SII

RP Series
OPOS Control
Application Programmer's Guide

U00131810510

Seiko Instruments Inc.

U00131810500	November 2013
U00131810501	September 2014
U00131810502	April 2015
U00131810503	October 2016
U00131810504	May 2017
U00131810505	August 2019
U00131810506	November 2019
U00131810507	October 2020
U00131810508	December 2021
U00131810509	October 2022
U00131810510	July 2023

Copyright© 2013-2023 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 registered trademarks 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 document describes the specifications, functions, and operating instructions of the RP series OPOS Control (hereinafter referred to as "the OPOS Control") provided by Seiko Instruments Inc. (hereinafter referred to as "SII") for the RP series POS printers (hereinafter referred to as "printer") and cash drawers that are connected to the printers.

This document basically written on the basis of the following conditions:

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

- **Target Printers**

  This section lists the printers (products) supported by the software.

Printer	Description in This Document
RP-D10	RP-D10
RP-E10	RP-E10
RP-E11	

- **Operating System Abbreviations**

  This section lists the operating system abbreviations used in this document.

Operating System	Abbreviation
General Microsoft® Windows®	Windows
Microsoft® Windows® 11	Windows 11
Microsoft® Windows® 11 IoT Enterprise	
Microsoft® Windows® 10	Windows 10
Microsoft® Windows® 10 IoT Enterprise	

- **Terms**

  This section describes terms used in this document.

Definition	Description
UPOS V1.13	"UnifiedPOS Retail Peripheral Architecture Version 1.13"
OPOS specification	Specification of the POS device interface defined in UPOS V1.13.
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 and cash drawer 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.
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.
Cash Drawer Control	OPOS Control of the Cash Drawer device class. Controlled device is the cash drawer which is connected to the printer.
RP Series OPOS Control	OPOS Control which controls the printer and its connected cash drawer. In this manual, this is used when the POS Printer Control and the Cash Drawer Control are not distinguished.
Printer	In this manual, this term is used to indicate the printer product (printer device).
Cash drawer	Cash drawer which is connected to the printer. In this manual, this term is used to indicate the cash drawer device.
Default	The value immediately after satisfying the availability condition.
Line spacing	In the OPOS Control, this means the height of each single-high print line (total value of the printed line height and the whitespace between each pair of lines).

Table of Contents

1 Overview 1-1

1.1 Scope of This Document 1-1
1.2 Overview of OPOS Control 1-2
1.3 Printer Settings 1-3
1.4 Limitation (POS Printer Control, Cash Drawer Control) 1-4

2 Installation 2-1

2.1 Conditions of Installation 2-1
2.2 Installation Procedure 2-2
2.3 Uninstallation Procedure 2-5
2.3.1 When [Uninstall] is Selected 2-5
2.3.2 When [Modify] is Selected to Uninstall 2-6
2.4 Repair Procedure 2-8
2.5 Configuration Program (OPOS Utility) 2-10
2.5.1 Addition of Device 2-11
2.5.2 Device Setting 2-20
2.5.3 Addition and Deletion of Logical Device Name 2-25
2.5.4 Test of Device 2-29
2.5.5 Deletion of Device 2-32

3 OPOS Control 3-1

3.1 General 3-1
3.2 POS Printer Control 3-1
3.3 Cash Drawer Control 3-2

4 OPOS Printer Control Interface Specification 4-1

4.1 Summary 4-1
4.1.1 Common Properties 4-1
4.1.2 Specific Properties 4-3
4.1.3 Common Methods 4-8
4.1.4 Specific Methods 4-8
4.1.5 Events 4-9
4.2 Data Characters and Escape Sequences 4-10
4.3 Common Properties 4-15
BinaryConversion Property R/W 4-15
CapCompareFirmwareVersion Property 4-16
CapPowerReporting Property 4-16
CapStatisticsReporting Property 4-16
CapUpdateFirmware Property 4-16
CapUpdateStatistics Property 4-17
CheckHealthText Property 4-17
Claimed Property 4-17
ControlObjectDescription Property 4-17
ControlObjectVersion Property 4-17
DeviceDescription Property 4-18
DeviceEnabled Property R/W 4-18
DeviceName Property 4-18
FreezeEvents Property R/W 4-19
OpenResult Property 4-19
OutputID Property 4-20
PowerNotify Property R/W 4-20
PowerState Property 4-21
ResultCode Property 4-21
ResultCodeExtended Property 4-22
ServiceObjectDescription Property 4-23
ServiceObjectVersion Property 4-23
State Property 4-23
4.4 Specific Properties 4-24
AsyncMode Property R/W 4-24
CapCharacterSet Property 4-24
CapCoverSensor Property 4-24
CapMapCharacterSet Property 4-24
CapRec2Color Property 4-25
CapRecBarCode Property 4-25
CapRecBitmap Property 4-25
CapRecBold Property 4-25
CapRecCartridgeSensor Property 4-25
CapRecColor Property 4-25
CapRecDhigh Property 4-26
CapRecDwide Property 4-26

CapRecDwideDhigh Property 4-26
CapRecEmptySensor Property 4-26
CapRecItalic Property 4-26
CapRecLeft90 Property 4-26
CapRecMarkFeed Property 4-27
CapRecNearEndSensor Property 4-27
CapRecPageMode Property 4-27
CapRecPapercut Property 4-27
CapRecPresent Property 4-27
CapRecRight90 Property 4-28
CapRecRotate180 Property 4-28
CapRecRuledLine Property 4-28
CapRecStamp Property 4-28
CapRecUnderline Property 4-28
CapTransaction Property 4-28
CartridgeNotify Property R/W 4-29
CharacterSet Property R/W 4-29
CharacterSetList Property 4-30
CoverOpen Property 4-30
ErrorLevel Property 4-30
ErrorStation Property 4-31
ErrorString Property 4-31
FlagWhenIdle Property R/W 4-32
FontTypefaceList Property 4-32
MapCharacterSet Property R/W 4-32
MapMode Property R/W 4-33
PageModeArea Property 4-34
PageModeDescriptor Property 4-34
PageModeHorizontalPosition Property R/W 4-35
PageModePrintArea Property R/W 4-35
PageModePrintDirection Property R/W 4-36
PageModeStation Property R/W 4-38
PageModeVerticalPosition Property R/W 4-39
RecBarCodeRotationList Property 4-39
RecBitmapRotationList Property 4-40
RecCartridgeState Property 4-40
RecCurrentCartridge Property R/W 4-40
RecEmpty Property 4-40
RecLetterQuality Property R/W 4-41
RecLineChars Property R/W 4-41
RecLineCharsList Property 4-42
RecLineHeight Property R/W 4-43
RecLineSpacing Property R/W 4-43
RecLinesToPaperCut Property 4-44
RecLineWidth Property 4-45
RecNearEnd Property 4-45
RecSidewaysMaxChars Property 4-45

RecSidewaysMaxLines Property 4-46
RotateSpecial Property R/W 4-46
4.5 Common Methods 4-47
CheckHealth Method 4-47
ClaimDevice Method 4-48
ClearOutput Method 4-48
Close Method 4-49
CompareFirmwareVersion Method 4-49
DirectIO Method 4-49
Open Method 4-52
ReleaseDevice Method 4-52
ResetStatistics Method 4-53
RetrieveStatistics Method 4-53
UpdateFirmware Method 4-56
UpdateStatistics Method 4-56
4.6 Specific Methods 4-57
BeginInsertion Method 4-57
BeginRemoval Method 4-57
ChangePrintSide Method 4-57
ClearPrintArea Method 4-57
CutPaper Method 4-58
DrawRuledLine Method 4-59
EndInsertion Method 4-60
EndRemoval Method 4-60
MarkFeed Method 4-61
PageModePrint Method 4-61
PrintBarCode Method 4-63
PrintBitmap Method 4-71
PrintImmediate Method 4-73
PrintMemoryBitmap Method 4-74
PrintNormal Method 4-75
PrintTwoNormal Method 4-76
RotatePrint Method 4-77
SetBitmap Method 4-78
SetLogo Method 4-80
TransactionPrint Method 4-81
ValidateData Method 4-82
4.7 Events 4-85
DirectIOEvent Event 4-85
ErrorEvent Event 4-85
OutputCompleteEvent Event 4-86
StatusUpdateEvent Event 4-86

5 Cash Drawer Control Interface Specification 5-1

5.1 Summary 1
5.1.1 Common Properties 1
5.1.2 Specific Properties 3
5.1.3 Common Methods 3
5.1.4 Specific Methods 3
5.1.5 Events 3
5.2 Common Properties 4
BinaryConversion Property R/W 4
CapCompareFirmwareVersion Property 4
CapPowerReporting Property 4
CapStatisticsReporting Property 5
CapUpdateFirmware Property 5
CapUpdateStatistics Property 5
CheckHealthText Property 5
Claimed Property 6
ControlObjectDescription Property 6
ControlObjectVersion Property 6
DeviceDescription Property 6
DeviceEnabled Property R/W 6
DeviceName Property 7
FreezeEvents Property R/W 7
OpenResult Property 8
PowerNotify Property R/W 8
PowerState Property 9
ResultCode Property 9
ResultCodeExtended Property 10
ServiceObjectDescription Property 10
ServiceObjectVersion Property 10
State Property 11
5.3 Specific Properties 12
CapStatus Property 12
CapStatusMultiDrawerDetect Property 12
DrawerOpened Property 12
5.4 Common Methods 13
CheckHealth Method 13
ClaimDevice Method 13
Close Method 14
CompareFirmwareVersion Method 14
DirectIO Method 15
Open Method 15
ReleaseDevice Method 16
ResetStatistics Method 16
RetrieveStatistics Method 16
UpdateFirmware Method 17

UpdateStatistics Method 17
5.5 Specific Methods 18
OpenDrawer Method 18
WaitForDrawerClose Method 18
5.6 Events 19
DirectIOEvent Event 19
StatusUpdateEvent Event 19

6 Registry Used by OPOS Control 6-1

6.1 POS Printer Control 6-1
6.2 Cash Drawer Control 6-4

7 Header File 7-1

7.1 POS Printer Header File 7-1

1 Overview

The OPOS Control is the OpenPOS Control which conforms to the OPOS Version 1.13 POS Printer device class and Cash Drawer device class which controls the printer and its connected cash drawer. See the UPOS V1.13 for using the OpenPOS Control.

In addition, refer to Appendix A - Section 4: OPOS Application Header Files in UPOS V1.13. for header files used by the control. The additional specific values of the OPOS Control are defined. For these values, refer to "7 Header File" in this manual.

1.1 Scope of This Document

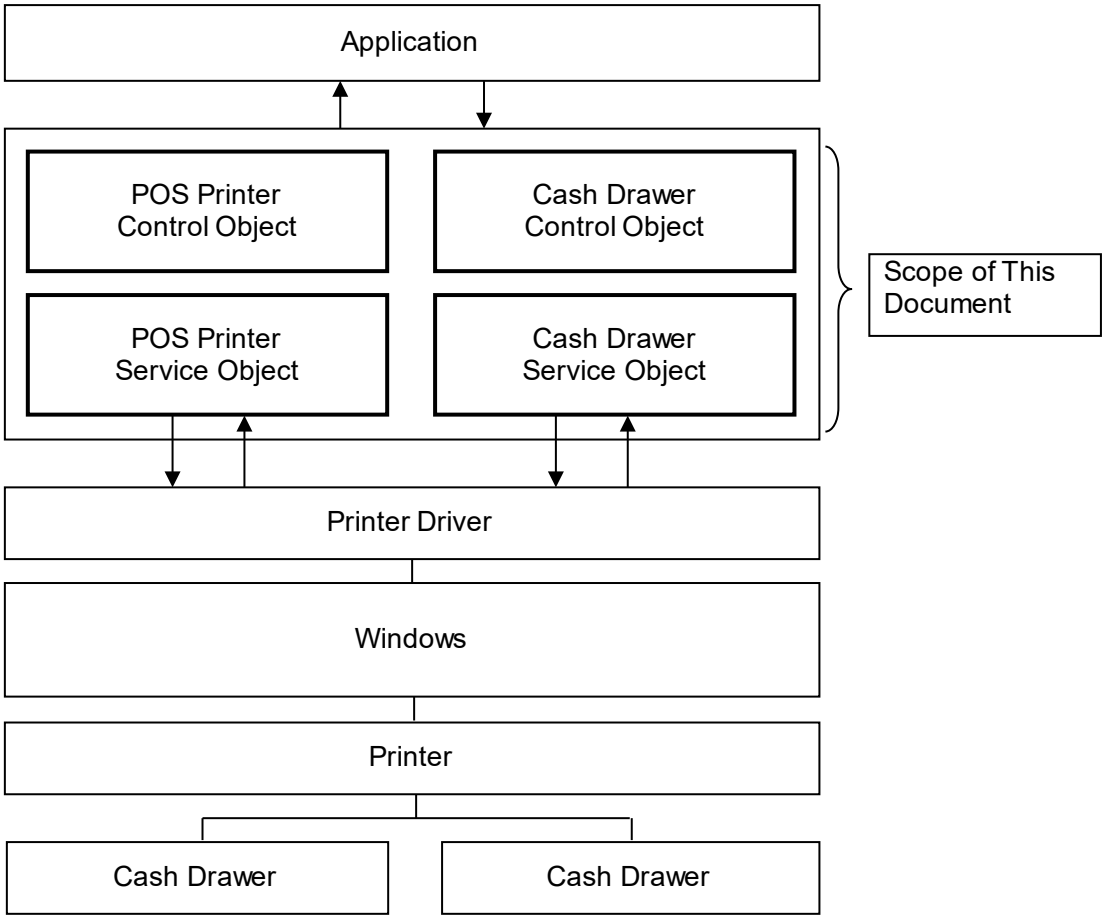
This manual is intended for reference mainly for application developers who use the OPOS Control and describes the following items, which are necessary for them.

- How to install the OPOS Control
- How to use the OPOS Control
- Limitation of the OPOS Control
- Description of the interface (property/method/event) of the OPOS Control
- Description of the setting items of the OPOS Control

1.2 Overview of the OPOS Control

(1) Structural Drawing of OPOS Control

The OPOS Control conforms to the ActiveX Control specifications and provides the properties, methods, and events to applications. The OPOS Control cannot be seen on the UI when an application is executed and only the application that uses it requests the process through methods or properties. Responses are issued to the application through method return values and parameters, properties, and events. The OPOS Control is implemented as an in-process server.



1.3 Printer Settings

The memory switches of the printer are set to [Value] in the following table when using the software.
See "RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" or "RP-E10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details about the memory switches.

- **POS Printer Control**

MS	Function	Value	Note
1-3	Mark Mode Selection*1 (Mark Mode)	0 : Disable	The execution of **ClaimDevice** method sets [Value] on the left forcibly.
1-6	Near-end Sensor Function Selection*1 (Near End Sensor)	0 : Disable 1 : Enable	Either one [Value] on the left can be set by [NearEndSensor] in the configuration program.
4-4	Paper Width Selection (Paper Width)	0 : 58 mm 1 : 80 mm	Either one [Value] on the left can be set by [PaperSize] in the configuration program.
4-5	Number of Effective Dots Selection (Number of Effective Dots)	0 : 360dots/512dots 1 : 432dots/576dots	
4-7	Maximum Print Speed Selection (Print Speed)	00B : Low 01B : Middle (Quality) 10B : Middle (Silent) 11B : High	Any one of [Value] on the left can be set by [PrintSpeed] in the configuration program.
5-1	Automatic Status Response Selection (Auto Status Back)	0 : Enable	The execution of **ClaimDevice** method sets [Value] on the left forcibly.
5-2	Initialized Response Selection (Init. Response)	0 : Enable	
5-3	Data Discard Selection When Error Occurs (Error Through)	1 : Disable	
5-5	Paper-near-end Sensor Error Selection*1 (Near End Error)	1 : Disable	

*1: In the case of RP-E10.

- **Cash Drawer Control**

MS	Function	Value	Note
5-1	Automatic Status Response Selection (Auto Status Back)	0 : Enable	By setting TRUE in **DeviceEnabled** property, [Value] on the left is set forcibly.
5-2	Initialized Response Selection (Init. Response)	0 : Enable	
5-3	Data Discard Selection When Error Occurs (Error Through)	1 : Disable	
13-3	Realtime Command Selection*1 (Realtime Command)	1 : Enable	

*1: In the case of RP-D10. In the case of RP-E10, the printer command "Enable/Disable Realtime Command" is executed as "Enable".

1.4 Limitation (POS Printer Control, Cash Drawer Control)

(1) General

Use a shared printer when all of the following are applied. (If a shared printer is not used, transmission data may interrupt.)

a. When using 1 printer from multiple computers via TCP/IP connections
b. When dividing the transmission data into multiple transmissions

For information on how to install and configure shared printers, see the "RP-D10 Series Printer Driver User's Guide" or "RP-E10 Series Printer Driver User's Guide."

(2) POS Printer Control

All interfaces of POS Printer device class defined by the OPOS specification 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.

- 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** Method
- **EndInsertion** Method
- **BeginRemoval** Method
- **EndRemoval** Method
- **ChangePrintSide** Method
- **MarkFeed** Method
- **PrintTwoNormal** Method

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

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

(f) When an error occurs, hardware reset is transmitted by the POS Printer Control to clear the remaining print data. However, a part of the remaining print data might be printed depending on the timing of the error release and the hardware reset transmission by the POS Printer Control.

(g) When the Bluetooth model is used, if the recoverable error has occurred during printout, it takes time before device is ready for use after recovering from the error.
Wait at least 10 seconds after recovering from the error.

(3) Cash Drawer Control

All interfaces of Cash Drawer device class defined by the OPOS specification are provided, with the following limitations.

(a) All the following methods always return OPOS_E_ILLEGAL(106) after they are enabled.
DirectIO Method
WaitForDrawerClose Method

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

2 Installation

Follow the procedure below to install the OPOS Control.

2.1 Conditions of Installation

■Operating Environment

Item	Specifications
Operating system	Microsoft® Windows® 11 (64 bits) Microsoft® Windows® 11 IoT Enterprise (64 bits) Microsoft® Windows® 10 (32 bits and 64 bits) Microsoft® Windows® 10 IoT Enterprise (32 bits and 64 bits)
Printer driver	In case of RP-D10: RP-D10 series printer driver In case of RP-E10: RP-E10 series printer driver

■Note

If RP-E10 Series OPOS Control (Ver.1.00 to Ver.1.11) is already installed when installing the OPOS Control, please be sure to uninstall RP-E10 Series OPOS Control and then install the OPOS Control.

2.2 Installation Procedure

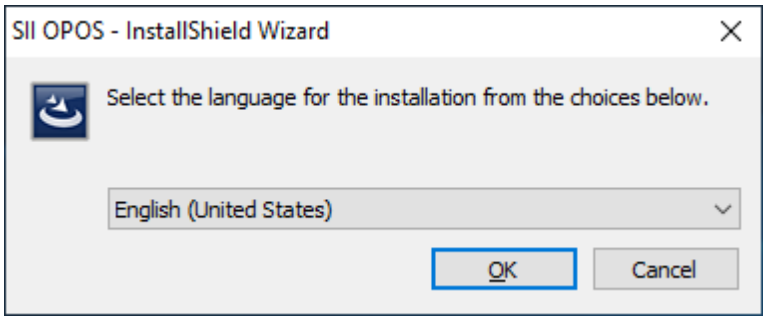
The installation procedure of this software is described below.

It is necessary to install the printer driver before installing this software.
For printer driver installation, please refer to "RP-E10 Series Printer Driver User's Guide" or "RP-D10 Series Printer Driver User's Guide".

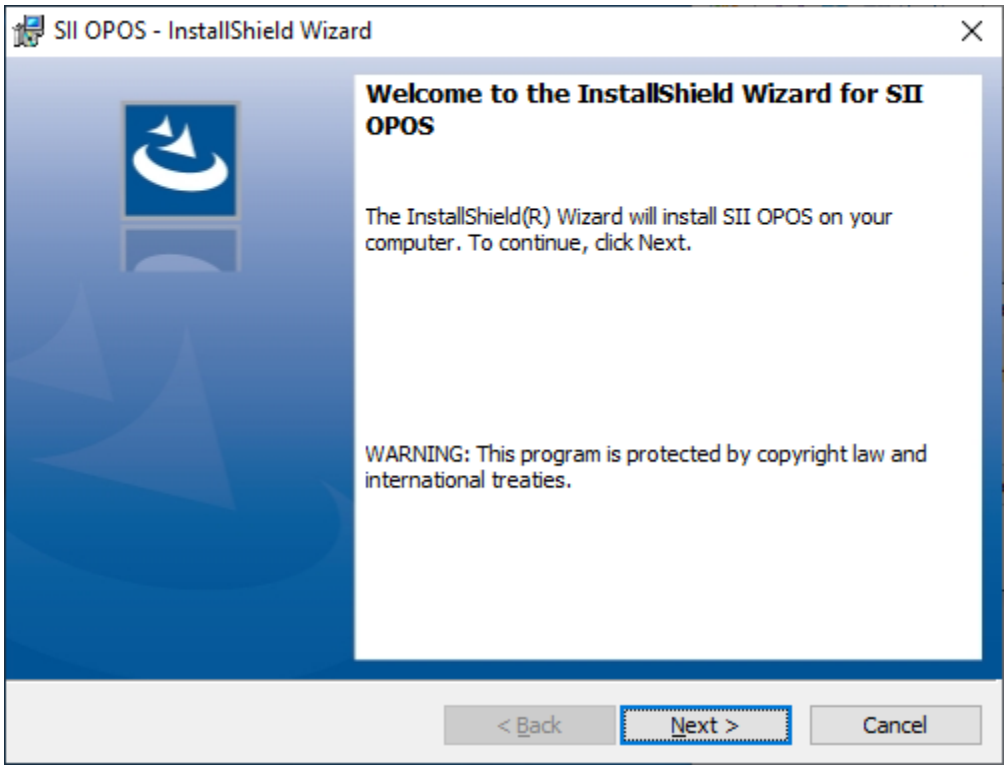
This installation requires logon to the computer with administrator privileges.

(1) Start the setup program (SetupOpos.exe).

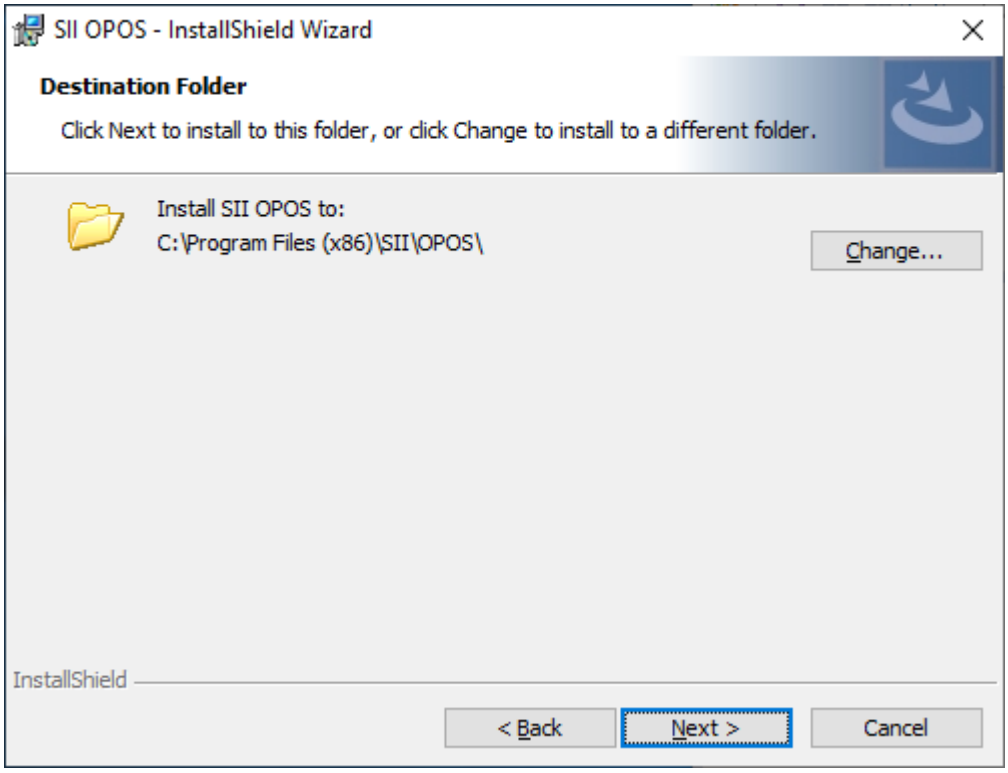
(2) The dialog box below is displayed and the setup program starts. Select [OK]. Hereinafter the installation procedure is explained with the installer window.



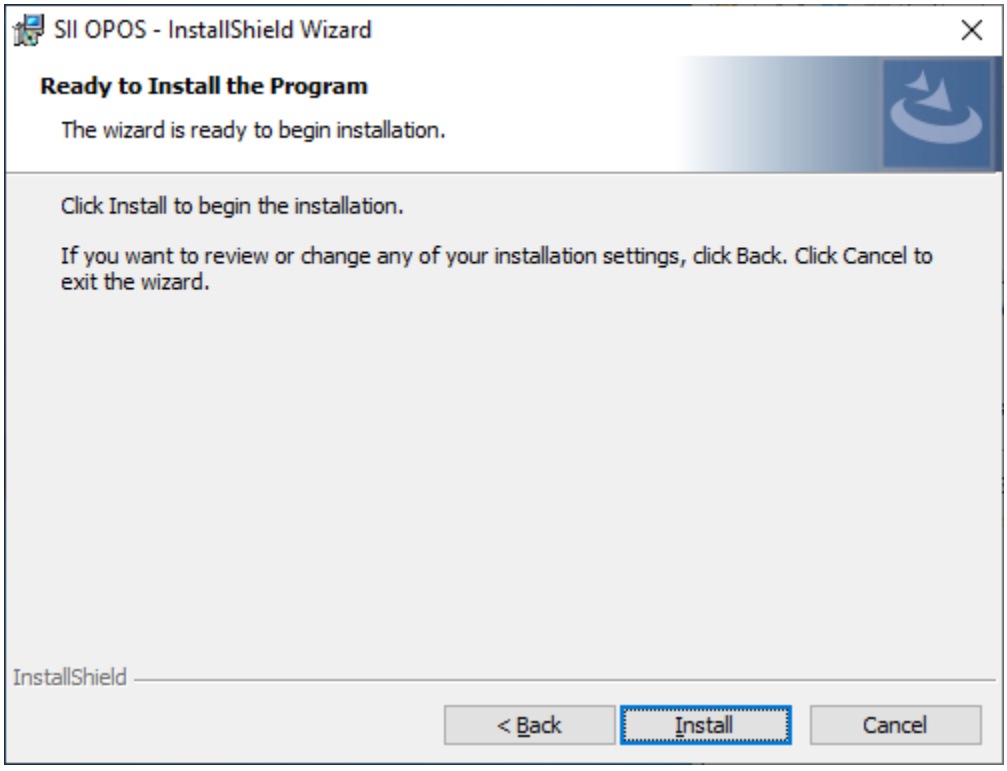
(3) The window below is displayed. Select [Next >].



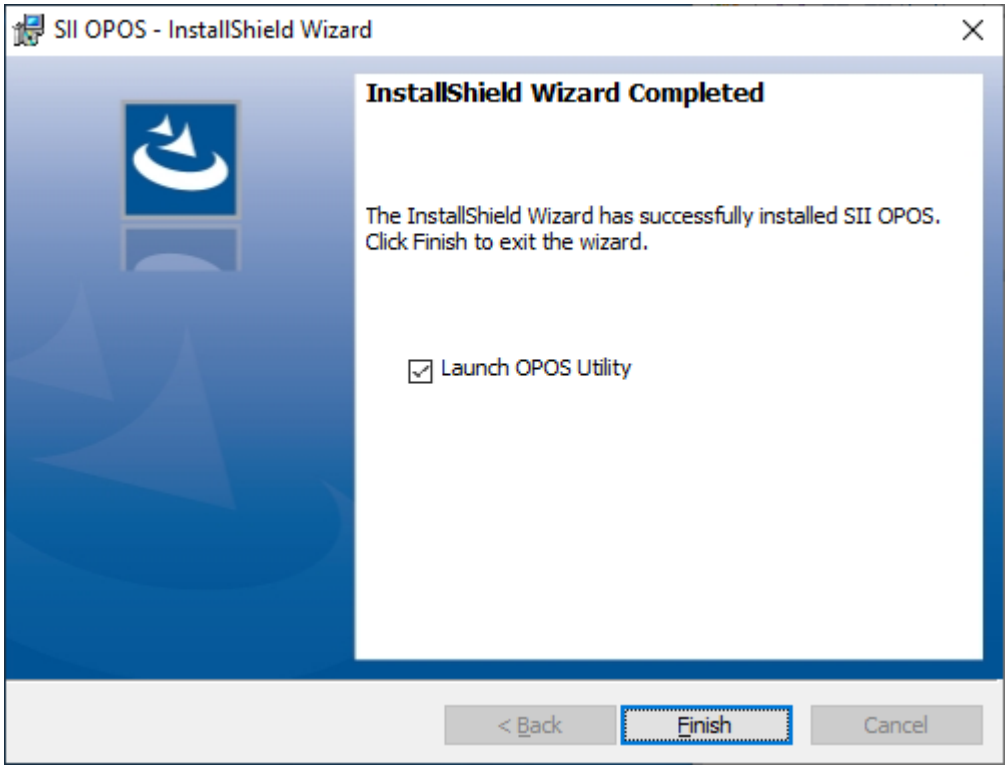
(4) Specify the installation folder. Installation folder is located in [System drive]:\Program Files\SII\OPOS by default. Select [Next >].



(5) After selecting the installation folder, the window below is displayed. If you select [Install], the installation starts.



(6) After data copy is completed, by keeping the check in [Launch OPOS Utility] box and selecting [Finish], the setup program is ended and the configuration program will be started.



This configuration program executes device addition and deletion, and another setting for each device. After setting is completed, the setting data will be saved in the registry.

Please refer to "2.5 Configuration Program (OPOS Utility)" for details about the configuration program. For the details about the registry, please refer to "6 Registry Used by the OPOS Control".

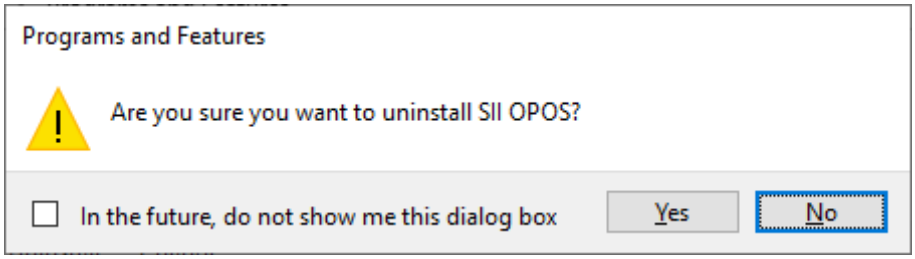
2.3 Uninstallation Procedure

Uninstall the OPOS Control by performing the following.

Open the control panel and select [Uninstall a program].
Then, select [SII OPOS] and click [Uninstall] or [Modify].

2.3.1 When [Uninstall] is Selected

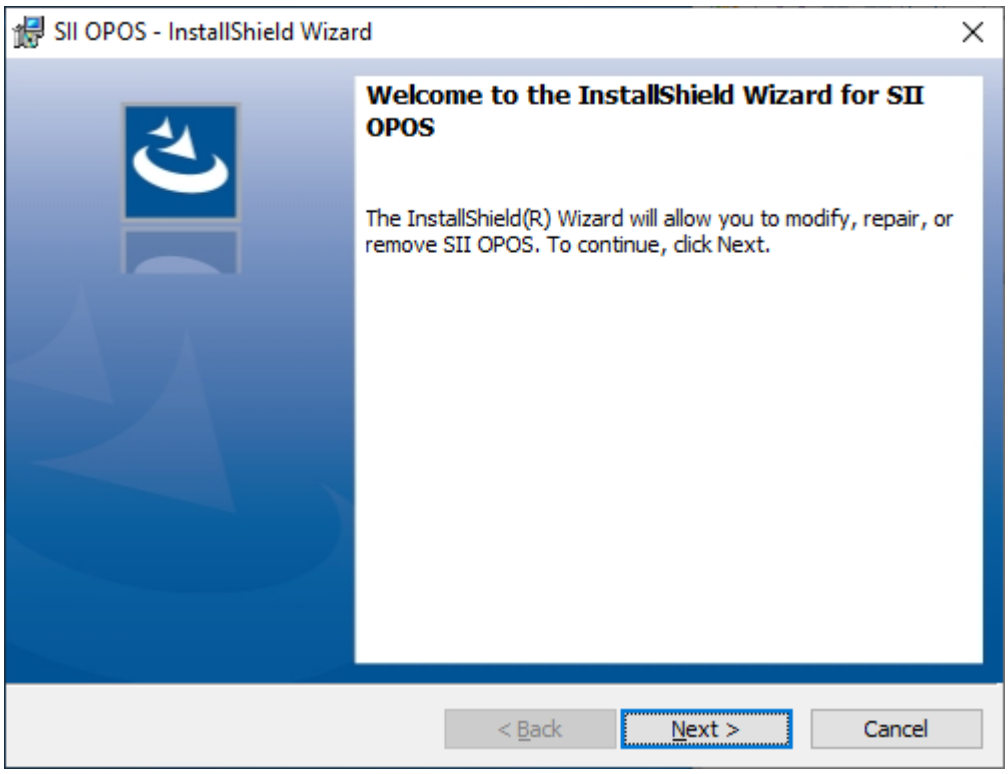
(1) The following dialog is displayed. Select [Yes].



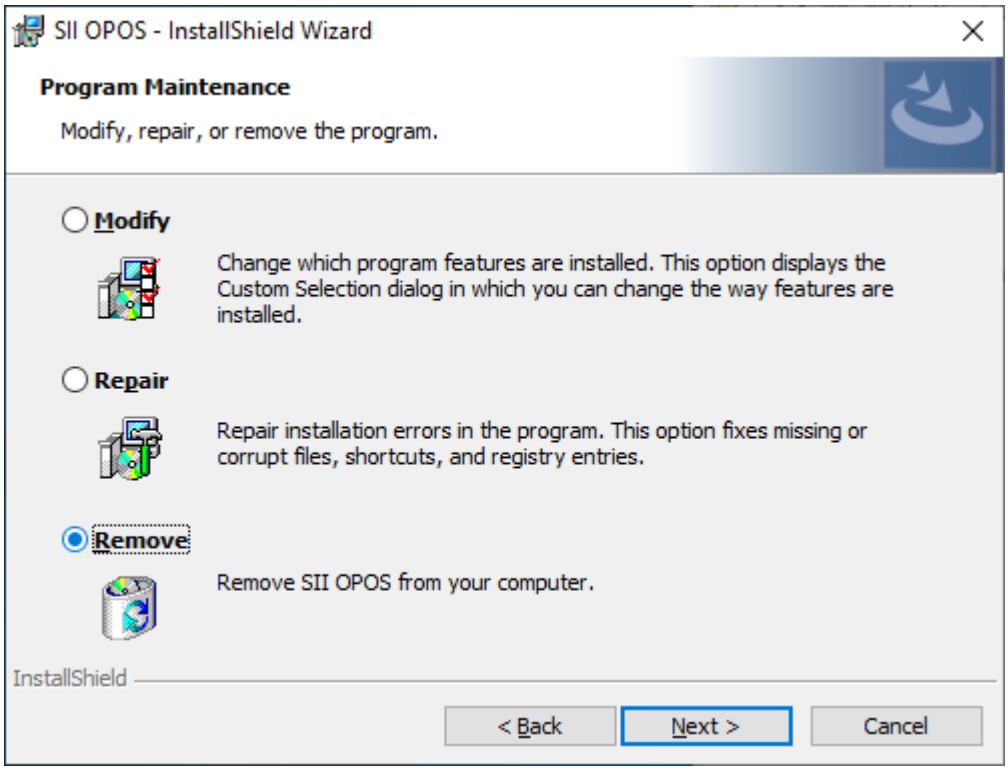
(2) Uninstallation is executed.

2.3.2 When [Modify] is Selected to Uninstall

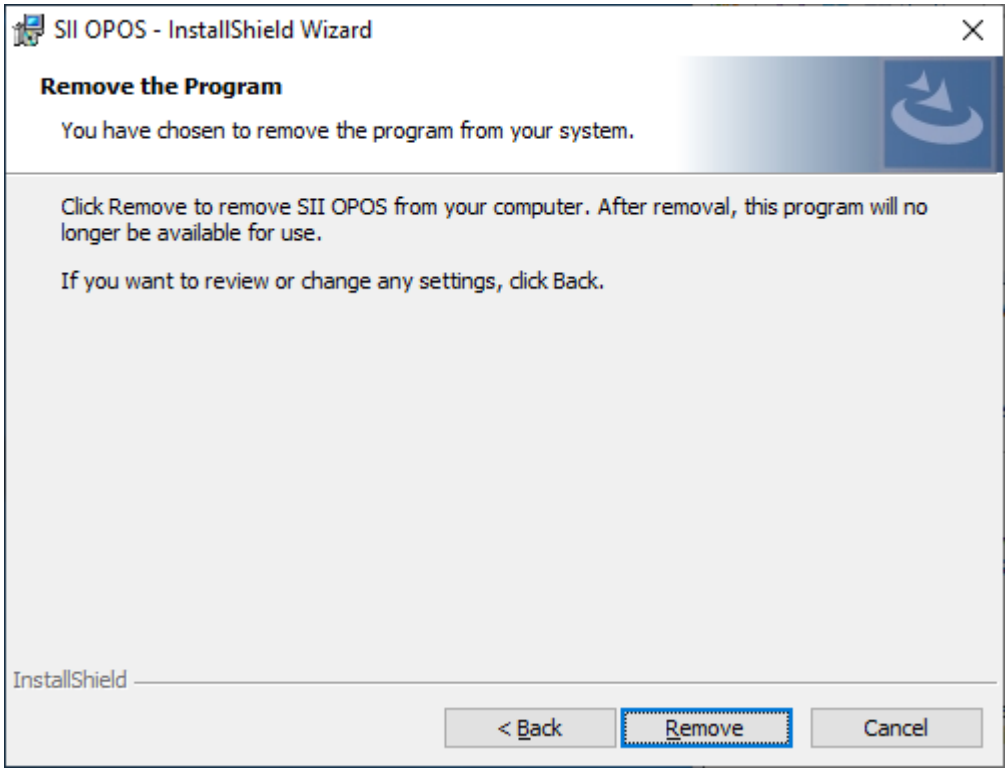
(1) The wizard of [Program Maintenance] is displayed. Select [Next >].



(2) The following window is displayed. Select [Remove] and click [Next >].



(3) The following window is displayed. Select [Remove].



(4) Uninstallation process is executed.

When the log files or other files are stored in the installed folder ([System drive]:\Program Files\SII\OPOS), they cannot be deleted by the uninstaller. Delete these files manually.

That completes the uninstallation process.

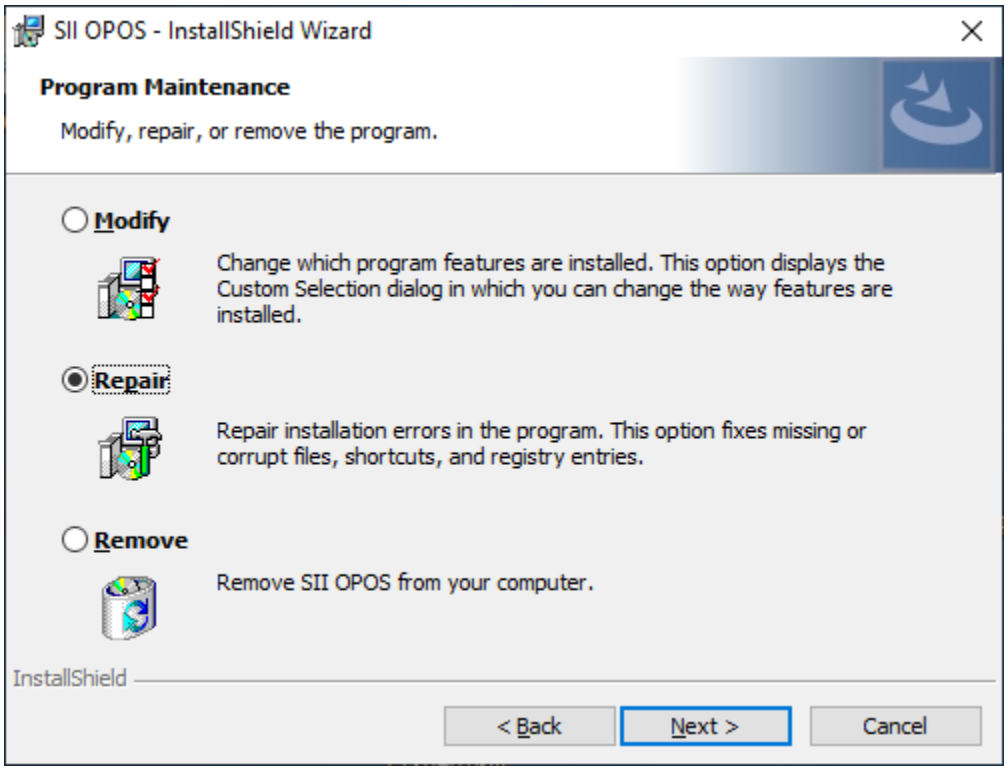
2.4 Repair Procedure

In the OPOS Control, the repair is executed by restoring the file copied to the PC at the installation.
The OPOS Control cannot change the function of the installed program. When [Modify] is selected in the following procedure, the repair is executed in the same way as [Repair].

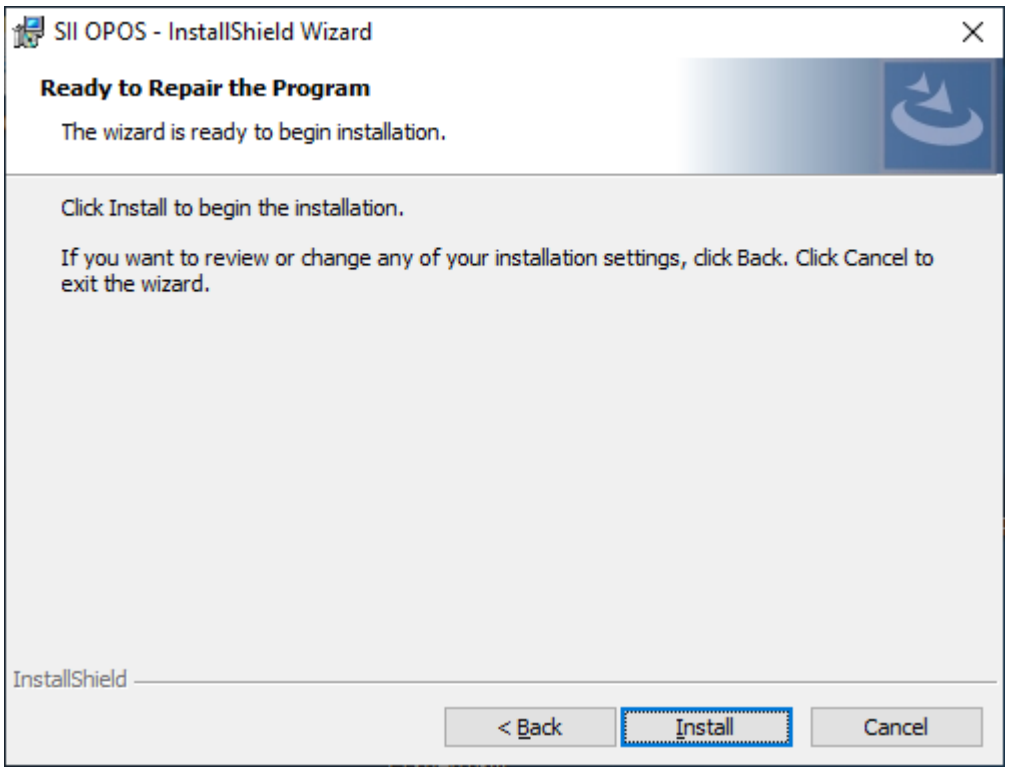
Repair the OPOS Control by the following operation.

Open the control panel and select [Uninstall a program].
Then, select [SII OPOS] and click [Modify].

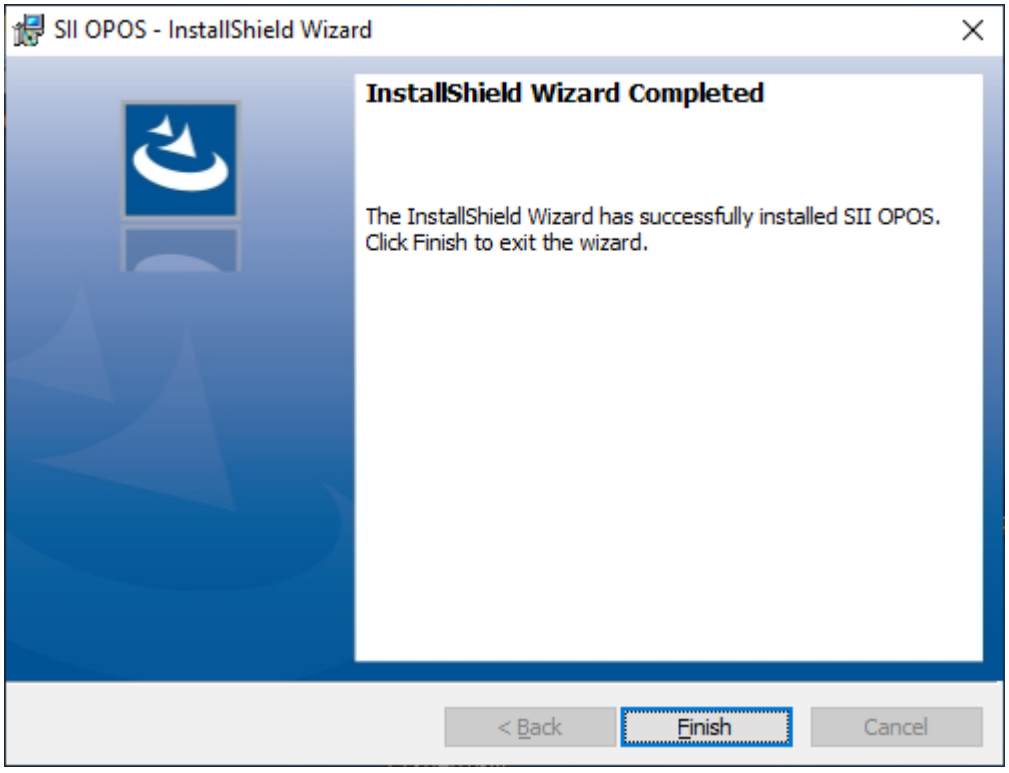
(1) The wizard of [Program Maintenance] is displayed. Select [Repair] and click [Next >].



(2) The following window is displayed. Select [Install].



(3) Repair process is executed. When the following window is displayed, select [Finish].



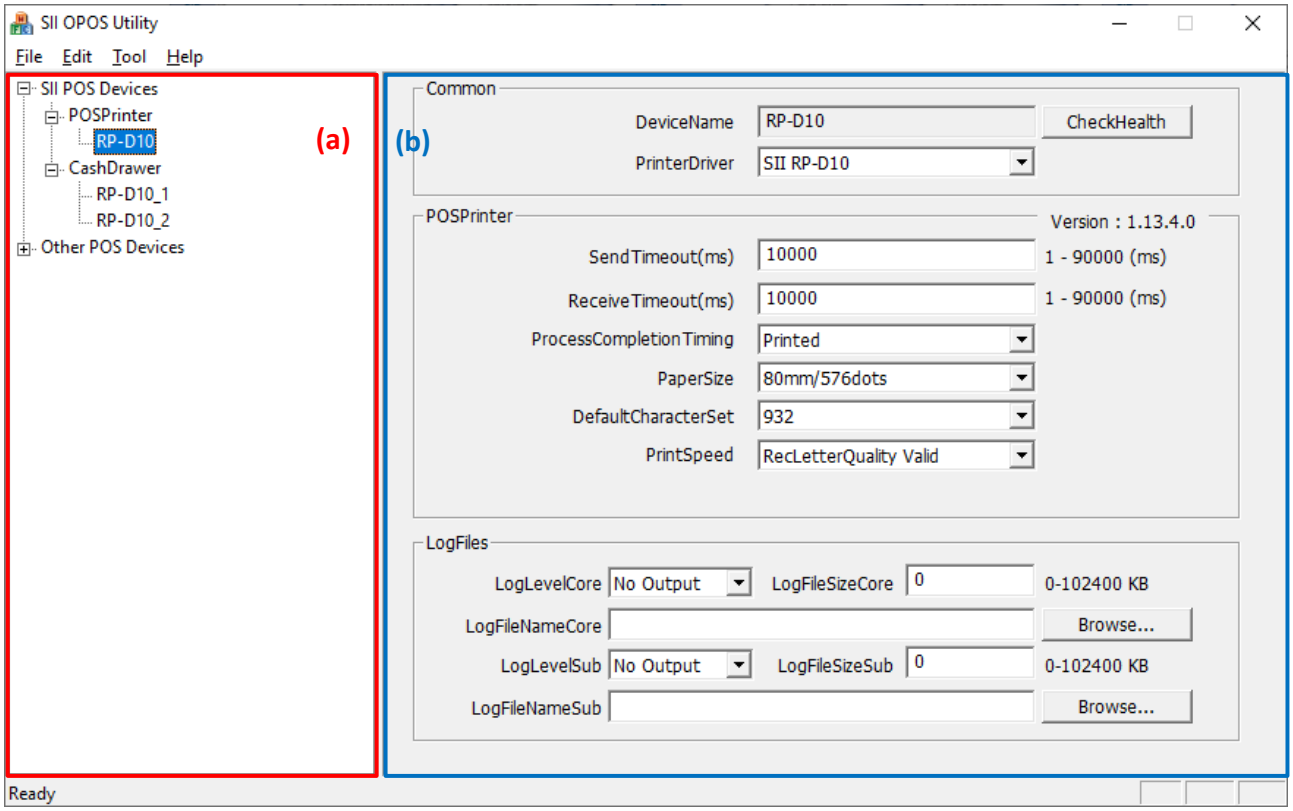
That completes the repair process.

2.5 Configuration Program (OPOS Utility)

The configuration program shows and changes the registry setting information for each device referenced by the OPOS Control. The program starts by the following procedure.

- Windows 11:
  Select [All apps] from the Start menu. Then select [OPOS Utility] from the displayed program list.
- Windows 10:
  Select [SII] - [OPOS Utility] from the Start menu.

The configuration program consists of "Device view" in the left and "Sitting view" in the right. The following figure shows an example of that RP-D10 is added to POSPrinter and two Drawers connected to RP-D10 are added to CashDrawer.



(a) "Device view": Displays the device name and the logical device name registered to the registry in a tree form.

(b) "Settings view": Shows or changes the setting information for the device name selected in the (a).

2.5.1 Addition of Device

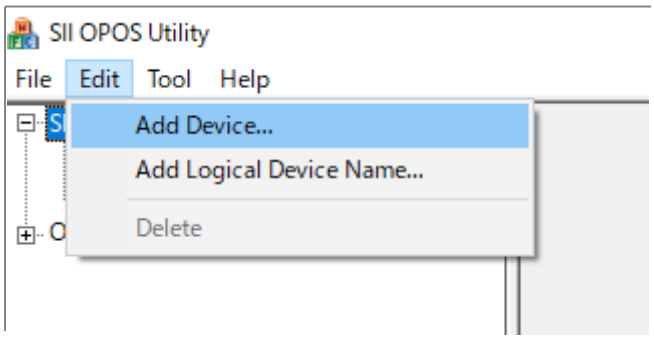
When the configuration program is started right after installing the OPOS Control, the device is not registered so the following screen appears.



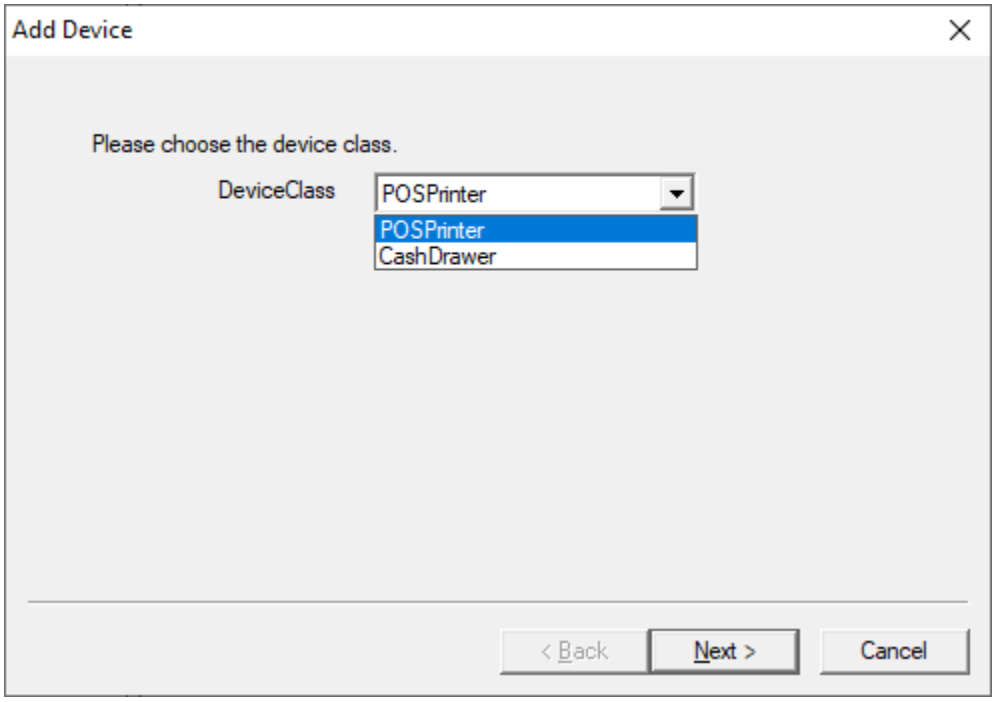
To use the OPOS Control, device addition is required at first. Click [Edit] or right click "Device view" and add the printer and drawer in the displayed menu. When adding the new printer, printer driver installation is required for applying communication port in advance. For printer driver installation, please refer to "RP-E10 Series Printer Driver User's Guide " or "RP-D10 Series Printer Driver User's Guide". For one printer, two drawers are available to add.

(1) Addition of printer

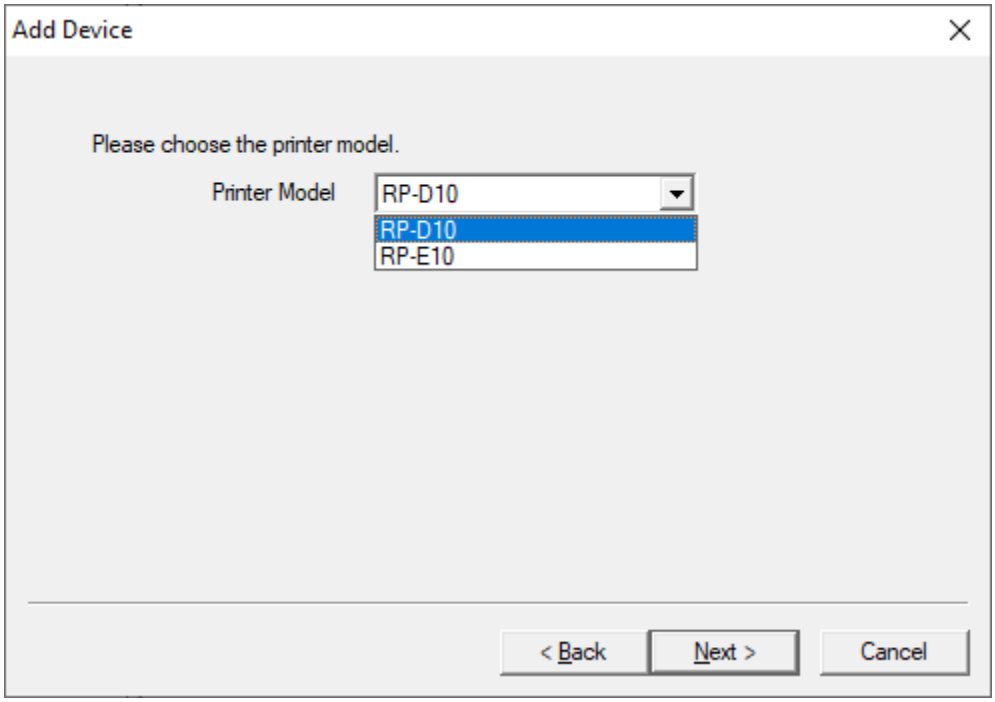
1) Select [Add Device...] in the menu opened by clicking [Edit] menu or right clicking "Device view".



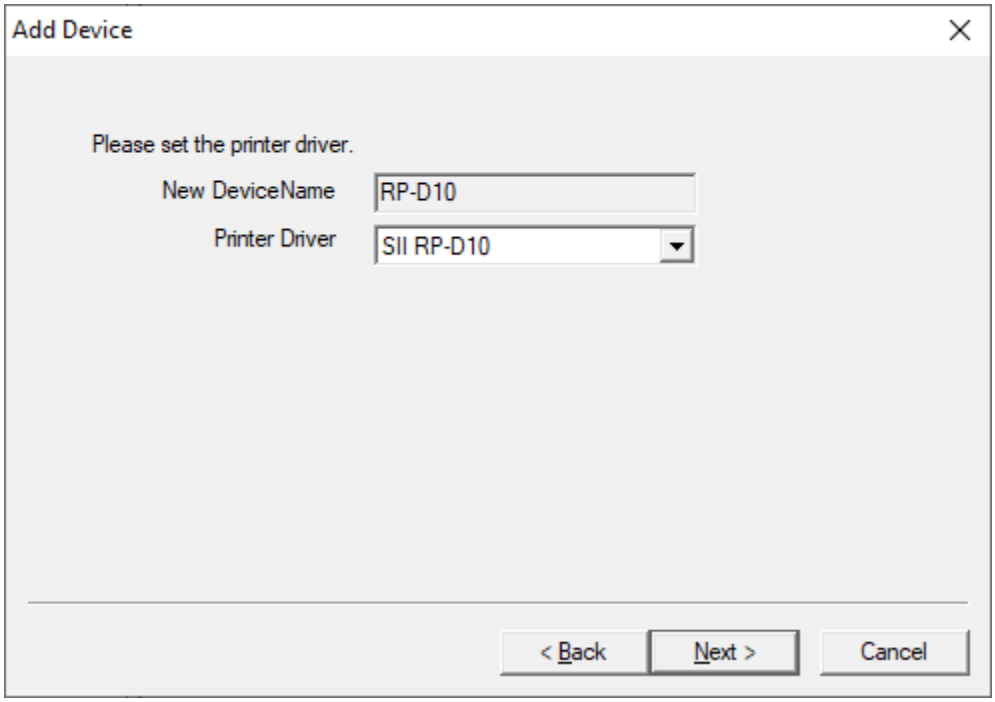
2) [Add Device] wizard is displayed. Select [POSPrinter] for [DeviceClass], and click [Next >].



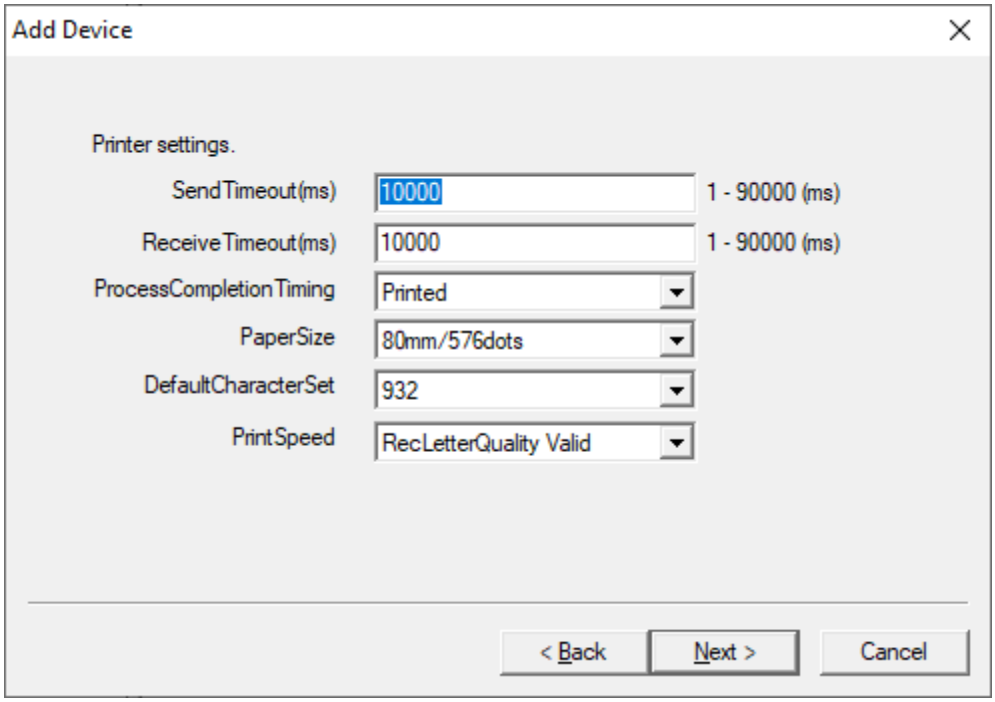
3) Select [Printer Model], and click [Next >]. (In the following examples shows the case of RP-D10 selection.)



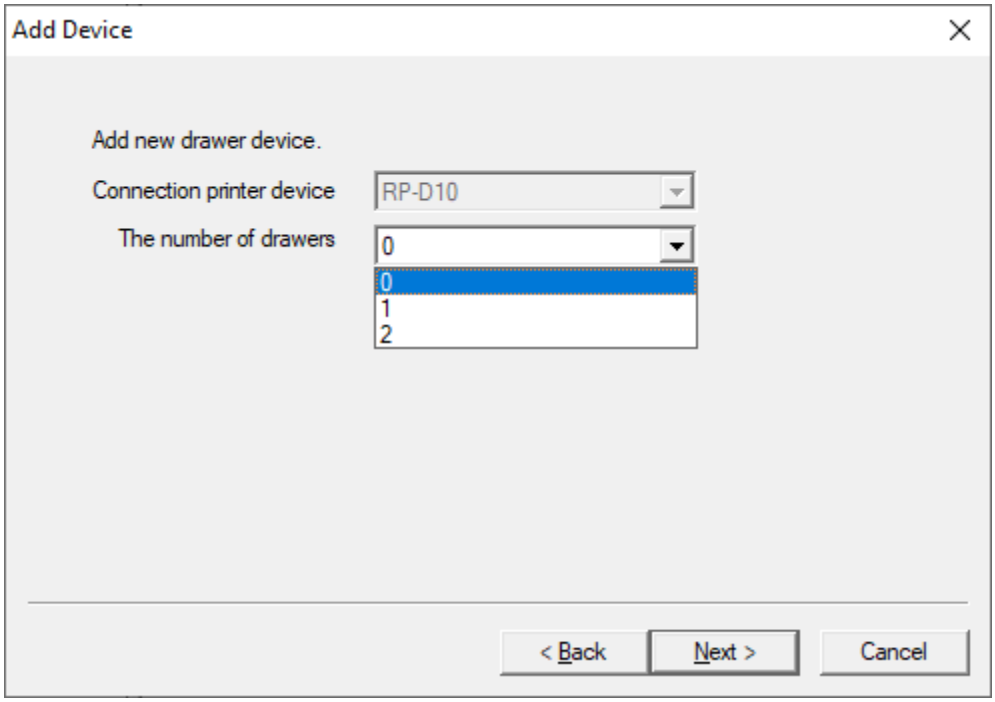
4) Select the printer driver using for the adding printer, and click [Next >].
([New DeviceName] is set automatically.)



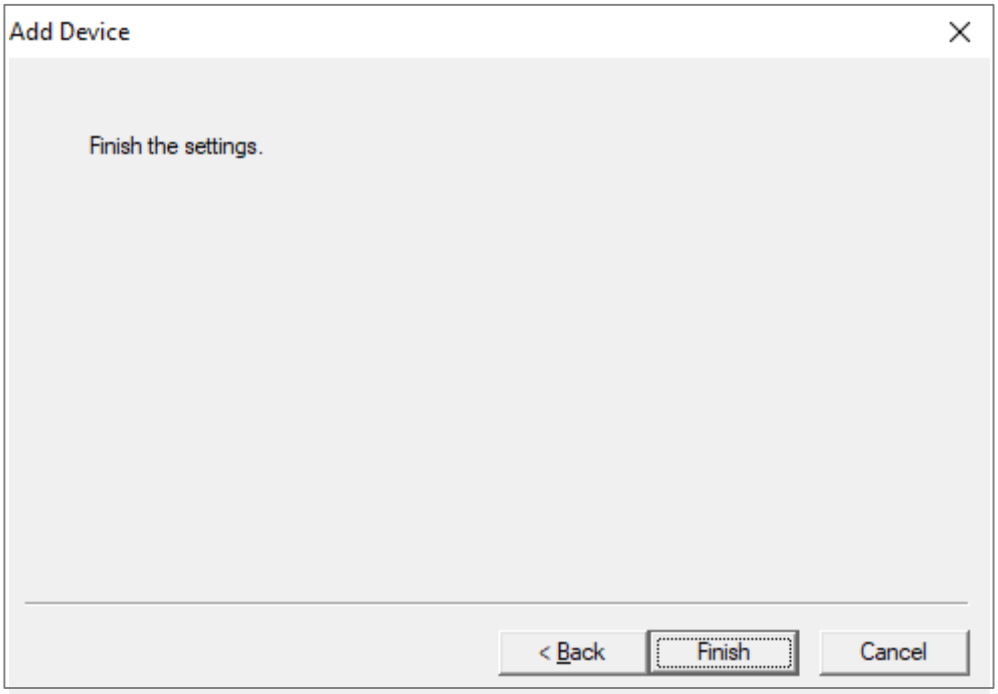
5) For the adding printer, details setting window is opened. Set each item, and select [Next >].
For details about the setting contents, please refer to "2.5.2 (1) 2) [POSPrinter]".



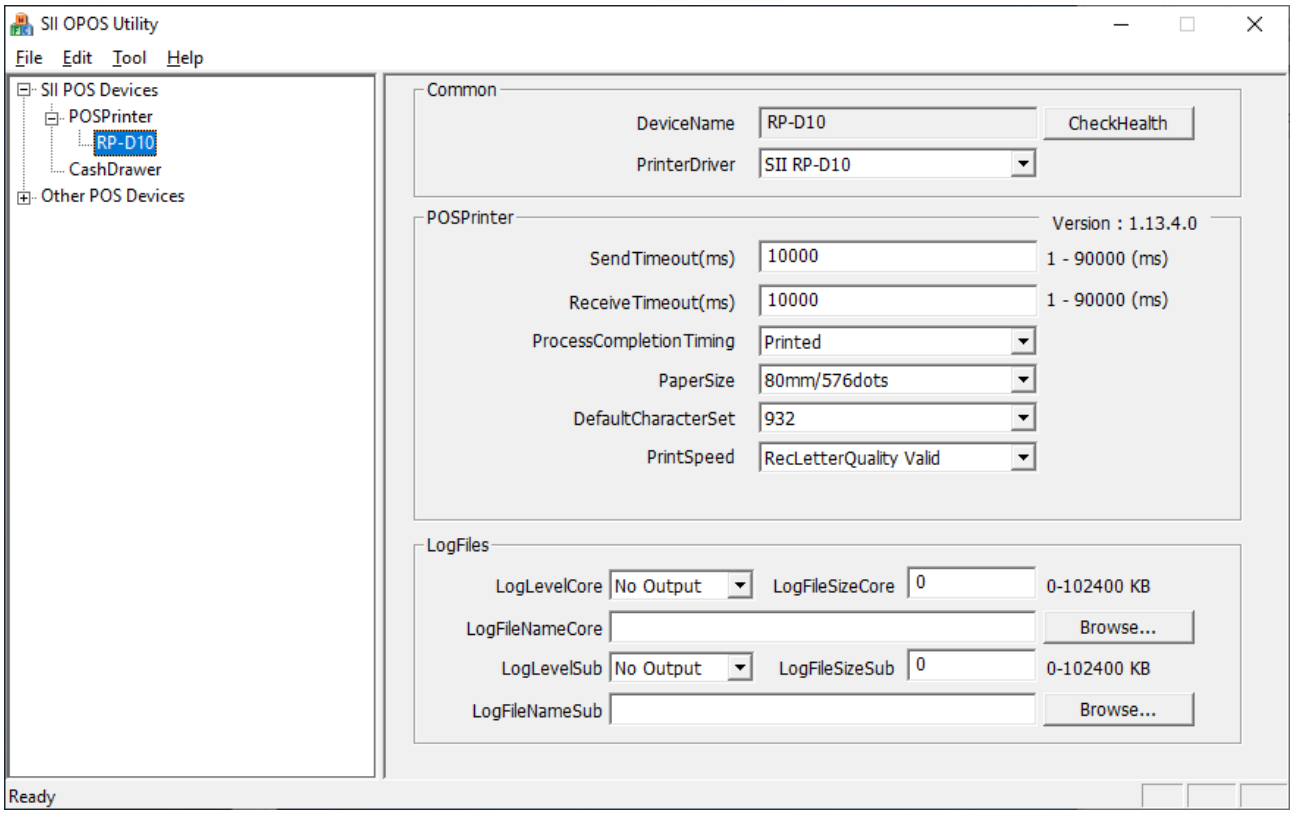
6) Subsequently, the following window is opened to select the number of drawers ([The number of drawers]) to connect the printer.
When adding the printer only: Select 0, and click [Next>]. Please refer to the description of 7) for subsequent settings.
When adding one cash drawer along with the printer: Select 1, and click [Next >]. Please refer to the description of 4) and 6) in (2) for subsequent settings.
When adding two cash drawers along with the printer: Select 2, and click [Next >]. Please refer to the description of 4), 5), and 6) in (2) for subsequent settings.



7) Finish adding the device. Select [Finish] to complete the addition of the device and the wizard.

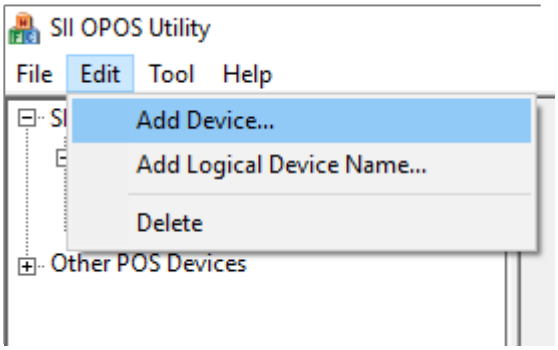


8) Added device name is shown in the state of selecting in "Device view" and "Settings view" of the configuration program. When the printer and the cash drawer are added at the same time, the printer is selected prior to the cash drawer.

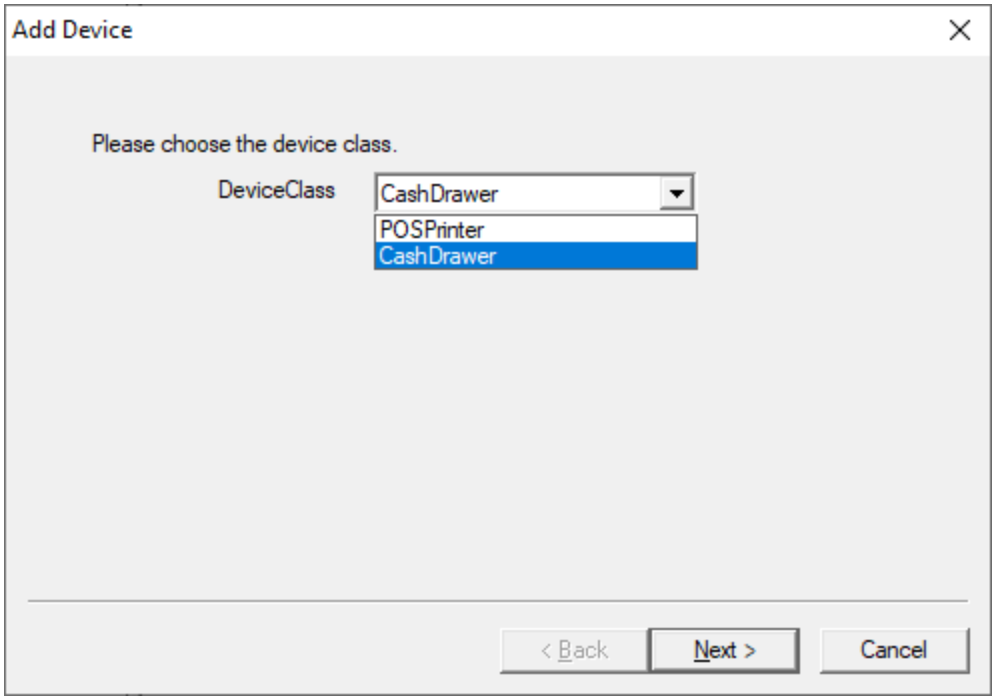


(2) Addition of cash drawer

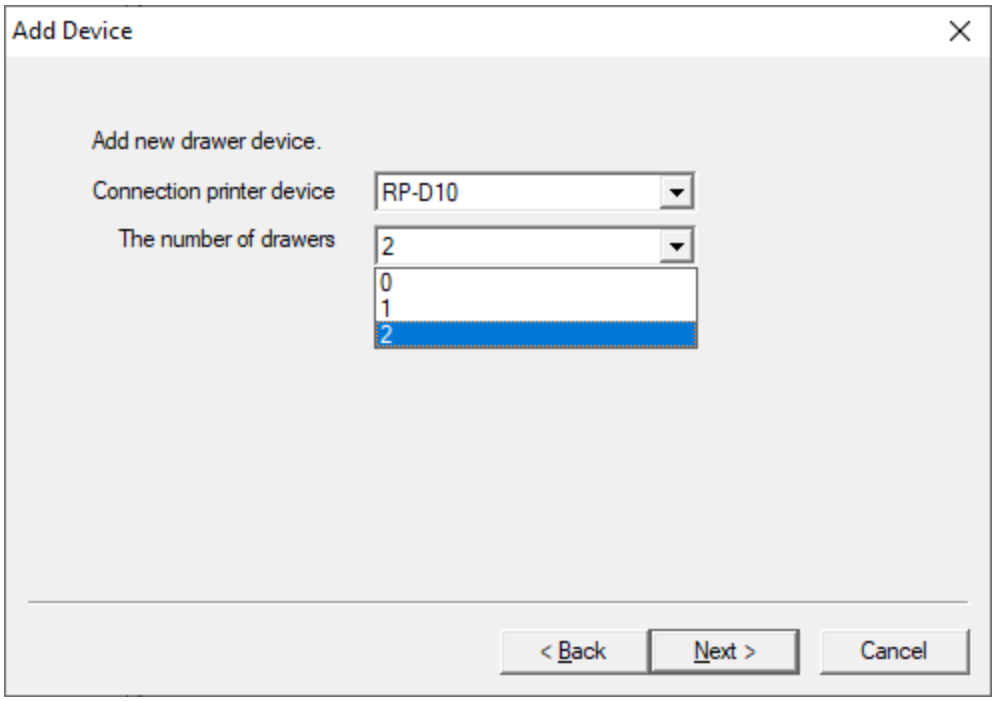
1) On "Device view", keep [SII OPOS Devices] selected in the top item of the tree, and select [Add Device...] in the menu opened by clicking [Edit] menu or right clicking "Device view".



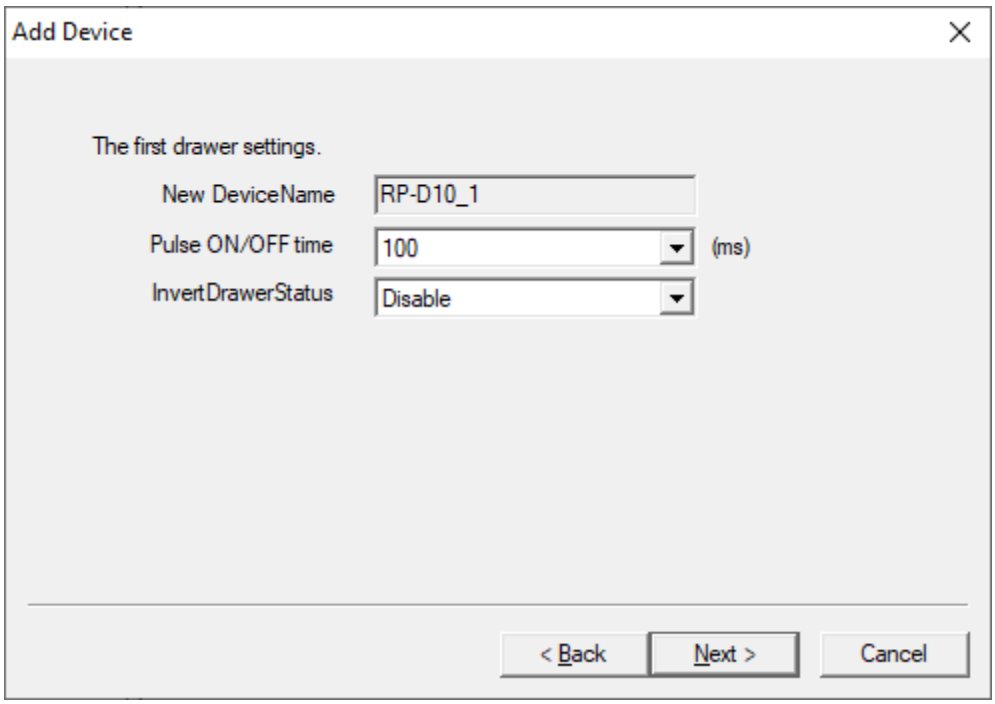
2) [Add Device] wizard is displayed. Select [CashDrawer] for [DeviceClass], and click [Next >].



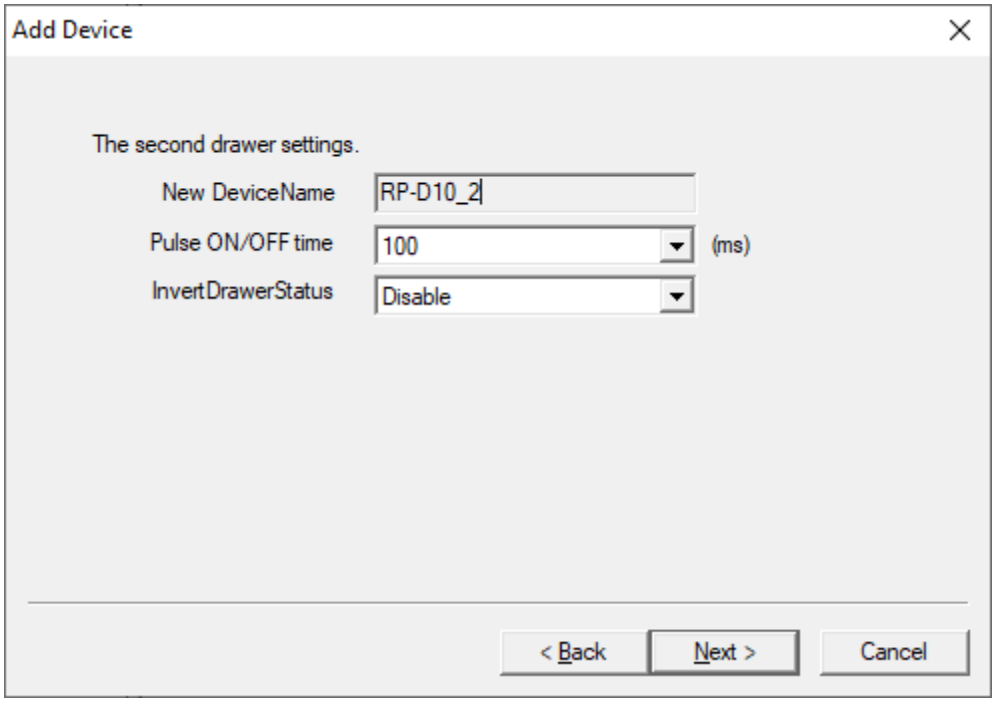
3) Select the device name of the printer as a connecting target, and then specify the number of drawers ([The Number of drawers]) to connected to the printer, and select [Next >].
When adding one cash drawer to the printer: Select 1, and click [Next >]. See the description of 4) and 6) for subsequent settings.
When adding two cash drawers to the printer: Select 2, and click [Next >]. See the description of 4), 5) and 6) for subsequent settings.



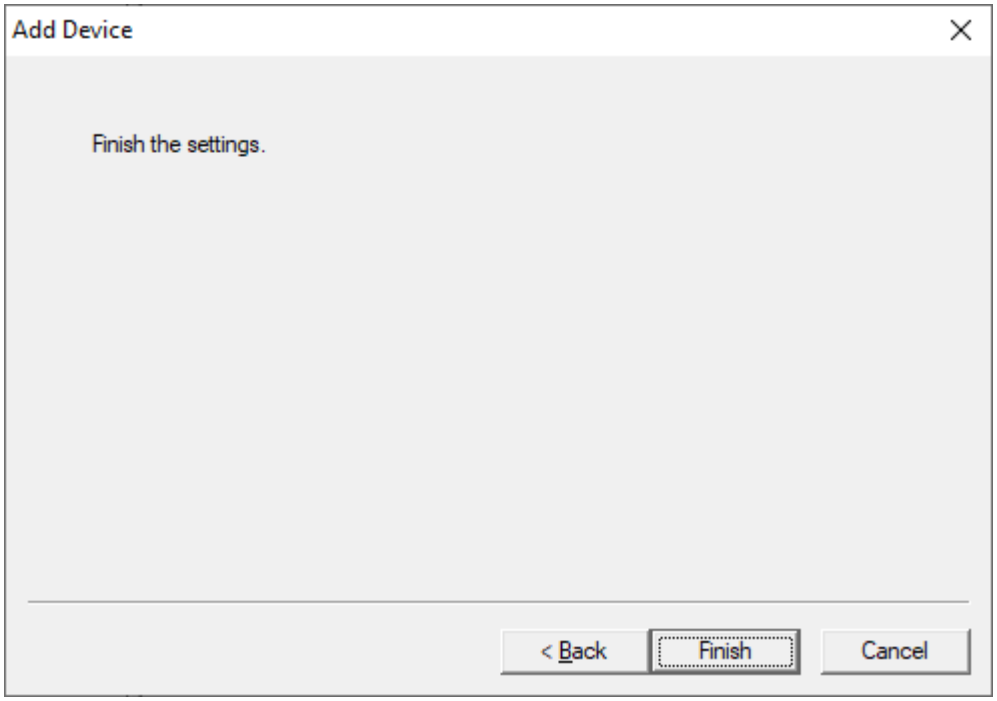
4) Advanced setting of the 1st cash drawer is shown. Set each item, and select [Next >].
Please refer to "2.5.2 (2) 2) [CashDrawer]" for details about the setting contents.
([New DeviceName] is set automatically.)



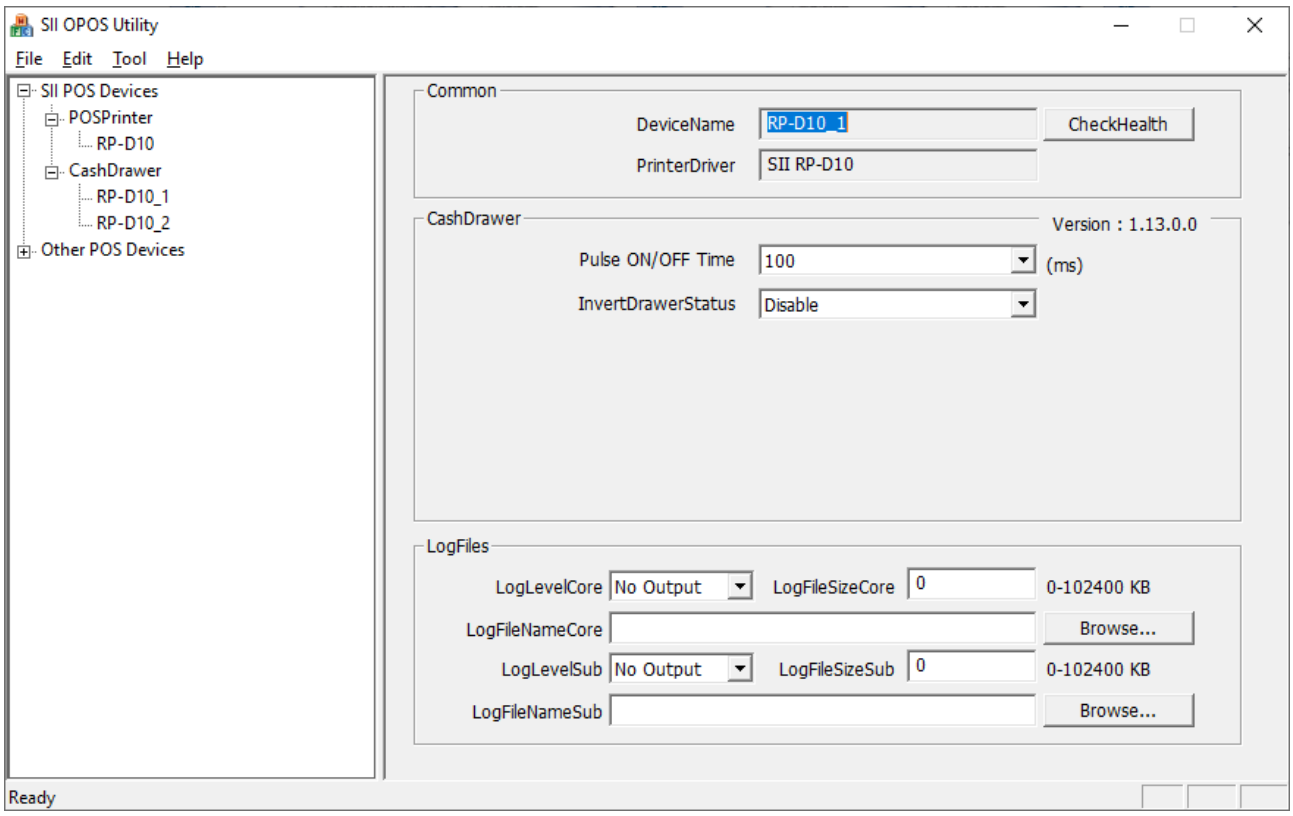
5) Detailed setting window is opened for the 2nd cash drawer. As with the setting of the 1st cash drawer, set each item and select [Next >].



6) Finish adding the cash drawer. Select [Finish] to complete the addition of the device and the wizard.



7) Added cash drawer is in the state of selecting, is shown in "Device view" and [Settings view] of the configuration program. When two drawers are added, the 1st drawer is selected prior to another.

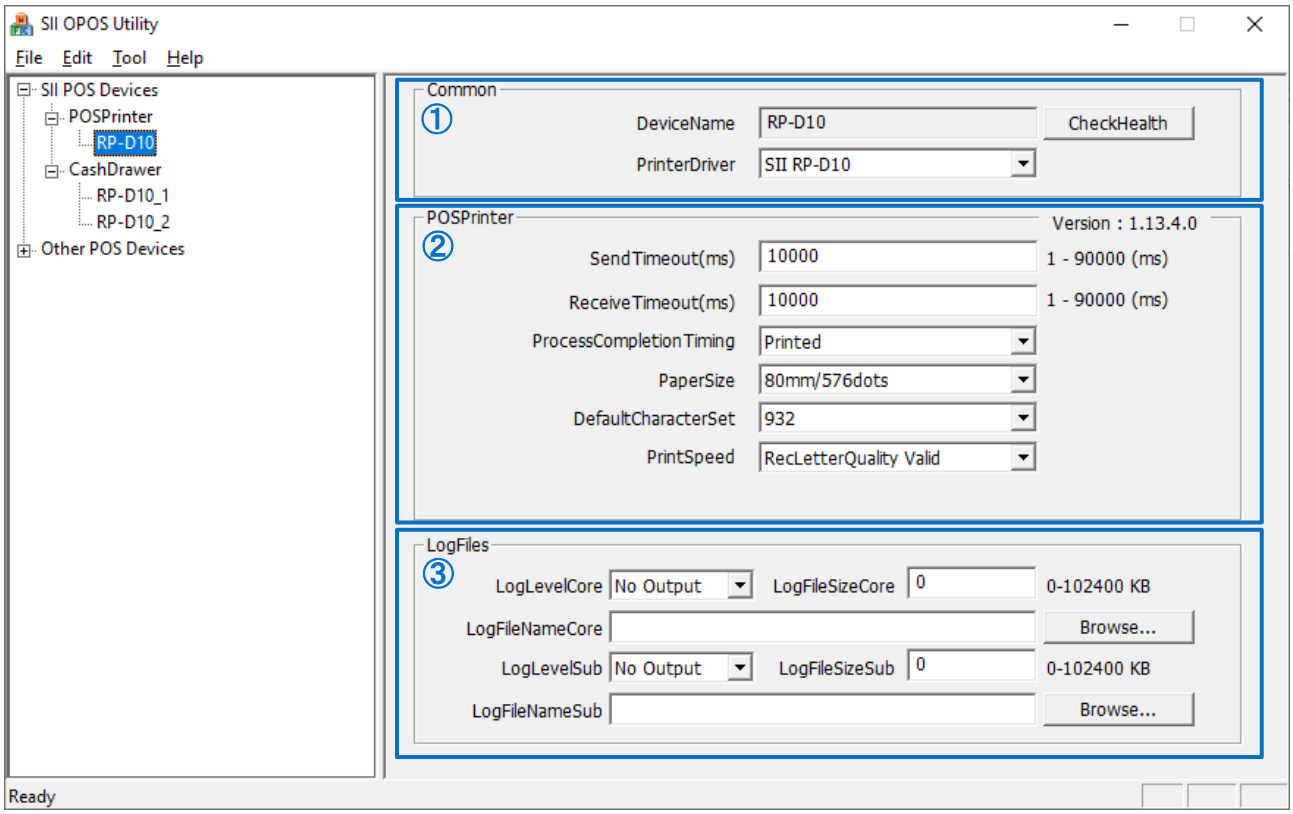


2.5.2 Device Setting

When a device name is selected in "Device view", changing the device setting is available in "Sitting view". However, in case of selecting a logical device name, the change is not available. For the details of logical device names, please refer to "2.5.3 Addition and Deletion of Logical Device Name".

(1) Printer setting

Setting contents of each item for the printer are as follows. It shows the state when the RP-D10 is selected in the initial state.



1) [Common]

Item	Default Value	Detail
DeviceName	-	The device name of the printer selected in "Device view".
PrinterDriver	-	Printer driver name.
CheckHealth	-	Executes an interactive test of the selected printer. **CheckHealth**(OPOS_CH_INTERACTIVE)

2) [POSPrinter]

Item	Default Value	Detail
SendTimeout	"10000"	Send Timeout (msec)
ReceiveTimeout	"10000"	Receive Timeout (msec)
ProcessCompletionTiming	"Printed"	Process Completion Timing setting Data Transmitted: Completion of the data transmission Printed: Completion of the printing
PaperSize	"80mm/576dots"	Selection of the paper width/number of effective dots 58mm/360dots 58mm/432dots 80mm/512dots 80mm/576dots
DefaultCharacterSet	"932"(Japanese) "999"(English)	The default value of the **CharacterSet** property 437: Code Page437 850: Code Page850 852: Code Page852 858: Code Page858 860: Code Page860 863: Code Page863 865: Code Page865 932 Code Page932 999: Windows ANSI Character 1250: Code Page1250 1251: Code Page1251 1252: Code Page1252 1253: Code Page1253 1254: Code Page1254
PrintSpeed	"RecLetterQuality Valid"	Print Speed Specify RecLetterQuality Valid : RecLetterQuality Setting is valid High : High speed mode is fixed. Middle(Quality) : Middle(Quality) speed mode is fixed. Middle(Silent) : Middle(Silent) speed mode is fixed. Low : Low speed mode is fixed.
NearEndSensor*1	"Enable"	Function of near end sensor Enable: Valid Disable: Invalid
Version	"1.13.x.x"	Version of dedicated POS Printer Service Object for RP-D10 or RP-E10

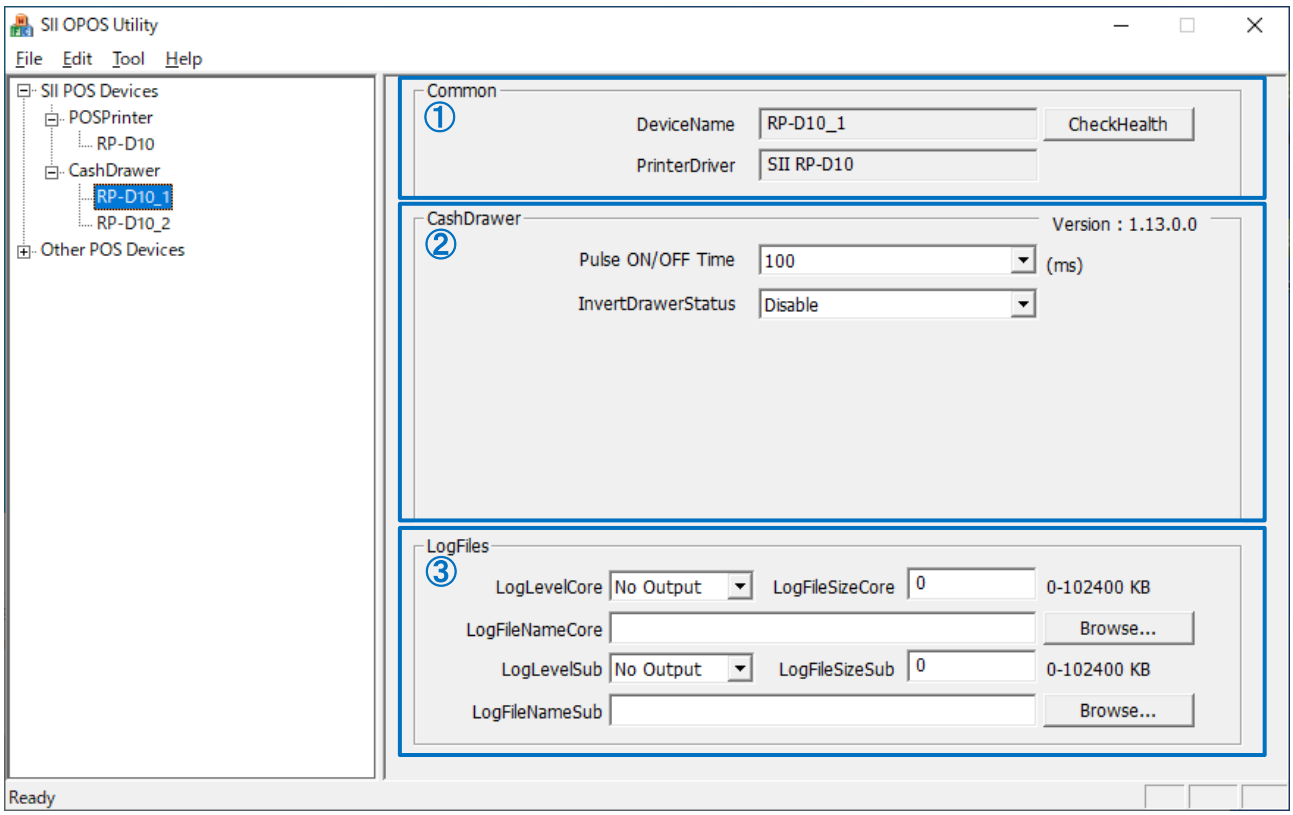
*1: In the case of RP-E10

3) [LogFiles]

Item	Default Value	Detail
LogLevelCore	" No Output "	Log output level of POS Printer Service Object No Output: No output Output(Lv.1): Error Output(Lv.5): Trace
LogFileSizeCore	"0"	Maximum log file size of POS Printer Service Object (KB)
LogFileNameCore	""	Log file name of POS Printer Service Object (Full path)
LogLevelSub	" No Output "	Log output level of dedicated POS Printer Service Object for RP-D10 or RP-E10 No Output: No output Output(Lv.1): Error Output(Lv.5): Trace
LogFileSizeSub	"0"	Maximum log file size of dedicated POS Printer Service Object for RP-D10 or RP-E10 (KB)
LogFileNameSub	""	Log file name of dedicated POS Printer Service Object for RP-D10 or RP-E10 (Full path)

(2) Drawer Setting

Setting contents of each item of the drawer are as follows. The figure below shows the state when the RP-D10_1 is selected.



1) [Common]

Item	Default Value	Detail
DeviceName	-	The device name of the cash drawer selected in "Device view".
PrinterDriver	-	Printer driver name. Change is not accepted due to referring the [PrinterDriver] setting contents of the connection targeted printer.
CheckHealth	-	Executes an interactive test of the selected cash drawer. **CheckHealth**(OPOS_CH_INTERACTIVE)

2) [CashDrawer]

Item	Default Value	Detail
Pulse ON/OFF Time	"100"	ON/OFF pulse time for cash drawer drive 100 to 800 (msec)
InvertDrawerStatus	"Disable"	Notification method of cash drawer open/close state Disable: Reversal is not active. The cash drawer is opened when the drawer switch input status is "Low". Enable: Reversal is active. The cash drawer is opened when the drawer switch input status is "High".
Version	"1.13.x.x"	Version of dedicated Cash Drawer Service Object for RP-D10 or RP-E10

3) [LogFiles]

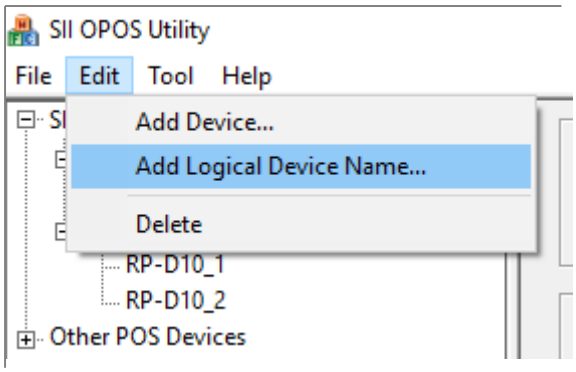
Item	Default Value	Detail
LogLevelCore	" No Output "	Log output level of Cash Drawer Service Object No Output: No output Output(Lv.1): Error Output(Lv.5): Trace
LogFileSizeCore	"0"	Maximum log file size of Cash Drawer Service Object (KB)
LogFileNameCore	""	Log file name of Cash Drawer Service Object (Full path)
LogLevelSub	" No Output "	Log output level of dedicated Cash Drawer Service Object for RP-D10 or RP-E10 No Output: No output Output(Lv.1): Error Output(Lv.5): Trace
LogFileSizeSub	"0"	Maximum log file size of dedicated Cash Drawer Service Object for RP-D10 or RP-E10 (KB)
LogFileNameSub	""	Log file name of dedicated Cash Drawer Service Object for RP-D10 or RP-E10 (Full path)

2.5.3 Addition and Deletion of Logical Device Name

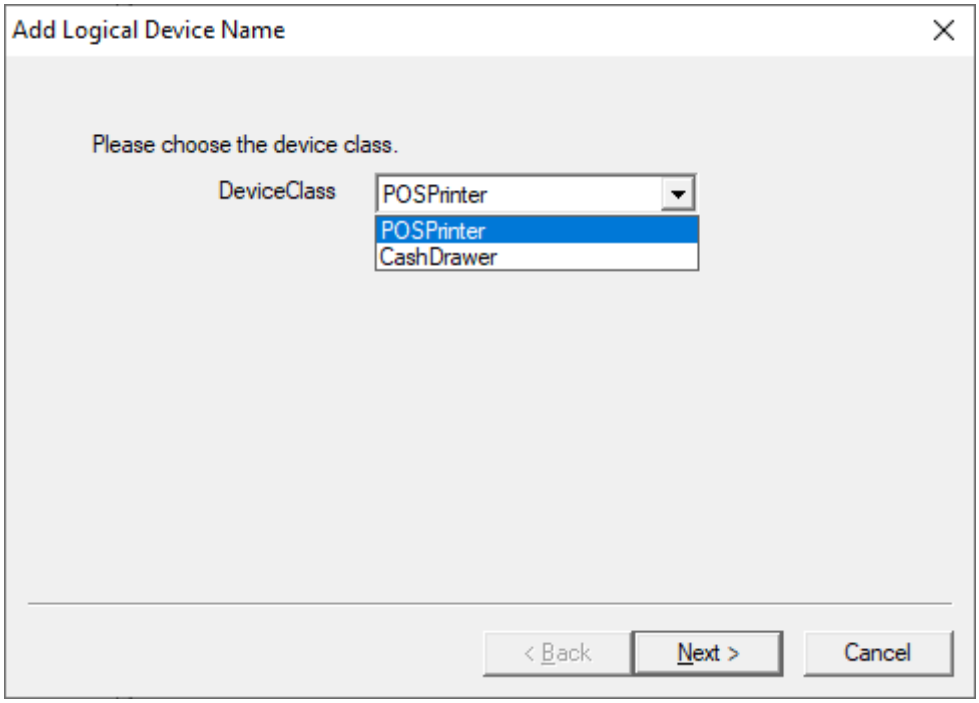
For specified devices, the logical device names are available to add or delete.

(1) Addition of Logical Device Name

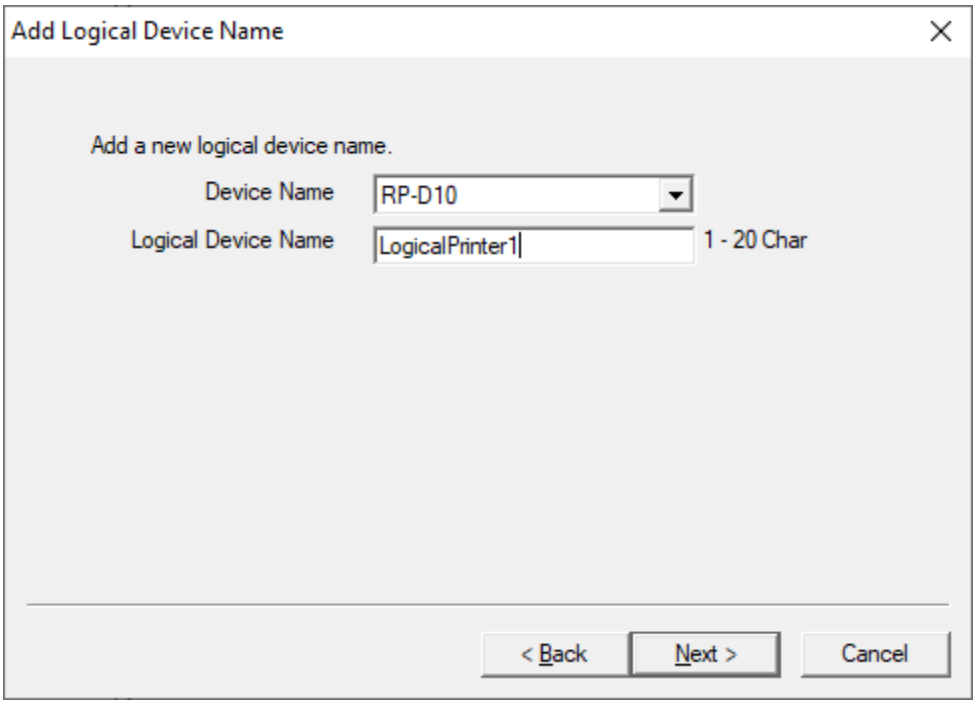
1) Select [Add Logical Device Name...] in the menu opened by clicking [Edit] menu or right clicking "Device view".



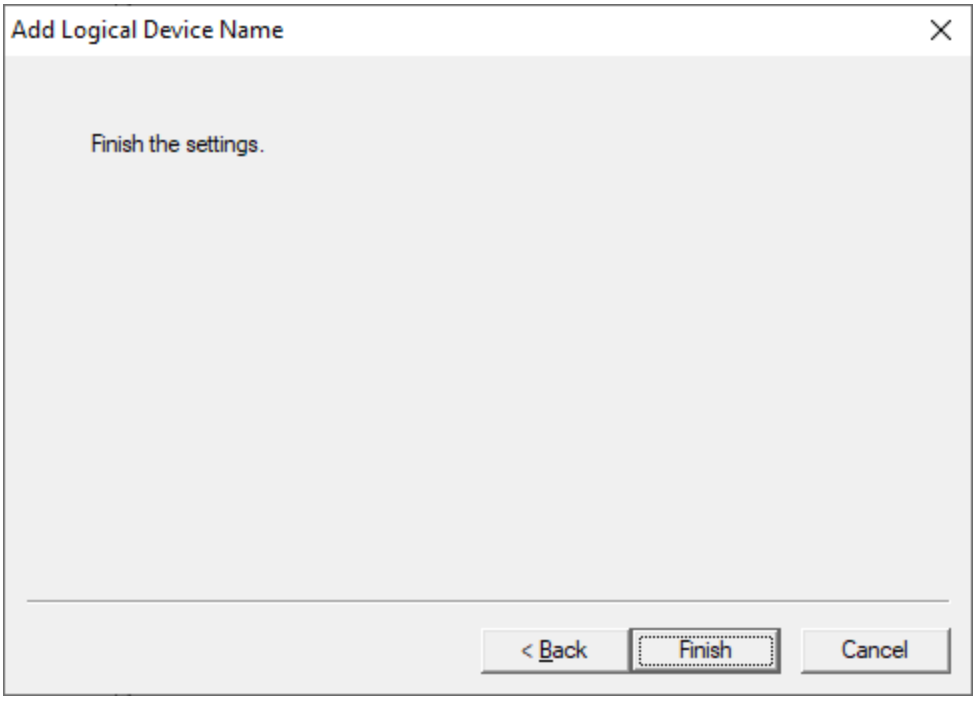
2) [Add Logical Device Name] wizard is displayed. Select [DeviceClass], and click [Next >].



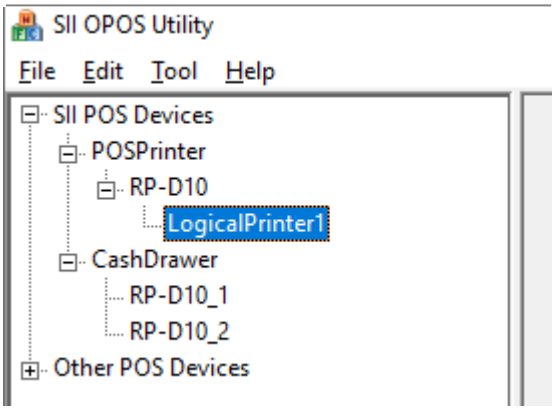
3) In the device name ([Device Name]) list, select the target device name for setting, and enter [Logical Device Name], and select [Next >].



4) The window below is displayed. Select [Finish] to complete the addition of the logical device name and the wizard.

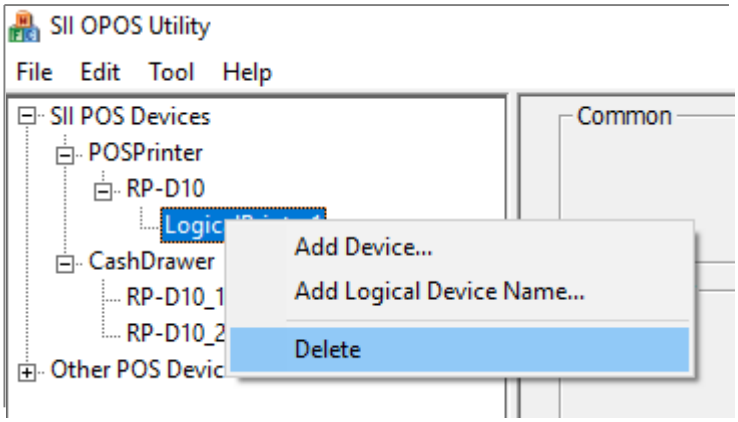


5) Added logical device name is shown under the targeted device name in a tree form.

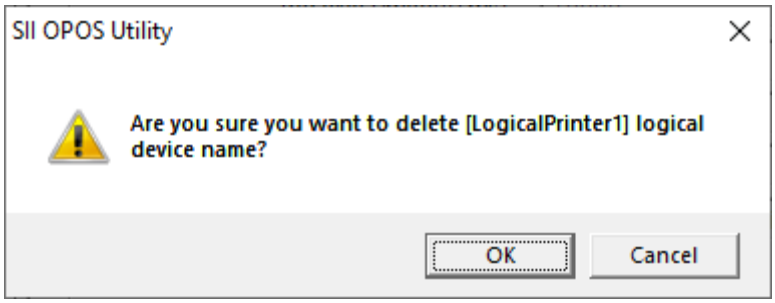


(2) Deletion of Logical Device Name

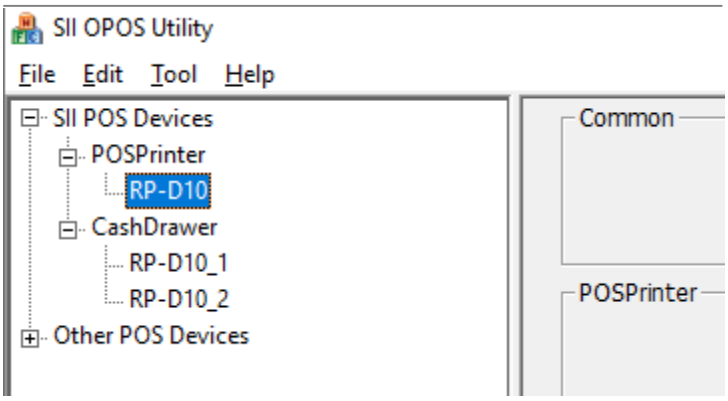
1) From "Device view", select the target logical device name to delete. Select [Delete] in the menu opened by clicking [Edit] menu or right clicking "Device view".



2) Confirm the warning message about deletion of the logical device name, and select [OK].



3) Selected device is deleted from "Device view".



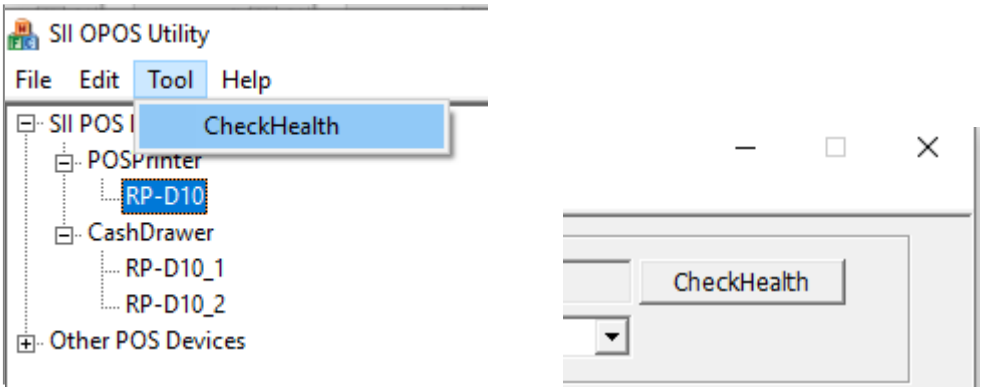
2.5.4 Test of Device

In the configuration program, it is possible to execute the operational test toward the device selected in "Device view".

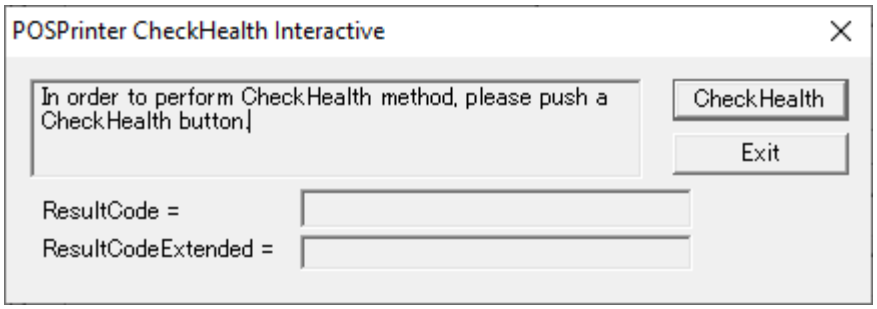
(1) Test of Printer

1) From "Device view", select the printer which the operational test is executed.

2) Select [CheckHealth] from the drop-down list in [Tool] menu, or select [CheckHealth] in [Settings view].

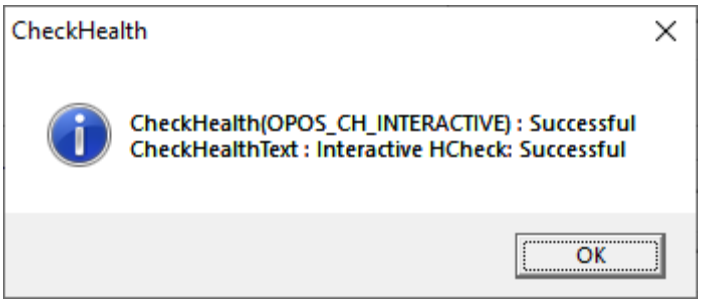


3) The dialog of [Interactive Test] is displayed. Select [CheckHealth] to execute the interactive test of the printer.



4) When the interactive test is executed successfully, the printer ROM version ID, **ServiceObjectVersion** property, and **DeviceName** property of the OPOS Printer Control are printed.

5) To finish the interactive test, select [Exit] in [Interactive Test] dialog. The result of the interactive test is displayed. The figure below shows the example when the interactive test is executed successfully. Click [OK] to finish.



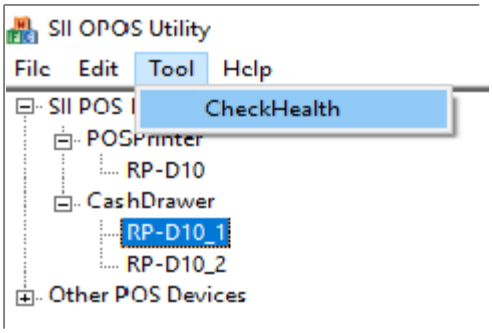
6) When the interactive test failed, the **ResultCode** property and the **ResultCodeExtended** property which show the error cause are displayed in the result of the interactive test. The figure below shows the example when the interactive test of the printer is not executed successfully. Click [OK] to finish. See the contents of return value for the **CheckHealth** method of the POS Printer Control for the **ResultCode** property and the **ResultCodeExtended** property.

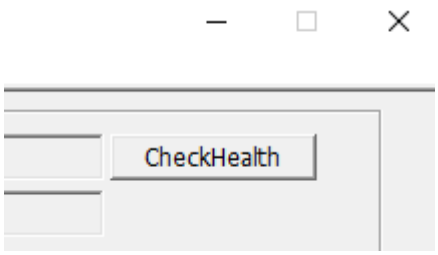


(2) Test of Cash Drawer

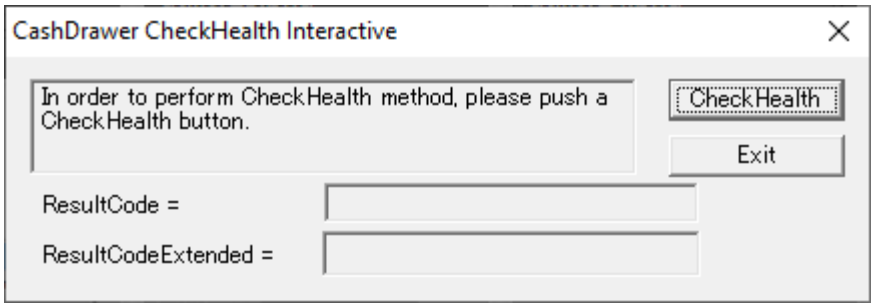
1) From "Device view", select the cash drawer which the operational test is executed.

2) Select [CheckHealth] from the drop-down list in [Tool] menu, or select [CheckHealth] in [Settings view].



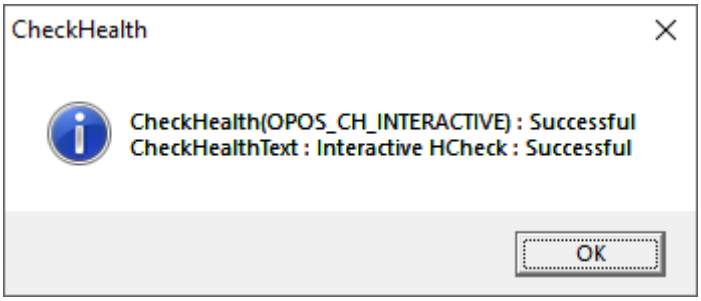


3) The dialog of [Interactive Test] is displayed. Select [CheckHealth] to execute the interactive test of the cash drawer.



4) When the interactive test succeeds, the selected cash drawer is opened.

5) To finish the interactive test, select [Exit] in [Interactive Test] dialog. The result of the interactive test is displayed. The figure below shows the example when the interactive test is executed successfully. Click [OK] to finish.



6) When the interactive test failed, the **ResultCode** property and the **ResultCodeExtended** property which show the error cause are displayed in the result of the interactive test. The figure below shows the example when the interactive test of the cash drawer is not executed successfully. Click [OK] to finish. See the contents of return value for the **CheckHealth** method of the Cash Drawer Control for the **ResultCode** property and the **ResultCodeExtended** property.

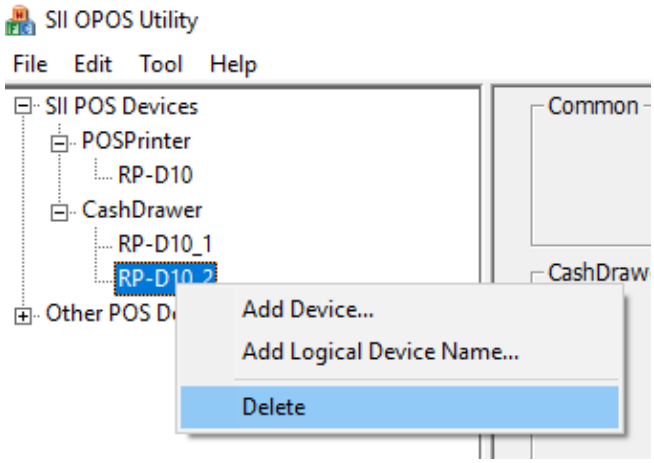


2.5.5 Deletion of Device

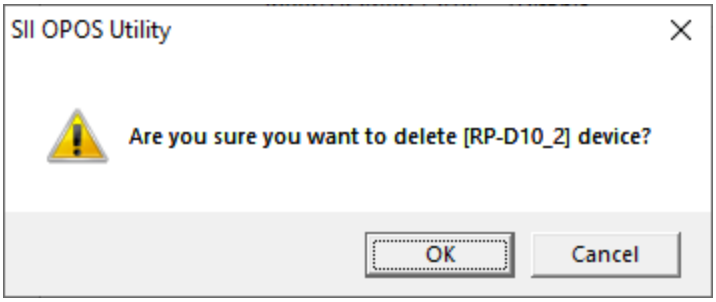
(1) Deletion of printer or cash drawer

In the following examples shows the case of RP-D10_2 drawer deletion.

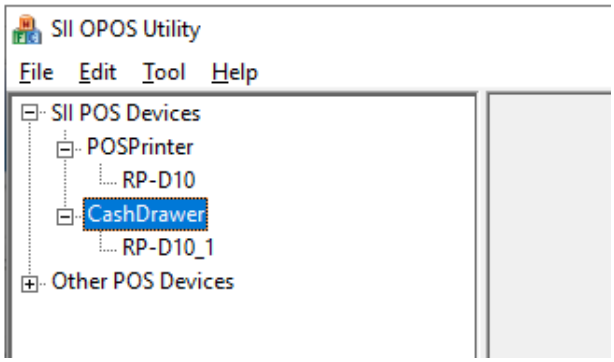
1) Select the printer or the cash drawer which is to be deleted from "Device view". Select [Delete] in the menu opened by clicking [Edit] menu or right clicking "Device view".



2) Confirm the warning message about deletion of the logical device name, and select [OK].



3) Selected device is deleted from "Device view".



When a printer is selected, not only the printer but also the entire printer-connected cash drawer and each logical device name are deleted. When a cash drawer is selected, the selected cash drawer and the logical device name are deleted.

3 OPOS Control

3.1 General

The application uses the OPOS Control according to the procedure below.

1) **Open** method: Call to link the Control Object to the Service Object.

2) **ClaimDevice** method: Call to gain exclusive access to the device.*1

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

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

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

6) **ReleaseDevice** method: Call to release exclusive access to the device.*1

7) **Close** method: Call to release the Service Object from the Control Object.

*1: This procedure is required for POS Printer Control but not for shared use by Cash Drawer Control.

For more information on the usage, refer to the document "UPOS V1.13".

3.2 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 performance is not supported.
POS Printer Control supports synchronous/asynchronous output based on the general output model.
Also, the printer controlled by the POS Printer Control is the device used exclusively.

3.3 Cash Drawer Control

The device controlled by the Cash Drawer Control can be shared. When the Cash Drawer Control is opened and enabled by the multiple applications, all these applications can use the properties and methods of the Cash Drawer Control.

When a certain application gains the exclusive control right of the Cash Drawer Control, only that application can call the **OpenDrawer** method and the **WaitForDrawerClose** method of the Cash Drawer Control.

4 POS Printer Control Interface Specification

4.1 Summary

4.1.1 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 POSPrinter Control 1.13.xxx [Public, by CRM/RCS-Dayton]"
ControlObjectVersion	Long	R	-	1013xxx
DeviceDescription	String	R	Open	"SII RP-D10 POS Printer"*1 "SII RP-E10 POS Printer"*2
DeviceEnabled	Boolean	R/W	Open & Claim	FALSE
DeviceName	String	R	Open	"RP-D10 POS Printer"*1 "RP-E10 POS Printer"*2
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

Property Name	Type	Access	Availability Condition	Default
ServiceObjectDescription	String	R	Open	"SII RP-D10 POS Printer Service Object, Copyright (C) 20xx Seiko Instruments Inc."*1
				"SII RP-E10 POS Printer Service Object, Copyright (C) 20xx Seiko Instruments Inc. "*2
ServiceObjectVersion	Long	R	Open	1013xxx
State	Long	R	-	OPOS_S_CLOSED(1)

*1: In case of RP-D10
*2: In case of RP-E10

4.1.2 Specific Properties

(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)
CapConcurrentJrnRec	Boolean	R	Open	FALSE
CapConcurrentJrnSlp	Boolean	R	Open	FALSE
CapConcurrentPageMode	Boolean	R	Open	FALSE
CapConcurrentRecSlp	Boolean	R	Open	FALSE
CapCoverSensor	Boolean	R	Open	TRUE
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
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
CapRecNearEndSensor	Boolean	R	Open	FALSE[*1]
				TRUE[*2,*3]
CapRecPageMode	Boolean	R	Open	TRUE

Property Name	Type	Access	Availability Condition	Default
CapRecPapercut	Boolean	R	Open	TRUE
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
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
CapTransaction	Boolean	R	Open	TRUE
CartridgeNotify	Long	R/W	Open	PTR_CN_DISABLED (0)
CharacterSet	Long	R/W	Open, Claim, & Enable	999*3
CharacterSetList	String	R	Open	"437,850,852,858,860,863,865,932, 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	""
JrnCartridgeState	Long	R	Open, Claim, & Enable	PTR_CART_UNKNOWN (0x10000000)
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
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	48
RecLineCharsList	String	R	Open	"36,41,44,48,57,64,72"
RecLineHeight	Long	R/W	Open, Claim, & Enable	24

Property Name	Type	Access	Availability Condition	Default
RecLineSpacing	Long	R/W	Open, Claim, & Enable	30
RecLinesToPaperCut	Long	R	Open, Claim, & Enable	3*1
				4*2
RecLineWidth	Long	R	Open, Claim, & Enable	576*3
RecNearEnd	Boolean	R	Open, Claim, & Enable	FALSE*1
				Depends on the printer state*2
RecSidewaysMaxChars	Long	R	Open, Claim, & Enable	200
RecSidewaysMaxLines	Long	R	Open, Claim, & Enable	19
RotateSpecial	Long	R/W	Open	PTR_RP_NORMAL(1)
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

*1: In case of RP-D10
*2: In case of RP-E10
*3: It is possible to change by the configuration program.

The following properties for POS Printer Control are not supported.

CapConcurrentJrnRec	CapConcurrentRecSlp	CapJrnCartridgeSensor
CapConcurrentJrnSlp	CapJrn2Color	CapJrnColor
CapConcurrentPageMode	CapJrnBold	CapJrnDhigh
CapJrnDwide	CapJrnItalic	CapJrnUnderline
CapJrnDwideDhigh	CapJrnNearEndSensor	CapSlp2Color
CapJrnEmptySensor	CapJrnPresent	CapSlpBarCode
CapSlpBitmap	CapSlpCartridgeSensor	CapSlpDwide
CapSlpBold	CapSlpColor	CapSlpDwideDhigh
CapSlpBothSidesPrint	CapSlpDhigh	CapSlpEmptySensor
CapSlpFullslip	CapSlpNearEndSensor	CapSlpRight90
CapSlpItalic	CapSlpPageMode	CapSlpRotate180
CapSlpLeft90	CapSlpPresent	CapSlpRuledLine
CapSlpUnderline	JrnEmpty	JrnLineCharsList
JrnCartridgeState	JrnLetterQuality	JrnLineHeight
JrnCurrentCartridge	JrnLineChars	JrnLineSpacing
JrnLineWidth	SlpBitmapRotationList	SlpEmpty
JrnNearEnd	SlpCartridgeState	SlpLetterQuality
SlpBarCodeRotationList	SlpCurrentCartridge	SlpLineChars
SlpLineCharsList	SlpLineSpacing	SlpNearEnd
SlpLineHeight	SlpLineWidth	SlpPrintSide
SlpLinesNearEndToEnd	SlpMaxLines	SlpSidewaysMaxChars
SlpSidewaysMaxLines		

4.1.3 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 1.13.

4.1.4 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

4.1.5 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 1.13.

4.2 Data Characters and Escape Sequences

(1) Escape Sequence operated when specified

Name	Data	Remarks
Paper cut	ESC|[#]P	Cuts receipt. The placeholder '#' is replaced by an ASCII decimal string indicating the cut percentage. If 0 is specified, paper cut is not executed. If 1 to 99 is specified, a partial cut is executed. If 100 is specified, a full cut is executed. If '#' is omitted, or if a value greater than 100 is specified, a full cut is executed. This is ignored during rotated 90° right/left mode by **RotatePrint** method or during Page Mode by **PageModePrint** method.
Feed and Paper cut	ESC|[#]fP	Cuts receipt, after feeding the paper by the **RecLinesToPaperCut** lines. The placeholder '#' is defined by "Paper cut" escape sequence. This is ignored during rotated 90° right/left mode by **RotatePrint** method or during Page Mode by **PageModePrint** method.
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. If '#' is omitted, the data is regarded as print data instead of escape sequence and OPOS_SUCCESS(0) is returned when the **ValidateData** method is used. A value from 1 to 20 can be specified for '#'.
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*1	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. If '#' is omitted, then one line is fed. A value from 0 to 255 can be specified for '#'. If '#' exceeds this range, 255 lines are fed. This is ignored during rotated 90° right/left mode by **RotatePrint** method.
Feed units*1	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, 255 units are fed. This is ignored during rotated 90° right/left mode by **RotatePrint** method.
Feed reverse	ESC|[#]rF	Not supported.

Name	Data	Remarks
Pass through embedded data	ESC|[*]#E	Sends the characters following "#E" through to the device 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 device. A value from 0 to 65535 can be specified for '#'. If the '#' exceeds this range, transmission of embedded data is not executed. The **ValidateData** method returns OPOS_E_ILLEGAL(106). If '#' is omitted, 0 is assumed to be specified, and OPOS_SUCCESS(0) is returned. If the number of byte in the print data specified by '#' is not set after the escape sequence is specified, the available print data to send is only sent. (Example: If ESC|2E"a" is specified, only "a" is sent since only 1 byte is set for the character string.) Also, during rotated 90° right/left mode by **RotatePrint** method, 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, or if the string following R is improper, all the data within the range specified by '#' is discarded. The **ValidateData** method returns OPOS_E_ILLEGAL(106). Also, during rotated 90° right/left mode by the **RotatePrint** method, the width cannot be calculated exactly because data string specified by 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** method returns OPOS_E_ILLEGAL(106). This is ignored during rotated 90° right/left mode by **RotatePrint** method or during Page Mode by **PageModePrint** method.

*1: In terms of RP-D10, the line spacing is reduced according to the setting value of the paper saving in the printer. Therefore, the following printing processes are changed.
-The amount of feed specified by the "Feed lines" escape sequence (ESC|[#]lF) is smaller than the setting value of the **RecLineSpacing** property.
-The amount of feed specified by the "Feed units" escape sequence (ESC|[#]uF) is processed based on the setting value of the paper saving (When Mode1 is selected, paper is not fed at all).
However, the line spacing from last print line to receipt cut position is not reduced because paper is cut after executing the paper feed for saved dot lines.
Cut operation which executes the paper feed for saved dot lines is executed for "Paper cut" escape sequence (ESC|[#]P), "Feed and Paper cut" escape sequence (ESC|[#]fP), and the **CutPaper** method.

- **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 following R (definition of the barcode characteristics).

  The string following R specifies the barcode characteristics using lowercase alphabet letters and numbers. The available numbers are the constant values defined for the **PrintBarCode** method.

  The following characters are used to indicate the attributes.

  s: symbology (barcode type)

  h: height (barcode height)

  w: width (barcode width)

  a: alignment (position of barcode)

  t: text position (position of HRI string)

  d: start of data (start position of barcode data)

  e: end of data (end position of barcode data)

  Attributes must be written in the above order. Every attribute is mandatory. If 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 the **PrintBarCode** method.

  The following example prints UPC-A with the conditions of center, HRI string printed below the barcode, 200 dots height, and 400 dots width.

  ESC|[*]33Rs101h200w400a-2t-13d123456789012e

- **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 following dL (definition of the ruled line characteristics).

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

  The following characters are used to indicate the attributes.

  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)

  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 the **DrawRuledLine** method.

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

(2) Escape Sequence operated during printing

It is characteristic that the state is kept until it is changed explicitly.

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

(3) Escape Sequence operated when printing

It is characteristic that it is reset at the end of each print method or by a "Normal" 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.
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 the ValidateData method returns OPOS_E_FAILURE(111).
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 the ValidateData method returns OPOS_E_FAILURE(111).

Name	Data	Remarks
RGB Color	ESC|[#]fC	Not supported.
Center	ESC|cA	Aligns following text 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 method or during Page Mode by **PageModePrint** method.
Right justify	ESC|rA	Aligns the subsequent texts 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 method or during Page Mode by **PageModePrint** method.
Left justify	ESC|lA	Aligns the subsequent texts 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 method or during Page Mode by **PageModePrint** method.
Normal	ESC|N	Restores printer characteristics to normal condition.
SubScript	ESC|[!]tbC	Not supported.
SuperScript	ESC|[!]tpC	Not supported.
Strike-through	ESC|[!][#]stC	Not supported.

4.3 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 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.

This document specifies those properties and method parameters that are affected by **BinaryConversion** in the individual property and method descriptions. The following line is added to their description:

"In the OPOS environment, the format of this data depends on the value of the **BinaryConversion** property. See the **BinaryConversion** property for details."

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 7-4 of the data byte. Second character = 0x30 + bits 3-0 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 the **Open** method.

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

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 FALSE: The function that compares firmware versions is not supported.

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

CapPowerReporting Property

Syntax **LONG CapPowerReporting;**

Remarks Identifies the reporting capabilities of the device.

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

Value	Meaning
OPOS_PR_STANDARD(1)	The Service Object can determine and report two of the power states – OFF_OFFLINE (that is, off or offline) and ONLINE.

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

CapStatisticsReporting Property

Syntax **BOOL CapStatisticsReporting;**

Remarks TRUE: The device can accumulate and provide various statistics regarding usage. The information accumulated and reported is device specific and is retrieved using the **RetrieveStatistics** method.

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

CapUpdateFirmware Property

Syntax **BOOL CapUpdateFirmware;**

Remarks FALSE: Firmware update is not supported.

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

CapUpdateStatistics Property

Syntax **BOOL CapUpdateStatistics;**

Remarks TRUE: Able to reset some of the device statistics to 0 by using the **ResetStatistics** method.
This property is initialized to TRUE by the **Open** method.

CheckHealthText Property

Syntax **BSTR CheckHealthText;**

Remarks Holds the results of the immediately preceding call to the **CheckHealth** method. The following examples illustrate 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** method.

Claimed Property

Syntax **BOOL Claimed;**

Remarks 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 the **Open** method.

ControlObjectDescription Property

Syntax **BSTR ControlObjectDescription;**

Remarks "OPOS POSPrinter Control 1.13.xxx [Public, by CRM/RCS-Dayton]" is set.
The property identifies the Control Object.
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 provides devices and related information.
Depends on the printer, the default is different.

Printer	Default
RP-D10	"SII RP-D10 POS Printer"
RP-E10	"SII RP-E10 POS Printer"

This property is initialized by the **Open** method.

DeviceEnabled Property R/W

Syntax **BOOL DeviceEnabled;**

Remarks TRUE: The device has been placed in an operational 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 disabled.

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

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

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

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 This property provides devices and related information.
Depends on the printer, the default is different.

Printer	Default
RP-D10	"RP-D10 POS Printer"
RP-E10	"RP-E10 POS Printer"

This property is initialized by the **Open** method.

FreezeEvents Property R/W

Syntax **BOOL FreezeEvents;**

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

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

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

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

OpenResult Property

Syntax **LONG OpenResult;**

Remarks Holds additional details about the most recent **Open** method. 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 the **Open** method.

OutputID Property

Syntax **LONG OutputID;**

Remarks Holds the identifier of the most recently started asynchronous output (call to an asynchronous method when the **AsyncMode** property 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.

PowerNotify Property R/W

Syntax **LONG PowerNotify;**

Remarks Contains the type of power notification selection made by the Application.
The values indicating power notification capabilities are as follows.

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 the **Open** method. This value provides compatibility with earlier releases.

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

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

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

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

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

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 the **Open** method is called, it returns the value OPOS_E_CLOSED(101).

The **ResultCode** values are:

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)	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 repaired.
OPOS_E_DISABLED(105)	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 device 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.

Value	Meaning
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.
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 the **ResultCodeExtended** property.

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 the OPOS Control-specific error information value are set.
When the **ResultCode** is set to other value, a value of 0 is set.

The OPOS Control takes the following values.

Value	Meaning
OPOS_EPTR_COVER_OPEN (201)	The 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 connected device during the bitmap is rotated 90° to the right/left by the **RotatePrint** method. For the height, the range not exceeding the maximum value of the connected device 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_AUTOCUTTER (1002)	Autocutter 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)

ServiceObjectDescription Property

Syntax **BSTR ServiceObjectDescription;**

Remarks A character string that identifies the Service Object is set to this property.
Depends on the printer, the default is different.

Printer	Default
RP-D10	Example: "SII RP-D10 POS Printer Service Object, Copyright (C) 20xx Seiko Instruments Inc."
RP-E10	Example: "SII RP-E10 POS Printer Service Object, Copyright (C) 20xx Seiko Instruments Inc."

This property is initialized by the **Open** method.

ServiceObjectVersion Property

Syntax **LONG ServiceObjectVersion;**

Remarks This property holds the Service Object version number.
This property is initialized by the **Open** method.

State Property

Syntax **LONG State;**

Remarks Contains the current state of the Control.

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 the **Open** method is successfully completed.

4.4 Specific Properties

AsyncMode Property R/W

Syntax **BOOL AsyncMode;**

Remarks TRUE: The print methods, **PrintNormal**, **CutPaper**, **PageModePrint**, **PrintBarCode**, **PrintBitmap**, **PrintMemoryBitmap**, **DrawRuledLine**, **RotatePrint**, and **TransactionPrint** are executed asynchronously.
FALSE: The methods are executed synchronously.

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

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

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 Page932, including ASCII characters 0x20 through 0x7F and the one-byte katakana characters 0xA1 through 0xDF. It also includes the Shift-JIS code characters, Levels 1 and 2.

This property is initialized to PTR_CCS_KANJI(11) by the **Open** method.

CapCoverSensor Property

Syntax **BOOL CapCoverSensor;**

Remarks TRUE: The printer has a "Cover open sensor".

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

CapMapCharacterSet Property

Syntax **BOOL CapMapCharacterSet;**

Remarks FALSE: The Service Object cannot exactly map the characters to the character sets defined in the **CharacterSetList** property.

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

CapRec2Color Property

Syntax **BOOL CapRec2Color;**

Remarks FALSE: The receipt cannot support 2 color printing.

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

CapRecBarCode Property

Syntax **BOOL CapRecBarCode;**

Remarks TRUE: The receipt has barcode printing capability.

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

CapRecBitmap Property

Syntax **BOOL CapRecBitmap;**

Remarks TRUE: The receipt can print bitmaps.

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

CapRecBold Property

Syntax **BOOL CapRecBold;**

Remarks TRUE: The receipt can print bold characters.

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

CapRecCartridgeSensor Property

Syntax **LONG CapRecCartridgeSensor;**

Remarks 0: Receipt Cartridge monitoring sensors are not supported.

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

CapRecColor Property

Syntax **LONG CapRecColor;**

Remarks PTR_COLOR_PRIMARY(0x00000001): Receipt color cartridges are not supported.

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

CapRecDhigh Property

Syntax **BOOL CapRecDhigh;**

Remarks TRUE: The receipt can print double high characters.

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

CapRecDwide Property

Syntax **BOOL CapRecDwide;**

Remarks TRUE: The receipt can print double wide characters.

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

CapRecDwideDhigh Property

Syntax **BOOL CapRecDwideDhigh;**

Remarks TRUE: The receipt can print double high/double wide characters.

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

CapRecEmptySensor Property

Syntax **BOOL CapRecEmptySensor;**

Remarks TRUE: The receipt has an "Out-of-paper sensor".

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

CapRecItalic Property

Syntax **BOOL CapRecItalic;**

Remarks FALSE: The receipt cannot support Italic characters.

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

CapRecLeft90 Property

Syntax **BOOL CapRecLeft90;**

Remarks TRUE: The receipt can print in a rotated 90° left mode.

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

CapRecMarkFeed Property

Syntax **LONG CapRecMarkFeed;**

Remarks 0: The receipt station cannot support the Control function for the marked thermal paper.

The Control function for the marked thermal paper is indicated. The Control function for the marked thermal paper is not supported.

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

CapRecNearEndSensor Property

Syntax **BOOL CapRecNearEndSensor;**

Remarks TRUE: The POS Printer doesn't have a "paper-near-end sensor".
FALSE: The POS Printer has a "paper-near-end sensor".

This property provides the different default depending on printers.

Printer	Default
RP-D10	FALSE.
RP-E10	TRUE: In the configuration program, the setting of NearEndSensor is enabled. FALSE: In the configuration program, the setting of NearEndSensor is disabled.

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

CapRecPageMode Property

Syntax **BOOL CapRecPageMode;**

Remarks TRUE: The printer is capable of supporting Page Mode for the receipt station.

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

CapRecPapercut Property

Syntax **BOOL CapRecPapercut;**

Remarks TRUE: The receipt can execute paper cuts.

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

CapRecPresent Property

Syntax **BOOL CapRecPresent;**

Remarks TRUE: Receipt can be specified for the station.

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

CapRecRight90 Property

Syntax **BOOL CapRecRight90;**

Remarks TRUE: The receipt can print in a rotated 90° right mode.

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

CapRecRotate180 Property

Syntax **BOOL CapRecRotate180;**

Remarks TRUE: The receipt can print in a rotated upside down mode.

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

CapRecRuledLine Property

Syntax **LONG CapRecRuledLine;**

Remarks Indicates the printable ruled lines of the receipt. A logical OR combination 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 combination of the above values(3) by the **Open** method.

CapRecStamp Property

Syntax **BOOL CapRecStamp;**

Remarks FALSE: The receipt cannot support stamp printing.

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

CapRecUnderline Property

Syntax **BOOL CapRecUnderline;**

Remarks TRUE: The receipt can print underlined characters.

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

CapTransaction Property

Syntax **BOOL CapTransaction;**

Remarks TRUE: The POS Printer Control transactions are supported.

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

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 values.

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

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

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

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.
This property is initialized to 999[*1] when the device is first enabled following the **Open** method.

It has one of the following values:

Value	Meaning
437	Selects Code Page437 character set.
850	Selects Code Page850 character set.
852	Selects Code Page852 character set.
858	Selects Code Page858 character set.
860	Selects Code Page860 character set.
863	Selects Code Page863 character set.
865	Selects Code Page865 character set.
932	Selects Katakana as the Code Page932 character set (Shift-JIS code).
PTR_CS_ANSI (999) / PTR_CS_WINDOWS (999)	Sets the Windows ANSI characters.
1250	Selects Code Page1250 character set.
1251	Selects Code Page1251 character set.
1252	Selects Code Page1252 character set.
1253	Selects Code Page1253 character set.
1254	Selects Code Page1254 character set.

・ This default can be changed by modifying registry setting.

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

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

CharacterSetList Property

Syntax **BSTR CharacterSetList;**

Remarks This is a character string of character set number.
"437,850,852,858,860,863,865,932,999,1250,1251,1252,1253,1254" is set.

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

CoverOpen Property

Syntax **BOOL CoverOpen;**

Remarks 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.
It has one of the following values:

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 the **ResultCode** property is OPOS_E_EXTENDED(114) and the **ResultCodeExtended** property 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)	Receipt printer

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

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

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)
"Vp voltage error"	OPOS_EPTR_VPPOWER(1001)
"Autocutter error"	OPOS_EPTR_AUTOCUTTER(1002)
"Cover open error"	OPOS_EPTR_COVER_OPEN(201)
"Out-of-paper error"	OPOS_EPTR_REC_EMPTY(203)
"Head temperature error"	OPOS_EPTR_HEAD_TEMP(1005)
"Failure"	OPOS_E_FAILURE(111)
"Time out"	OPOS_E_TIMEOUT(112)

Values in the above table are arranged in order of priority. If multiple errors occur at the same time, the higher priority item is set to **ErrorString** property.

This property is set by the Control just before the notification of **ErrorEvent**. If no description is 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 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 the **State** property is already set to OPOS_S_IDLE(2) when the **FlagWhenIdle** property is set to TRUE, then a **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 the **Open** method.

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

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

FontTypefaceList Property

Syntax **BSTR FontTypefaceList;**

Remarks Holds the string to specify the fonts and/or font typefaces that are supported by the printer. An empty string indicates that only the default typeface is supported. This property is initialized to an empty string by the **Open** method.

MapCharacterSet Property R/W

Syntax **BOOL MapCharacterSet;**

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

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

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 spacing.

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 Property	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)

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

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

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

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

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.
When PTR_S_RECEIPT(2) is specified for **PageModeStation** property, one of the following values is set to this property. (When **MapMode** = PTR_MM_DOTS(1))

Value	Meaning
"360,2400"	The **RecLineWidth** is the **PageModeArea** value when 360 is set.
"432,2400"	The **RecLineWidth** is the **PageModeArea** value when 432 is set.
"512,2400"	The **RecLineWidth** is the **PageModeArea** value when 512 is set.
"576,2400"	The **RecLineWidth** is the **PageModeArea** value when 576 is set.

Specify PTR_S_RECEIPT(2) for the **PageModeStation** property before accessing this property. When a valid station is not specified, an empty string is returned.

PageModeDescriptor Property

Syntax **LONG PageModeDescriptor;**

Remarks The Page Mode functionality available on the station specified for the **PageModeStation** property is indicated by OR of the following values.

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.

Specify PTR_S_RECEIPT(2) for the **PageModeStation** property before accessing this property. This property returns 0 until a valid station is specified. It is initialized to OR of the above values when PTR_S_RECEIPT(2) is specified.

PageModeHorizontalPosition Property R/W

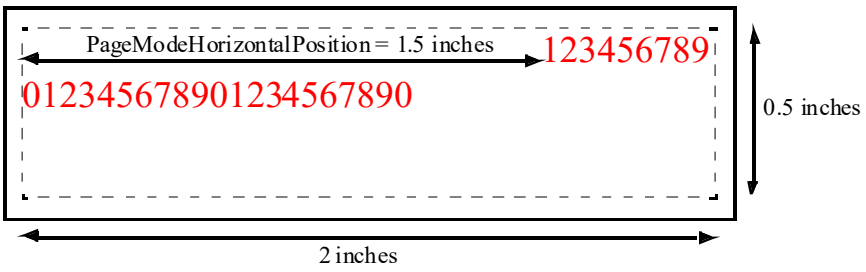
Syntax **LONG PageModeHorizontalPosition;**

Remarks Holds the offset value for correcting the horizontal start position within the Page Mode print area for the selected **PageModeStation**. This is expressed in the unit specified for the **MapMode** property. The horizontal direction is the same as the actual **PageModePrintDirection** property. 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** property before accessing this property. When a valid station is not specified, 0 is returned.

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

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to 2 inches by 0.5 inches
myptr.setPageModePrintArea("0,0,2000,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.setPageModeHorizontalPosition(1500);
myptr.printNormal(PTR_S_RECEIPT, "12345678901234567890123456789O\n");
```

The code sample above will generate the following receipt.



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

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

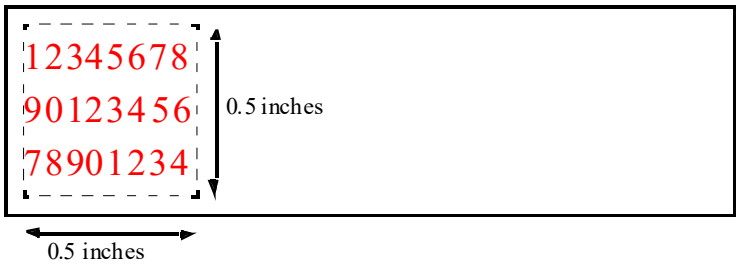
PageModePrintArea Property R/W

Syntax **BSTR PageModePrintArea;**

Remarks Holds the Page Mode print area for the selected **PageModeStation** property expressed in the unit specified by **MapMode**. The maximum print area is the page area.
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 "50,100,200,400," then the station print area is a rectangle beginning at the top left position (50,100), and the bottom right position (249,499).
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 the **PageModeStation** property before accessing this property. This property returns an empty string until a valid station 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 value is stored in the **ResultCode** property.

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;**

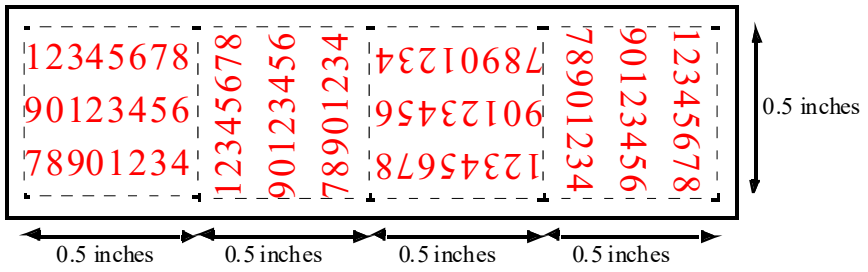
Remarks Holds the print direction. The print direction shall be as follows:

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 the **PageModeHorizontalPosition** and **PageModeVerticalPosition** properties. 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("0,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789012345678901234567890\n");
myptr.setPageModePrintArea("500,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_BOTTOM_TO_TOP);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789012345678901234567890\n");
myptr.setPageModePrintArea("1000,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_RIGHT_TO_LEFT);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789012345678901234567890\n");
myptr.setPageModePrintArea("1500,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_TOP_TO_BOTTOM);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789012345678901234567890\n");
```

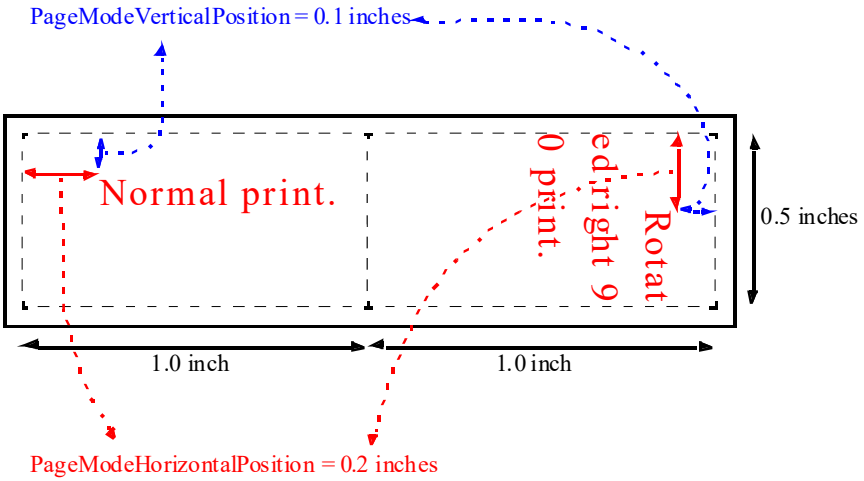
The code sample above will generate the following receipt.



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.pageModeVerticalPosition(100);
myptr.pageModeHorizontalPosition(200);
myptr.setPageModePrintArea("0,0,1000,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.printNormal(PTR_S_RECEIPT, "Normal print.\n");
myptr.setPageModePrintArea("1000,0,1000,500");
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 the **PageModeStation** property before accessing this property. This property returns 0 until a valid station 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 value is stored in the **ResultCode** property.

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. This will change the property values that are specific to each station. The target station for the **PageModePrint** method is also changed. The available station is PTR_S_RECEIPT(2) only.
This property is initialized to 0 by the **Open** method. Be sure to specify PTR_S_RECEIPT(2) for this property before accessing the property or method of the Page Mode function.

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

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

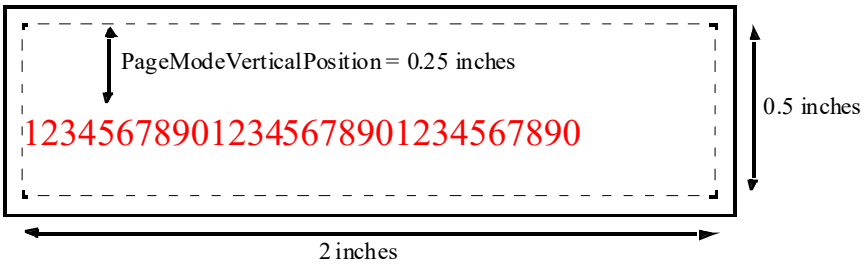
PageModeVerticalPosition Property R/W

Syntax **LONG PageModeVerticalPosition;**

Remarks Holds the offset value for correcting the vertical start position within the Page Mode print area for the selected **PageModeStation**. This is expressed in the unit specified for the **MapMode** property. The vertical direction is perpendicular to the direction specified in the actual **PageModePrintDirection** property. If the exact position cannot be supported then the position is set to the closest supported value. 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 the **PageModeVerticalPosition** property.

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to 2 inches by 0.5 inches
myptr.setPageModePrintArea("0,0,2000,500");
myptr.setPageModePrintDirection(PTR_PD_LEFT_TO_RIGHT);
myptr.setPageModeVerticalPosition(250);
myptr.printNormal(PTR_S_RECEIPT,"12345678901234567890123456789O\n");
```

The code sample above will generate the following receipt.



Specify PTR_S_RECEIPT(2) for the **PageModeStation** property before accessing this property. When a valid station is not specified, 0 is returned.

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

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 the **Open** method. 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 the **Open** method. 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 following the **Open** method.

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 following the **Open** method.

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

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 TRUE: The receipt is out of paper.
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 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 following the **Open** method.

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

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

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

RecLineWidth	RecLineChars	Print Font (H × W)	Character space	RecLineHeight
360	30 (Default)	Font A (24 × 12 dots)	0 dots	24
	40	Font B (16 × 8 dots)	1 dot	16
432	27	Font A (24 × 12 dots)	4 dots	24
	30		2 dots	
	33		1 dot	
	36 (default)		0 dots	
	43	Font B (16 × 8 dots)	2 dots	16
	48		1 dot	
	54		0 dots	
512	42 (default)	Font A (24 × 12 dots)	0 dots	24
	56	Font B (16 × 8 dots)	1 dot	16

RecLineWidth	RecLineChars	Print Font (H × W)	Character space	RecLineHeight
576	36	Font A (24 × 12 dots)	4 dots	24
	41		2 dots	
	44		1 dot	
	48 (default)		0 dots	
	57	Font B (16 × 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 when the paper width is 58mm/432dots, then the Service Object selects a **RecLineChars** = 43.)

If the character width is not supported, then an error is returned. (For example, if 60 is set when the paper width is 58mm/432dots, then an error occurs.)
Setting **RecLineChars** may also update **RecLineHeight**, and **RecLineSpacing**.
The value of **RecLineChars** is initialized to the default value when the device is first enabled following the **Open** method.

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

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

RecLineCharsList Property

Syntax **BSTR RecLineCharsList;**

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

Value	Meaning
"30,40"	The **RecLineCharsList** value when the **RecLineWidth** is set to 360.
"27,30,33,36,43,48,54"	The **RecLineCharsList** value when the **RecLineWidth** is set to 432.
"42,56"	The **RecLineCharsList** value when the **RecLineWidth** is set to 512.
"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 receipt printed line height expressed in the unit of measure given by **MapMode**.
When setting a value to this property cause an error because this property is automatically updated by the change of **RecLineChars**.
When **RecLineChars** is changed, **RecLineHeight** is updated to the height of the print font that corresponds to the set **RecLineChars**. The relationship between **RecLineHeight** and the print font is as follows.
(When the **MapMode** property is PTR_MM_DOTS(1))

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

See the **RecLineChars** property for the relationship **RecLineChars** and the print font. The value of **RecLineHeight** is initialized to 24 when the device is first enabled following the **Open** method.

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

Value	Meaning
OPOS_E_ILLEGAL(106)	The content of property cannot be changed.

RecLineSpacing Property R/W

Syntax **LONG RecLineSpacing;**

Remarks Holds the spacing of each single-high print line, the printed line height plus the line spacing height. This property is expressed in the unit defined 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.
Also, the configurable range differs depending on the setting of **RecLineChars**. The configurable ranges are as follows. (When the **MapMode** property is PTR_MM_DOTS(1))

RecLineWidth	RecLineChars	RecLineSpacing	Default Value
360	30	24 to 255	30
	40	16 to 255	
432	27	24 to 255	
	30		
	33		
	36		
	43	16 to 255	
	48		
	54		
512	42	24 to 255	
	56	16 to 255	

RecLineWidth	RecLineChars	RecLineSpacing	Default Value
576	36	24 to 255	30
	41		
	44		
	48		
	57	16 to 255	
	64		
	72		

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 and if 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 the **MapMode** property is PTR_MM_DOTS(1)) when the device is first enabled following the **Open** method.

In terms of RP-D10, when the setting of paper saving is enabled, the line spacing and the height of the whitespace between each pair of lines is printed with the values specified in the setting of paper saving. Therefore, the line spacing will not be as specified in the **RecLineSpacing** property. For details about the setting of paper saving, see "RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

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

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property 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 fed before the receipt is cut.
It depends on the printer. The distance between the print head and the cutter is obtained by dividing this distance by the amount of line count shown on **RecLineSpacing** is set. Therefore, this property can be changed by changing **RecLineSpacing**. This property provides the different default depending on printers.

Printer	Default
RP-D10	3
RP-E10	4

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

RecLineWidth Property

Syntax **LONG RecLineWidth;**

Remarks Holds the width of a line of **RecLineChars**. Expressed in the unit given by **MapMode**.
This property is initialized to one of the following values depending on the setting of PaperSize in the configuration program when the device is first enabled following the **Open** method. (When the **MapMode** property is PTR_MM_DOTS(1))

PaperSize	RecLineWidth
58 mm/360 dots	360
58 mm/432 dots	432
80 mm/512 dots	512
80 mm/576 dots	576

RecNearEnd Property

Syntax **BOOL RecNearEnd;**

Remarks TRUE: The receipt is low.
FALSE: The receipt is not low.

This property provides different default depending on the printer.

Printer	Default
RP-D10	FALSE
RP-E10	Depends on the printer

This property is initialized while the device is enabled.

In terms of RP-E10, if **CapRecNearEndSensor** is FALSE, this property is always set in FALSE. If **CapRecNearEndSensor** is TRUE, keep the near end state. If **RecEmpty** property is TRUE, **RecNearEnd** 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** property and print font when the device is first enabled following the **Open** method.

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

Example: When the **PageModeArea** property is "576,2400", and the **RecLineChars** property is "48"

RecSidewaysMaxChars Property
= 2400 / (12 + 0) = 200 (The decimal part is truncated.)

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).
The **RecLineWidth** property is the value calculated from the **RecLineSpacing** property. Therefore, this property can be changed by changing the **RecLineSpacing** property.
This property is initialized when the device is first enabled following the **Open** method.

RotateSpecial Property R/W

Syntax **LONG RotateSpecial;**

Remarks Holds the rotation orientation for barcodes.
This property is initialized to PTR_RP_NORMAL(1) by the **Open** method.
It has one of the following values:

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

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

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

4.5 Common Methods

CheckHealth Method

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

The *Level* parameter 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** property, and **DeviceName** property of the OPOS 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** property, and **DeviceName** property of the OPOS Printer Control.

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

Return One of the following values is returned and stored in the **ResultCode** property:

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. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property to TRUE.
OPOS_E_ILLEGAL(106)	Unsupported *Level* parameter is specified.
OPOS_E_NOHARDWARE(107)	The device 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 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 the **ResultCodeExtended** property is notified.

ClaimDevice Method

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

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

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

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	Exclusive access is granted. The **Claimed** property 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* parameter is specified.
OPOS_E_TIMEOUT(112)	Another application has exclusive access to the device and the *Timeout* (in millisecond) has elapsed before the device is released. Or, the device is not available before the *Timeout* (in millisecond) has elapsed.

ClearOutput Method

Syntax **LONG ClearOutput ();**

Remarks Calls 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.

When the Bluetooth model is used, it takes about 10 seconds before device is ready for use after **ClearOutput**.

Return One of the following values is returned by the method and also stored in the **ResultCode** property:

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

Close Method

Syntax **LONG Close ();**

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

Return One of the following values is returned and stored in the **ResultCode** property:

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 stored in the **ResultCode** property:

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

DirectIO Method

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

Parameter	Remarks
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 the **BinaryConversion** property. See **BinaryConversion** property for details.

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

1. Remaining memory capacity response

Command	PTR_DI_GET_REMAINING_MEMORY (3)
pData	Not used
pString	OUT　Remaining memory
	Issues the remaining memory response command and returns its response in character (numeral) strings. Asynchronous output is not possible. When **AsyncMode** = TRUE, synchronous output is possible if the **State** property is OPOS_S_IDLE(2).

2. Status response

Command	PTR_DI_GET_STATUS_DATA (501)
pData	IN　Status type 1:Status response of paper sensor
pString	OUT　Status
	Paper sensor status (n=1) RP-D10 '0': Paper is ready. '1': No paper. RP-E10

	Paper end	**Paper-near-end**
'0'	Paper is ready.	Paper is ready.
'1'	No paper.	Paper is ready.
'2'	Paper is ready.	No paper.
'3'	No paper.	No paper.

If the paper-near-end sensor is set in disabled, the paper-near-end sensor becomes Paper is ready status.
When **AsyncMode** = TRUE, synchronous output is possible if the **State** property is OPOS_S_IDLE(2).

3. International character select

Command	PTR_DI_SET_INTERNATIONAL_CHARACTER(201)
pData	IN International character number: n 0 ≤ n ≤ 12
pString	Not used
	Selects the international character. International character select by this function continues until the **CharacterSet** property is changed. When the **CharacterSet** property is changed, the default is set. When **AsyncMode** = TRUE, synchronous output is possible if the **State** property is OPOS_S_IDLE(2). The corresponding countries for international character number (n) are as below. 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

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The **DirectIO** method is successfully completed.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property to TRUE.
OPOS_E_ILLEGAL(106)	Parameter has an error or an invalid command number is specified.
OPOS_E_NOHARDWARE(107)	The device 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 the **ResultCodeExtended** property is notified.

Open Method

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

The *DeviceName* parameter specifies the device name to open. Specify the registered device name (such as "RP-D10") or the Logical Device Name of the POS printer to execute this method.

Remarks Calls this method to open the device.
When the **Open** method is successful, the common property and other class-specific properties are initialized.
When the Bluetooth model is used, wait at least 15 seconds after **Close** method before **Open** method.

Return One of the following values is returned and stored in the **ResultCode** property:

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_FAILER(111)	Initialization of the OPOS Driver is failed.

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

1. When OPOS Control is closed and the **Open** method is failed:
   The **ResultCode** property will continue to return OPOS_E_CLOSED(101).

2. When the OPOS Control is already opened:
   The **Open** method will return OPOS_E_ILLEGAL(106), but the **ResultCode** property may continue to return the value held before the **Open** method.

ReleaseDevice Method

Syntax **LONG ReleaseDevice ();**

Remarks Calls this method to release exclusive access to the device. If the **DeviceEnabled** property is TRUE and the device is an exclusive-use device, then the device is first disabled. Do not execute this while the event is in progress (or in the event handler).

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	Exclusive access has been released. The **Claimed** property 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	Remarks
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 manufacturer 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	Remarks
ReceiptLineFeedCount	Number of receipt line feeds performed (unit: 100 dot-line)
HoursPoweredCount	Number of hours powered on (unit: hour)
PaperCutCount	Number of paper cuts

Return One of the following values is returned and stored in the **ResultCode** property.

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) At least one of the specified statistics could not be reset.

RetrieveStatistics Method

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

Parameter	Remarks
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 manufacturer 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. The **CapStatisticsReporting** property 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="1.13.0" 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 the manufacturer-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 empty.

The available statistics are as follows.

XML Definition Name	Remarks
UnifiedPOSVersion	Version of the UnifiedPOS specification supported
DeviceCategory	Device category
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
InstallationDate	Device installation date
HoursPoweredCount	Number of hours powered on (unit: hour)
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
PaperCutCount	Number of paper cuts
FailedPaperCutCount	Number of failed paper cuts
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)
PaperCutCount_Accumulated	Number of paper cuts (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 stored in the **ResultCode** property:

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

UpdateFirmware Method

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

Remarks This method is not supported.

Return The following value is returned by the method and also stored in the **ResultCode** property:

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 stored in the **ResultCode** property:

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

4.6 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 stored in the **ResultCode** property:

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 stored in the **ResultCode** property:

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 stored in the **ResultCode** property:

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 the **PageModePrintArea** property. The entire page may be cleared by setting the **PageModePrintArea** to be the same as the **PageModeArea** and then by using **ClearPrintArea** method.
Specify PTR_S_RECEIPT(2) for the **PageModeStation** property before calling this method.

Return One of the following values is returned and stored in the **ResultCode** property:

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

CutPaper Method

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

Parameter	Remarks
Percentage	Indicates the percentage of the paper to be cut. The value 100 causes a full paper cut. Other values between 1 and 99 request a partial cut percentage. If 0 is specified, paper cut is not executed.

Remarks Calls to cut the receipt.
This method is executed synchronously if **AsyncMode** is FALSE, and asynchronously if **AsyncMode** is TRUE.
Paper cut can also be executed by using "Paper cut" escape sequence (ESC|[#]P) when calling **PrintNormal** or **PrintImmediate** methods.
If printing data remains in the printer buffer, paper cut is executed after all buffered data is printed.
During rotated 90° right/left mode by **RotatePrint** method, or while Page Mode by **PageModePrint** method is selected, the method is failed and paper cut is not executed.

Due to the positions of printer head and cutter, paper cut might be executed at the middle of printing data. To avoid this, call this method after feeding paper for the value of **RecLinesToPaperCut** property.

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property to TRUE.
OPOS_E_ILLEGAL(106)	An invalid percentage was specified. Specify the number between 0 and 100. Or, it is also notified when the method is not supported, for example, during rotated 90° right/left mode by **RotatePrint** method, or while in the Page Mode by **PageModePrint** method.
OPOS_E_NOHARDWARE(107)	The device 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** property 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 the **ResultCodeExtended** property is notified.

DrawRuledLine Method

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

Parameter	Remarks
Station	Station to be used. PTR_S_RECEIPT(2) is specified.
PositionList	Specifies the position of ruled line.
LineDirection	Specifies the direction of ruled line. Available values are PTR_RL_HORIZONTAL(1) (horizontal ruled line) and PTR_RL_VERTICAL(2) (vertical ruled line).
LineWidth	Specifies the width of ruled line. Specify 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 for the specified station.
This method is executed synchronously if **AsyncMode** is FALSE, and asynchronously if **AsyncMode** is TRUE.
How to specify the *PositionList* parameter 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 the **MapMode** property.

These are examples of the *PositionList* parameter.

· "0,500"
This setting draws a horizontal ruled line 500 dots long starting at the 0th dot from the left.

· "0,200;300,100"
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 100 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 the **MapMode** property. These are examples of the *PositionList* parameter.

· "0,100,400,500"
This setting draws vertical ruled lines at the 0th, 100th, 400th, and 500th dot from the left. The valid range of the drawing position of vertical ruled line is 0 to (**RecLineWidth** property value – *LineWidth* parameter 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.

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

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

Return One of the following values is returned and stored in the **ResultCode** property:

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

EndInsertion Method

Syntax **LONG EndInsertion ();**

Remarks This method is not supported.

Return The following value is returned by the method and also stored in the **ResultCode** property:

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 stored in the **ResultCode** property:

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

MarkFeed Method

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

Remarks This method is not supported.

Return The following value is returned by the method and also stored in the **ResultCode** property:

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

PageModePrint Method

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

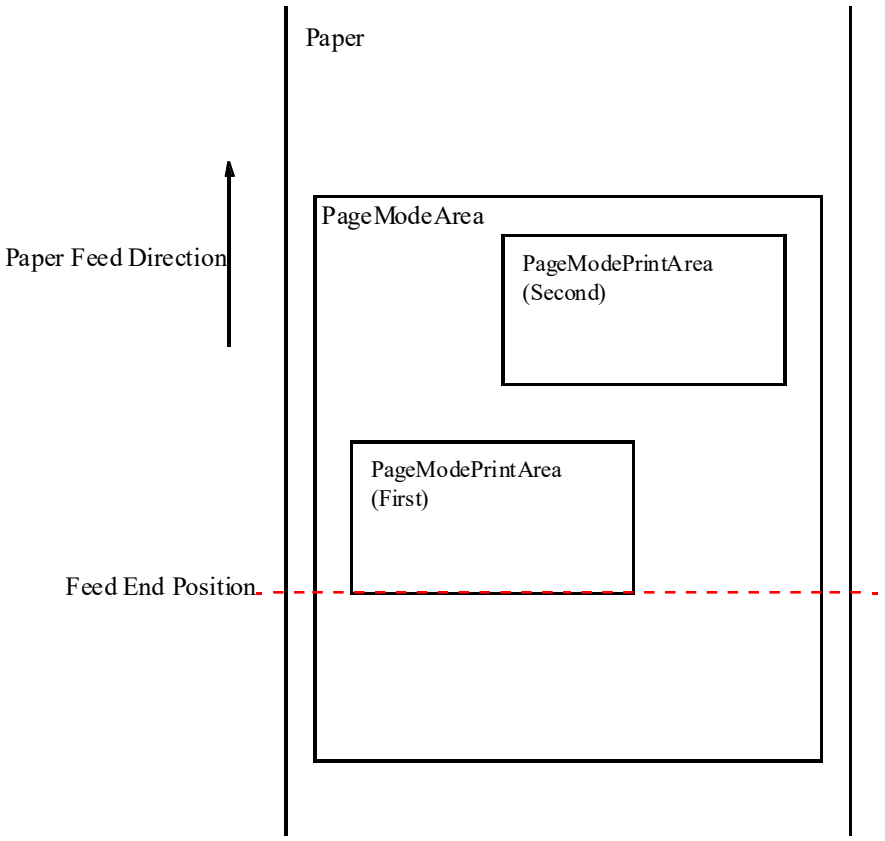
The values of the *Control* parameter are as follows.

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

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

If *Control* is PTR_PM_PAGE_MODE(1), then Page Mode is entered. Subsequent calls to **PrintNormal**, **PrintBarCode**, **PrintBitmap**, and **PrintMemoryBitmap** will buffer the print data until **PageModePrint** is called with the Control parameter set to PTR_PM_PRINT_SAVE(2), PTR_PM_NORMAL(3), or PTR_PM_CANCEL(4). (In this case, the print methods only validate the method parameters and buffer the data they do not initiate printing. Also, the value of the **AsyncMode** property does not affect their operation. No **OutputID** will be assigned to the request, nor will an **OutputCompleteEvent** be queued.)
If *Control* is specified to PTR_PM_PRINT_SAVE(2), then Page Mode is not exited. If some data is buffered by calls to the methods **PrintNormal**, **PrintBarCode**, **PrintBitmap**, and **PrintMemoryBitmap**, then the buffered data is saved and printed. This control is used to print the same page layout with additional print items inside of the page.

If *Control* is PTR_PM_NORMAL(3), then Page Mode is exited. If some data is buffered by calls to the methods **PrintNormal**, **PrintBarCode**, **PrintBitmap**, and **PrintMemoryBitmap**, then the buffered data is printed. The buffered data will not be saved.
If *Control* is PTR_PM_CANCEL(4), then Page Mode is exited. If some data is buffered by calls to the methods **PrintNormal**, **PrintBarCode**, **PrintBitmap**, and **PrintMemoryBitmap**, then the buffered data is not printed and is not saved.
Note that when the **PageModePrint** method is called, 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 Page Mode print area. If more than one **PageModePrintArea** is defined, then after the **PageModePrint** method is called, all of the data that is to be printed in their respective **PageModePrintArea**(s) will be printed and the paper will be fed to the end of page of the **PageModePrintArea** located the farthest "down" the sheet of paper. (See figure below).



The entire Page Mode transaction is treated as one message. This method is executed synchronously if **AsyncMode** property is FALSE, and asynchronously if **AsyncMode** is TRUE.
Calling the **ClearOutput** method cancels Page Mode to return to the normal state. Any buffered print lines are also cleared.
Page Mode can be used within a transaction print, but not within a rotate print.
Specify PTR_S_RECEIPT(2) for the **PageModeStation** property before calling this method.

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_ILLEGAL(106)	The station specified by the **PageModeStation** property does not exist, or **CapRecPageMode** is FALSE. Or, while the station specified by the **PageModeStation** property 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 the *Control* parameter.
OPOS_E_NOHARDWARE(107)	The device 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. (It is returned only when **AsyncMode** is FALSE, and PTR_PM_NORMAL(3), PTR_PM_PRINT_SAVE(2), or PTR_PM_CANCEL(4) is specified to the *Control* parameter.)
OPOS_E_EXTENDED(114)	Except for OPOS_EPTR_TOOBIG(206) and OPOS_EPTR_BADFORMAT(207), the error defined by the **ResultCodeExtended** property is notified.

Value	Meaning
Other values	Refer to the description of the **ResultCode** property.

PrintBarCode Method

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

Parameter	Remarks
Station	Station to be used. PTR_S_RECEIPT(2) is specified.
Data	Character string of barcode. The format of this data depends on the value of the **BinaryConversion** property. See **BinaryConversion** property for details.
Symbology	Barcode type to be used. See the table of *Symbology* below.
Height	Barcode height. Expressed in the unit given by **MapMode**. For PTR_MM_DOTS(1), specify the value between 1 and 255. For QR Code, although *Height* is ignored, specify the value between 1 and 255 as well. During Page Mode by **PageModePrint** method, specify the value within the range of print area defined by the **PageModePrintArea** property and the **PageModeVerticalPosition** property.
Width	Barcode width. Expressed in the unit given by **MapMode**. The actual printed barcode width is the best size that fits into the width specified by *Width*. For PTR_MM_DOTS(1), specify the value between 1 and **RecLineWidth**. When the **RotateSpecial** property and printing barcode in a rotated 90° to right/left mode by **RotatePrint** method are executed, specify the value not exceeding the maximum value of the connected devices (see the **PageModeArea** property for the maximum value). During Page Mode by **PageModePrint** method, specify the value within the range of print area defined by the **PageModePrintArea** property and the **PageModeHorizontalPosition** property.
Alignment	Position of the barcode. See the table of *Alignment* below.
TextPosition	Position of the character string. See the table of *TextPosition* below.

The values of the *Symbology* parameter are as follows.

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)

Value	Meaning
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_PDF417(201)	PDF417
PTR_BCS_QRCODE(204)	QR Code (Mixed mode)

The values of the *Alignment* parameter are as follows.

Value	Meaning
PTR_BC_LEFT(-1)	Left justify
PTR_BC_CENTER(-2)	Center
PTR_BC_RIGHT(-3)	Right justify
Other value	Distance from the left edge where barcode printing starts. Expressed in the unit given by **MapMode**.

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

The values of the *TextPosition* parameter are as follows.

Value	Meaning
PTR_BC_TEXT_NONE(-11)	Does not print the text. 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 Calls this method in order to print the barcode at the specified POS Printer.
This method is executed synchronously if **AsyncMode** is FALSE, and asynchronously if **AsyncMode** is TRUE.

Hereinafter, the limitations for each parameter of the barcode are described when the **MapMode** property 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* = 113 × X 226 ≤ *Width* ≤ 678 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* = 65 × X 130 ≤ *Width* ≤ 390 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' - '9'. The 8th letter does not affect the barcode printing data.
Width	*Width* = 81 × X 162 ≤ *Width* ≤ 486 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' - '9'. The 13th letter does not affect the barcode printing data.
Width	*Width* = 113 × X 226 ≤ *Width* ≤ 678 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' - '9'. When 15 letters or 18 letters are inputted, the 13th character does not affect the printing data.
Width	- For 14 or 15 letters *Width* = 138 × X 276 ≤ *Width* ≤ 828 - For 17 or 18 letters *Width* = 165 × X 330 ≤ *Width* ≤ 990 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' - '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+(10×2)) × X) (14 × D+56) ≤ *Width* ≤ (54 × D+174) D: the number of barcode character 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	One of 'A' - 'D' must be specified for the end of the 1st line and at least one of '0' - '9,' '$,' '+,' ':,' '-,' .',' or '/' must be specified.
Width	*Width* = (((6 × X)+(2 × X × N)) × D)+((X × N-X) × D')+ (X × (10×2-1)) (20 × D+2 × D'+38) ≤ *Width* ≤ (72 × D+12 × D'+114) D: the number of barcode character D': the number of 'A' to 'D', '+', '.', '/', ':' included in the barcode character 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 1 letter consisting of '0' - '9', 'A' - 'Z', 'space', '$', '%', '+', '-', '.', '/' must be specified.
Width	*Width* = (((X × 7)+(X × N × 3))×(D+2))+((10×2-1) × X) (26 × D+90) ≤ *Width* ≤ (96 × D+306) D: the number of barcode character 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	See the parameter list below.
Width	*Width* = X × ((10×2)+((D+2+2)×9)+1) (18 × D+114) ≤ *Width* ≤ (54 × D+342) D: the number of barcode character X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *Width*.

・ When printing Code93, if all the configurable data parameter are specified, the value of the **BinaryConversion** property must be specified by OPOS_BC_NIBBLE(1) or OPOS_BC_DECIMAL(2). This is because the data includes the symbology character. Regarding the decimal numbers specified for *Data*, 0x00 represents 0 and 0x01 represents 1.

Number	Character	Number	Character
0	0	24	O
1	1	25	P
2	2	26	Q
3	3	27	R
4	4	28	S
5	5	29	T
6	6	30	U
7	7	31	V
8	8	32	W
9	9	33	X
10	A	34	Y
11	B	35	Z
12	C	36	-
13	D	37	.
14	E	38	SPACE
15	F	39	$

Number	Character	Number	Character
16	G	40	/
17	H	41	+
18	I	42	%
19	J	43	($)
20	K	44	(%)
21	L	45	(/)
22	M	46	(+)
23	N	-	-

[Code128]

Parameter	Limitation
Data	See the parameter list below. The first letter of the first line should be decimal number, 103(0x67), 104(0x68), 105(0x69), after that at least one letter should be included.
Width	*Width* = X × ((10×2)+((D+2)×11)+2) (22 × D+88) ≤ *Width* ≤ (66 × D+264) 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, if all the configurable data parameter are specified, the value of the **BinaryConversion** property must be specified by OPOS_BC_NIBBLE(1) or OPOS_BC_DECIMAL(2). This is because the data includes the symbology character. Regarding the decimal numbers specified for *Data*, 0x00 represents 0 and 0x01 represents 1.

Number	Character			Number	Character		
	CODE A	CODE B	CODE C		CODE A	CODE B	CODE C
0	SPACE	SPACE	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

Number	Character			Number	Character		
	CODE A	CODE B	CODE C		CODE A	CODE B	CODE C
15	/	/	15	68	EOT	d	68
16	0	0	16	69	ENG	e	69
17	1	1	17	70	ACK	f	70
18	2	2	18	71	BEL	g	71
19	3	3	19	72	BS	h	72
20	4	4	20	73	HT	i	73
21	5	5	21	74	LF	j	74
22	6	6	22	75	VT	k	75
23	7	7	23	76	FF	l	76
24	8	8	24	77	CR	m	77
25	9	9	25	78	SO	n	78
26	:	:	26	79	SI	o	79
27	;	;	27	80	DLE	p	80
28	<	<	28	81	DC1	q	81
29	=	=	29	82	DC2	r	82
30	>	>	30	83	DC3	s	83
31	?	?	31	84	DC4	t	84
32	@	@	32	85	NAK	u	85
33	A	A	33	86	SYN	v	86
34	B	B	34	87	ETB	w	87
35	C	C	35	88	CAN	x	88
36	D	D	36	89	EM	y	89
37	E	E	37	90	SUB	z	90
38	F	F	38	91	ESC	{	91
39	G	G	39	92	FS	|	92
40	H	H	40	93	GS	}	93
41	I	I	41	94	RS	~	94
42	J	J	42	95	US	DEL	95
43	K	K	43	96	FNC3	FNC3	96
44	L	L	44	97	FNC2	FNC2	97
45	M	M	45	98	SHIFT	SHIFT	98
46	N	N	46	99	CODE C	CODE C	99
47	O	O	47	100	CODE B	FNC4	CODE B
48	P	P	48	101	FNC4	CODE A	CODE A
49	Q	Q	49	102	FNC1	FNC1	FNC1
50	R	R	50	103	START(CODE A)		
51	S	S	51	104	START(CODE B)		
52	T	T	52	105	START(CODE C)		

[PDF417]

Parameter	Limitation
Data	0x00 to 0x7F must follow the ASCII code and 0x80 to 0xFF must follow the extended character set of PC437 English list.
Width	*Width* = (17C+69) × X+(2X × 2) *Height* = RY+(2X × 2) X: module width Y: module height R: the number of row C: the number of vertical column For the number of row and the number of vertical column, 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 row and the number of vertical column are determined.

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

[QR Code]

Parameter	Limitation
Data	ASCII 0x00 to 0xFF and Shift JIS code based on JIS X 0208.
Width	*Width* = (4V+17) × X+(4X × 2) V: Version of QR Code (1 to 40) X: Module width and height (2 to 11) 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* parameter is selected after the version is determined.

・ QR Code model is fixed at 2 and an 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.

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property 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 large. - This *Symbology* is not supported. - The values of *Alignment* is improper. - The values of *TextPosition* is improper. - The specified data is not printable. - The data stored in the specified *Width* parameter cannot be printed.

Value	Meaning
OPOS_E_NOHARDWARE(107)	The device 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** property 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 the **ResultCodeExtended** property is notified.

PrintBitmap Method

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

Parameter	Remarks
Station	Station to be used. PTR_S_RECEIPT(2) is specified.
FileName	Name of Windows bitmap file. Only the uncompressed format is supported (Full path or relative path must be specified). Monochrome or full color (24 bit) format file can be specified.
Width	Print width of bitmap. See values below.
Alignment	Print position of bitmap. See values below.

The values of *Width* parameter are as follows.

Value	Meaning
PTR_BM_ASIS(-11)	Prints the bitmap with one pixel per printer dot of the printer.
Other values	Bitmap width. Expressed in the unit given by **MapMode**. If **MapMode** is PTR_MM_DOTS(1), specify between 1 and **RecLineWidth** property. When printing bitmap in a rotated 90° to right/left mode by **RotatePrint** method is executed, specify the value not exceeding the maximum value of the connected devices (see the **PageModeArea** property for the maximum value). During Page Mode by **PageModePrint** method, specify the value within the range of print area defined by the **PageModePrintArea** property and the **PageModeHorizontalPosition** property.

・ The value of bitmap width can be rounded up to a multiple of 8 within the OPOS Control. Specify the bitmap width within the range so that the converted value does not exceed the print area.

The values of the *Alignment* parameter are as follows.

Value	Meaning
PTR_BM_LEFT(-1)	Left justify
PTR_BM_CENTER(-2)	Center
PTR_BM_RIGHT(-3)	Right justify
Other values	Distance from the left edge where bitmap printing starts. Expressed in the unit given by **MapMode**.

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

Remarks This method is called when bitmap is printed on the specified printer.
The highest performance cannot be achieved since the bitmap data is transferred to the printer after **PrintBitmap** is called. It is recommended to print the bitmap data using **SetBitmap** and escape sequence.
This method is executed synchronously if **AsyncMode** is FALSE, and asynchronously if **AsyncMode** is TRUE.
The *Width* parameter 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 dot of the POS Printer. 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 and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property 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 large. - *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** property. - 32 bits colored bitmap is set.
OPOS_E_NOHARDWARE(107)	The device 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. (The **State** property 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 the **ResultCodeExtended** is notified.

PrintImmediate Method

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

Parameter	Remarks
Station	Station to be used. PTR_S_RECEIPT(2) is specified.
Data	The 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 the **BinaryConversion** property. See **BinaryConversion** property for details.

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

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

In terms of RP-D10, when the setting of paper saving is enabled in the printer, the value specified in the setting of paper saving in the printer is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed. (The value specified by the **RecLineSpacing** property is ignored).
However, when any of the "Paper cut" escape sequence (ESC|[#]P), the "Feed and Paper cut" escape sequence (ESC|[#]fP), or the **CutPaper** method is executed after the paper is fed by the carriage return (CR) or line feed (LF), distance from the last print line to the cut position is not reduced because paper is cut after executing the paper feed for saved dot lines.

Remarks Calls this method to print the *Data* on the printer immediately. The print 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** method and **RotatePrint** method. If printing data remains in the printer buffer, printing is executed after all the buffered data is printed.

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property 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	Remarks
Station	Station to be used. PTR_S_RECEIPT(2) is specified.
Data	Byte array holding the bitmap data. For the supported image file, see the **PrintBitmap** method. The format of this data depends on the value of the **BinaryConversion** property. See **BinaryConversion** property for details.
Type	Bitmap format. Only PTR_BMT_BMP (1) is supported for the format of the data indicated by the *Data* parameter.
Width	Print width of bitmap. See values below.
Alignment	Print position of bitmap. See values below.

The values of *Width* parameter are as follows.

Value	Meaning
PTR_BM_ASIS(-11)	Prints the bitmap with one pixel per printer dot of the printer.
Other values	Bitmap width. Expressed in the unit given by **MapMode**. If **MapMode** is PTR_MM_DOTS(1), specify between 1 and **RecLineWidth** property. When printing bitmap in a rotated 90° to right/left mode by **RotatePrint** method is executed, specify the value not exceeding the maximum value of the connected devices (see the **PageModeArea** property for the maximum value). During Page Mode by **PageModePrint** method, specify the value within the range of print area defined by the **PageModePrintArea** property and the **PageModeHorizontalPosition** property.

・ The value of bitmap width can be rounded up to a multiple of 8 within the OPOS Control. Specify the bitmap width within the range so that the converted value does not exceed the print area.

The values of the *Alignment* parameter are as follows.

Value	Meaning
PTR_BM_LEFT(-1)	Left justify
PTR_BM_CENTER(-2)	Center
PTR_BM_RIGHT(-3)	Right justify
Other values	Distance from the left edge where bitmap printing starts. Expressed in the unit given by **MapMode**.

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

Remarks This method is called when the bitmap held on the memory is printed on the specified printer.

For the operation specifications, see the **PrintBitmap** method.

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property 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 large. - *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** property. - 32 bits colored bitmap is set.
OPOS_E_NOHARDWARE(107)	The device 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. (The **State** property 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 the **ResultCodeExtended** is notified.

PrintNormal Method

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

Parameter	Remarks
Station	Station to be used. PTR_S_RECEIPT(2) is specified.
Data	The 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 the **BinaryConversion** property. See **BinaryConversion** property for details.

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

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

Value	Meaning
CR & LF	Carriage return (CR) is ignored. Operation of line feed (LF) is executed once.

In terms of RP-D10, when the setting of paper saving is enabled in the printer, the value specified in the setting of paper saving in the printer is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed (The value specified by the **RecLineSpacing** property is ignored).
However, when any of the "Paper cut" escape sequence (ESC|[#]P), the "Feed and Paper cut" escape sequence (ESC|[#]fP), or the **CutPaper** method is executed after the paper is fed by the carriage return (CR) or line feed (LF), distance from the last print line to the cut position is not reduced because paper is cut after executing the paper feed for saved dot lines.

Remarks Calls this method to print the *Data* on POS Printer. The print data that exceeds the maximum number of characters per line is printed on the next print line.
If printing 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 and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property to TRUE.
OPOS_E_ILLEGAL(106)	The specified printer does not exist.
OPOS_E_NOHARDWARE(107)	The device 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. (The **State** property 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 the **ResultCodeExtended** property 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 stored in the **ResultCode** property:

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

RotatePrint Method

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

Parameter	Remarks
Station	Station to be used. PTR_S_RECEIPT(2) is specified.
Rotation	Direction of rotation. See values below.

The values of *Rotation* are as follows.

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 ORed with one of the above start rotated print values.
PTR_RP_BITMAP(8192)	Starts rotated bitmap printing. This value is ORed with one of the above start rotated print values. Rotates the bitmap printed by **PrintBitmap**.
PTR_RP_NORMAL(1)	End of 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* parameter set to PTR_RP_NORMAL(1). Lines are printed in the order that they are sent, 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** method 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** method makes upside-down bitmap.

* Caution for rotating 180°

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

If *Rotation* is PTR_RP_RIGHT90(257) or PTR_RP_LEFT90(258), the horizontal writing mode starts. Until **RotatePrint** is called by setting *Rotation* parameter to PTR_RP_NORMAL(1), the data called by **PrintNormal** method is buffered. (In this case, the data of the above method is only buffered and not printed. Also, the value of **AsyncMode** property does not affect its operation. In other words, no **OutputID** is assigned and no **OutputCompleteEvent** is informed.) If PTR_RP_BARCODE (4096) is set in *Rotation*, barcode printing by **PrintBarCode** method is buffered. If PTR_RP_BITMAP(8192) is set in *Rotation*, bitmap printing by the **PrintBitmap** method is buffered.
If *Rotation* is PTR_RP_NORMAL(1), rotated print mode is end. If some print data is buffered during the sideways rotated print is valid, the buffered data is printed.

Service Object calculates so that the width in the horizontal writing mode becomes best size. The maximum width is 2400 dots in the horizontal writing 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 escape sequence is specified on the **PrintNormal** method during the horizontal writing mode, the print data is buffered regardless of the *Rotation* parameter.

If the transaction by **TransactionPrint** method is started, the horizontal writing mode is cleared as well as **ClearOutput** method. This is because buffering by the **TransactionPrint** method is prior to buffering by the **RotatePrint** method. In order to batch print in the horizontal writing mode, execute the **TransactionPrint** method, execute the **RotatePrint** method to complete the rotated print mode, and then clear the batch print by the **TransactionPrint** method.

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

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property to TRUE.
OPOS_E_ILLEGAL(106)	The specified printer does not exist.
OPOS_E_NOHARDWARE(107)	The device 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. (The **State** property 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 the **ResultCodeExtended** property is notified.

SetBitmap Method

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

Parameter	Remarks
BitmapNumber	The number to be assigned to this bitmap. The valid bitmap numbers are 1 through 20.
Station	Station to be used. PTR_S_RECEIPT(2) is specified.
FileName	Name of Windows bitmap file. The compressed format is not supported (Full path or relative path must be specified). Monochrome and full color (24bit) format is also available. If an empty string is set, the bitmap setting is canceled.
Width	Print width of bitmap. See the following values.
Alignment	Print position of bitmap. See **PrintBitmap** for values.

The values of *Width* parameter are as follows.

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

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

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property to TRUE.
OPOS_E_ILLEGAL(106)	One of the following parameter errors occurred. - The values of *BitmapNumber* is improper. - *Station* does not exist. - *Station* does not support bitmap printing. - *Width* is too large. - *Alignment* is invalid or too big. When absolute position is specified, the total including print bitmap width exceeds the range of **RecLineWidth**. - 32 bits colored bitmap is set.
OPOS_E_NOHARDWARE(107)	The device 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.
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 the **State** property is OPOS_S_BUSY(3) or OPOS_S_ERROR(4)).
OPOS_E_EXTENDED(114)	The error defined by the **ResultCodeExtended** is notified.

SetLogo Method

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

Parameter	Remarks
Location	The logo to be set. PTR_L_TOP(1) or PTR_L_BOTTOM(2).
Data	The 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 the **BinaryConversion** property. See **BinaryConversion** property for details.

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

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

In terms of RP-D10, when the setting of paper saving is enabled in the printer, the value specified in the setting of paper saving in the printer is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed (The value specified by the **RecLineSpacing** property is ignored).
However, when any of the "Paper cut" escape sequence (ESC|[#]P), the "Feed and Paper cut" escape sequence (ESC|[#]fP), or the **CutPaper** method is executed after the paper is fed by the carriage return (CR) or line feed (LF), distance from the last print line to the cut position is not reduced because paper is cut after executing the paper feed for saved dot lines.

Remarks Calls this method to save a data string as a top or bottom logo.
The logo is printed by calling the **PrintNormal** or **PrintImmediate** methods including the escape sequence of the top or bottom logo into the print data.

The *Data* registered by this method is kept by the character of **CharacterSet** property at the time when the method is executed. Therefore, **SetLogo** method is executed by the **CharacterSet** property of 932 to register the data including kanji character(s). After that, kanji character(s) will be printed even if the **CharacterSet** property is set to 999 when the data is printed by escape sequence.

Return One of the following values is returned and stored in the **ResultCode** property:

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

TransactionPrint Method

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

Parameter	Remarks
Station	Station to be used. PTR_S_RECEIPT(2) is specified.
Control	Transaction control. See values below.

The values of *Control* are as follows.

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

Remarks Calls 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**, and **DrawRuledLine** will buffer the print data at the Service Object until **TransactionPrint** is called with the *Control* parameter set to PTR_TP_NORMAL(12). (In this case, the print methods only validate the method parameters and buffer the data – they do not initiate printing. Also, the value of **AsyncMode** property does not affect its operation. In other words, no **OutputID** is assigned 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**, and **DrawRuledLine**, then the buffered data is printed. This method is executed synchronously if **AsyncMode** is FALSE, and asynchronously if **AsyncMode** is TRUE.
Calling **ClearOutput** method cancels transaction mode. Any buffered print lines are also cleared.
For the combination with **RotatePrint** method, refer to **RotatePrint** method.

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The method was successful.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnabled** property to TRUE.
OPOS_E_ILLEGAL(106)	The specified printer does not exist.
OPOS_E_NOHARDWARE(107)	The device 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. (The **State** property 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 the **ResultCodeExtended** property is notified.

ValidateData Method

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

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

Remarks Before calling the **PrintImmediate** or **PrintNormal** methods, this method is called when determining whether a data sequence, which 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 and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	The data is valid.
OPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Calls the **ClaimDevice** method to gain exclusive access.
OPOS_E_DISABLED(105)	Not enabled. Calls after setting **DeviceEnable**d property 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 90° to the left or right print mode.
OPOS_E_NOHARDWARE(107)	The device 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 returned are as follows.

Escape Sequence	Condition
Paper cut	One of the following statuses occurs. - Percentage '#' 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** method. - It is not supported during Page Mode by **PageModePrint** method.
Feed and Paper cut	One of the following statuses occurs. - Percentage '#' 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** method. - It is not supported during Page Mode by **PageModePrint** method.

Escape Sequence	Condition
Feed lines	One of the following statuses occurs. - The line number '#' 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** method. - It is not supported during Page Mode by **PageModePrint** method.
Feed units	- Feed unit number '#' is not precisely supported due to occurrence of rounding error of 1 dot depending on the setting of the **MapMode** property. - The feed unit number '#' 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** method. - It is not supported during Page Mode by **PageModePrint** method.
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** method. - It is not supported during Page Mode by **PageModePrint** method.
Right justify	One of the following statuses occurs. - It is not supported during rotated 90° right/left mode by **RotatePrint** method. - It is not supported during Page Mode by **PageModePrint** method.
Left justify	One of the following statuses occurs. - It is not supported during rotated 90° right/left mode by **RotatePrint** method. - It is not supported during Page Mode by **PageModePrint** method.

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

Escape Sequence	Condition
(General)	The format of the escape sequence is invalid.
Feed, Paper cut, and Stamp	Not supported.
Fire stamp	Not supported.
Print bitmap	The bitmap number '#' is not precisely supported.
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 in one line, and the data2 is printed in the next line.

4.7 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	Remarks
ResultCode	Factor code causing the error event. See *Result Code* 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 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_AUTOCUTTER(1002)	Autocutter error has occurred.
OPOS_EPTR_HEAD_TEMP(1005)	Head-temperature error has occurred.
OPOS_EPTR_UNRECOVERABLE(1010)	A non-recoverable error has occurred.

The content of the location specified by the *pErrorResponse* is preset to the default value 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 "ProcessCompletionTiming" is set in "0", this event prints the data not being transmitted to the printer. If "ProcessCompletionTiming" is set in "1", this event prints the unprinted data.
OPOS_ER_CLEAR(12)	Clears the asynchronous output. The error state is exited.

Remarks The error is detected and is notified when the POS Printer Control state shifts to the error state.

OutputCompleteEvent Event

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

The *OutputID* parameter indicates the ID number of the completed asynchronous output request.

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

StatusUpdateEvent Event

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

Status is set to one of the following values.

Value	Meaning
PTR_SUE_COVER_OPEN(11)	The printer cover is open.
PTR_SUE_COVER_OK(12)	The printer cover is closed.
PTR_SUE_REC_EMPTY(24)	The receipt is out of paper.
PTR_SUE_REC_NEAREMPTY(25)*1	The receipt is low.
PTR_SUE_REC_PAPEROK(26)	The 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). The **FlagWhenIdle** property 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)).

*1: Only RP-E10 is supported.

Remarks This event is reported when the printer device has an important state change.

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.

5 Cash Drawer Control Interface Specification

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	FALSE
CapUpdateFirmware	Boolean	R	Open	FALSE
CapUpdateStatistics	Boolean	R	Open	FALSE
CheckHealthText	String	R	Open	""
Claimed	Boolean	R	Open	FALSE
ControlObjectDescription	String	R	-	"OPOS CashDrawer Control 1.13.xxx [Public, by CRM/RCS-Dayton]."
ControlObjectVersion	Long	R	-	1013xxx
DeviceDescription	String	R	Open	"SII RP-D10 Cash Drawer"*1
				"SII RP-E10 Cash Drawer"*2
DeviceEnabled	Boolean	R/W	Open	FALSE
DeviceName	String	R	Open	"RP-D10 Cash Drawer"*1
				"RP-E10 Cash Drawer"*2
FreezeEvents	Boolean	R/W	Open	FALSE
OpenResult	Long	R	-	OPOS_SUCCESS(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

Property Name	Type	Access	Availability Condition	Default
ServiceObjectDescription	String	R	Open	"SII RP-D10 Cash Drawer Service Object, Copyright (C) 20xx Seiko Instruments Inc."*1
				"SII RP-E10 Cash Drawer Service Object, Copyright (C) 20xx Seiko Instruments Inc."*2
ServiceObjectVersion	Long	R	Open	1013xxx
State	Long	R	-	OPOS_S_CLOSED(1)

*1: In case of RP-D10
*2: In case of RP-E10

5.1.2 Specific Properties

Property Name	Type	Access	Availability Condition	Default*1
CapStatus	Boolean	R	Open	TRUE
CapStatusMultiDrawerDetect	Boolean	R	Open	FALSE
DrawerOpened	Boolean	R	Open & Enable	Depends on the cash drawer status

5.1.3 Common Methods

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

*1: The availability condition differs from that of UPOS V 1.13.

5.1.4 Specific Methods

Method Name	Availability Condition
OpenDrawer	Open & Enable
WaitForDrawerClose	Open & Enable

5.1.5 Events

Event Name	Occurrence Condition
DirectIOEvent	Open & Enable
StatusUpdateEvent	Open & Enable

5.2 Common Properties

BinaryConversion Property R/W

Syntax **LONG BinaryConversion;**

Remarks The value of **BinaryConversion** can be modified to one of the following values, but this affects no method.

Value	Meaning
OPOS_BC_NONE(0)	Data is placed one byte per BString character, with no conversion. (Default)
OPOS_BC_NIBBLE(1)	Each byte is converted into two characters.
OPOS_BC_DECIMAL(2)	Each byte is converted into three characters.

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

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

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 FALSE: The function that compares firmware versions is not supported.

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

CapPowerReporting Property

Syntax **LONG CapPowerReporting;**

Remarks Identifies the power notification capabilities of the Device. The values of **CapPowerReporting** are as follows.

Value	Meaning
OPOS_PR_STANDARD(1)	Two types of power states can be determined and notified.

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

CapStatisticsReporting Property

Syntax **BOOL CapStatisticsReporting;**

Remarks FALSE: No statistical data regarding the device is available.

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

CapUpdateFirmware Property

Syntax **BOOL CapUpdateFirmware;**

Remarks FALSE: Firmware update is not supported.

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

CapUpdateStatistics Property

Syntax **BOOL CapUpdateStatistics;**

Remarks FALSE: None of the statistical data can be reset/updated by the application.

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

CheckHealthText Property

Syntax **BSTR CheckHealthText;**

Remarks Holds the results of the immediately preceding call to the **CheckHealth** method. The results of diagnosis are as follows.

Value	Meaning
"Internal HCheck: Successful "	Succeeded in the health check without using the device.
"External HCheck: Successful"	Succeeded in the communication confirmation of the cash drawer and opening the drawer using the device.
"External HCheck: Failure"	Failed in the communication confirmation of the cash drawer or opening the drawer 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 the **CheckHealth** method.

Claimed Property

Syntax **BOOL Claimed;**

Remarks TRUE: The device is claimed for exclusive access.
FALSE: The device is released for sharing with other applications.

The value of this property is initialized to FALSE by the **Open** method.

ControlObjectDescription Property

Syntax **BSTR ControlObjectDescription;**

Remarks "OPOS CashDrawer Control 1.13.xxx [Public, by CRM/RCS-Dayton]" is set.
The property identifies the Control Object.
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 provides devices and related information.

Depends on the printer, the default is different.

Printer	Default
RP-D10	"SII RP-D10 Cash Drawer"
RP-E10	"SII RP-E10 Cash Drawer "

This property is initialized by the **Open** method.

DeviceEnabled Property R/W

Syntax **BOOL DeviceEnabled;**

Remarks TRUE: The device has been placed in an operational 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 disabled.

The application must set this property to TRUE before using output devices.
This property is initialized to FALSE by the **Open** method.

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

Value	Meaning
OPOS_SUCCESS(0)	The property was set successfully.
OPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
OPOS_E_FAILURE(111)	The connection port cannot be opened. Try again after checking that the port is not used by other programs or that the port exists.

DeviceName Property

Syntax **BSTR DeviceName;**

Remarks This property provides devices and related information.

Depends on the printer, the default is different.

Printer	Default
RP-D10	"RP-D10 Cash Drawer"
RP-E10	"RP-E10 Cash Drawer"

This property is initialized by the **Open** method.

FreezeEvents Property R/W

Syntax **BOOL FreezeEvents;**

Remarks 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 were 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.
This property is initialized to FALSE by the **Open** method.

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

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

OpenResult Property

Syntax **LONG OpenResult;**

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

Value	Meaning
OPOS_SUCCESS(0)	Successful open.
OPOS_OR_ALREADYOPEN(301)	The Control is already open.
OPOS_OR_REGBADNAME(302)	The registry does not contain a key for the specified device name.
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.

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

PowerNotify Property R/W

Syntax **LONG PowerNotify;**

Remarks Contains the type of power notification selection made by the Application.
The values indicating power notification capabilities are as follows..

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 the **Open** method. This value provides compatibility with earlier releases.

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

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 The device's current power state is set when it can be determined.
The values of **PowerState** are as follows.

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

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

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

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 the **Open** method is called, it returns the value OPOS_E_CLOSED(101). The **ResultCode** values are:

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_CLAIMED(102)	Attempt was made to access a device that is exclusively accessed by another process.
OPOS_E_NOSERVICE(104)	The Control cannot communicate with the Service Object. The software must be repaired.
OPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.

Value	Meaning
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 device is not connected to the system or is not powered on.
OPOS_E_FAILURE(111)	A communication error has occurred.
OPOS_E_TIMEOUT(112)	Timeout has occurred when the **ClaimDevice** method is executed. The exclusive access cannot be obtained within the preset time.

ResultCodeExtended Property

Syntax **LONG ResultCodeExtended;**

Remarks This property is initialized to 0 by the **Open** method. The extended error code is not supported.

Reference **ResultCode** Property

ServiceObjectDescription Property

Syntax **BSTR ServiceObjectDescription;**

Remarks A character string that identifies the Service Object is set to this property.
Depends on the printer, the default is different.

Printer	Default
RP-D10	Example: "SII RP-D10 Cash Drawer Service Object, Copyright (C) 20xx Seiko Instruments Inc."
RP-E10	Example]: "SII RP-E10 Cash Drawer Service Object, Copyright (C) 20xx Seiko Instruments Inc."

This property is initialized by the **Open** method.

ServiceObjectVersion Property

Syntax **LONG ServiceObjectVersion;**

Remarks This property holds the Service Object version number.
This property is initialized by the **Open** method.

State Property

Syntax **LONG State;**

Remarks Contains the current state of the Control.

Value	Meaning
OPOS_S_CLOSED(1)	The Control is closed. (Default)
OPOS_S_IDLE(2)	The Control is in a good state and is not busy.

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

5.3 Specific Properties

CapStatus Property

Syntax **BOOL CapStatus;**

Remarks TRUE: The open/close state of the cash drawer can be reported.

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

CapStatusMultiDrawerDetect Property

Syntax **BOOL CapStatusMultiDrawerDetect;**

Remarks FALSE: Not supported.

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

DrawerOpened Property

Syntax **BOOL DrawerOpened;**

Remarks TRUE: The cash drawer is open.*1
FALSE: The cash drawer is closed.*1

This property is initialized to an appropriate value when the device is enabled.

*1: The open/close state of the cash drawer can be reversed and then notified using the InvertDrawerStatus in the configuration program.

5.4 Common Methods

CheckHealth Method

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

The *Level* parameter 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 a health check without using the device physically. OPOS_SUCCESS is always returned.
OPOS_CH_EXTERNAL(2)	Opens the cash drawer, if possible, after confirming the communication with the drawer using the device. OPOS_SUCCESS is returned when it is opened successfully. This method fails when another application has exclusive access to the device.
OPOS_CH_INTERACTIVE(3)	Executes an interactive test of the device. The Service Object displays the modal dialog and opens the cash drawer, if possible. OPOS_SUCCESS is returned when it is opened successfully. This method fails when another application has exclusive access to the device.

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

Return One of the following values is returned and stored in the **ResultCode** property:

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_CLAIMED(102)	Another device has exclusive access.
OPOS_E_ILLEGAL(106)	Unsupported Level parameter is specified.
OPOS_E_NOHARDWARE(107)	The printer to which the cash drawer is connected is powered off or the cable is not connected. This value is notified only when OPOS_CH_EXTERNAL(2) is set.
OPOS_E_FAILURE(111)	A communication error has occurred. This value is notified only when OPOS_CH_EXTERNAL(2) is set.

ClaimDevice Method

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

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

Remarks Calls this method to request exclusive access to the device.
Acquisition of exclusive access is not essential since the cash drawer device is a sharable device.
When it is successful, the **Claimed** property is set to TRUE.

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	Exclusive access is granted and available device connection is established. The **Claimed** property 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* parameter is specified.
OPOS_E_TIMEOUT(112)	Another application has exclusive access to the device and the *Timeout* (in millisecond) has elapsed before the device is released. Or, the device is not available before the *Timeout* (in millisecond) has elapsed.

Close Method

Syntax **LONG Close ();**

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

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	Device has been disabled and closed.
Other values	Refer to the description of the **ResultCode** property.

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 stored in the **ResultCode** property:

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

DirectIO Method

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

Remarks Calls this method to communicate directly with the Service Object.
This method is not supported.

Return The following value is returned by the method and also stored in the **ResultCode** property:

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

Open Method

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

The *DeviceName* parameter specifies the device name to open. Specify the registered device name (such as "RP-D10_1") or the cash drawer Logical Device Name.

Remarks Calls this method to open the device.
When the **Open** method is successful, the common property and other class-specific properties are initialized.
When the Bluetooth model is used, wait at least 15 seconds after **Close** method before **Open** method.

Return One of the following values is returned and stored in the **ResultCode** property:

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. This includes the cases when *DeviceName* is set using an "(empty string)".
OPOS_E_FAILURE(111)	Initialization of the OPOS Driver is failed.

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

1. When OPOS Control is closed and the **Open** method is failed:
   The **ResultCode** property will continue to return OPOS_E_CLOSED(101).

2. When the OPOS Control is already opened:
   The **Open** method will return OPOS_E_ILLEGAL(106), but the **ResultCode** property may continue to return the value held before the **Open** method.

ReleaseDevice Method

Syntax **LONG ReleaseDevice ();**

Remarks Calls this method to release exclusive access to the device.
If the **DeviceEnabled** property is TRUE and the device is an exclusive-use device, then the device is first disabled.
Do not execute this while the event is in progress (or in the event handler).

Return One of the following values is returned and stored in the **ResultCode** property:

Value	Meaning
OPOS_SUCCESS(0)	Access has been released. The **Claimed** property is now FALSE.
OPOS_E_ILLEGAL(106)	The application does not have exclusive access to the device.
Other values	Refer to the description of the **ResultCode** property.

ResetStatistics Method

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

Remarks This method is not supported.

Return The following value is returned by the method and also stored in the **ResultCode** property:

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

RetrieveStatistics Method

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

Remarks This method is not supported.

Return The following value is returned by the method and also stored in the **ResultCode** property:

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

UpdateFirmware Method

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

Remarks This method is not supported.

Return The following value is returned by the method and also stored in the **ResultCode** property:

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 stored in the **ResultCode** property:

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

5.5 Specific Methods

OpenDrawer Method

Syntax **LONG OpenDrawer ();**

Remarks Opens the cash drawer. This method fails when another application has exclusive access to the device.

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

Value	Meaning
OPOS_SUCCESS(0)	The cash drawer was opened successfully.
OPOS_E_CLAIMED(102)	Another device has exclusive access.
OPOS_E_DISABLED (105)	Not enabled.
OPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
OPOS_E_FAILURE(111)	Communication with the device is failed.

WaitForDrawerClose Method

Syntax **LONG WaitForDrawerClose (LONG** *BeepTimeout*, **LONG** *BeepFrequency*, **LONG** *BeepDuration*, **LONG** *BeepDelay*);

Remarks This method is not supported.

Return The following value is returned and stored in the **ResultCode** property.

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

5.6 Events

DirectIOEvent Event

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

Remarks This event is not notified.

StatusUpdateEvent Event

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

Status is set to one of the following values.

Value	Meaning
CASH_SUE_DRAWERCLOSED(0)	The cash drawer is closed.*1
CASH_SUE_DRAWEROPEN(1)	The cash drawer is open.*1
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)).

*1: The open/close state of the cash drawer can be reversed and then notified using the InvertDrawerStatus in the configuration program.

Remarks This event is notified when the open/close state of the cash drawer is changed.

To notify the open/close state of the cash drawer, the following conditions must be satisfied.
- The **CapStatus** property is set to TRUE.
- The **DeviceEnabled** property is set to TRUE.

To notify the power state, the following conditions must be satisfied.
- The **PowerNotify** property is set to OPOS_PN_ENABLED(1).
- The **DeviceEnabled** property is set to TRUE.

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.

6 Registry Used by OPOS Control

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

Although the registry can be set manually, set it using the configuration program (OPOS Utility). The contents of registry are read at the time when the OPOS Control executes the **Open** method. Therefore, the changes made during the OPOS Control operation do not affect the operation. In order to update the new setting, call the **Open** method after calling the **Close** method in the OPOS Control once.

6.1 POS Printer Control

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

Device name key depends on the printer. It is added every POS printer by the configuration program.

Printer	Device Name key	Meaning
RP-D10	Example: "RP-D10", "RP-D10-xx"	Device Name of POS printer for RP-D10 which is added by the configuration program.
RP-E10	Example: "RP-E10", "RP-E10-xx"	Device Name of POS printer for RP-E10 which is added by the configuration program.

[Registry value list]

Value Name	Value	Meaning
Default	OPOS.POSPRINTER.SO.SII.CORE113.1	-
CoreSODLLPath	C:\Program Files\SII\OPOS \POSPrinterCoreSO113.dll	File name of POS Printer Service Object (Full path)
DefaultCharacterSet	999	Default of **CharacterSet** property
Description	SII RP-D10 POS Printer Service Object, Copyright (C) 20xx Seiko Instruments Inc.*1	Details of dedicated POS Printer Service Object for printer.
	SII RP-E10 POS Printer Service Object, Copyright (C) 20xx Seiko Instruments Inc.*2	
DriverName	SII RP-D10*1	Printer driver name Used for printer driver setting
	SII RP-E10*2	
LogFileNameCore	-	Log file name of POS Printer Service Object (Full path)
LogFileNameSub	-	Log file name of dedicated POS Printer Service Object for RP-D10 or RP-E10 (Full path)
LogFileSizeCore	0	Maximum log file size of POS Printer Service Object (KB)
LogFileSizeSub	0	Maximum log file size of dedicated POS Printer Service Object for RP-D10 or RP-E10 (KB)
LogLevelCore	-1	Log output level of POS Printer Service Object -1: No output 0: Error 1: Warning 2: Information 3: Debug 4: Trace
LogLevelSub	-1	Log output level of dedicated POS Printer Service Object for RP-D10 or RP-E10 -1: No output 0: Error 1: Warning 2: Information 3: Debug 4: Trace
NearEndSensor*2	1	Near-end sensor function 0: Enable 1: Disable
PaperSize	3	Paper width and number of effective dots selection 0: 58mm/360dots 1: 58mm/432dots 2: 80mm/512dots 3: 80mm/576dots

Value Name	Value	Meaning
PrintSpeed	0	Print Speed Specify 0: RecLetterQuality setting is Enable 1: Hi-speed mode fixed 2: Middle-speed (quality) mode fixed 3: Middle-speed (silent) mode fixed 4: Low-speed mode fixed
SendTimeout	10000	Send timeout (msec)
ReceiveTimeout	10000	Receive timeout (msec)
ProcessCompletion Timing	1	Process Completion 0: Data transmission 1: Timing at completion of the data printing
SubSODLLPath	C:\Program Files\SII\OPOS \RPD10POSPrinterSubSO113.dll*1	File name of dedicated POS Printer Service Object for RP-D10 or RP-E10 (Full path)
	C:\Program Files\SII\OPOS \RPE10POSPrinterSubSO113.dll*2	
Version	1.13.xx	Version of dedicated POS Printer Service Object for RP-D10 or RP-E10

*1: In case of RP-D10
*2: In case of RP-E10

6.2 Cash Drawer Control

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

Device name key depends on the printer. It is added every POS printer by the configuration program.

Printer	Device Name key	Meaning
RP-D10	Example: "RP-D10_ 1", "RP-D10_ 2", "RP-D10-xx_1", "RP-D10-xx_2"	Device Name of Cash Drawer for RP-D10 which is added by the configuration program.
RP-E10	Example: "RP-E10_ 1", "RP-E10_ 2", "RP-E10-xx_1", "RP-E10-xx_2"	Device Name of Cash Drawer for RP-E10 which is added by the configuration program.

[Registry value list]

Value Name	Value	Meaning
Default	OPOS.CASHDRAWER.SO.SII.CORE113.1	-
CoreSODLLPath	C:\Program Files\SII\OPOS \CashDrawerCoreSO113.dll	File name of Cash Drawer Service Object (Full path)
Description	SII RP-D10 Cash Drawer Service Object, Copyright (C) 20xx Seiko Instruments Inc.*1	Details of dedicated Cash Drawer Service Object for Printer
	SII RP-E10 Cash Drawer Service Object, Copyright (C) 20xx Seiko Instruments Inc.*2	
DrawerNo	1 or 2	Cash drawer number 1: Cash drawer1 2: Cash drawer2
DriverName	SII RP-D10*1	Printer driver name Used for printer driver setting
	SII RP-E10*2	
InvertDrawerStatus	F	Notification method of cash drawer open/close state F: Reversal is not active. The cash drawer is opened when the drawer switch input statusis Low. T: Reversal is active. The cash drawer is opened when the drawer switch input status is High.
LogFileNameCore	-	Log file name of Cash Drawer Service Object (Full path)
LogFileNameSub	-	Log file name of dedicated Cash Drawer Service Object for RP-D10 or RP-E10 (Full path)
LogFileSizeCore	0	Maximum log file size of Cash Drawer Service Object (KB)

Value Name	Value	Meaning
LogFileSizeSub	0	Maximum log file size of dedicated Cash Drawer Service Object for RP-D10 or RP-E10 (KB)
LogLevelCore	-1	Log output level of Cash Drawer Service Object -1: No output 0: Error 1: Warning 2: Information 3: Debug 4: Trace
LogLevelSub	-1	Log output level of dedicated Cash Drawer Service Object for RP-D10 or RP-E10 -1: No output 0: Error 1: Warning 2: Information 3: Debug 4: Trace
PulseTimer	1	ON/OFF pulse time for cash drawer drive 1 to 8 (× 100 millisecond)
SubSODLLPath	C:\Program Files\SII\OPOS \RPD10CashDrawerSubSO113.dll*1	File name of dedicated Cash Drawer Service Object for RP-D10 or RP-E10 (Full path)
	C:\Program Files\SII\OPOS \RPE10CashDrawerSubSO113.dll*2	
Version	1.13.xx	Version of dedicated Cash Drawer Service Object for RP-D10 or RP-E10

*1: In case of RP-D10
*2: In case of RP-E10

7 Header File

7.1 POS Printer Header File

The header file used in the OPOS Control is described below.
The constants defined uniquely in the OPOS Control are as follows.

- OPOS_EPTR_VPPOWER

- OPOS_EPTR_AUTOCUTTER

- OPOS_EPTR_HEAD_TEMP

- OPOS_EPTR_IMAGEAREA_FULL

- OPOS_EPTR_UNRECOVERABLE

- 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_AUTOCUTTER          =   PTRERREXT+2;
const LONG OPOS_EPTR_HEAD_TEMP            =   PTRERREXT+5;
const LONG OPOS_EPTR_IMAGEAREA_FULL   =   PTRERREXT+9;
const LONG OPOS_EPTR_UNRECOVERABLE   =   PTRERREXT+10;


//////////////////////////////////////////////////////////////////
// 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)
```