIPT(2).
Data	Character string of barcode. The format of this data depends on the value of **BinaryConversion**. See **BinaryConversion** for details.
Symbology	Barcode type to be used. See the table of *Symbology*.
Height	Barcode height. Expressed in the unit given by **MapMode**. For PTR_MM_DOTS(1), specify the value 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	Description
Width	Barcode width. Expressed in the unit given by **MapMode**. The width of the barcode actually printed is the best fit within the width specified by *Width*. For PTR_MM_DOTS(1), specify the value 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	Position of the barcode. See the table of *Alignment*.
TextPosition	Position of the character string. See the table of *TextPosition*.

*1: Not supported in MP-B30.

- Values of *Symbology*

Value	Meaning
PTR_BCS_UPCA(101)	UPC-A
PTR_BCS_UPCE(102)	UPC-E
PTR_BCS_EAN8(103)	EAN8 (JAN8)
PTR_BCS_JAN8(103)	JAN8 (EAN8)
PTR_BCS_EAN13(104)	EAN13 (JAN13)
PTR_BCS_JAN13(104)	JAN13 (EAN13)
PTR_BCS_EAN13_S(119)	EAN13 (JAN13) with supplemental barcode
PTR_BCS_ITF(106)	Interleaved 2 of 5
PTR_BCS_Codabar(107)	Codabar (NW-7)
PTR_BCS_Code39(108)	Code39
PTR_BCS_Code93(109)	Code93
PTR_BCS_Code128(110)	Code128
PTR_BCS_Code128_Parsed(123)	Code128 Parsed
PTR_BCS_GS1DATABAR(131)	GS1 Databar Omni-directional
PTR_BCS_GS1DATABAR_E(132)	GS1 Databar Expanded
PTR_BCS_GS1DATABAR_S(133)	GS1 Databar Stacked Omni-directional
PTR_BCS_GS1DATABAR_E_S(134)	GS1 Databar Expanded Stacked
PTR_BCS_PDF417(201)	PDF417
PTR_BCS_QRCODE(204)	QR Code (Mixed mode)
PTR_BCS_AZTEC(206)	Aztec
PTR_BCS_OTHER + 6(507)	GS1 Databar Limited
PTR_BCS_OTHER + 7(508)	GS1 Databar Stacked
PTR_BCS_OTHER + 10(511)	GS1 Databar Truncated

- Values of *Alignment*

Value	Meaning
PTR_BC_LEFT(-1)	Align with the left-most print column.
PTR_BC_CENTER(-2)	Align in the center of the station.
PTR_BC_RIGHT(-3)	Align with the right-most print column.
Other value	Distance from the left-most print column to the start of the barcode. Expressed in the unit of measure 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
PTR_BC_TEXT_NONE(-11)	No text is printed. Only prints the barcode.
PTR_BC_TEXT_ABOVE(-12)	Prints the text above the barcode.
PTR_BC_TEXT_BELOW(-13)	Prints the text below the barcode.

Remarks Call this method in order to print the barcode at 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, if PTR_BC_TEXT_BELOW is specified, the text is placed under the barcode, and 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 the parameters of each barcode are described as follows when **MapMode** is PTR_MM_DOTS(1).

[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	*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	*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 abbreviated code is actually printed as UPC-E. If the specified UPC-A initial character is other than 0 or a specified character is not included in the following list, OPOS_E_ILLEGAL(106) is returned.

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

[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	*Width* = 67 × X 134 ≤ *Width* ≤ 402 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	*Width* = 95 × X 190 ≤ *Width* ≤ 570 X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *Width.*

[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	- For 14 or 15 letters *Width* = 122 × X 244 ≤ *Width* ≤ 732 - For 17 or 18 letters *Width* = 149 × X 298 ≤ *Width* ≤ 894 X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *Width.*

[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	*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*.

[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 '/'.
Width	*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 'A' to 'D', '+', '.', '/', ':' included in the 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*.

[Code39]

Parameter	Limitation
Data	At least one of '0' to '9', 'A' to 'Z', space, '$', '%', '+', '-', '.', '/' must be specified.
Width	*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	*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*.

When printing Code93, to specify all the configurable data parameter, the value of **BinaryConversion** must be specified with OPOS_BC_NIBBLE(1) or OPOS_BC_DECIMAL(2). This is because the data includes the symbology character. Regarding the number specified for *Data*, 0x00 represents 0 and 0x01 represents 1.

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.

[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	*Width* = X × ((D + 2) × 11 + 2) 22 × D + 48 ≤ *Width* ≤ 66 × D + 144 D: the number of barcode character (including start code) X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *Width*.

When printing Code128, to specify all the configurable data parameter, the value of **BinaryConversion** must be specified with OPOS_BC_NIBBLE(1) or OPOS_BC_DECIMAL(2). This is because the data includes the symbology character. Regarding the number specified for *Data*, 0x00 represents 0 and 0x01 represents 1.

- 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
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

Number	Character			Number	Character		
	Code A	Code B	Code C		Code A	Code B	Code C
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
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 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	*Width* = X × ((D + 2) × 11 + 2) 1[*1] ≤ *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
"{{"	'{'

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.

Input example of *Data*

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"

[GS1 Databar Omni-directional]

Parameter	Limitation
Data	Specify 13 letters consisting of '0' to '9'.
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.

[GS1 Databar Expanded]

Parameter	Limitation
Data	Specify with 2 or more letters of '0' to '9', 'A' to 'Z', 'a' to 'z', space, '! ', '"', '%', '&', '', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '>', '=', '?', '_' 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	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	Specify with 2 or more letters of '0' to '9', 'A' to 'Z', 'a' to 'z', space, '! ', '"', '%', '&', '', '(', ')', '*', '+', ',', '-', '.', '/', ':', ';', '<', '>', '=', '?', '_'. 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.

[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.
Height *Width*	*Width* = (17 × C + 69) × X *Height* = R × Y X: module width Y: module height R: the number of rows C: the number of vertical columns 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 the *Width* and *Height* parameters 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.

[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) × X 42 ≤ *Width* V: version of QR Code (1 to 40) X: module width and height (2 to 16) For version, the smallest value that input data can be converted to barcode is selected. Also, for module width and height, the maximum size that does not exceed the *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, OPOS_E_ILLEGAL(106) is notified.

[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 When data to send cannot be specified as the character string, specify the value of **BinaryConversion** with OPOS_BC_NIBBLE(1) or OPOS_BC_DECIMAL(2), and input data with the specified format. 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	*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	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	*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.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	One of the following parameter errors occurred. - *Station* does not exist. - *Station* does not support barcode printing. - *Height* or *Width* is 0 or too big. - This *Symbology* is not supported. - The value of *Alignment* is improper. - The value of *TextPosition* is improper. - The specified data is not printable. - The data stored in the specified *Width* cannot be printed.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_TIMEOUT(112)	Data transmission timeout has occurred.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress or an error occurs. (The **State** is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if **AsyncMode** is FALSE, and to OPOS_S_ERROR(4) if **AsyncMode** is TRUE.)
OPOS_E_EXTENDED(114)	Except for OPOS_EPTR_TOOBIG(206) and OPOS_EPTR_BADFORMAT(207), the error defined by **ResultCodeExtended** is notified.

PrintBitmap Method

Syntax **LONG PrintBitmap (LONG** *Station*, **BSTR** *FileName*, **LONG** *Width*, **LONG** *Alignment*);

Parameter	Description
Station	Station to be used. Specify PTR_S_RECEIPT(2).
FileName	Name of bitmap file. For the supported image file, see below.
Width	Print width of bitmap. See values below.
Alignment	Print position of bitmap. See values below.

- Supported bitmap file

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

- Values of *Width*

Value	Meaning
PTR_BM_ASIS(-11)	Prints the bitmap with one bitmap pixel per printer dot.
Other values	Bitmap width. Expressed in the unit given by **MapMode**. If **MapMode** is PTR_MM_DOTS(1), specify between 1 and **RecLineWidth**. When printing bitmap in a rotated 90° to right/left mode by **RotatePrint** is executed, specify the value not exceeding the maximum value of the printer (see **PageModeArea** for the maximum value). 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 by this software. Specify the bitmap width within the range so that the converted value does not exceed the print area.

- Values of *Alignment*

Value	Meaning
PTR_BM_LEFT(-1)	Align with the left-most print column.
PTR_BM_CENTER(-2)	Align in the center of the station.
PTR_BM_RIGHT(-3)	Align with the right-most print column.
Other values	Distance from the left-most print column to the start of the bitmap. Expressed in the unit of measure 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.

Remarks Call this method when printing a bitmap on the specified printer.
The highest performance cannot be achieved since the bitmap data is sent to the printer after **PrintBitmap** is called. It is recommended to print the bitmap data using **SetBitmap** and escape sequence.
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 PTR_BM_ASIS(-11), then no transformation is executed. The bitmap is printed with one 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.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.

Value	Meaning
OPOS_E_ILLEGAL(106)	One of the following parameter errors occurred. • *Station* does not exist. • *Station* does not support bitmap printing. • *Width* is too big. • *Alignment* is invalid or too big. When position is specified during normal or upside down mode, the total value including print bitmap width exceeds the range of **RecLineWidth**. • 32-bit colored bitmap is set.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_NOEXIST(109)	The file specified by *FileName* was not found.
OPOS_E_TIMEOUT(112)	Data transmission timeout has occurred.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress or an error occurs. (**State** is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if **AsyncMode** is FALSE, and to OPOS_S_ERROR(4) if **AsyncMode** is TRUE.)
OPOS_E_EXTENDED(114)	The error defined by **ResultCodeExtended** is notified.

PrintImmediate Method

Syntax **LONG PrintImmediate (LONG** *Station*, **BSTR** *Data*);

Parameter	Description
Station	Station to be used. Specify PTR_S_RECEIPT(2).
Data	Characters to be printed. Consists of printable characters, escape sequences, line feeds (LF), and carriage returns (CR). See the following values and meanings of special characters within *Data* for LF and CR. The format of this data depends on the value of **BinaryConversion**. See **BinaryConversion** for details.

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

Value	Meaning
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.

Remarks Call this method to print the data in *Data* on the printer immediately. The data that exceeds the maximum number of characters per line is printed on the next print line. This method tries to print its data immediately without buffering by **TransactionPrint** and **RotatePrint**.
If unprinted data remains in the printer buffer, printing is executed after all the buffered data is printed.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	The specified printer does not exist.
OPOS_E_FAILURE(111)	A communication error has occurred.
OPOS_E_TIMEOUT(112)	Data transmission timeout has occurred.

PrintMemoryBitmap Method

Syntax **LONG PrintMemoryBitmap (LONG** *Station*, **BSTR** *Data*, **LONG** *Type*, **LONG** *Width*, **LONG** *Alignment*);

Parameter	Description
Station	Station to be used. Specify PTR_S_RECEIPT(2).
Data	Byte array holding the bitmap data. For the supported image file, see **PrintBitmap**. The format of this data depends on the value of **BinaryConversion**. See **BinaryConversion** for details.
Type	Bitmap format. Only PTR_BMT_BMP (1) is supported for the format of the data indicated by *Data*.
Width	Print width of bitmap. See values below.
Alignment	Print position of bitmap. See values below.

- Values of *Width*

Value	Meaning
PTR_BM_ASIS(-11)	Prints the bitmap with one bitmap pixel per printer dot.
Other values	Bitmap width. Expressed in the unit given by **MapMode**. If **MapMode** is PTR_MM_DOTS(1), specify between 1 and **RecLineWidth**. When printing bitmap in a rotated 90° to right/left mode by **RotatePrint** is executed, specify the value 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 defined by the **PageModePrintArea** and the **PageModeHorizontalPosition**.

The value is rounded up to a multiple of 8 by this software. Specify the bitmap width within the range so that the converted value does not exceed the print area.

- Values of *Alignment*

Value	Meaning
PTR_BM_LEFT(-1)	Align with the left-most print column.
PTR_BM_CENTER(-2)	Align in the center of the station.
PTR_BM_RIGHT(-3)	Align with the right-most print column.
Other values	Distance from the left-most print column to the start of the bitmap. Expressed in the unit of measure given by **MapMode**.

During rotated 90° right/left mode by **RotatePrint** and **PageModePrint**, the setting of *Alignment* is invalid and the data is always left-justified.

Remarks Call this method when printing a memory-stored bitmap on the specified printer.
For the operation specifications, see **PrintBitmap**.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	One of the following parameter errors occurred. • *Station* does not exist. • *Station* does not support bitmap printing. • *Width* is too big. • *Alignment* is invalid or too big. When position is specified during normal or upside down mode, the total value including print bitmap width exceeds the range of **RecLineWidth**. • 32-bit colored bitmap is set.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_NOEXIST(109)	There is no data in *Data*.
OPOS_E_TIMEOUT(112)	Data transmission timeout has occurred.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress or an error occurs. (**State** is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if **AsyncMode** is FALSE, and to OPOS_S_ERROR(4) if **AsyncMode** is TRUE.)
OPOS_E_EXTENDED(114)	The error defined by **ResultCodeExtended** is notified.

PrintNormal Method

Syntax **LONG PrintNormal (LONG** *Station*, **BSTR** *Data*);

Parameter	Description
Station	Station to be used. Specify PTR_S_RECEIPT(2).
Data	Characters to be printed. Consists of printable characters, escape sequences, line feeds (LF), and carriage returns (CR). See the following values and meanings of special characters within *Data* for LF and CR. The format of this data depends on the value of **BinaryConversion**. See **BinaryConversion** for details.

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

Value	Meaning
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.

Remarks Call this method to print the data in *Data*. The print data that exceeds the maximum number of characters per line is printed on the next print line.
If unprinted data remains 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.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	The specified printer does not exist.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_FAILURE(111)	A communication error has occurred.
OPOS_E_TIMEOUT(112)	Data transmission timeout or data receive timeout has occurred.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress or an error occurs. (**State** is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if **AsyncMode** is FALSE, and to OPOS_S_ERROR(4) if **AsyncMode** is TRUE.)
OPOS_E_EXTENDED(114)	Except for OPOS_EPTR_TOOBIG(206) and OPOS_EPTR_BADFORMAT(207), the error defined by **ResultCodeExtended** is notified.

PrintTwoNormal Method

Syntax **LONG PrintTwoNormal (LONG** *Stations*, **BSTR** *Data1*, **BSTR** *Data2*);

Remarks This method is not supported.

Return The following value is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_E_ILLEGAL(106)	This method is not supported.

RotatePrint Method

Syntax **LONG RotatePrint (LONG** *Station*, **LONG** *Rotation*);

Parameter	Description
Station	Station to be used. Specify PTR_S_RECEIPT(2).
Rotation	Direction of rotation. See values below.

- Values of *Rotation*.

Value	Meaning
PTR_RP_RIGHT90(257)	Starts rotated printing 90° to the right (clockwise).
PTR_RP_LEFT90(258)	Starts rotated printing 90° to the left (counterclockwise).
PTR_RP_ROTATE180(259)	Starts rotated printing 180°, that is, print upside down.
PTR_RP_BARCODE(4096)	Starts rotated barcode printing. This value is OR with one of the above start rotated print values.
PTR_RP_BITMAP(8192)	Starts rotated bitmap printing. This value is OR with one of the above start rotated print values. Rotates the bitmap printed by **PrintBitmap**.
PTR_RP_NORMAL(1)	Ends rotated printing.

Remarks This method is executed synchronously if **AsyncMode** is FALSE, and asynchronously if **AsyncMode** is TRUE.

If *Rotation* is PTR_RP_ROTATE180(259), then upside down print mode starts. Subsequent calls to **PrintNormal** or **PrintImmediate** will print the data upside down until **RotatePrint** is called with the *Rotation* set to PTR_RP_NORMAL(1). Lines are printed in the order that they are sent to the POS Printer Control, with the start of each line justified at the right margin of the printer. If PTR_RP_BARCODE(4096) is set as OR of PTR_RP_ROTATE180(259) in *Rotation*, the barcode printing by **PrintBarCode** makes upside down barcode. Also, if PTR_RP_BITMAP(8192) is set as OR of PTR_RP_ROTATE180(259) in *Rotation*, the bitmap printing by **PrintBitmap** makes upside down bitmap.

- **Caution for rotating 180°**

  If the transaction by **TransactionPrint** is started, the upside down print mode is cleared as well as **ClearOutput**.

  If *Rotation* is PTR_RP_RIGHT90(257) or PTR_RP_LEFT90(258), the sideways print mode starts. Subsequent calls to **PrintNormal** will buffer the print data until **RotatePrint** is called with *Rotation* set to PTR_RP_NORMAL(1). (In this case, the data of the above method is only buffered and not printed. Also, the value of **AsyncMode** does not affect its operation. In other words, no **OutputID** is assigned to the request and no **OutputCompleteEvent** is informed.)
  If PTR_RP_BARCODE (4096) is set in *Rotation*, barcode printing by **PrintBarCode** is buffered. If PTR_RP_BITMAP(8192) is set in *Rotation*, bitmap printing by **PrintBitmap** is buffered.
  If *Rotation* is PTR_RP_NORMAL(1), rotated print mode is exited. If some print data is buffered while the sideways rotated print is in effect, the buffered data is printed.
  Service Object calculates so that the width in the sideways print mode becomes the optimum size. The maximum width is 2400 dots in the sideways print mode. If the print data per line exceeds this range, the width is 2400 dots and non-printed data is printed by feeding to the next print line.

- **Caution for rotating 90°**

  If the bitmap print by "Print Bitmap" escape sequence (ESC|#B) is specified on **PrintNormal** during the sideways print mode, the print data is buffered regardless of *Rotation*.

  If the transaction by **TransactionPrint** is started, the sideways print mode is canceled as well as **ClearOutput**. This is because buffering by **TransactionPrint** is prior to buffering by the **RotatePrint**. In order to perform batch print in the sideways print mode, execute **TransactionPrint**, execute **RotatePrint** to exit the rotated print mode, and then cancel the batch print by **TransactionPrint**.

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

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	The specified printer does not exist.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress or an error occurs. (**State** is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if **AsyncMode** is FALSE, and to OPOS_S_ERROR(4) if **AsyncMode** is TRUE.)
OPOS_E_EXTENDED(114)	Except for OPOS_EPTR_TOOBIG(206) and OPOS_EPTR_BADFORMAT(207), the error defined by **ResultCodeExtended** is notified.

SetBitmap Method

Syntax **LONG SetBitmap (LONG** *BitmapNumber*, **LONG** *Station*, **BSTR** *FileName*, **LONG** *Width*, **LONG** *Alignment*);

Parameter	Description
BitmapNumber	The number to be assigned to this bitmap. The valid bitmap numbers are 1 through 20.
Station	Station to be used. Specify PTR_S_RECEIPT(2).
FileName	Name of bitmap file. If an empty string is set, the bitmap setting is canceled. For the supported image file, see **PrintBitmap**.
Width	Print width of bitmap. See the following values.
Alignment	Print position of bitmap. See **PrintBitmap** for values.

- Values of *Width*

Value	Meaning
PTR_BM_ASIS(-11)	Prints the bitmap with one bitmap pixel per printer dot.
Other values	Bitmap width. Expressed in the unit given by **MapMode**. If **MapMode** is PTR_MM_DOTS(1), specify between 1 and **RecLineWidth**.

Remarks Call this method to save the information about the bitmap to be printed.
The bitmap may then be printed by calling **PrintNormal** or **PrintImmediate** with the "Print Bitmap" escape sequence (ESC|#B) in the print data.
Service Object prepares for printing with downloading bitmap data to the downloaded bit image area and NV graphics area of the printer. When bitmap print is specified by escape sequence, only command to instruct printing is transmitted, so optimum performance can be obtained.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call the **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	One of the following parameter errors occurred. • The value of *BitmapNumber* is improper. • *Station* does not exist. • *Station* does not support bitmap printing. • *Width* is too big. • *Alignment* is invalid or too big. When absolute position is specified, the total including print bitmap width exceeds the range of **RecLineWidth**. • 32-bit colored bitmap is set.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_NOEXIST(109)	The file specified by *FileName* was not found.
OPOS_E_FAILURE(111)	Failed to register bitmap due to the shortage of memory on the printer.

Value	Meaning
OPOS_E_TIMEOUT(112)	Data transmission timeout has occurred or any response from the printer is not made before timeout.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress or an error occurs (This can be set when **State** is OPOS_S_BUSY(3) or OPOS_S_ERROR(4)).
OPOS_E_EXTENDED(114)	The error defined by **ResultCodeExtended** is notified.

SetLogo Method

Syntax **LONG SetLogo (LONG** *Location***, BSTR** *Data***);**

Parameter	Description
Location	Logo to be set. Specify PTR_L_TOP(1) or PTR_L_BOTTOM(2).
Data	Characters that produce the logo. Consists of printable characters, escape sequences, line feeds (LF), and carriage returns (CR). See the following values and meanings of special characters within *Data* for LF and CR. The format of this data depends on the value of **BinaryConversion**. See **BinaryConversion** for details.

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

Value	Meaning
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.

Remarks Call this method to save a data string as the top or bottom logo.
The logo can be printed by calling **PrintNormal** or **PrintImmediate** with "Print Top Logo" escape sequence (ESC|tL) or "Print Bottom Logo" escape sequence (ESC|bL) in the print data.
Data registered by this method holds the data in the character set of **CharacterSet** at the time when the method is executed. Therefore, execute **SetLogo** with **CharacterSet** 932 to register the data including kanji character(s). After that, kanji character(s) will be printed even if **CharacterSet** is set to 999 when the data is printed by escape sequence.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	Improper *Location* is specified.
Other values	Refer to the description of **ResultCode**.

TransactionPrint Method

Syntax **LONG TransactionPrint(LONG** *Station*, **LONG** *Control*);

Parameter	Description
Station	Station to be used. Specify PTR_S_RECEIPT(2).
Control	Transaction control. See values below.

- Values of *Control*

Value	Meaning
PTR_TP_TRANSACTION(11)	Starts a transaction.
PTR_TP_NORMAL(12)	Ends a transaction by printing the buffered data.

Remarks Call this method to enter or exit transaction mode.
If *Control* is PTR_TP_TRANSACTION (11), then transaction mode is entered. Subsequent calls to **PrintNormal**, **CutPaper**, **RotatePrint**, **PrintBarCode**, **PrintBitmap**, **PrintMemoryBitmap**, or **DrawRuledLine** will buffer the print data at the Service Object until **TransactionPrint** is called with the *Control* set to PTR_TP_NORMAL(12). (In this case, the print methods only validate the method parameters and buffer the data, and they do not initiate printing. Also, the value of **AsyncMode** does not affect its operation. In other words, no **OutputID** is assigned to the request and no **OutputCompleteEvent** is informed.)

If *Control* is PTR_TP_NORMAL(12), then transaction mode is exited. If some data was buffered by calls to **PrintNormal**, **CutPaper**, **RotatePrint**, **PrintBarCode**, **PrintBitmap**, **PrintMemoryBitmap**, or **DrawRuledLine**, then the buffered data is printed. 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.
For the combination with **RotatePrint**, refer to **RotatePrint**.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	The specified printer does not exist.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_BUSY(113)	Cannot be executed while output is in progress or an error occurs. (**State** is set to OPOS_S_BUSY(3) or OPOS_S_ERROR(4) if **AsyncMode** is FALSE, and to OPOS_S_ERROR(4) if **AsyncMode** is TRUE.)
OPOS_E_EXTENDED(114)	Except for OPOS_EPTR_TOOBIG(206) and OPOS_EPTR_BADFORMAT(207), the error defined by **ResultCodeExtended** is notified.

ValidateData Method

Syntax **LONG ValidateData(LONG** *Station*, **BSTR** *Data*);

Parameter	Description
Station	Station to be used. Specify PTR_S_RECEIPT(2).
Data	Data to be validated. May include printable data and escape sequences. The format of this data depends on the value of **BinaryConversion**. See **BinaryConversion** for details.

Remarks Before calling **PrintImmediate** or **PrintNormal**, call this method when determining whether a data sequence, possibly including one or more escape sequences, is valid for the specified printer.
This method does not cause any printing but is used to determine the capability of the printer.

Return One of the following values is returned by the method and also placed in **ResultCode**.

Value	Meaning
OPOS_SUCCESS(0)	The data is valid.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **ClaimDevice** to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Call after setting **DeviceEnabled** to TRUE.
OPOS_E_ILLEGAL(106)	At least one of the escape sequences is out of the range. However, the Control can select valid alternatives. Also, this value is stored when the escape sequence is not supported by the Page Mode function or rotated right/left 90° print mode.
OPOS_E_NOHARDWARE(107)	The printer is not connected to the system or is not powered on.
OPOS_E_FAILURE(111)	At least one of the escape sequences is not supported. No alternatives can be selected.

The cases in which OPOS_E_ILLEGAL(106) is notified are as follows.

Escape Sequence	Condition
Feed lines	One of the following statuses occurs. • The line count '#' is not precisely supported. The Control will select the nearest value that is supported. • It is not supported during rotated 90° right/left mode by **RotatePrint**. • It is not supported during Page Mode by **PageModePrint**.

Escape Sequence	Condition
Feed units	• The feed unit count '#' is not precisely supported due to occurrence of rounding error of 1 dot depending on the setting of **MapMode**. • The feed unit count '#' is not precisely supported. The Control will select the nearest value that is supported. • It is not supported during rotated 90° right/left mode by **RotatePrint**. • It is not supported during Page Mode by **PageModePrint**.
Pass through embedded data	The number of bytes of embedded data '#' is not precisely supported.
Print in-line barcode	One of the following statuses occurs. • The number of bytes of barcode data '#' is not precisely supported. • The characters following R and indicating the attribute, or the parameter value are not precisely supported.
Print in-line ruled line	One of the following statuses occurs. • The number of bytes of ruled line data '#' is not precisely supported. • The characters following dL and indicating the attribute, or the parameter value are not precisely supported.
Underline	The thickness '#' is not precisely supported. The Control will select the nearest value that is supported.
Scale vertically	The scale factor '#' is not precisely supported. The Control will select the nearest value that is supported.
Scale horizontally	The scale factor '#' is not precisely supported. The Control will select the nearest value that is supported.
Center	One of the following statuses occurs. • It is not supported during rotated 90° right/left mode by **RotatePrint**. • It is not supported during Page Mode by **PageModePrint**.
Right justify	One of the following statuses occurs. • It is not supported during rotated 90° right/left mode by **RotatePrint**. • It is not supported during Page Mode by **PageModePrint**.
Left justify	One of the following statuses occurs. • It is not supported during rotated 90° right/left mode by **RotatePrint**. • It is not supported during Page Mode by **PageModePrint**.

The cases in which OPOS_E_FAILURE(111) is notified are as follows.

Escape Sequence	Condition
(General)	The format of the escape sequence is invalid.
Paper cut	Not supported.
Feed and Paper cut	Not supported.
Feed, Paper cut, and Stamp	Not supported.
Fire stamp	Not supported.
Print bitmap	The bitmap number '#' is not correct.
Feed reverse	Not supported.
Font typeface	Not supported.
Italic	Not supported.
Alternate color (Custom)	Not supported.
Shading	Not supported.
RGB Color	Not supported.
SubScript	Not supported.
SuperScript	Not supported.
Strike-through	Not supported.

Data	Condition
data1CRdata2LF	(Where CR is a Carriage Return and LF is a Line Feed.) Cannot remain on the same line after the data is printed. The data1 is printed on one line, and the data2 is printed on the next line.

5.7 Events

DirectIOEvent Event

Syntax **void DirectIOEvent(LONG** *EventNumber***, LONG*** *pData***, BSTR*** *pString***);**

Remarks This event is not supported.

ErrorEvent Event

Syntax **void ErrorEvent (LONG** *ResultCode***, LONG** *ResultCodeExtended***, LONG** *ErrorLocus***, LONG*** *pErrorResponse***);**

Parameter	Description
ResultCode	Factor code causing the error event. See **ResultCode** for the values.
ResultCodeExtended	Extended code of the factor causing the error event. See values below.
ErrorLocus	Set to OPOS_EL_OUTPUT(1). The error occurred while processing asynchronous output.
pErrorResponse	Pointer for error event response. See values below.

If *ResultCode* is OPOS_E_EXTENDED(114), then *ResultCodeExtended* is set to one of the following values.

Value	Meaning
OPOS_EPTR_COVER_OPEN(201)	The printer cover is open.
OPOS_EPTR_REC_EMPTY(203)	The receipt is out of paper.
OPOS_EPTR_VPPOWER(1001)	Vp voltage error has occurred.
OPOS_EPTR_HEAD_TEMP(1005)	Head-temperature error has occurred.
OPOS_EPTR_UNRECOVERABLE(1010)	A non-recoverable error has occurred.
OPOS_EPTR_BATTERY(1013)	A battery error has occurred.
OPOS_EPTR_MARK_JAM(1014)	Paper jam error has occurred at the mark or the gap detection.

The content of the location specified by *pErrorResponse* is preset to the default of OPOS_ER_RETRY(11).

The application sets one of the following values.

Value	Meaning
OPOS_ER_RETRY(11)	Retries the asynchronous output. The error state is exited. If the registry "ProcessCompletionTiming" is "0", all data not being transmitted to the printer is printed. If the registry "ProcessCompletionTiming" is "1", all unprinted data is printed.
OPOS_ER_CLEAR(12)	Clears all buffered print data. The error state is exited.

Remarks The event is notified when an error is detected and the POS Printer Control state transitions into the error state.

OutputCompleteEvent Event

Syntax **void OutputCompleteEvent (LONG** *OutputID***);**

OutputID indicates the ID number of the completed asynchronous output request.

Remarks This event is notified when the previously started asynchronous output request has completed successfully.

StatusUpdateEvent Event

Syntax **void StatusUpdateEvent (LONG** *Status***);**

Status is set to one of the following values.

Value	Description
PTR_SUE_COVER_OPEN(11)	Printer cover is open.
PTR_SUE_COVER_OK(12)	Printer cover is closed.
PTR_SUE_REC_EMPTY(24)	Receipt is out of paper.
PTR_SUE_REC_PAPEROK(26)	Receipt is ready.
PTR_SUE_IDLE(1001)	All the asynchronous outputs finished either successfully or by cleared. The POS Printer Control's **State** is now OPOS_S_IDLE(2). **FlagWhenIdle** must be TRUE for this event to be notified. And the POS Printer Control automatically resets the property to FALSE before the event is notified.
OPOS_SUE_POWER_ONLINE(2001)	The device is powered on and ready (This is notified when **PowerNotify** = OPOS_PN_ENABLED(1)).
OPOS_SUE_POWER_OFF_OFFLINE(2004)	The device is powered off or offline (This is notified when **PowerNotify** = OPOS_PN_ENABLED(1)).

Remarks This event is notified when an important state change has occurred in the printer device.

When the Bluetooth model is used, it takes about 30 seconds to notify the OPOS_SUE_POWER_OFF_OFFLINE(2004) after the power condition of the device is either power off or offline.
And it takes about 10 seconds to notify the OPOS_SUE_POWER_ONLINE(2001) after the power condition of the device is on and ready.

Chapter 6 Registry Used by This Software

The registry set by the configuration program is described as follows.

Although the registry values can be set manually, set them using the configuration program. The contents of registry are read at the time when this software executes the **Open**. Therefore, changing the value during operation of this software will not be reflected in the operation. In order to reflect new setting values, call the **Open** after calling the **Close** in this software.

6.1 POS Printer Control

[Registry key]
HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\POSPrinter\DeviceNameKey

The device name key depends on the printer. It is added by the configuration program for each printer device.

Printer	Device Name key	Meaning
MP-B21L	Example: "MP-B21L"	Device name of printer which is added by the configuration program.
MP-B30L	Example: "MP-B30L"	
MP-B30	Example: "MP-B30"	

[Registry value list]

Value Name	Value	Meaning
Default	OPOS.POSPRINTER.SO.SII.CORE.GEN	-
CoreSODLLPath	C:\Program Files\SII\OPOS\SPSWO_Prn.dll	File name of POS Printer Service Object (Full path)
DefaultCharacterSet	932 (Japanese) 999 (English)	Default of **CharacterSet**
Description	SII MP-Bxx POS Printer Service Object, Copyright (C) 20xx Seiko Instruments Inc.	Details of dedicated POS Printer Service Object for printer
DriverName	SII MP-B21L SII MP-B30L SII MP-B30	Printer driver name Used for printer driver setting
PortName	-	Communication port name
Interface	Driver	First time setup interface

Value Name	Value	Meaning
LogRetentionPeriod	1	Retention period of log file 1: 1 day 3: 3 days 10: 10 days 30: 30 days 90: 90 days
LogLevelCore	-1	Log output level of POS Printer Service Object -1: No output 0: Error 4: Trace
LogLevelSub	-1	Log output level of dedicated POS Printer Service Object for MP-Bxx -1: No output 0: Error 4: Trace
MarkMode	0	Mark detection mode 0: Disabled 1: Enabled
PrintWidth	384	Print width and number of effective dots selection MP-B21L 168 : 21mm/168dots 176 : 22mm/176dots 192 : 24mm/192dots 208 : 26mm/208dots 224 : 28mm/224dots 240 : 30mm/240dots 256 : 32mm/256dots 272 : 34mm/272dots 288 : 36mm/288dots 304 : 38mm/304dots 320 : 40mm/320dots 336 : 42mm/336dots 352 : 44mm/352dots 360 : 45mm/360dots 368 : 46mm/368dots 384 : 48mm/384dots

Value Name	Value	Meaning
PrintWidth	384	MP-B30L 360 : 45mm/360dots 368 : 46mm/368dots 384 : 48mm/384dots 400 : 50mm/400dots 416 : 52mm/416dots 432 : 54mm/432dots 448 : 56mm/448dots 464 : 58mm/464dots 480 : 60mm/480dots 496 : 62mm/496dots 512 : 64mm/512dots 528 : 66mm/528dots 544 : 68mm/544dots 560 : 70mm/560dots 576 : 72mm/576dots
PaperSize	3	MP-B30 Receipt width and number of effective dots selection 3: 80mm/576dots
PrintQuality	0	Print Quality Specify 0: Enabled **RecLetterQuality** 1: Standard 2: Quality 1 3: Quality 2
SendTimeout	10000	Send timeout (ms)
ReceiveTimeout	10000	Receive timeout (ms)
ProcessCompletion Timing	1	Process Completion Timing 0: Completion of data transmission 1: Completion of printing
SubSODLLPath	MP-B21L: C:\Program Files\SII\OPOS \SPSWO_Prn_MPB21L.dll MP-B30L: C:\Program Files\SII\OPOS \SPSWO_Prn_MPB30L.dll MP-B30: C:\Program Files\SII\OPOS \SPSWO_Prn_MPB30.dll	File name of dedicated POS Printer Service Object for MP-Bxx (Full path)
Version	1.14.xx	Version of dedicated POS Printer Service Object for MP-Bxx

Chapter 7 Header File

7.1 POS Printer Header File

The header file used in this software is described below.
The constants defined uniquely in this software are as follows.

- OPOS_EPTR_VPPOWER
- OPOS_EPTR_HEAD_TEMP
- OPOS_EPTR_IMAGEAREA_FULL
- OPOS_EPTR_UNRECOVERABLE
- OPOS_EPTR_BATTERY
- OPOS_EPTR_MARK_JAM
- PTR_DI_GET_REMAINING_MEMORY
- PTR_DI_SET_INTERNATIONAL_CHARACTER
- PTR_DI_GET_STATUS_DATA

Header file: siicommptr.h

```
////////////////////////////////////////////////////////////////////
//
// SIICommPtr.h
//
// POS Printer header file for OPOS Applications.
//
// Modification history
// -----------------------------------------------------------------
//
////////////////////////////////////////////////////////////////////
#if   !defined(SIICOMMPTR_H)
#define SIICOMMPTR_H
////////////////////////////////////////////////////////////////////
// OPOS "ResultCodeExtended" Property Base Constants
////////////////////////////////////////////////////////////////////
const LONG PTRERREXT = 1000; // POS Printer specific error base
////////////////////////////////////////////////////////////////////
```

```
// "ResultCodeExtended" Property Constants for Printer
//////////////////////////////////////////////////////////////////
const LONG OPOS_EPTR_VPPOWER            =  PTRERREXT+1;
const LONG OPOS_EPTR_HEAD_TEMP          =  PTRERREXT+5;
const LONG OPOS_EPTR_IMAGEAREA_FULL     =  PTRERREXT+9;
const LONG OPOS_EPTR_UNRECOVERABLE      =  PTRERREXT+10;
const LONG OPOS_EPTR_BATTERY            =  PTRERREXT+13;
const LONG OPOS_EPTR_MARK_JAM           =  PTRERREXT+14;


//////////////////////////////////////////////////////////////////
// Parameter Constants of "DirectIO" Method
//////////////////////////////////////////////////////////////////
const LONG PTR_DI_GET_REMAINING_MEMORY              = 3;
const LONG PTR_DI_SET_INTERNATIONAL_CHARACTER       = 201;
const LONG PTR_DI_GET_STATUS_DATA                   = 501;
#endif   //!defined(SIICOMMPTR_H)
```