SII JavaPOS
Application Programmer's Guide

Rev.01

[Products]

RP-F10 Series

RP-E10 Series

RP-D10 Series

Seiko Instruments Inc.

Rev.01 May 2026

Copyright© 2026 by Seiko Instruments Inc. All rights reserved.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the U.S., Japan, and other countries.
Java® brand name is a registered trademark of Oracle.

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

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

SII is a trademark of Seiko Instruments Inc.

Introduction

This manual describes "SII JavaPOS" (hereinafter referred to as the "software") that controls the printer provided by Seiko Instruments Inc. (hereinafter referred to as "SII").

Target Products

This section lists the products supported by the software.

	Device Name	Description in This Manual
POSPrinter	RP-F10 RP-E10 RP-D10	Printer

See "UnifiedPOS Retail Peripheral Architecture Version 1.13 APPENDIX B Java for Retail POS" (hereinafter referred to as "UPOS APG V1.13") before using the software.

Notation in This Manual

The notation in this manual is described.

Operation and Display

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

- Screenshots and display layouts of Windows 11
- Operating instructions with a mouse and a keyboard

Operating System Abbreviations

The operating system abbreviations used in this manual are listed below.

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

Terms

The terms used in this manual are defined as below.

Definition	Description
UPOS APG V1.13	"UnifiedPOS Retail Peripheral Architecture Version 1.13 APPENDIX B Java for Retail POS"
POSPrinter	JavaPOS driver of POSPrinter Device Class. The device to be controlled is a printer.
CashDrawer	JavaPOS Driver of the CashDrawer Device Class. The device to be controlled is a drawer connected to the printer.
Printer	Printer body (printer device).
Drawer	Drawer body (drawer device) connected to the printer.
JCL	"Java Configuration Loader"
JER	"JavaPOS Entry Registry"
JVM	"Java Virtual Machine"
OS	Operating system
Initial value	The value immediately after satisfying the availability condition.
Line spacing	The height of each print line (total value of the printed line height and the whitespace between each pair of lines).

Symbols

The symbols used in this manual are described below.

Caution

◆ Notes and limitations are described.

Reference

- Supplemental information and related matters are described.

Table of Contents

Chapter 1 Overview 1-1

1.1 Configuratioin 1-1
1.1.1 Structural Diagram 1-1
1.2 Operating Environment 1-2
1.2.1 System Environment 1-2
1.2.2 Supported Models 1-2
1.3 Limitations 1-3
1.3.1 POSPrinter 1-3
1.3.2 CashDrawer 1-3

Chapter 2 Installation 2-1

Chapter 3 How to Operate Configuration Program 3-1

3.1 Startup 3-1
3.2 Screen Layout 3-2
3.2.1 Menu Bar 3-2
3.2.2 Device View 3-3
3.2.3 Setting View 3-4
3.3 Functions 3-6
3.3.1 Addition of JposEntry 3-6
3.3.2 Modification of JposEntry 3-16
3.3.3 Deletion of JposEntry 3-17
3.3.4 Log Setting 3-17

Chapter 4 POS Printer Control Interface Specifications 4-1

4.1 Summary 4-1
4.1.1 Common Properties 4-1
4.1.2 Specific Properties 4-2
4.1.3 Common Methods 4-6

4.1.4 Specific Methods 4-6
4.1.5 Events 4-7
4.2 Data Characters and Escape Sequences 4-8
4.2.1 Escape Sequence Operated When Specified 4-8
4.2.2 Escape Sequence Operated During Printing 4-10
4.2.3 Escape Sequence Operated When Printing 4-11
4.3 Details 4-13
4.3.1 Common Properties 4-13
CapCompareFirmwareVersion Property 4-13
CapPowerReporting Property 4-13
CapStatisticsReporting Property 4-13
CapUpdateFirmware Property 4-14
CapUpdateStatistics Property 4-14
CheckHealthText Property 4-14
Claimed Property 4-15
DeviceEnabled Property R/W 4-15
FreezeEvents Property R/W 4-15
OutputID Property 4-16
PowerNotify Property R/W 4-16
PowerState Property 4-17
State Property 4-17
DeviceControlDescription Property 4-17
DeviceControlVersion Property 4-18
DeviceServiceDescription Property 4-18
DeviceServiceVersion Property 4-18
PhysicalDeviceDescription Property 4-18
PhysicalDeviceName Property 4-19
4.3.2 Specific Properties 4-20
AsyncMode Property R/W 4-20
CapCharacterSet Property 4-20
CapCoverSensor Property 4-20
CapMapCharacterSet Property 4-21
CapRec2Color Property 4-21
CapRecBarCode Property 4-21
CapRecBitmap Property 4-21
CapRecBold Property 4-22
CapRecCartridgeSensor Property 4-22
CapRecColor Property 4-22
CapRecDhigh Property 4-22
CapRecDwide Property 4-23
CapRecDwideDhigh Property 4-23
CapRecEmptySensor Property 4-23
CapRecItalic Property 4-23
CapRecLeft90 Property 4-24
CapRecMarkFeed Property 4-24
CapRecNearEndSensor Property 4-24

CapRecPageMode Property 4-25
CapRecPaperCut Property 4-25
CapRecPresent Property 4-25
CapRecRight90 Property 4-25
CapRecRotate180 Property 4-26
CapRecRuledLine Property 4-26
CapRecStamp Property 4-26
CapRecUnderline Property 4-26
CapTransaction Property 4-27
CartridgeNotify Property R/W 4-27
CharacterSet Property R/W 4-27
CharacterSetList Property 4-28
CoverOpen Property 4-28
ErrorLevel Property 4-28
ErrorStation Property 4-29
ErrorString Property 4-29
FlagWhenIdle Property R/W 4-30
FontTypefaceList Property 4-30
MapCharacterSet Property R/W 4-31
MapMode Property R/W 4-31
PageModeArea Property 4-32
PageModeDescriptor Property 4-33
PageModeHorizontalPosition Property R/W 4-33
PageModePrintArea Property R/W 4-34
PageModePrintDirection Property R/W 4-35
PageModeStation Property R/W 4-37
PageModeVerticalPosition Property R/W 4-37
RecBarCodeRotationList Property 4-38
RecBitmapRotationList Property 4-38
RecCartridgeState Property 4-38
RecCurrentCartridge Property R/W 4-39
RecEmpty Property 4-39
RecLetterQuality Property R/W 4-39
RecLineChars Property R/W 4-40
RecLineCharsList Property 4-41
RecLineHeight Property R/W 4-41
RecLineSpacing Property R/W 4-42
RecLinesToPaperCut Property 4-43
RecLineWidth Property 4-43
RecNearEnd Property 4-44
RecSidewaysMaxChars Property 4-44
RecSidewaysMaxLines Property 4-44
RotateSpecial Property R/W 4-45
4.3.3 Common Methods 4-46
checkHealth Method 4-46
claim Method 4-46

clearOutput Method 4-46
close Method 4-47
compareFirmwareVersion Method 4-47
directIO Method 4-47
open Method 4-49
release Method 4-49
resetStatistics Method 4-49
retrieveStatistics Method 4-49
updateFirmware Method 4-50
updateStatistics Method 4-50
4.3.4 Specific Methods 4-51
beginInsertion Method 4-51
beginRemoval Method 4-51
changePrintSide Method 4-51
clearPrintArea Method 4-51
cutPaper Method 4-51
drawRuledLine Method 4-52
endInsertion Method 4-52
endRemoval Method 4-52
markFeed Method 4-52
pageModePrint Method 4-53
printBarCode Method 4-54
printBitmap Method 4-68
printImmediate Method 4-70
printMemoryBitmap Method 4-71
printNormal Method 4-71
printTwoNormal Method 4-72
rotatePrint Method 4-72
setBitmap Method 4-74
setLogo Method 4-75
transactionPrint Method 4-76
validateData Method 4-76
4.3.5 Events 4-78
DirectIOEvent Event 4-78
ErrorEvent Event 4-78
OutputCompleteEvent Event 4-79
StatusUpdateEvent Event 4-79

Chapter 5 CashDrawer 5-1

5.1 Summery 5-1
5.1.1 Common Properties 5-1
5.1.2 Specific Properties 5-2
5.1.3 Common Methods 5-2
5.1.4 Specific Methods 5-2
5.1.5 Events 5-2

5.2 Detail 5-3
5.2.1 Common Properties 5-3
CapCompareFirmwareVersion Property 5-3
CapPowerReporting Property 5-3
CapStatisticsReporting Property 5-3
CapUpdateFirmware Property 5-4
CapUpdateStatistics Property 5-4
CheckHealthText Property 5-4
Claimed Property 5-5
DeviceEnabled Property R/W 5-5
FreezeEvents Property R/W 5-5
PowerNotify Property R/W 5-6
PowerState Property 5-6
State Property 5-7
DeviceControlDescription Property 5-7
DeviceControlVersion Property 5-7
DeviceServiceDescription Property 5-7
DeviceServiceVersion Property 5-7
PhysicalDeviceDescription Property 5-8
PhysicalDeviceName Property 5-8
5.2.2 Specific Properties 5-9
CapStatus Property 5-9
CapStatusMultiDrawerDetect Property 5-9
DrawerOpened Property 5-9
5.2.3 Common Methods 5-10
checkHealth Method 5-10
claim Method 5-10
close Method 5-11
compareFirmwareVersion Method 5-11
directIO Method 5-11
open Method 5-11
release Method 5-11
resetStatistics Method 5-11
retrieveStatistics Method 5-12
updateFirmware Method 5-12
updateStatistics Method 5-12
5.2.4 Specific Methods 5-13
openDrawer Method 5-13
waitForDrawerClose Method 5-13
5.2.5 Events 5-14
DirectIOEvent Event 5-14
StatusUpdateEvent Event 5-14

Appendix A Exceptions A-1

A.1 POSPrinter Control A-1
A.1.1 Common Properties....A-1
A.1.2 Specific Properties....A-3
A.1.3 Common Methods A-13
A.1.4 Specific Methods A-17
A.1.5 Escape Sequence A-28
A.2 CashDrawer....A-30
A.2.1 Common Properties....A-30
A.2.2 Specific Properties....A-31
A.2.3 Common Methods A-32
A.2.4 Specific Methods A-33

Appendix B Statistics B-1

B.1 Statistics Defined by UnifiedPOS B-1
B.2 Statistics Defined by Manufacturer B-2

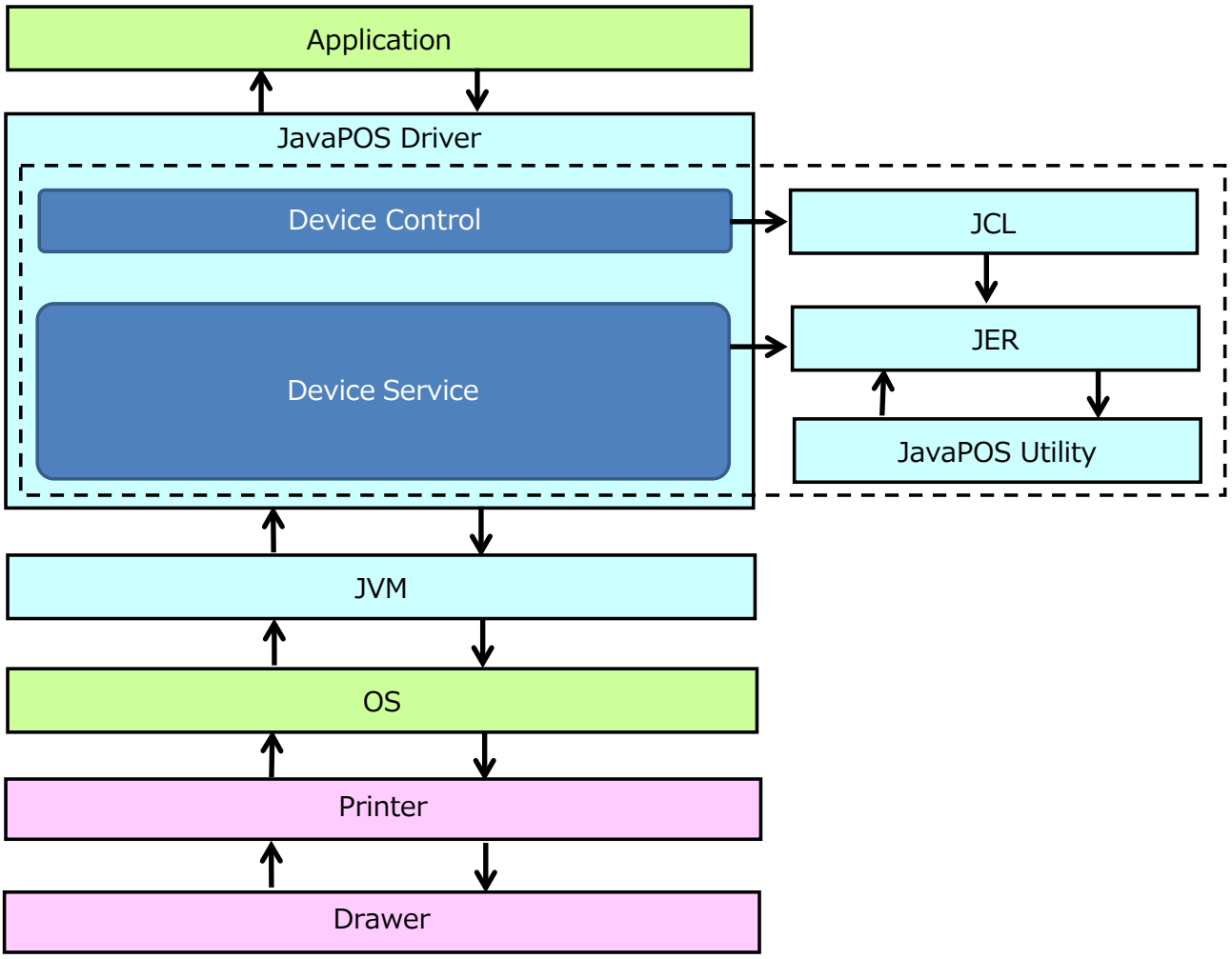
Chapter 1 Overview

This chapter describes the overview of the software.

1.1 Configuration

1.1.1 Structural Diagram

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



1.2 Operating Environment

1.2.1 System Environment

The system environment of the software is shown below.

Item	Specifications
Operating System	Microsoft® Windows® 10 (32 bits and 64 bits) Microsoft® Windows® 10 IoT Enterprise (32 bits and 64 bits) Microsoft® Windows® 11 (64 bits) Microsoft® Windows® 11 IoT Enterprise (64 bits)
Java Operation Environment	Java Standard Edition 8 or later
Printer Driver*1	For RP-F10: RP-F10 series printer driver For RP-E10: RP-E10 series printer driver For RP-D10: RP-D10 series printer driver

*1: Use dedicated installer for each model to install the printer driver.
See "RP-D10 Series Printer Driver User's Guide" or "SII Software Package for Windows Installation Guide" of each model for the installation procedure.
When renaming the printer driver, the name should be included the printer model name "SII RP-x10".

1.2.2 Supported Models

Description in this document	Model	Interface
RP-F10	RP-F10-K27J1 RP-F10-W27J1	USB, Ethernet, Bluetooth
RP-E10	RP-E10-K3FJ1 RP-E10-W3FJ1 RP-E11-K3FJ1 RP-E11-W3FJ1	USB, Serial, Ethernet
RP-D10	RP-D10-K27J1 RP-D10-W27J1	USB, Serial, Ethernet

1.3 Limitations

The limitations of the software are described.

1.3.1 POSPrinter

All interfaces of POSPrinter Device Class 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
- Cartridge function

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

- **beginInsertion**
- **beginRemoval**
- **changePrintSide**
- **endInsertion**
- **endRemoval**
- **markFeed**
- **printTwoNormal**

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

(e) Do not disconnect the connection cable while printing is in progress. If do so, printing may not be performed correctly after reconnecting the cable.

1.3.2 CashDrawer

All interfaces of CashDrawer device are provided, with the following limitations.

(a) The following method always returns JPOS_E_ILLEGAL after it is opened.

- **directIO**

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

Chapter 2 Installation

For installation and uninstallation, see "RP-D10 Series Printer Driver User's Guide" or "SII Software Package for Windows Installation Guide.".

It is necessary to install Java execution environment before installing the software.

Caution

◆ This installation requires logon to the computer with administrator privileges.

Chapter 3 How to Operate Configuration Program

This chapter describes the configuration program provided by the software.
The explanations and screenshots in this chapter are based on the RP-F10.

3.1 Startup

The startup procedure of the configuration program is described.

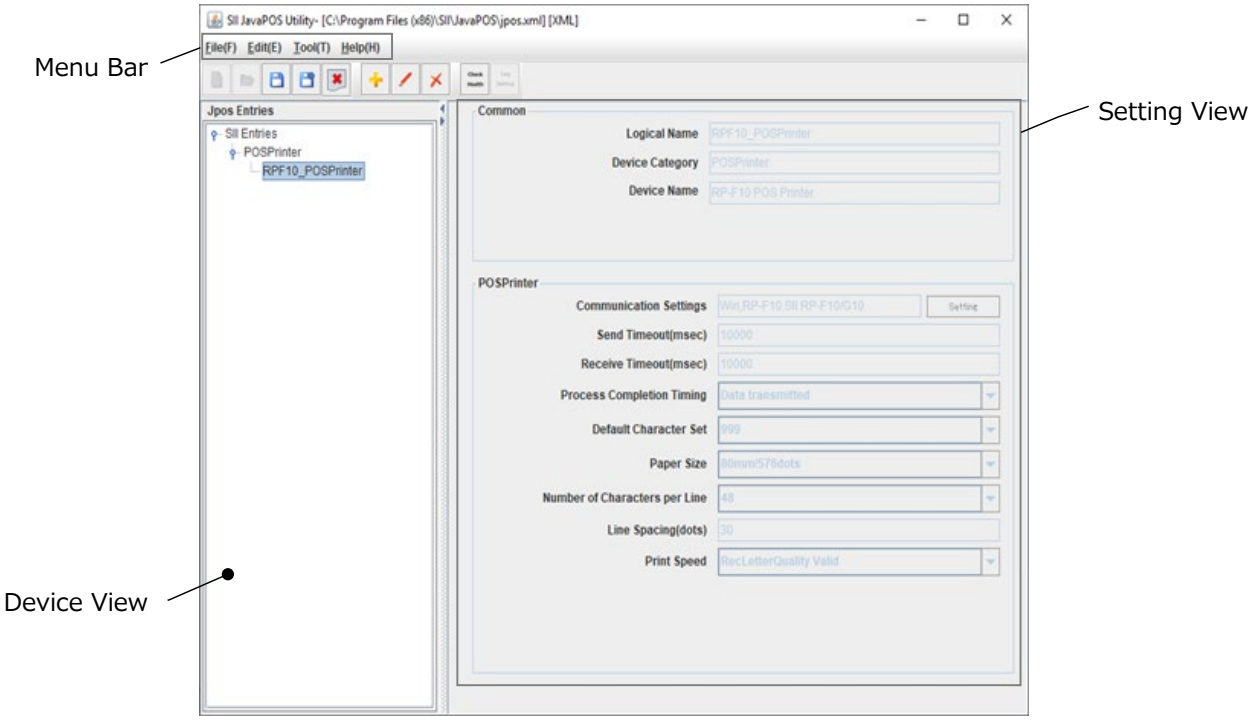
Select [SII Printer Software] - [JavaPOS] - [JavaPosUtility.bat] from the Start menu, and then the configuration program starts up.

Caution

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

3.2 Screen Layout

The screen layout of the configuration program is described.



Item	Description
Menu Bar	The menu bar of the configuration program. See "3.2.1 Menu Bar" for items in the menu bar.
Device View	The type, the name, and the logical name of the device registered in the system are displayed in a tree.
Setting View	Displays setting contents of the device selected in "Device View". See "3.2.3(1) POSPrinter setting items", or "3.2.3(2)CashDrawer setting items" for items of each device.

3.2.1 Menu Bar

Item		Description
File(F)	New(N)	Starts creating a new xml file.
	Open(O)	Loads the saved data.
	Save(S)	Saves the data being edited.
	SaveAs(A)	Saves the data being edited with a name. The file name should be jpos.xml.
	Close(C)	Closes the configuration program.
	Exit	Ends the configuration program.
Edit(E)	Add(A)	Adds a new device.
	Modify(M)	Changes the setting contents for the selected device.
	Delete(D)	Deletes the device being selected.

Item		Description
Tool(T)	CheckHealth	Executes an interactive test for the device being selected.
	LogSetting(L)	Sets the log. See "3.3.4 Log Setting" for the log setting.
Help(H)	About SII JavaPOS Utility(A)	Displays the version information of the configuration program.
	Language(L)	Japanese (J) Displays the configuration program in Japanese. English (E) Displays the configuration program in English.

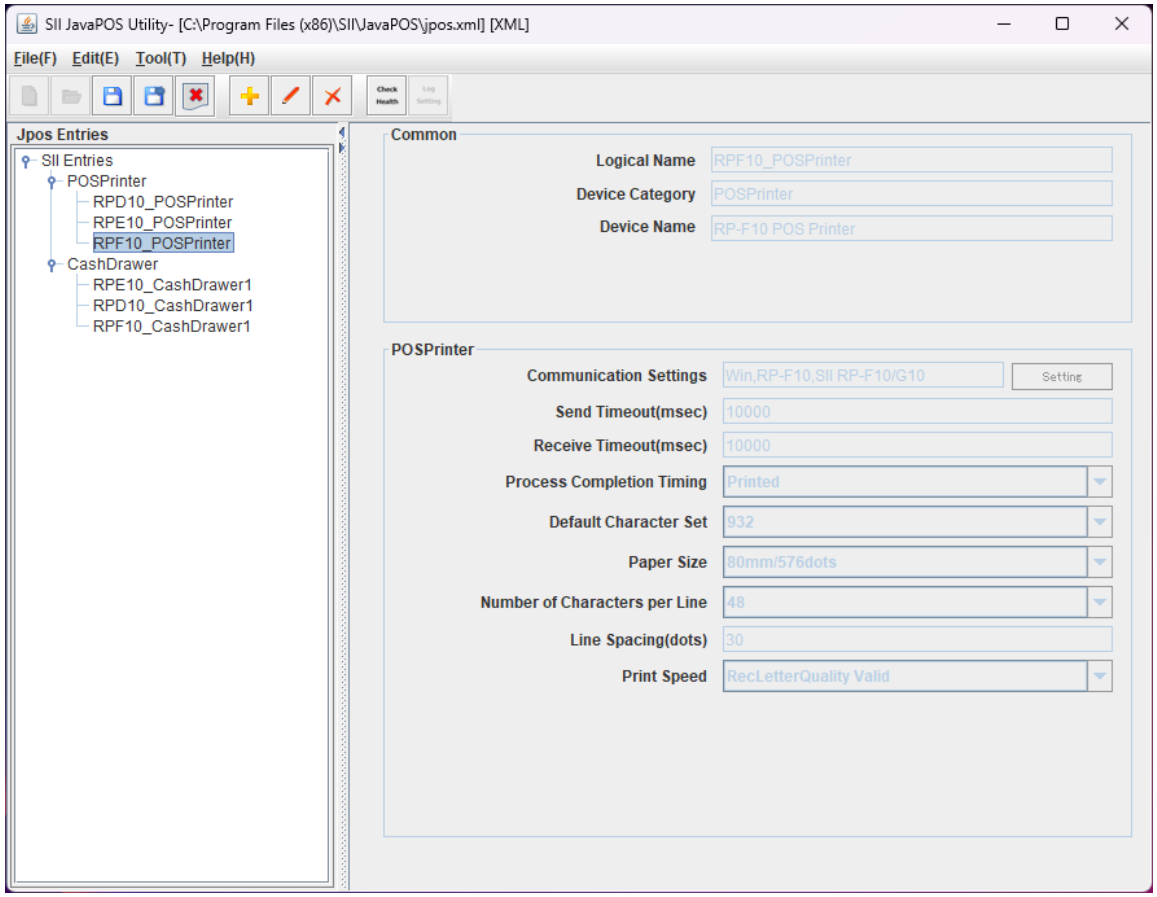
3.2.2 Device View

Name	Description
SII Entries	Displays SII devices. When a logical name is selected in "Device View", the setting contents of the device can be changed and deleted.

3.2.3 Setting View

(1) POSPrinter setting items

The items displayed in "Setting View" when the printer is selected and setting contents are described below.



Item	Description	Setting Content (" " : Initial Value)
Common		
Logical Name	Any logical name entered	For RP-F10: RPF10_POSPrinter For RP-E10: RPE10_POSPrinter For RP-D10: RPD10_POSPrinter
Device Category	Device type	POSPrinter
Device Name	Device name	RP-F10: RP-F10 POS Printer RP-E10: RP-E10 POS Printer RP-D10: RP-D10 POS Printer

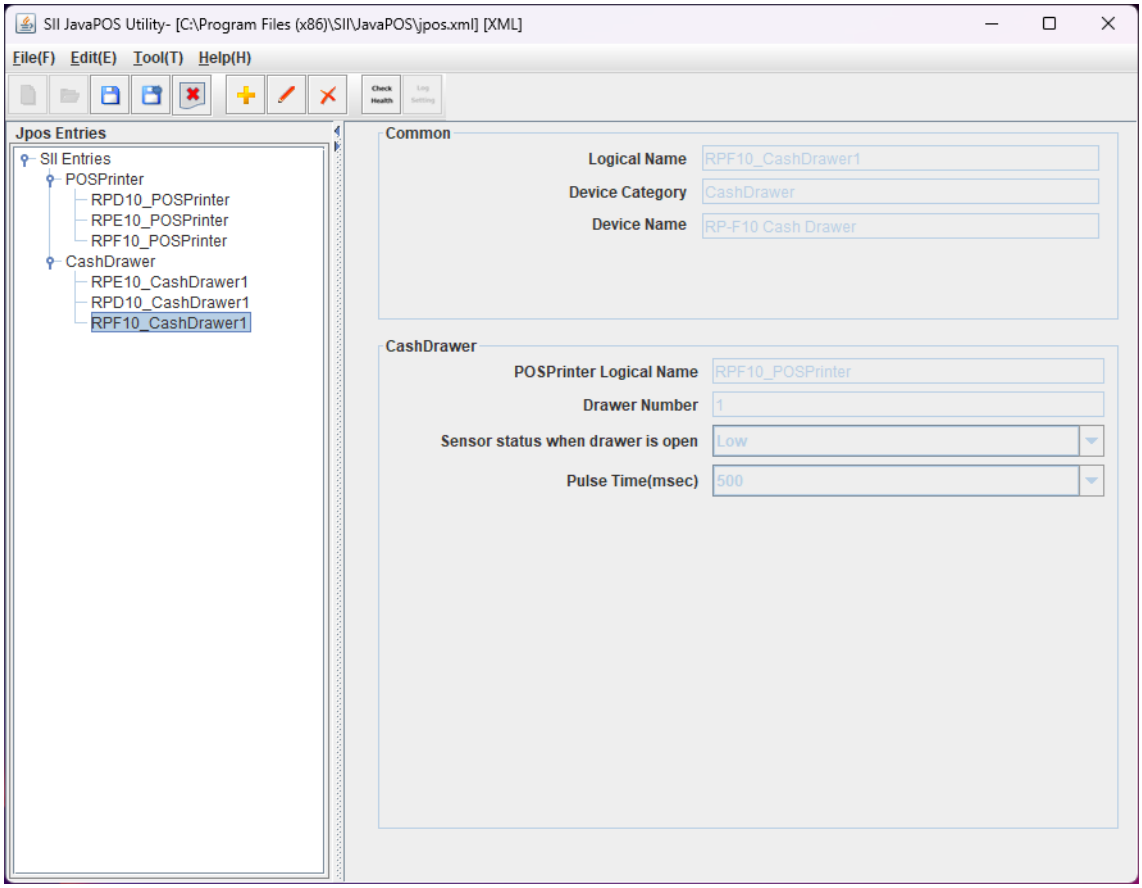
Item	Description	Setting Content (" " : Initial Value)
POSPrinter		
Communication Settings	Printer driver used for communication with the printer	Selectable from the [Setting] button. Example: Windows
Send Timeout(msec)	Send timeout value in communication with the printer (milliseconds)	3000 to 60000 (10000)
Receive Timeout(msec)	Receive timeout value in communication with the printer (milliseconds)	3000 to 60000 (10000)
Process Completion Timing	Timing of method completion	Completion of data transmission Data transmitted
Default Character Set	Character set type **CharacterSet** is initialized with this value. See **CharacterSet** for details.	437 737*1 850 852 855*1 857*1 858 860 863 865*1 866*1 932*2 999*3 1250 1251 1252 1253 1254
Paper Size	Paper width / Number of effective dots	58mm/360dots 58mm/432dots 80mm/512dots 80mm/576dots
Number of Characters per Line	Number of 1-byte characters printed on 1 line **RecLineChars** is initialized with this value.	360 dots / 30, 40 432 dots / 27, 30, 33, 36, 43, 48, 54 512 dots / 42, 56 576 dots / 36, 41, 44, 48, 57, 64, 72
Line Spacing(dots)	Line spacing per 1 line **RecLineSpacing** is initialized with this value.	0 to 255 (30)
Near End Sensor*4	Select enable/disable for near end sensor of the printer.	0: Disable 1: Enable
Print Speed	Print speed of the printer It is decided by **RecLetterQuality** when selecting RecLetterQuality Valid.	RecLetterQuality Valid: Enabled **RecLetterQuality** High: High speed*5 Middle (Quality): Middle speed (quality)*5 Middle (Silent): Middle speed (silent)*5 Low: Low speed*5*6

*1: Supported only by RP-F10.
*2: Initial value for Japanese
*3: Initial value for English
*4: Supported only by RP-E10.
*5: See "RP-F10 SERIES THERMAL PRINTER TECHNICAL REFERENCE", "RP-E10 SERIES THERMAL PRINTER TECHNICAL REFERENCE", and "RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details of the print speed.

*6: Supported only by RP-R10 and RP-D10.

(2) CashDrawer setting items

The items displayed in "Setting View" when the cash drawer is selected and setting contents are described below.



Item	Description	Setting Content (" " : Initial Value)
Common		
Logical Name	Any logical name entered	For RP-F10: RPF10_CashDrawer1 For RP-E10: RPE10_CashDrawer1 For RP-D10: RPD10_CashDrawer1
Device Category	Device type	CashDrawer
Device Name	Device name	RP-F10: RP-F10 Cash Drawer RP-E10: RP-E10 Cash Drawer RP-D10: RP-D10 Cash Drawer

Item	Description	Setting Content (" " : Initial Value)
CashDrawer		
POSPrinter Logical Name	Logical name of the printer to which the drawer is connected	POSPrinter devices registered in JavaPOS Entry Registry can be selected. Example: RP-F10_POSPrinter
Drawer Number	Drawer that open when **OpenDrawer** is executed	1 2
Sensor status when drawer is open	Drawer sensor status when the drawer is open	Low High
Pulse Time (msec)	Time of the pulse to open the drawer	100 200 300 400 500 600 700 800

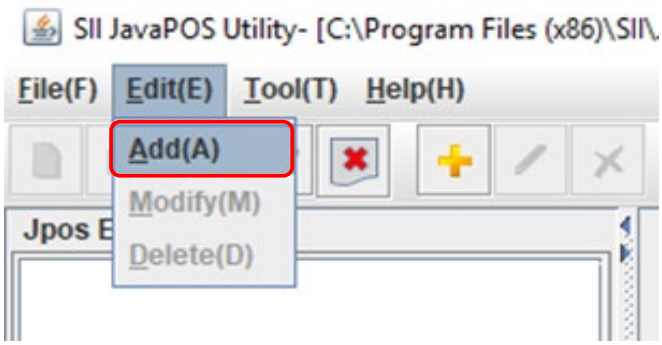
3.3 Functions

The functions of the configuration program are described.

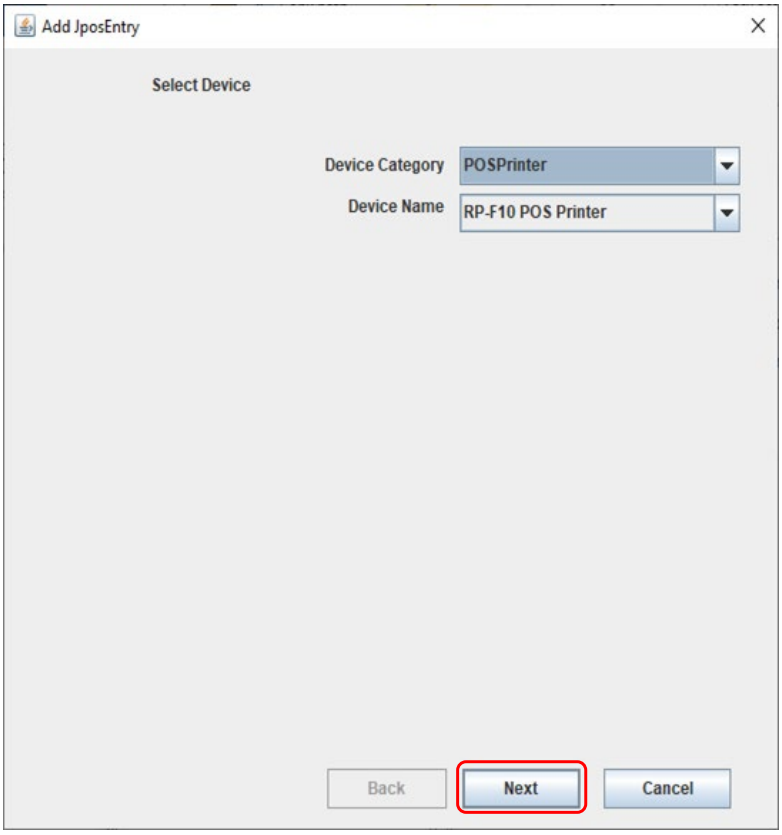
3.3.1 Addition of JposEntry

(1) Addition of POSPrinter

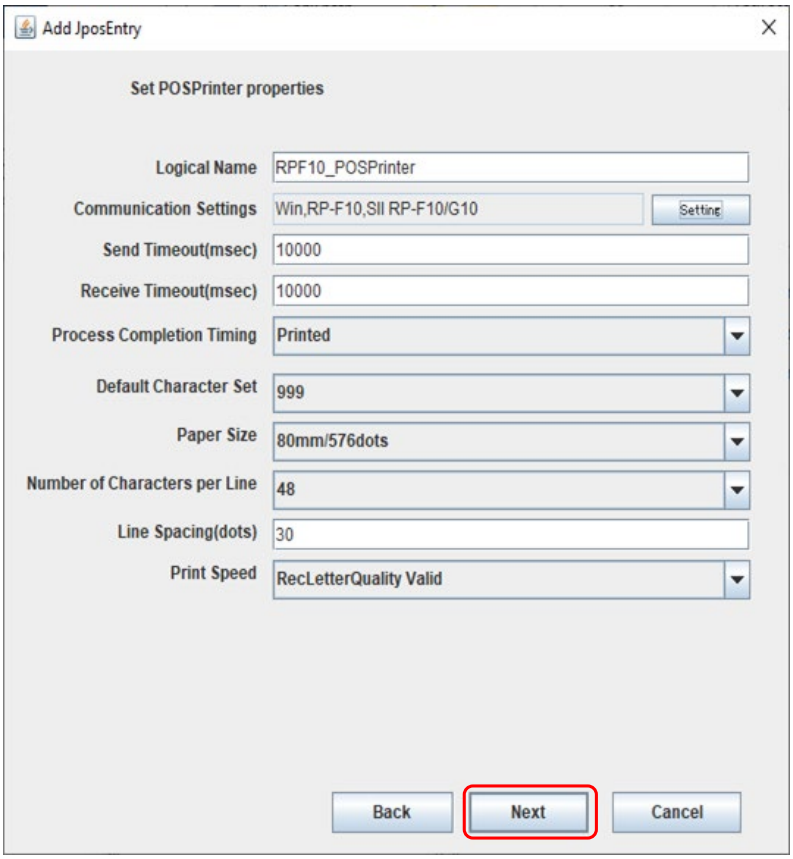
1. When the configuration program starts, the following window is displayed. Select [Edit(E)] – [Add(A)] from "Menu Bar".



2. Select "POSPrinter" for [Device Category], and "RP-F10 POS Printer" for [Device Name], then click the [Next] button.



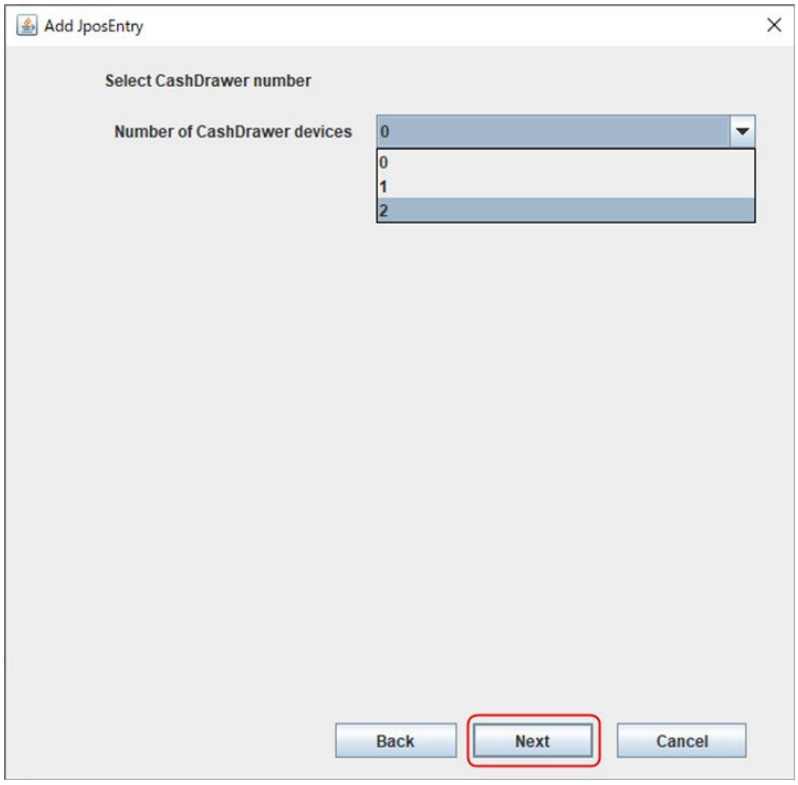
3. The detailed settings of the POSPrinter to be added are displayed. Set each item, and click the [Next] button. See "3.2.3(1) POSPrinter setting items" for the setting contents.



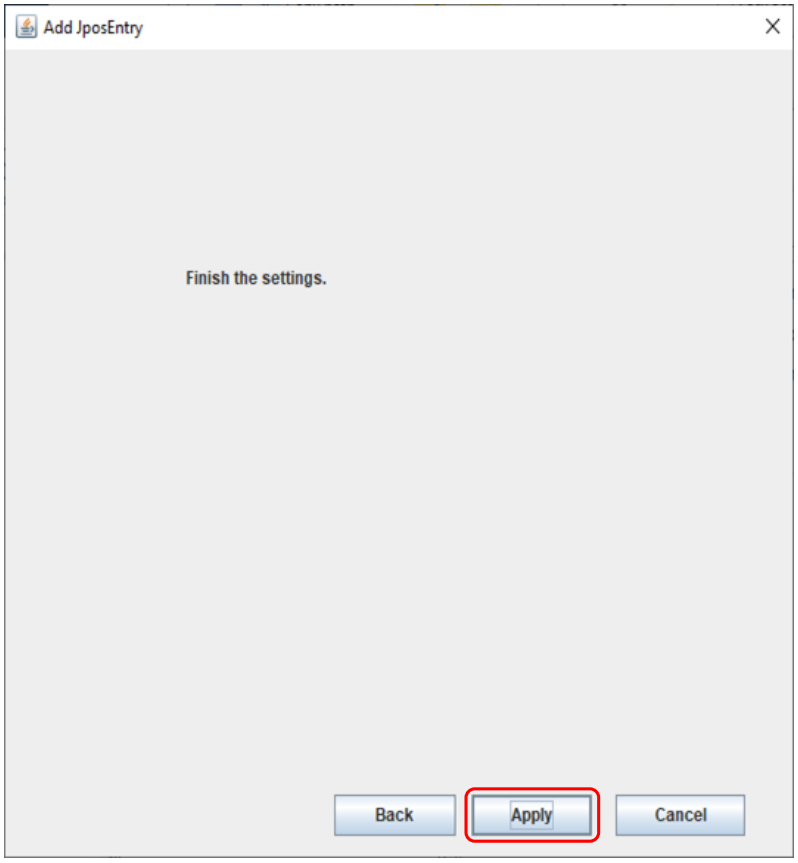
Caution

◆ The same logical name cannot be set for multiple POSPrinter device.

4. The select CashDrawer number ([Number of CashDrawer devices]) window is displayed.
   When adding the POSPrinter device only, select [0], and click the [Next] button.
   Refer to 5 for subsequent settings.
   When adding 1 CashDrawer device:
   select [1], and click the [Next] button.
   Refer to 4 in "(2) Addition of CashDrawer" for subsequent settings.
   When adding 2 Cash Drawer devices:
   select [2], and click the [Next] button.
   Refer to 5 in "(2) Addition of CashDrawer" for subsequent settings.



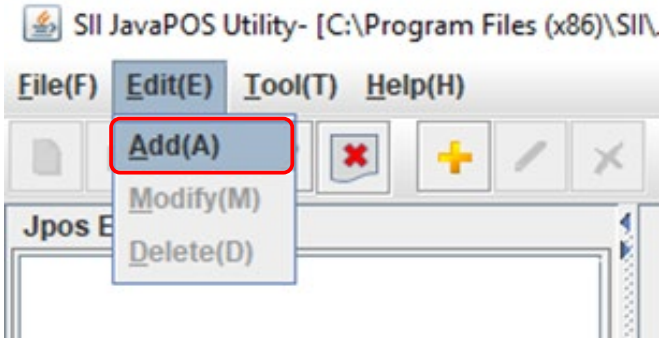
5. Click the [Apply] button.



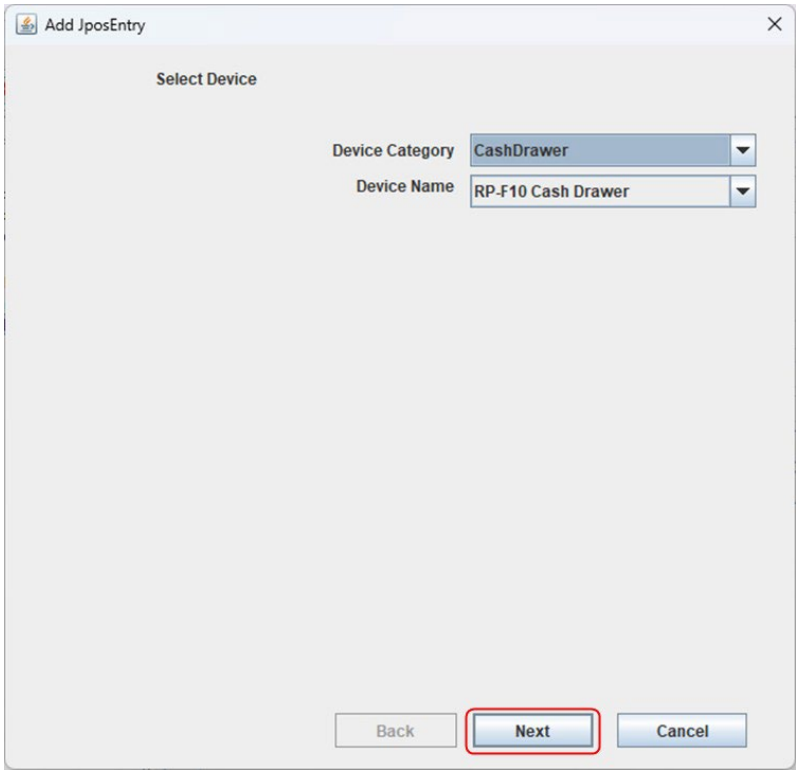
6. Select [File(F)] – [Save(S)] or [SaveAs(A)] from "Menu Bar".
When [SaveAs(A)] is selected, save the file name as jpos.xml.

(2) Addition of CashDrawer

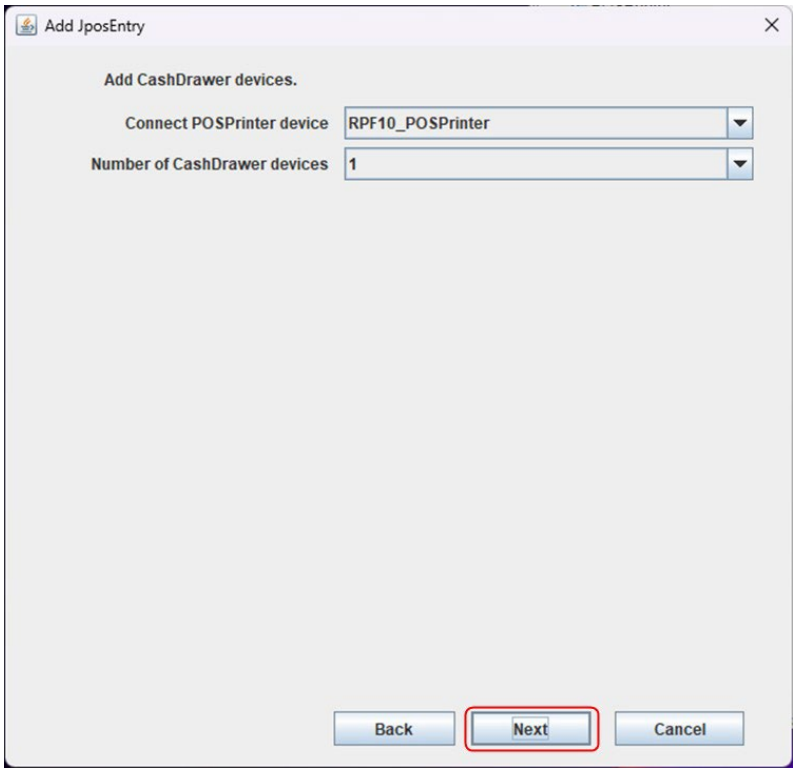
1. When the configuration program starts, the following window is displayed. Select [Edit(E)] – [Add(A)] from "Menu Bar".



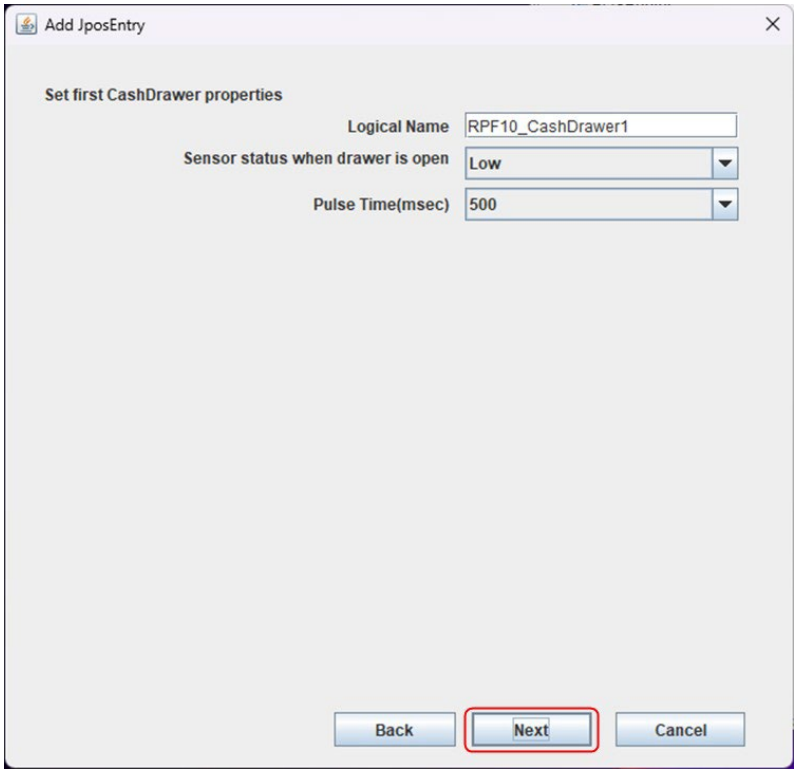
2. Select "CashDrawer" for [Device Category], and "RP-F10 Cash Drawer" for [Device Name], then click the [Next] button.



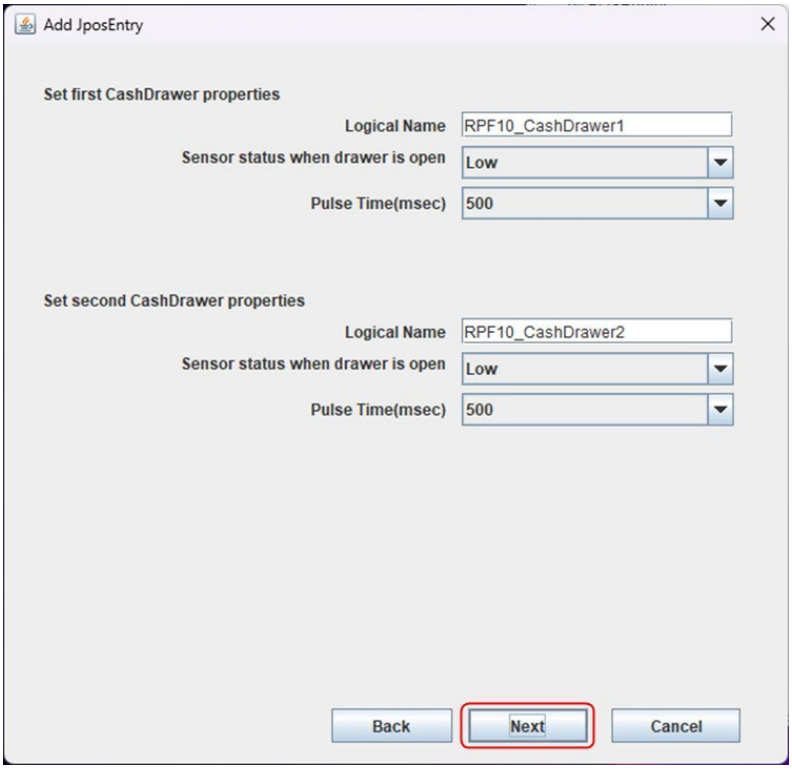
3. The next window to select the connect POSPrinter logical name and the number of CashDrawer devices, is displayed.
   Select the logical name of the printer that connects drawer in [Connect POSPrinter device].
   When adding 1 CashDrawer:
   Select [1] in [Number of CashDrawer devices], and click the [Next] button.
   Refer to 4 for subsequent settings.
   When adding 2 CashDrawer:
   Select [2] in [Number of CashDrawer devices], and click the [Next] button.
   Refer to 5 for subsequent settings.



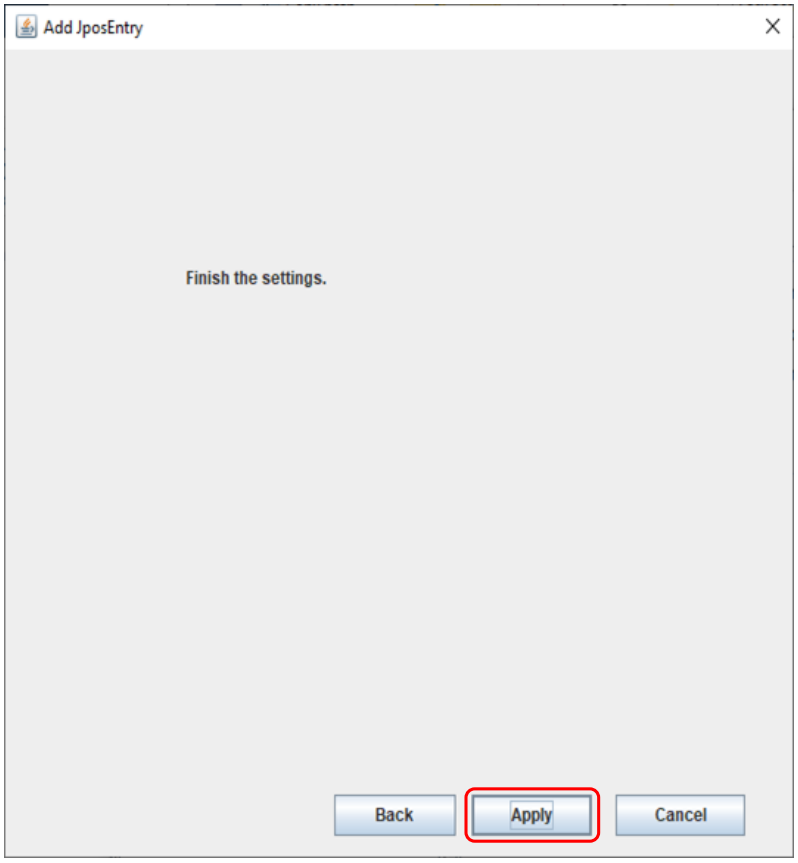
4. The advanced setting window for driving 1 drawer is displayed.
   Set each item, and click the [Next] button.
   For details about the items, refer to "3.2.3(2) CashDrawer setting items".



5. The advanced setting window for driving 2 drawers is displayed.
   Set each item, and click the [Next] button.
   For details about the items, refer to "3.2.3(2) CashDrawer setting items".

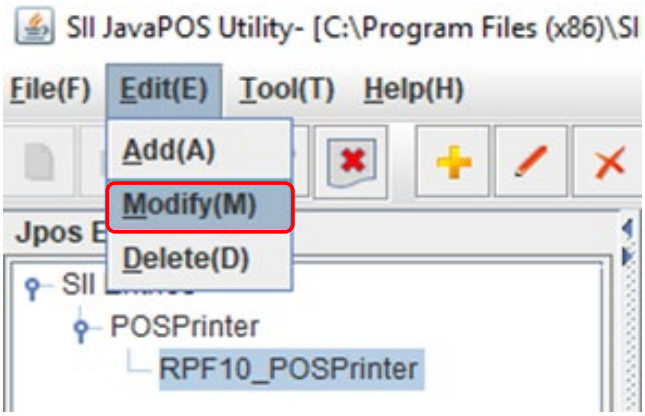


6. Click the [Apply] button.

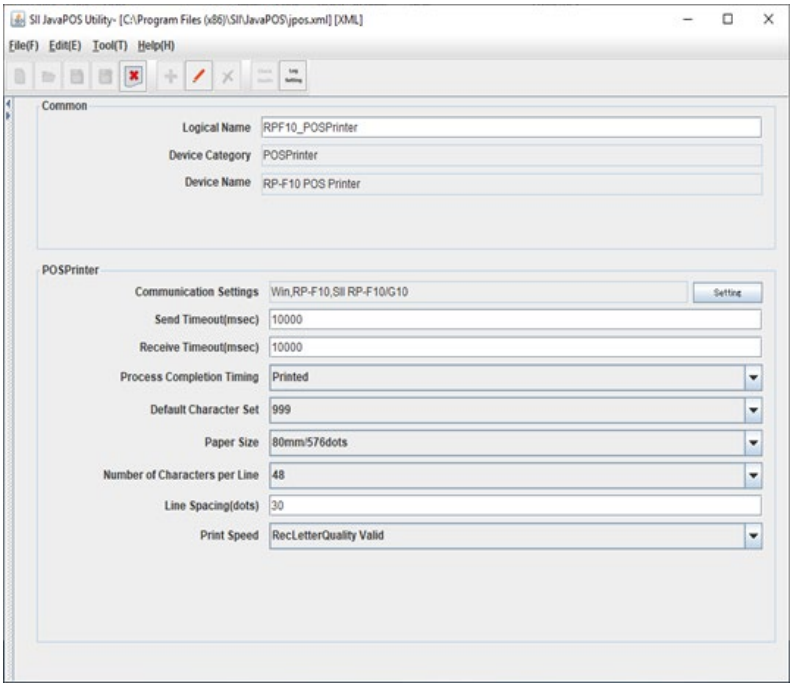


3.3.2 Modification of JposEntry

1. Select the device name of the device to be changed from "Device View".
   Select [Edit(E)] – [Modify(M)] from "Menu Bar".



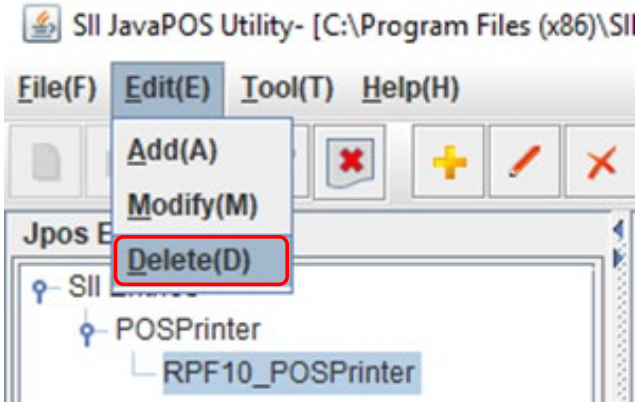
2. "Setting View" is displayed in editable state. Change the contents.
   Select [Edit(E)] – [Modify(M)] from "Menu Bar" to end editing.



3. Select [File(F)] – [Save(S)] or [SaveAs(A)] from "Menu Bar".
   When [SaveAs(A)] is selected, save the file name as jpos.xml.

3.3.3 Deletion of JposEntry

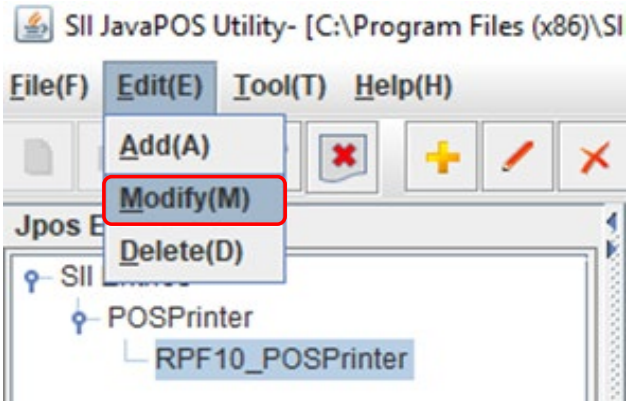
1. Select the device name of the device to be deleted from "Device View".
   Select [Edit(E)] – [Delete(D)] from "Menu Bar".



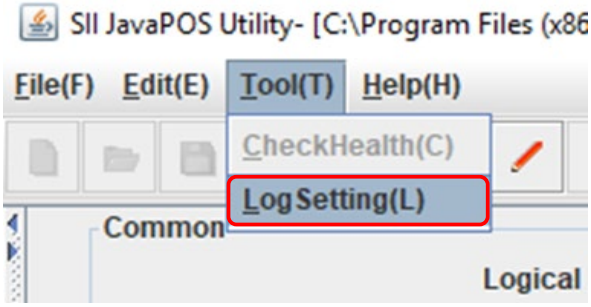
2. Select [File(F)] – [Save(S)] or [SaveAs(A)] from "Menu Bar".
   When [SaveAs(A)] is selected, save the file name as jpos.xml.

3.3.4 Log Setting

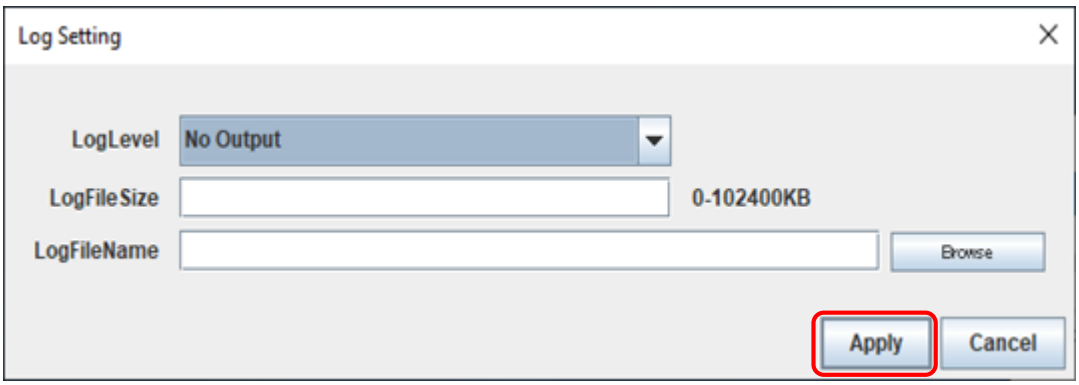
1. Select the device name of the device to be set the log from "Device View".
   Select [Edit(E)] – [Modify(M)] from "Menu Bar".



2. Select [Tool(T)] – [LogSetting(L)] from "Menu Bar".



3. The [Log Setting] wizard is displayed. Set the contents, and click the [Apply] button.



Item			Description
LogLevel			
	No Output		No logs are output.
	Error		Error level logs are output.
	Trace		Trace level logs are output.
LogFileSize			
	0 to 102400		Set the maximum log file size (KB).
LogFileName			
	File name		Set the full path of the file name.
		Browse	Open the file.

4. Select [File(F)] – [Save(S)] or [SaveAs(A)] from "Menu Bar".
When [SaveAs(A)] is selected, save the file name as jpos.xml.

Chapter 4 POSPrinter

4.1 Summary

4.1.1 Common Properties

Property Name	Type	Access	Availability Condition	Initial Value
CapCompareFirmwareVersion	**boolean**	R	open	false
CapPowerReporting	**int**	R	open	JPOS_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
DeviceEnabled	**boolean**	R/W	open & claim	false
FreezeEvents	**boolean**	R/W	open	false
OutputID	**int**	R	open	0
PowerNotify	**int**	R/W	open	JPOS_PN_DISABLED(0)
PowerState	**int**	R	open	JPOS_PS_UNKNOWN(2000)
State	**int**	R	-	JPOS_S_CLOSED(1)
DeviceControlDescription	**String**	R	-	"JavaPOS POSPrinter Device Control"
DeviceControlVersion	**int**	R	-	"1013000"
DeviceServiceDescription	**String**	R	open	"SII RP-x10 Series JavaPOS POSPrinter Service Driver, Copyright (C) 20xx Seiko Instruments Inc."
DeviceServiceVersion	**int**	R	open	"1013XXX"
PhysicalDeviceDescription	**String**	R	open	"SII RP-F10 POS Printer"*1
				"SII RP-E10 POS Printer"*2
				"SII RP-D10 POS Printer"*3
PhysicalDeviceName	**String**	R	open	"RP-F10 POS Printer"*1
				"RP-E10 POS Printer"*2
				"RP-D10 POS Printer"*3

*1: In case of RP-F10
*2: In case of RP-E10

*3: In case of RP-D10

4.1.2 Specific Properties

(When **RecLineWidth**=576, **RecLineChars**=48, **RecLineSpacing**=30, **CharacterSet**=999)

Property Name	Type	Access	Availability Condition	Initial Value
AsyncMode	**boolean**	R/W	open	false
CapCharacterSet	**int**	R	open	PTR_CCS_KANJI(11)
CapCoverSensor	**boolean**	R	open	true
CapMapCharacterSet	**boolean**	R	open	false
CapRec2Color	**boolean**	R	open	false
CapRecBarCode	**boolean**	R	open	true
CapRecBitmap	**boolean**	R	open	true
CapRecBold	**boolean**	R	open	true
CapRecCartridgeSensor	**int**	R	open	0
CapRecColor	**int**	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	**int**	R	open	0
CapRecNearEndSensor	**boolean**	R	open	false[*1*2] Depends on JavaPOS Entry Registry[*3]
CapRecPageMode	**boolean**	R	open	true
CapRecPapercut	**boolean**	R	open	true
CapRecPresent	**boolean**	R	open	true
CapRecRight90	**boolean**	R	open	true
CapRecRotate180	**boolean**	R	open	true
CapRecRuledLine	**int**	R	open	0
CapRecStamp	**boolean**	R	open	false
CapRecUnderline	**boolean**	R	open	true
CapTransaction	**boolean**	R	open	true
CartridgeNotify	**int**	R/W	open	PTR_CN_DISABLED(0)
CharacterSet	**int**	R/W	open, claim, & enable	999[*4]
CharacterSetList	**String**	R	open	"437,737[*1],850,852,855[*1],857[*1],858,860,863,865,866[*1],932,999,1250,1251,1252,1253,1254"
CoverOpen	**boolean**	R	open, claim, & enable	Depends on status of this method
ErrorLevel	**int**	R	open	PTR_EL_NONE(1)

Property Name	Type	Access	Availability Condition	Initial Value
ErrorStation	**int**	R	open	0
ErrorString	**String**	R	open	""
FlagWhenIdle	**boolean**	R/W	open	false
FontTypefaceList	**String**	R	open	""
MapCharacterSet	**boolean**	R/W	open	false
MapMode	**int**	R/W	open	PTR_MM_DOTS(1)
PageModeArea	**String**	R	open	""
PageModeDescriptor	**int**	R	open	0
PageModeHorizontalPosition	**int**	R/W	open	0
PageModePrintArea	**String**	R/W	open	""
PageModePrintDirection	**int**	R/W	open	0
PageModeStation	**int**	R/W	open	0
PageModeVerticalPosition	**int**	R/W	open	0
RecBarCodeRotationList	**String**	R	open	"0,R90,L90,180"
RecBitmapRotationList	**String**	R	open	"0,R90,L90,180"
RecCartridgeState	**int**	R	open, claim, & enable	PTR_CART_UNKOWN (0x10000000)
RecCurrentCartridge	**int**	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	**int**	R/W	open, claim, & enable	48
RecLineCharsList	**String**	R	open	"36,41,44,48,57,64,72"
RecLineHeight	**int**	R/W	open, claim, & enable	24
RecLineSpacing	**int**	R/W	open, claim, & enable	30
RecLinesToPaperCut	**int**	R	open, claim, & enable	3
RecLineWidth	**int**	R	open, claim, & enable	576*4
RecNearEnd	**boolean**	R	open, claim, & enable	false*1*2
				Depends on JavaPOS Entry Registry*3
RecSidewaysMaxChars	**int**	R	open, claim, & enable	200
RecSidewaysMaxLines	**int**	R	open, claim, & enable	19
RotateSpecial	**int**	R/W	open	PTR_RP_NORMAL(1)

*1: In case of RP-F10
*2: In case of RP-E10
*3: In case of RP-D10
*4: Can be modified by the configuration program.

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

Property Name	Type	Access	Availability Condition	Initial Value
CapConcurrentJrnRec	**boolean**	R	open	false
CapConcurrentJrnSlp	**boolean**	R	open	false
CapConcurrentPageMode	**boolean**	R	open	false
CapConcurrentRecSlp	**boolean**	R	open	false
CapJrn2Color	**boolean**	R	open	false
CapJrnBold	**boolean**	R	open	false
CapJrnCartridgeSensor	**int**	R	open	0
CapJrnColor	**int**	R	open	0
CapJrnDhigh	**boolean**	R	open	false
CapJrnDwide	**boolean**	R	open	false
CapJrnDwideDhigh	**boolean**	R	open	false
CapJrnEmptySensor	**boolean**	R	open	false
CapJrnItalic	**boolean**	R	open	false
CapJrnNearEndSensor	**boolean**	R	open	false
CapJrnPresent	**boolean**	R	open	false
CapJrnUnderline	**boolean**	R	open	false
CapSlp2Color	**boolean**	R	open	false
CapSlpBarCode	**boolean**	R	open	false
CapSlpBitmap	**boolean**	R	open	false
CapSlpBold	**boolean**	R	open	false
CapSlpBothSidesPrint	**boolean**	R	open	false
CapSlpCartridgeSensor	**int**	R	open	0
CapSlpColor	**int**	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	**int**	R	open	0
CapSlpUnderline	**boolean**	R	open	false
JrnCartridgeState	**int**	R	open, claim, & enable	PTR_CART_UNKNOWN (0x10000000)
JrnCurrentCartridge	**int**	R/W	open, claim, & enable	0

Property Name	Type	Access	Availability Condition	Initial Value
JrnEmpty	**boolean**	R	open, claim, & enable	false
JrnLetterQuality	**boolean**	R/W	open, claim, & enable	false
JrnLineChars	**int**	R/W	open, claim, & enable	0
JrnLineCharsList	**String**	R	open	""
JrnLineHeight	**int**	R/W	open, claim, & enable	0
JrnLineSpacing	**int**	R/W	open, claim, & enable	0
JrnLineWidth	**int**	R	open, claim, & enable	0
JrnNearEnd	**boolean**	R	open, claim, & enable	false
SlpBarCodeRotationList	**String**	R	open	""
SlpBitmapRotationList	**String**	R	open	""
SlpCartridgeState	**int**	R	open, claim, & enable	PTR_CART_UNKNOWN (0x10000000)
SlpCurrentCartridge	**int**	R/W	open, claim, & enable	0
SlpEmpty	**boolean**	R	open, claim, & enable	false
SlpLetterQuality	**boolean**	R/W	open, claim, & enable	false
SlpLineChars	**int**	R/W	open, claim, & enable	0
SlpLineCharsList	**String**	R	open	""
SlpLineHeight	**int**	R/W	open, claim, & enable	0
SlpLinesNearEndToEnd	**int**	R	open, claim, & enable	0
SlpLineSpacing	**int**	R/W	open, claim, & enable	0
SlpLineWidth	**int**	R	open, claim, & enable	0
SlpMaxLines	**int**	R	open, claim, & enable	0
SlpNearEnd	**boolean**	R	open, claim, & enable	false
SlpPrintSide	**int**	R	open, claim, & enable	PTR_PS_UNKNOWN(0)
SlpSidewaysMaxChars	**int**	R	open, claim, & enable	0
SlpSidewaysMaxLines	**int**	R	open, claim, & enable	0

4.1.3 Common Methods

Method Name	Availability Condition
checkHealth	open, claim, & enable
claim	open
clearOutput	open, claim, & enable*1
close	open
compareFirmwareVersion	open, claim, & enable
directIO	open, claim, & enable*1
open	-
release	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 JavaPOS V1.13.

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	Availability Condition
DirectIOEvent	open, claim, & enable
ErrorEvent	open, claim, & enable
OutputCompleteEvent	open, claim, & enable
StatusUpdateEvent	open, claim, & enable

4.2 Data Characters and Escape Sequences

4.2.1 Escape Sequence Operated When Specified

Name	Data	Remarks
Paper cut	ESC|[#]P	• Cuts receipt paper. The placeholder '#' is replaced by an ASCII decimal string indicating the cut percentage. If a value greater than 100 is specified for '#', then a full cut is executed. If a value from 1 to 99 is specified for '#', a partial cut is executed. If 0 is specified for '#', no cut is executed. If '#' is omitted, a full cut is executed. • This is ignored during rotated 90° right/left mode by **rotatePrint** or during Page Mode by **pageModePrint**.
Feed and Paper cut	ESC|[#]fP	• Cuts receipt paper after feeding the paper by the **RecLinesToPaperCut** lines. The placeholder '#' is defined by "Paper cut" escape sequence (ESC|[#]P). • This is ignored during rotated 90° right/left mode by **rotatePrint** or during Page Mode by **pageModePrint**.
Feed, Paper cut, and Stamp	ESC|[#]sP	Not supported.
Print bitmap	ESC|#B	• Prints the pre-stored bitmap. The placeholder '#' is replaced by the bitmap number. A value from 1 to 20 can be specified for '#'. If values other than 1 to 20 are specified for '#', they are ignored. If the character '#' is omitted, the data is regarded as print data instead of an escape sequence.
Print top logo	ESC|tL	Prints the pre-stored top logo.
Print bottom logo	ESC|bL	Prints the pre-stored bottom logo.
Fire stamp	ESC|sL	Not supported.
Feed lines[*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. A value from 0 to 255 can be specified for '#'. If '#' exceeds this range, the maximum supported number of 255 lines is fed. If '#' is omitted, then 1 line is fed. • This is ignored during rotated 90° right/left mode by **rotatePrint** or during Page Mode by **pageModePrint**.
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 1 unit is fed. • If **MapMode** is set to PTR_MM_DOTS(1), a value from 0 to 255 can be specified for '#'. If '#' exceeds this range, the maximum supported number of 255 units is fed. • This is ignored during rotated 90° right/left mode by **rotatePrint** method or during Page Mode by **pageModePrint**.
Feed reverse	ESC|[#]rF	Not supported.

Name	Data	Remarks
Pass through embedded data	ESC|[*]#E	• Sends the characters following "#E" through to the printer without modifying any of them. The placeholder '#' is replaced by an ASCII decimal string indicating the number of bytes following the escape sequence that should be passed through as-is to the printer. A value from 0 to 65535 can be specified for '#'. If '#' exceeds this range, transmission of embedded data is not executed. If '#' is omitted, the data is regarded as print data instead of an escape sequence. If the print data for the number of bytes specified by '#' is not set after the escape sequence is specified, only the available print data to send is sent. (Example: If ESC|2Ea is specified, only "a" is sent since only 1 byte is set for the character string.) • During rotated 90° right/left mode by **rotatePrint**, the width cannot be calculated exactly because data string specified by transmission of embedded data is not counted as character string. Therefore, make an appropriate adjustment by inserting blanks.
Print in-line barcode (See the following "In-Line Barcode Printing".)	ESC|[*]#R	• Prints a barcode. The placeholder '#' is replaced by an ASCII decimal string indicating the number of characters of the string following R (definition of the barcode characteristics). If '#' is omitted, the data is regarded as print data instead of an escape sequence. • If the number of characters specified by '#' does not match the number of bytes following R, all the data within the range specified by '#' is discarded. • During rotated 90° right/left mode by **rotatePrint**, the width cannot be calculated exactly because data string specified by transmission of barcode printing is not counted as character string. Therefore, make an appropriate adjustment by inserting blanks.
Print in-line ruled line	ESC|*#dL	Not supported.

*1: In RP-F10 and RP-D10, the line spacing is reduced according to the setting value of the printer function setting "Paper Saving". 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 **RecLineSpacing**.
- The amount of feed specified by the "Feed units" escape sequence (ESC|[#]uF) is processed on the basis of the setting value of the printer function setting "Paper Saving". (When Mode1 is selected, paper is not fed at all.)

However, the line spacing from last print line to cut position is not reduced because paper is cut after executing the paper feed for saved dot lines.
Then cut operations in which the paper feed for saved dot lines is executed are "Paper cut" escape sequence (ESC|[#]P), "Feed and Paper cut" escape sequence (ESC|[#]fP), and **cutPaper**.

- **In-Line Barcode Printing**

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

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

  The characters indicating the attributes are as follows.

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

  Attributes must be written in the above order.
  Every attribute is mandatory.
  Both or either of the 2 conditions are violated or the value outside of the range is specified for the number subsequent to each attribute, it may cause unpredictable print results.

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

  ESC|33Rs101h200w400a-2t-13d123456789012e

  or

  ESC|*33Rs101h200w400a-2t-13d123456789012e

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

4.2.2 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.

4.2.3 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 3 or larger, then a thickness of 2 is used. If '#' is omitted, then a thickness of 1 is used for the underline. 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 values less than 1 are specified for '#', print in 1 scale. If values greater than 8 are specified for '#', print in 8 scale. If '#' is omitted, the data is regarded as print data instead of an escape sequence.
Scale vertically	ESC|#vC	• A supported value for the placeholder '#' is 1 to 8. If values less than 1 are specified for '#', print in 1 scale. If values greater than 8 are specified for '#', print in 8 scale. If '#' is omitted, the data is regarded as print data instead of an escape sequence.
RGB Color	ESC|[#]fC	Not supported.
Center	ESC|cA	• Aligns the characters after ESC|cA in the center. This must be specified at the head of the line. If not, this is invalid. Also, if there is a linefeed on the print data, the center is valid after linefeed. • This specification is ignored during rotated 90° right/left mode by **rotatePrint** or during Page Mode by **pageModePrint**.
Right justify	ESC|rA	• Aligns the characters after ESC|rA to the right. This must be specified at the head of the line. If not, this is invalid. Also, if there is a linefeed on the print data, the right justify is valid after linefeed. • This specification is ignored during rotated 90° right/left mode by **rotatePrint** or during Page Mode by **pageModePrint**.

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

4.3 Details

This section describes the details of the POSPrinter device.
For details of the thrown exception errors, see "Appendix A Exceptions".

4.3.1 Common Properties

CapCompareFirmwareVersion Property

Type **boolean**

Remarks Indicates whether a firmware file's version can be compared against the firmware version of the device.
The following table shows the valid property values.

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

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

CapPowerReporting Property

Type **int**

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

Value	Meaning
JPOS_PR_STANDARD(1)	2 types of power states, JPOS_PS_OFF_OFFLINE(2004) (power off or offline) and JPOS_PS_ONLINE(2001), can be determined and reported.

This property is initialized to JPOS_PR_STANDARD(1) by **open**.

CapStatisticsReporting Property

Type **boolean**

Remarks Indicates the statistics accumulation function of the device.
The following table shows the valid property values.

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

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

CapUpdateFirmware Property

Type **boolean**

Remarks Indicates whether the device supports firmware updating.
The following table shows the valid property values.

Value	Meaning
false	Firmware update is not supported.

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

CapUpdateStatistics Property

Type **boolean**

Remarks Indicates the function that some or all device statistics can be reset.
The following table shows the valid property values.

Value	Meaning
true	The device statistics, or some of the statistics, can be reset to 0 using **resetStatistics**.

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

CheckHealthText Property

Type **String**

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

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

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

Claimed Property

Type **boolean**

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

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

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

DeviceEnabled Property R/W

Type **boolean**

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

Value	Meaning
false	The device has been disabled. If changed to false, then the device is physically disabled.
true	The device has been in an operational state. If changed to true, then the device is brought to an operational state.

The application must set this property to true before using the device.
If **State** is other than JPOS_S_IDLE(2), **DeviceEnabled** cannot be changed from true to false.

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

FreezeEvents Property R/W

Type **boolean**

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

Value	Meaning
false	The application allows events to be delivered. If some events have been held while events were frozen and all other conditions are correct for delivering the events, then changing **FreezeEvents** to false will cause these events to be delivered.
true	Events will not be delivered. Events will be enqueued until **FreezeEvents** is set to false.

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

If an error occurs while the print method such as the **printNormal** is operated under the **AsyncMode** is true, **ErrorEvent** is frozen and the **State** turns to JPOS_S_BUSY(3). In this case, clear the frozen event by the **clearOutput** or conduct the **close** after **ErrorEvent** is occurred by setting false since the POSPrinter device cannot be closed under this circumstance.

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

OutputID Property

Type **int**

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

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

Output ID is numbered from 1 to 2147483646 cyclically.

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

PowerNotify Property R/W

Type **int**

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

Value	Meaning
JPOS_PN_DISABLED(0)	The POSPrinter device will not provide any power notifications to the application. No power notification **StatusUpdateEvents** will be fired, and **PowerState** is not be set.
JPOS_PN_ENABLED(1)	The POSPrinter device will fire power notification **StatusUpdateEvents** and update **PowerState**, when **DeviceEnabled** is set to true. The level of functionality depends upon **CapPowerReporting**.

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

This property is initialized to JPOS_PN_DISABLED(0) by **open**.

PowerState Property

Type **int**

Remarks Indicates the current power condition.
The following table shows the valid property values.

Value	Meaning
JPOS_PS_UNKNOWN(2000)	Cannot determine the device's power state, for one of the following reasons: •**PowerNotify** is JPOS_PN_DISABLED(0). •**DeviceEnabled** is false.
JPOS_PS_ONLINE(2001)	The device is powered on and ready.
JPOS_PS_OFF_OFFLINE(2004)	The device is powered off or offline.

This property is initialized to JPOS_PS_UNKNOWN(2000) by **open**.

State Property

Type **int**

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

Value	Meaning
JPOS_S_CLOSED(1)	The device is closed.
JPOS_S_IDLE(2)	The device is in a good state and is not busy.
JPOS_S_BUSY(3)	The device is in a good state and is busy executing output.
JPOS_S_ERROR(4)	An error has been reported, and the application must recover the device to a normal 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 JPOS _S_CLOSED(1) until **open** is successfully completed.

DeviceControlDescription Property

Type **String**

Remarks "JavaPOS POSPrinter Device Control" is set.
Holds an identifier for the Device Control.
This property is always readable.

DeviceControlVersion Property

Type **String**

Remarks "1013000" is set.
Holds the Device Control version number.
This property is always readable.

DeviceServiceDescription Property

Type **String**

Remarks Holds an identifier for the Device Service.
This property is initialized to "SII RP-x10 Series JavaPOS POSPrinter Service Driver, Copyright (C) 20xx Seiko Instruments Inc." by **open**.

DeviceServiceVersion Property

Type **String**

Remarks Holds the Device Service version number.
This property is initialized to "1013XXX" by **open**.

PhysicalDeviceDescription Property

Type **String**

Remarks Holds an identifier for the physical device.
This property depends on the printer.
One of the following values is set to this property by **open**.

Printer	Value
RP-F10	"SII RP-F10 POS Printer"
RP-E10	"SII RP-E10 POS Printer"
RP-D10	"SII RP-D10 POS Printer"

PhysicalDeviceName Property

Type **String**

Remarks Holds a short name identifying the physical device.
This property depends on the printer.
One of the following values is set to this property by **open**.

Printer	Value
RP-F10	"RP-F10 POS Printer"
RP-E10	"RP-E10 POS Printer"
RP-D10	"RP-D10 POS Printer"

4.3.2 Specific Properties

AsyncMode Property R/W

Type **boolean**

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

Value	Meaning
false	**printNormal**, **cutPaper**, **printBarCode**, **printBitmap**, **printMemoryBitmap**, **rotatePrint**, **transactionPrint**, and **pageModePrint** print methods are executed synchronously.
true	The above methods are executed asynchronously.

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

CapCharacterSet Property

Type **int**

Remarks Holds the printable character setting of the printer.
The following table shows the valid property values.

Value	Meaning
PTR_CCS_KANJI(11)	The character setting supports Code Page932, including ASCII characters 0x20 through 0x7F and the 1-byte katakana characters 0xA1 through 0xDF. It also includes the Shift-JIS code characters defined by JIS Levels 1 and 2.

This property is initialized to PTR_CCS_KANJI(11) by **open**.

CapCoverSensor Property

Type **boolean**

Remarks Indicates whether the printer has a "cover open" sensor.
The following table shows the valid property values.

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

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

CapMapCharacterSet Property

Type **boolean**

Remarks Indicates that the POSPrinter device is able to map the characters of the application to a character set.
The following table shows the valid property values.

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

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

CapRec2Color Property

Type **boolean**

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

Value	Meaning
false	2 color printing of the receipt is not supported.

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

CapRecBarCode Property

Type **boolean**

Remarks Indicates whether the receipt has barcode printing capability.
The following table shows the valid property values.

Value	Meaning
true	The receipt has barcode printing capability.

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

CapRecBitmap Property

Type **boolean**

Remarks Indicates whether the receipt can print bitmaps.
The following table shows the valid property values.

Value	Meaning
true	The receipt can print bitmaps.

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

CapRecBold Property

Type **boolean**

Remarks Indicates whether the receipt can print bold characters.
The following table shows the valid property values.

Value	Meaning
true	The receipt can print bold characters.

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

CapRecCartridgeSensor Property

Type **int**

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

Value	Meaning
0	Receipt cartridge monitoring sensors are not supported.

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

CapRecColor Property

Type **int**

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

Value	Meaning
PTR_COLOR_PRIMARY (0x00000001)	The receipt supports primary color.

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

CapRecDhigh Property

Type **boolean**

Remarks Indicates whether the receipt can print double high characters.
The following table shows the valid property values.

Value	Meaning
true	The receipt can print double high characters.

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

CapRecDwide Property

Type **boolean**

Remarks Indicates whether the receipt can print double wide characters.
The following table shows the valid property values.

Value	Meaning
true	The receipt can print double wide characters.

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

CapRecDwideDhigh Property

Type **boolean**

Remarks Indicates whether the receipt can print double high / double wide characters.
The following table shows the valid property values.

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

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

CapRecEmptySensor Property

Type **boolean**

Remarks Indicates whether the receipt has an out-of-paper sensor.
The following table shows the valid property values.

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

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

CapRecItalic Property

Type **boolean**

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

Value	Meaning
false	Italic characters of the receipt are not supported.

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

CapRecLeft90 Property

Type **boolean**

Remarks Indicates whether the receipt can print in a rotated 90° left mode.
The following table shows the valid property values.

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

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

CapRecMarkFeed Property

Type **int**

Remarks Indicates the type of mark sensed paper handling available.
The following table shows the valid property values.

Value	Meaning
0	Control function for marked thermal paper is not supported.

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

CapRecNearEndSensor Property

Type **boolean**

Remarks Indicates whether the receipt has a paper-near-end sensor.
The following table shows the valid property values.

Value	Meaning
false	A paper-near-end sensor is not supported.
true	The printer has a "paper-near-end" sensor.

This property depends on the printer and **NearEndSensor** property of JposEntry.

In case of RP-F10 and RP-D10:
This property is initialized to false by **open**.

In case of RP-E10:
When **NearEndSensor** of JposEntry is set to "0", this property is initialized to false by **open**.
When **NearEndSensor** of JposEntry is set to "1", this property is initialized to true by **open**.

CapRecPageMode Property

Type **boolean**

Remarks Indicates whether the printer can support Page Mode for the receipt station.
The following table shows the valid property values.

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

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

CapRecPaperCut Property

Type **boolean**

Remarks Indicates whether the receipt can perform paper cuts.
The following table shows the valid property values.

Value	Meaning
true	The receipt can perform paper cuts.

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

CapRecPresent Property

Type **boolean**

Remarks Indicates whether the receipt print station is present.
The following table shows the valid property values.

Value	Meaning
true	The receipt print station is present.

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

CapRecRight90 Property

Type **boolean**

Remarks Indicates whether the receipt can print in a rotated 90° right mode.
The following table shows the valid property values.

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

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

CapRecRotate180 Property

Type **boolean**

Remarks Indicates whether the receipt can print in a rotated upside down mode.
The following table shows the valid property values.

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

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

CapRecRuledLine Property

Type **int**

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

Value	Meaning
0	Ruled line drawing printing is not supported.

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

CapRecStamp Property

Type **boolean**

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

Value	Meaning
false	A stamp printing is not supported.

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

CapRecUnderline Property

Type **boolean**

Remarks Indicates whether the receipt can print underlined characters.
The following table shows the valid property values.

Value	Meaning
true	The receipt can print underlined characters.

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

CapTransaction Property

Type **boolean**

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

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

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

CartridgeNotify Property R/W

Type **int**

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 available.

Only PTR_CN_DISABLED(0) can be set.
This property is initialized to PTR_CN_DISABLED(0) by **open**.

CharacterSet Property R/W

Type **int**

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

Value	Meaning
437	Selects Code Page437 character set.
737[*1]	Selects Code Page737 character set.
850	Selects Code Page850 character set.
852	Selects Code Page852 character set.
855[*1]	Selects Code Page855 character set.
857[*1]	Selects Code Page857 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.
866[*1]	Selects Code Page866 character set.
932	Selects Katakana as Code Page932 character set (Shift-JIS code).

Value	Meaning
999	Selects Windows ANSI character set.*2
1250	Selects Code Page1250 character set.
1251	Selects Code Page1251 character set.
1252	Selects Code Page1252 character set.*2
1253	Selects Code Page1253 character set.
1254	Selects Code Page1254 character set.

*1: Supported only by RP-F10.
*2: Windows ANSI character set is equal to Code Page1252 character set.

This property is initialized to the value set to **DefaultCharacterSet** of the configuration program when the device is first enabled by **open**.

CharacterSetList Property

Type **String**

Remarks Holds the character set numbers.
This property is initialized to "437,737*1,850,852,855*1,857*1,858,860,863,865,866*1, 932,999,1250,1251,1252,1253,1254" by **open**.

*1: Supported only by RP-F10.

CoverOpen Property

Type **boolean**

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

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

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

ErrorLevel Property

Type **int**

Remarks Holds the severity of the error condition.
The following table shows the valid property 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 property is set just before **ErrorEvent** is notified.
When the error is cleared, then the property is changed to PTR_EL_NONE(1).
This property is initialized to PTR_EL_NONE(1) by **open**.

ErrorStation Property

Type **int**

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

Value	Meaning
0	The error was not detected.
PTR_S_RECEIPT(2)	The error is detected at the receipt station.

This property is set just before **ErrorEvent** is notified.
When the error is cleared, then the property is changed to 0.
This property is initialized to 0 by **open**.

ErrorString Property

Type **String**

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

Value	Meaning
"The power supply of the device is off."	Error message in the next condition. Priority is 1. ErrorCode=JPOS_E_NOHARDWARE(107)
"Unrecoverable error occurred."	Error message in the next condition. Priority is 2. ErrorCode=JPOS_E_EXTENDED(114) ErrorCodeExtended=1010
"VP power error occurred."	Error message in the next condition. Priority is 3. ErrorCode=JPOS_E_EXTENDED(114) ErrorCodeExtended=1001
"Cutter error."	Error message in the next condition. Priority is 4. ErrorCode=JPOS_E_EXTENDED(114) ErrorCodeExtended=1002
"The cover is open."	Error message in the next condition. Priority is 5. ErrorCode=JPOS_E_EXTENDED(114) ErrorCodeExtended=201
"Out of receipt form."	Error message in the next condition. Priority is 6. ErrorCode=JPOS_E_EXTENDED(114) ErrorCodeExtended=203

Value	Meaning
"Head temperature error."	Error message in the next condition. Priority is 7. ErrorCode=JPOS_E_EXTENDED(114) ErrorCodeExtended=1005
"Communication error occurred."	Error message in the next condition. Priority is 8. ErrorCode=JPOS_E_FAILURE(111)
"Windows system error occurred."	
"Time out."	

When multiple errors occur simultaneously, the higher-priority value is set.
This property is set just before **ErrorEvent** is notified.
When the error is cleared, then the property is changed to empty string.
This property is initialized to empty string by **open**.

FlagWhenIdle Property R/W

Type **boolean**

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

Value	Meaning
false	**StatusUpdateEvent** is not notified.
true	**StatusUpdateEvent** will be sent when **State** is JPOS_S_IDLE(2).

This property is automatically reset to false when the 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 enqueued if the outputs were completed successfully or if they were cleared by **clearOutput** or by an **ErrorEvent** handler.

If **State** is already set to JPOS_S_IDLE(2) when this property is set to true, then a **StatusUpdateEvent** is enqueued immediately. The application can therefore depend upon the event, with no race condition between the starting of its last asynchronous output and the setting of this flag.
This property is initialized to false by **open**.

FontTypefaceList Property

Type **String[]**

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

MapCharacterSet Property R/W

Type **boolean**

Remarks Indicates whether character mapping is supported or not.
The following table shows the valid property values.

Value	Meaning
false	Mapping is not supported.

Only false can be set.
This property is initialized to false by **open**.

MapMode Property R/W

Type **int**

Remarks Holds the mapping mode of the printer.
The mapping mode defines the unit of measure used for other properties, such as line heights and line spacing.
The following mapping modes are supported.
The values in the brackets indicate the value converted into dot.

Parameter	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 formula.

MapMode	Conversion
PTR_MM_DOTS(1)	No conversion
PTR_MM_TWIPS(2)	k=1/1440 ■PTR_MM_DOTS(1) to PTR_MM_TWIPS(2) conversion twips=dot/(dpi * k) ■PTR_MM_TWIPS(2) to PTR_MM_DOTS(1) conversion dot=twips * dpi * k
PTR_MM_ENGLISH(3)	k=1/1000 ■PTR_MM_DOTS(1) to PTR_MM_ENGLISH(3) conversion english=dot/(dpi * k) ■PTR_MM_ENGLISH(3) to PTR_MM_DOTS(1) conversion dot=english * dpi * k

MapMode	Conversion
PTR_MM_METRIC(4)	k=1/100, mmpi=25.4 ■PTR_MM_DOTS(1) to PTR_MM_METRIC(4) conversion metric=(mmpi * dot)/(dpi * k) ■PTR_MM_METRIC(4) to PTR_MM_DOTS(1) conversion dot=(metric * dpi * k)/mmpi

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

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

This property is initialized to PTR_MM_DOTS(1) when the device is first enabled by **open**.

PageModeArea Property

Type **String**

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 2 ASCII numbers separated by a comma, in the following order: horizontal size, vertical size.
When PTR_S_RECEIPT(2) is specified for **PageModeStation**, one of the following values is set to this property. (When **MapMode** is PTR_MM_DOTS(1))

Value	Meaning
"360,2400"	Page Mode area when **RecLineWidth** is 360.
"432,2400"	Page Mode area when **RecLineWidth** is 432.
"512,2400"	Page Mode area when **RecLineWidth** is 512.
"576,2400"	Page Mode area when **RecLineWidth** is 576.

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

PageModeDescriptor Property

Type **int**

Remarks The Page Mode functionality available on the station specified for **PageModeStation** 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 **PageModeStation** before accessing this property.
This property returns 0 until a valid station is specified.
This property is initialized to OR of the above values after PTR_S_RECEIPT(2) is set.

PageModeHorizontalPosition Property R/W

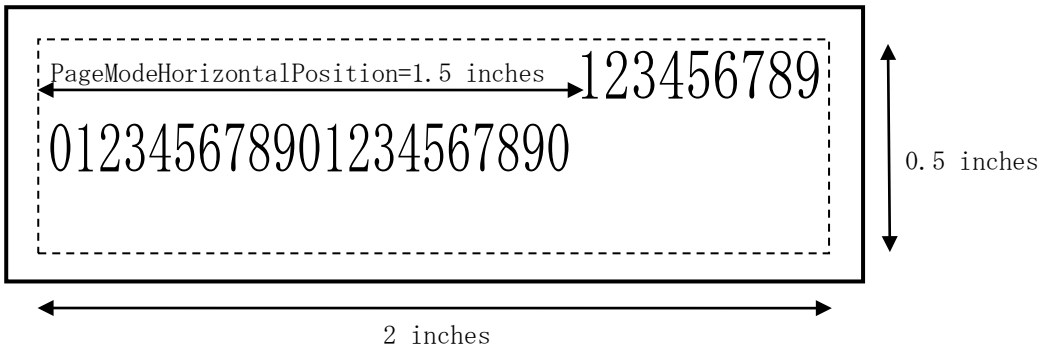
Type **int**

Remarks Holds the horizontal start position offset within the print area for the selected by **PageModeStation**.
This property is expressed in the unit of specified for **MapMode**.
The horizontal direction is the same as the actual **PageModePrintDirection**.
A read/get on this property will return the horizontal position offset set by the last write/set and not the current position.
Specify PTR_S_RECEIPT(2) for **PageModeStation** before accessing this property, otherwise the value 0 is returned.

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

```
myptr.setMapMode(PTR_MM_ENGLISH);
myptr.setPageModeStation(PTR_S_RECEIPT);
myptr.pageModePrint(PTR_PM_PAGE_MODE);
// Set print area to 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");
myptr.pageModePrint(PTR_PM_NORMAL);
```

The code sample above will generate the following receipt.



PageModePrintArea Property R/W

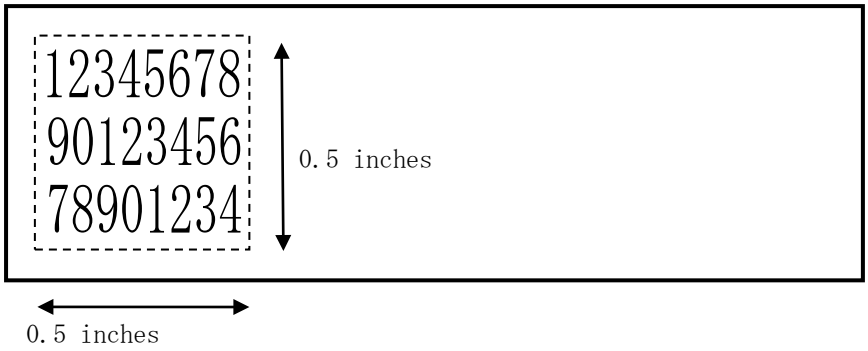
Type **String**

Remarks Holds the Page Mode print area for the selected **PageModeStation** expressed in the unit specified by **MapMode**.
The maximum print area is the page area.
The property consists of 4 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.

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

```
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,"123456789012345678901234\n");
myptr.pageModePrint(PTR_PM_NORMAL);
```

The code sample above will generate the following receipt.



Specify PTR_S_RECEIPT(2) for **PageModeStation** 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.

PageModePrintDirection Property R/W

Type **int**

Remarks Holds the print direction of the Page Mode print area.
The following table shows the valid property values.

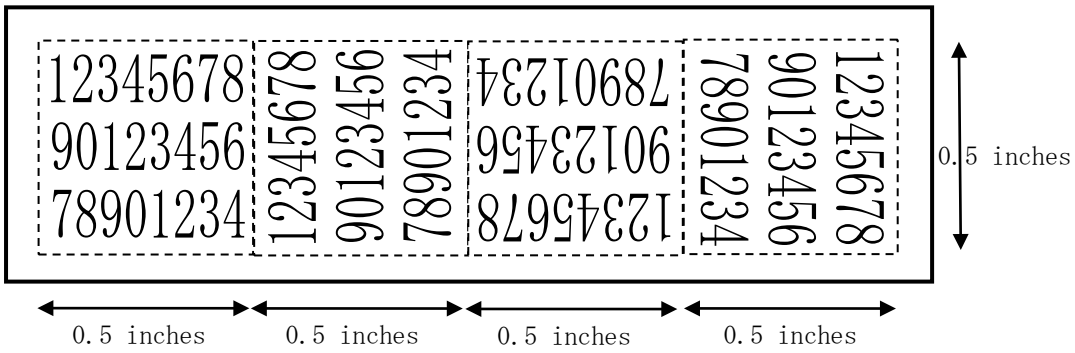
Value	Meaning
PTR_PD_LEFT_TO_RIGHT(1)	Normal direction printing. Print from left to right, starting at the top left corner of the Page Mode print area.
PTR_PD_BOTTOM_TO_TOP(2)	Rotated left 90° printing. Print from bottom to top, starting at the bottom left corner of the Page Mode print area.
PTR_PD_RIGHT_TO_LEFT(3)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

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

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

```
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,"123456789012345678901234\n");
myptr.setPageModePrintArea("500,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_BOTTOM_TO_TOP);
myptr.printNormal(PTR_S_RECEIPT,"123456789012345678901234\n");
myptr.setPageModePrintArea("1000,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_RIGHT_TO_LEFT);
myptr.printNormal(PTR_S_RECEIPT,"123456789012345678901234\n");
myptr.setPageModePrintArea("1500,0,500,500");
myptr.setPageModePrintDirection(PTR_PD_TOP_TO_BOTTOM);
myptr.printNormal(PTR_S_RECEIPT,"123456789012345678901234\n");
myptr.pageModePrint(PTR_PM_NORMAL);
```

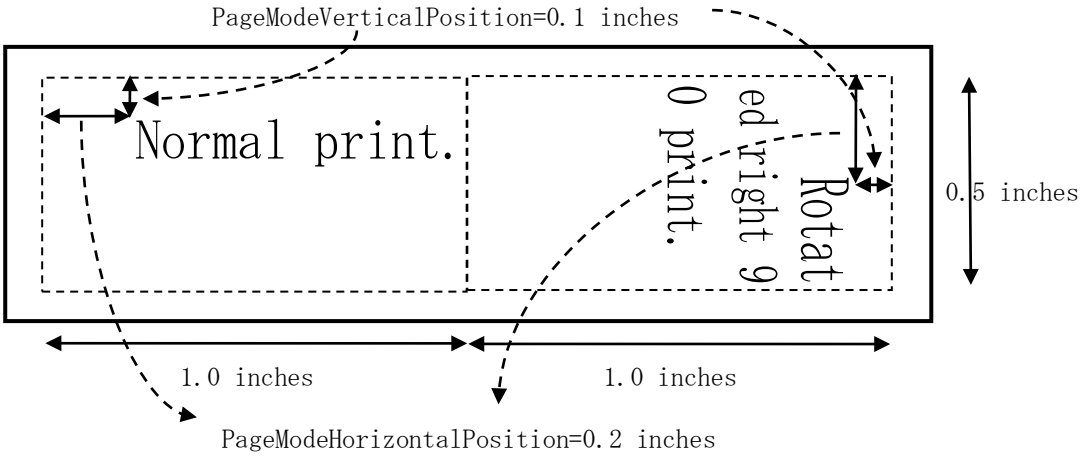
The code sample above will generate the following receipt.



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.setPageModeVerticalPosition(100);
myptr.setPageModeHorizontalPosition(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.pageModePrint(PTR_PM_NORMAL);
```

The code sample above will generate the following receipt.



Specify PTR_S_RECEIPT(2) for **PageModeStation** 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.

PageModeStation Property R/W

Type **int**

Remarks Sets the print station for subsequent Page Mode properties.
The available station is PTR_S_RECEIPT(2) only.
This property is initialized to 0 by **open**.
Be sure to specify PTR_S_RECEIPT(2) for this property before accessing the property or method of the Page Mode function.

PageModeVerticalPosition Property R/W

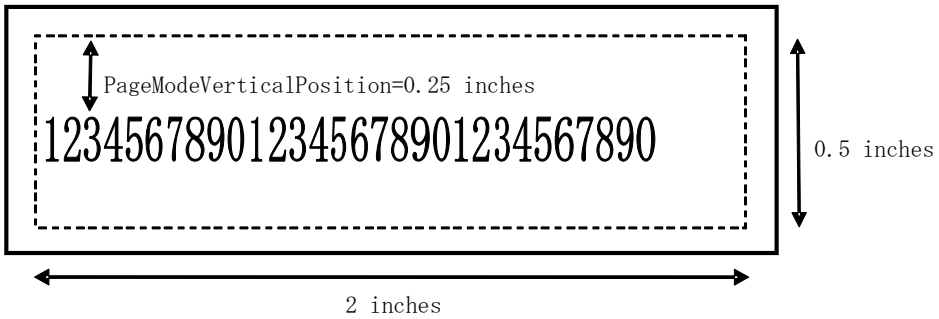
Type **int**

Remarks Holds the vertical start position offset within the print area for the selected **PageModeStation**.
This is expressed in the unit specified for **MapMode**.
The vertical direction is perpendicular to the direction specified in the actual **PageModePrintDirection**. A read/get on this property will return the vertical position offset set by the last write/set and not the current position.

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

```
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");
myptr.pageModePrint(PTR_PM_NORMAL);
```

The code sample above will generate the following receipt.



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

RecBarCodeRotationList Property

Type **String**

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

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

RecBitmapRotationList Property

Type **String**

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

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

RecCartridgeState Property

Type **int**

Remarks Contains the status of the currently selected receipt cartridge (ink, ribbon, or toner).
The following table shows the valid property values.

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

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

RecCurrentCartridge Property R/W

Type **int**

Remarks Specifies the currently selected receipt cartridge.
The following table shows the valid property values.

Value	Meaning
PTR_COLOR_PRIMARY (0x00000001)	Primary cartridge is selected.

Only PTR_COLOR_PRIMARY(0x00000001) can be set.
This property is initialized to PTR_COLOR_PRIMARY(0x00000001) when the device is first enabled by **open**.

RecEmpty Property

Type **boolean**

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

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

This property is initialized while the device is disabled and keeps the current state.
When **CoverOpen** is true, **RecEmpty** is not updated.

RecLetterQuality Property R/W

Type **boolean**

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

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

When **PrintSpeed** of JposEntry is set to "0", this property is valid.
Then the speed of printer is changed depending on the print mode.
When **PrintSpeed** of JposEntry is besides "0", this property is invalid.
Then the speed of printer is set in fixed speed and does not depend on this property.
This property is initialized to false when the device is first enabled by **open**.

RecLineChars Property R/W

Type **int**

Remarks Holds the number of characters that may be printed on a receipt line.
This property is set to one of the values which **RecLineCharsList** has.
Depending on the specified value, the printer prints in the following print font.

・When **RecLineWidth** is 360:

Value	Print Font (H × W)	Character space	RecLineHeight	Available Range to specify
30	24 × 12 dots	0 dots	24	1 to 30
40	16 × 8 dots	1 dot	16	31 to 40

・When **RecLineWidth** is 432:

Value	Print Font (H × W)	Character space	RecLineHeight	Available Range to specify
27	24 × 12 dots	4 dots	24	1 to 27
30	24 × 12 dots	2 dots	24	28 to 30
33	24 × 12 dots	1 dot	24	31 to 33
36	24 × 12 dots	0 dots	24	34 to 36
43	16 × 8 dots	2 dots	16	37 to 43
48	16 × 8 dots	1 dot	16	44 to 48
54	16 × 8 dots	0 dots	16	49 to 54

・When **RecLineWidth** is 512:

Value	Print Font (H × W)	Character space	RecLineHeight	Available Range to specify
42	24 × 12 dots	0 dots	24	1 to 42
56	16 × 8 dots	1 dot	16	43 to 56

・When **RecLineWidth** is 576:

Value	Print Font (H × W)	Character space	RecLineHeight	Available Range to specify
36	24 × 12 dots	4 dots	24	1 to 36
41	24 × 12 dots	2 dots	24	37 to 41
44	24 × 12 dots	1 dot	24	42 to 44
48	24 × 12 dots	0 dots	24	45 to 48
57	16 × 8 dots	2 dots	16	49 to 57
64	16 × 8 dots	1 dot	16	58 to 64
72	16 × 8 dots	0 dots	16	65 to 72

If the setting value is not supported, then the exception is thrown.
For example, when **RecLineWidth** is 360 and "41" is set, and then an exception is thrown.
Setting **RecLineChars** may also update **RecLineHeight**, **RecLineSpacing**, **RecSidewaysMaxChars** and **RecSidewaysMaxLines**.
This property is initialized to the value set to **RecLineChar** of JposEntry when the device is first enabled by **open**.

RecLineCharsList Property

Type **String**

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

Value	Meaning
"30,40"	The value of **RecLineCharsList** when PaperSize of JposEntry is 0.
"27,30,33,36,43,48,54"	The value of **RecLineCharsList** when PaperSize of JposEntry is 1.
"42,56"	The value of **RecLineCharsList** when PaperSize of JposEntry is 2.
"36,41,44,48,57,64,72"	The value of **RecLineCharsList** when PaperSize of JposEntry is 3.

RecLineHeight Property R/W

Type **int**

Remarks Holds the print line height.
This property is expressed in the unit specified for **MapMode**.
This property is not settable. This property is automatically set by **RecLineWidth** and **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 **MapMode** is PTR_MM_DOTS(1))

Value	Meaning
24	The value of **RecLineHeight** when font type is FONT A (24 × 12 dots).
16	The value of **RecLineHeight** when font type is FONT B (16 × 8 dots).

For the relationship between **RecLineChars** and the print font, see **RecLineChars**.
This property is initialized when the device is first enabled by **open**.

RecLineSpacing Property R/W

Type **int**

Remarks Holds the spacing of each single-high print line. This includes both the printed line height plus the whitespace between each pair of lines.
This property is expressed in the unit specified for **MapMode**.
The configurable range differs depending on the setting of **RecLineWidth** and **RecLineChars**.
The following table shows the valid configurable ranges. (When **MapMode** is PTR_MM_DOTS(1))

RecLineWidth	RecLineChars	Value
360	30	24 to 255
	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
576	36	24 to 255
	41	
	44	
	48	
	57	16 to 255
	64	
	72	

If the value other than the above is specified, the exception is notified.
This property is initialized to the value set to **RecLineSpacing** of JposEntry when the device is first enabled by **open**.

In RP-F10 and RP-D10, when the printer function setting "Paper Saving" is enabled, the line spacing and the space between lines are printed with the specified value in the printer function setting "Paper Saving". Therefore, the line spacing will not be as specified in the **RecLineSpacing**.
For details about "Paper Saving", see "RP-F10 SERIES THERMAL PRINTER TECHNICAL REFERENCE", or "RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE".

RecLinesToPaperCut Property

Type **int**

Remarks Holds the number of lines that must be advanced before cutting the receipt paper.
This property is determined by the following calculation based on the printer and **RecLineSpacing**.

[Calculation formula]
RP-F10: **RecLinesToPaperCut** = 76 / **RecLineSpacing**
RP-E10: **RecLinesToPaperCut** = 100 / **RecLineSpacing**
RP-D10: **RecLinesToPaperCut** = 88 / **RecLineSpacing**

Example:
When **RecLineSpacig** is "30" (**MapMode**=PTR_MM_DOTS(1))
RecLinesToPaperCut = 76 / 30=2.53… = 3
(round a number up to the nearest integer)

This property is initialized based on the above calculation when the device is first enabled by **open**.

RecLineWidth Property

Type **int**

Remarks Holds the width of a line of the printer.
The property is expressed in the unit specified for **MapMode**.
This property is initialized to one of the following values when the device is first enabled by **open**. (When **MapMode** is PTR_MM_DOTS(1))

Value	Meaning
360	The value of **RecLineWidth** when PaperSize of JposEntry is 0.
432	The value of **RecLineWidth** when PaperSize of JposEntry is 1.
512	The value of **RecLineWidth** when PaperSize of JposEntry is 2.
576	The value of **RecLineWidth** when PaperSize of JposEntry is 3.

RecNearEnd Property

Type **boolean**

Remarks Notifies whether the receipt is low.
The following table shows the valid property values.

Value	Meaning
false	The receipt is not low.
true	The receipt is low.

This property is initialized and kept current while the device is enabled.
When **CapRecNearEndSensor** is false, the value of this property is always false.
When **CapRecNearEndSensor** is true and **RecEmpty** is true, **RecNearEnd** always indicates true.

RecSidewaysMaxChars Property

Type **int**

Remarks Holds the maximum number of 1-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 **PageModeArea** and print font.

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

Example:
When **PageModeArea** is "576,2400", and **RecLineChars** is "48"
RecSidewaysMaxChars = 2400 / ((24 / 2) + 0) = 200
(round a number down to the nearest integer)

This property is initialized based on the above calculation when the device is first enabled by **open**.

RecSidewaysMaxLines Property

Type **int**

Remarks Holds the maximum number of lines that may be printed in sideways mode (rotated 90° to the left or right).
This property is determined by the following calculation based on **RecLineWidth**, **RecLineSpacing** and **RecLineHeight**.

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

Example:

When **RecLineWidth** is 576, **RecLineHeight** is 24 and **RecLineSpacing** is 30.

RecSidewaysMaxLines = (576 - 24) / 30 + 1 = 19

(round a number down to the nearest integer)

This property is initialized based on the above calculation when the device is first enabled by **open**.

RotateSpecial Property R/W

Type **int**

Remarks Holds the rotation orientation for barcodes.
The following table shows the valid property values.

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

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

While the *rotation* that is **rotatePrint** includes PTR_RP_BARCODE(4096), the value of this property is ignored.

4.3.3 Common Methods

checkHealth Method

Syntax **void checkHealth(int** *level*) throws **JposException**;

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

Value	Meaning
JPOS_CH_INTERNAL(1)	Executes an internal test without changing the physical state of the device. Confirm if the printer is in printable condition.
JPOS_CH_EXTERNAL(2)	Executes the test print after confirming the communication with the printer. ROM version ID of the printer, **DeviceServiceVersion**, and, **PhysicalDeviceName** are printed.
JPOS_CH_INTERACTIVE(3)	Executes an interactive test of the device. The software displays the modal dialog and prints the ROM version ID of the printer, **DeviceServiceVersion**, and **PhysicalDeviceName**.

Remarks Calls this method to test the state of a device.
A text description of the results of this method is placed in **CheckHealthText**.
checkHealth is always executed synchronously.

claim Method

Syntax **void claim(int** *timeout*) throws **JposException**;

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

Remarks Requests exclusive access to the device.
The POSPrinter device cannot be used until exclusive access is obtained.
When successful, **Claimed** is changed to true.
When the power is OFF or the cable is not connected, this method is failed.

clearOutput Method

Syntax **void clearOutput**() throws **JposException**;

Remarks Clears all device output that has been buffered.
Any output error events that are enqueued (usually waiting for **FreezeEvents** to be set to false) are also cleared.

The printer itself will not be reset.

close Method

Syntax **void close**() throws **JposException**;

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

compareFirmwareVersion Method

Syntax **void compareFirmwareVersion**(**String** *firmwareFileName*) throws **JposException**;

Remarks This method is not supported.

directIO Method

Syntax **void directIO**(**int** *command*, **int**[] *data*, **Object** *object*) throws **JposException**;

Parameter	Remarks
command	Command number. Specific values assigned by the POSPrinter device.
data	An array of 1 mutable integer. Specific values or usage vary by *command* and the POSPrinter device.
object	Additional data. Usage vary by *command* and the POSPrinter device.

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

directIO is always executed synchronously.

- **Remaining memory capacity response**

Parameter	Remarks
command	3
data	OUT Remaining memory
object	*null*

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

- **Status response**

Parameter	Remarks
command	501
data	OUT Status
object	*null*

Returns the paper sensor status in a numeric value. The response data is stored in *data*.

Out Status	Status	
	RP-F10, RP-D10	RP-E10[*1]
0	Paper ready	Paper ready / no near-end
1	No paper	No paper / no paper-near-end
2	-	Paper ready / paper-near-end
3	-	No paper / paper-near-end

*1: When **CapRecNearEndSensor** is false, the status is always "not paper-near-end".

- **International character selection**

Parameter	Remarks
command	201
data	International character number *n* 0 ≤ *n* ≤ 12
object	*null*

Selects the international character.
The following international characters are supported.

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

International character select by this function continues until **CharacterSet** is changed.
When **CharacterSet** is changed, the initial value is set.

open Method

Syntax **void open**(**String** *logicalDeviceName*) throws **JposException**;

logicalDeviceName specifies the device name to open.
Specify the registered logical name of the POSPrinter device to execute this method.

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

release Method

Syntax **void release**() throws **JposException**;

Remarks Releases exclusive access to the device.
If **DeviceEnabled** is true, then the device is also disabled.
Do not execute this method while the event is in progress (or in the event handler).

resetStatistics Method

Syntax **void resetStatistics**(**String** *statisticsBuffer*) throws **JposException**;

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

retrieveStatistics Method

Syntax **void retrieveStatistics**(**String**[1] *statisticsBuffer*) throws **JposException**;

Parameter	Remarks
statisticsBuffer	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 specified device statistics.
When this method is successful, the XML string of the statistics is returned.
For the statistics that can be retrieved, see "Appendix B Statistics".
retrieveStatistics is always executed synchronously.

updateFirmware Method

Syntax **void updateFirmware**(**String** *firmwareFileName*) throws **JposException**;

Remarks This method is not supported.

updateStatistics Method

Syntax **void updateStatistics**(**String** *statisticsBuffer*) throws **JposException**;

Remarks This method is not supported.

4.3.4 Specific Methods

beginInsertion Method

Syntax **void beginInsertion(int** *timeout*) throws **JposException**;

Remarks This method is not supported.

beginRemoval Method

Syntax **void beginRemoval(int** *timeout*) throws **JposException**;

Remarks This method is not supported.

changePrintSide Method

Syntax **void changePrintSide(int** *side*) throws **JposException**;

Remarks This method is not supported.

clearPrintArea Method

Syntax **void clearPrintArea**() throws **JposException**;

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

cutPaper Method

Syntax **void cutPaper(int** *percentage*) throws **JposException**;

Parameter	Remarks
percentage	Specifies the percentage of the paper to be cut. Valid values are 0 to 100.

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 Cuts the receipt paper.
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**.
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**, or while Page Mode by **pageModePrint** is selected, this method fails 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**.

In RP-F10 and RP-D10, when the printer function setting "Paper Saving" is enabled, the value specified in the printer function setting "Paper Saving" is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed (The value specified by **RecLineSpacing** is ignored).
However, when the "Paper cut" escape sequence (ESC|[#]P), the "Feed and Paper cut" escape sequence (ESC|[#]fP), or **cutPaper** 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.
See "RP-F10 SERIES THERMAL PRINTER TECHNICAL REFERENCE", or "RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details.

drawRuledLine Method

Syntax **void drawRuledLine(int** *station*, **String** *positionList*, **int** *lineDirection*, **int** *lineWidth*, **int** *lineStyle*, **int** *lineColor*) throws **JposException**;

Remarks This method is not supported.

endInsertion Method

Syntax **void endInsertion**() throws **JposException**;

Remarks This method is not supported.

endRemoval Method

Syntax **void endRemoval**() throws **JposException**;

Remarks This method is not supported.

markFeed Method

Syntax **void markFeed(int** *type*) throws **JposException**;

Remarks This method is not supported.

pageModePrint Method

Syntax **void pageModePrint(int** *control*) throws **JposException**;

The values of the *control* are as follows.

Value	Meaning
PTR_PM_PAGE_MODE(1)	Enters Page Mode.
PTR_PM_PRINT_SAVE(2)	Prints the print data of the Page Mode print area and saves the data. This is used for repeated printings.
PTR_PM_NORMAL(3)	Prints the print area and destroys the canvas and 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 **PageModeStation**.

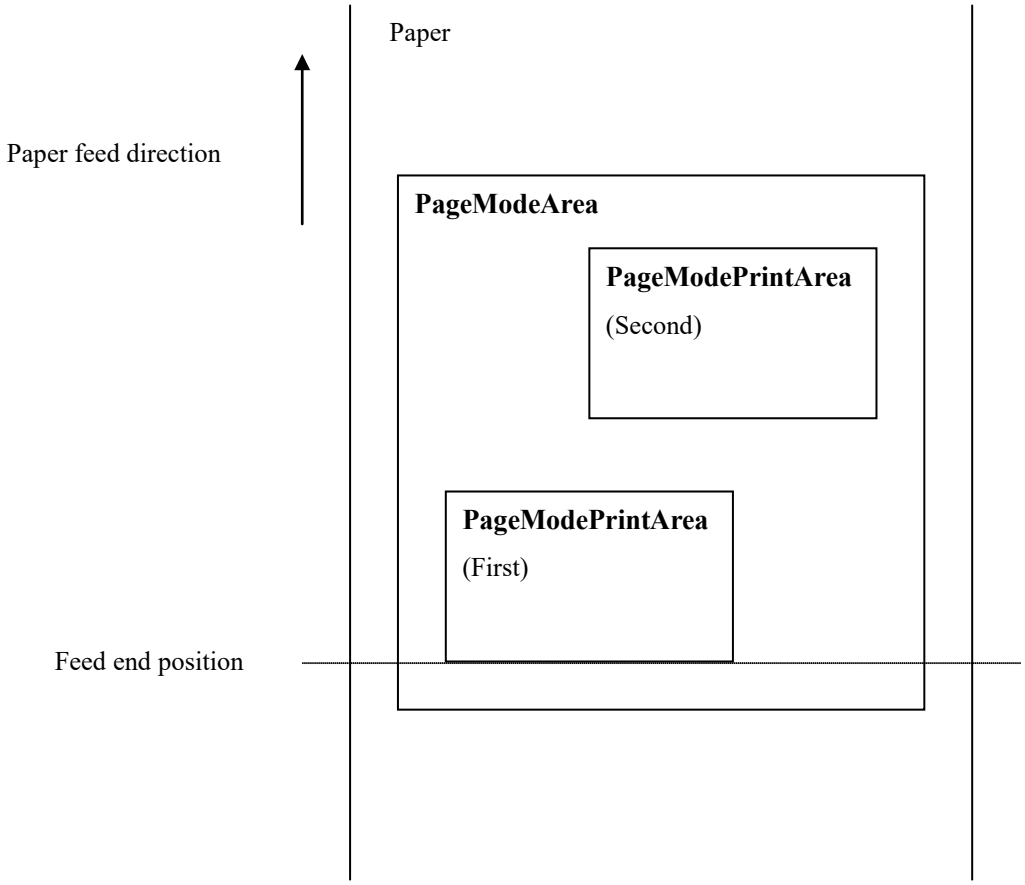
If PTR_PM_PAGE_MODE(1) is specified for *control*, then Page Mode is entered. Subsequently, the print data can be buffered with **printNormal**, **printBarCode**, **printBitmap**, or **printMemoryBitmap** until **pageModePrint** is called by specifying PTR_PM_PRINT_SAVE(2), PTR_PM_NORMAL(3), or PTR_PM_CANCEL(4).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

If PTR_PM_PRINT_SAVE(2) is specified for *control*, then Page Mode is continued. If some data is buffered by one of **printNormal**, **printBarCode**, **printBitmap**, and **printMemoryBitmap**, then the data is saved and printed. This control is used to print the same page layout with additional print items inside of the page.

If PTR_PM_NORMAL(3) is specified for *control*, then Page Mode is exited to return to the normal state.
If some data is buffered by one of **printNormal**, **printBarCode**, **printBitmap**, and **printMemoryBitmap**, then the data is printed. The buffered data will not be saved.

If PTR_PM_CANCEL(4) is specified for *control*, then Page Mode is exited to return to the normal state.
If some data is buffered by one of **printNormal**, **printBarCode**, **printBitmap**, and **printMemoryBitmap**, then the data is not printed or saved.

When **pageModePrint** is called with PTR_PM_NORMAL(3) or PTR_PM_PRINT_SAVE(2) specified for *control*, all of the print data on the Page Mode print area defined by **PageModePrintArea** is printed, and the paper is fed to the end of the area. If more than 1 Page Mode print area is defined, then after **PageModePrint** is called, all of the data that is to be printed in the respective Page Mode print area(s) is printed, and the paper is fed to the end of the Page Mode print area located the farthest "down" the sheet of paper. (See figure below.)



The entire Page Mode area is processed in a batch as 1 transaction.
This method is executed synchronously if **AsyncMode** is false, and asynchronously if **AsyncMode** is true.
Calling **clearOutput** cancels Page Mode to return to the normal state. The buffered print data is also cleared.
The Page Mode function can be used within a transaction print, but not within a rotate print.
Specify PTR_S_RECEIPT(2) for **PageModeStation** before calling this method.

printBarCode Method

Syntax **void printBarCode(int** *station*, **String** *data*, **int** *symbology*, **int** *height*, **int** *width*, **int** *alignment*, **int** *textPosition*) throws **JposException**;

Parameter	Remarks
station	Station to be used. Specify PTR_S_RECEIPT(2).
data	Character string of the barcode.
symbology	Barcode type to be used. See the table of *symbology*.

Parameter	Remarks
height	Barcode height. Expressed in the unit specified for **MapMode**. For PTR_MM_DOTS(1), specify the value from 1 to 255. *height* of the following barcodes are ignored and set automatically by *width*. Specify from 1 to 255. • QR Code • GS1 Databar Omni-directional*1 • GS1 Databar Expanded*1 • GS1 Databar Expanded Stacked*1 • GS1 Databar Limited*1 • GS1 Databar Truncated*1 *height* of the following barcodes are ignored and set with a fixed value. Specify from 1 to 255. • GS1 Databar Stacked Omni-directional*1 • GS1 Databar Stacked*1 During Page Mode by **pageModePrint**, specify the value within the range of print area specified by **PageModePrintArea** and **PageModeVerticalPosition**.
width	Barcode width. Expressed in the unit specified for **MapMode**. The width of the barcode actually printed is the best fit within the width specified by *width*. For PTR_MM_DOTS(1), specify the value from 1 to **RecLineWidth** for upright position. When rotating the barcode 90° right/left by **RotateSpecial** or rotatePrint, specify in the range not exceeding the maximum value of the printer. (See **PageModeArea** for the maximum value of the printer.) During Page Mode by **pageModePrint**, specify the value within the range of print area specified by **PageModePrintArea** and **PageModeHorizontalPosition**.
alignment	Position of the barcode. See the table of *alignment* below.
textPosition	Position of the HRI character. See the table of *textPosition* below.

*1: Supported only by RP-F10.

・ Values of *symbology*

Value	Meaning
PTR_BCS_UPCA(101)	UPC-A
PTR_BCS_UPCE(102)	UPC-E
PTR_BCS_EAN8(103)	EAN8 (JAN8)
PTR_BCS_JAN8(103)	JAN8 (EAN8)
PTR_BCS_EAN13(104)	EAN13 (JAN13)
PTR_BCS_JAN13(104)	JAN13 (EAN13)
PTR_BCS_EAN13_S(119)	EAN13 (JAN13) with supplemental barcode
PTR_BCS_ITF(106)	Interleaved 2 of 5
PTR_BCS_Codabar(107)	Codabar (NW-7)

Value	Meaning
PTR_BCS_Code39(108)	Code39
PTR_BCS_Code93(109)	Code93
PTR_BCS_Code128(110)	Code128
PTR_BCS_Code128_Parsed(123) *1	Code128 Parsed
PTR_BCS_GS1DATABAR(131) *1	GS1 Databar GS1 Databar Omni-directional
PTR_BCS_GS1DATABAR_E(132) *1	GS1 Databar Expanded
PTR_BCS_GS1DATABAR_S(133) *1	GS1 Databar Stacked Omni-directional
PTR_BCS_GS1DATABAR_E_S(134) *1	GS1 Databar Expanded Stacked
PTR_BCS_PDF417(201)	PDF417
PTR_BCS_QRCODE(204)	QR Code (Mixed mode)
PTR_BCS_OTHER+6(507) *1	GS1 Databar Limited
PTR_BCS_OTHER+7(508) *1	GS1 Databar Stacked
PTR_BCS_OTHER+10(511) *1	GS1 Databar Truncated

*1: Supported only by RP-F10.

・ Values of *alignment*

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 specified for **MapMode**.

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

・ Values of *textPosition* parameter

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 when printing the barcode at the specified printer.
This method is executed synchronously if **AsyncMode** is false, and asynchronously if **AsyncMode** is true.
If **RotateSpecial** indicates that the barcode is rotated, the barcode is printed in rotated mode. *height*, *width*, and *textPosition* are applied to the barcode before it is rotated.
For example, when specifying PTR_RP_LEFT90(258) for **RotateSpecial** and calling this method specifying PTR_BC_TEXT_BELOW(-13) for *textPosition* of this method, the text is placed below the barcode, and the text and barcode are rotated 90° to the left and printed.

In RP-F10, the barcode quiet zone is not secured. Verify that the barcode can be read with your actual device beforehand.
However, the following barcode quiet zones are unnecessary or secured:

- GS1 Databar Omni-directional
- GS1 Databar Expanded
- GS1 Databar Stacked Omni-directional
- GS1 Databar Expanded Stacked
- GS1 Databar Limited
- GS1 Databar Stacked
- GS1 Databar Truncated

The limitations for the parameters of each barcode are described as follows when **MapMode** is PTR_MM_DOTS(1).

[UPC-A] PTR_BCS_UPCA(101)

Parameter	Limitation
data	Specify 11 or 12 letters consisting of '0' to '9'. The 12th letter does not affect the barcode printing data.
width	RP-F10: *width* = 95 × X 190 ≤ *width* ≤ 570 RP-E10, RP-D10: *width* = 113 × X 226 ≤ *width* ≤ 678 X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *width.*

[UPC-E] PTR_BCS_UPCE(102)

Parameter	Limitation
data	Specify 11 or 12 letters consisting of '0' to '9'. The 12th letter does not affect the barcode printing data.
width	RP-F10: *width* = 51 × X 102 ≤ *width* ≤ 306 RP-E10, RP-D10: *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, JPOS_E_ILLEGAL(106) is thrown.

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

[EAN8 (JAN8)] PTR_BCS_EAN8(103), PTR_BCS_JAN8(103)

Parameter	Limitation
data	Specify 7 or 8 letters consisting of '0' to '9'. The 8th letter does not affect the barcode printing data.
width	RP-F10: *width* = 67 × X 134 ≤ *width* ≤ 402 RP-E10, RP-D10: *width* = 81 × X 162 ≤ *width* ≤ 486 X: fine element width 2 ≤ X ≤ 6 X is automatically set according to *width.*

[EAN13 (JAN13)] PTR_BCS_EAN13(104), PTR_BCS_JAN13(104)

Parameter	Limitation
data	Specify 12 or 13 letters consisting of '0' to '9'. The 13th letter does not affect the barcode printing data.
width	RP-F10: *width* = 95 × X 190 ≤ *width* ≤ 570 RP-E10, RP-D10: *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] PTR_BCS_EAN13_S(119)

Parameter	Limitation
data	Specify 14, 15, 17, or 18 letters consisting of '0' to '9'. When 15 letters or 18 letters are entered, the 13th character does not affect the printing data.
width	RP-F10: • For 14 or 15 letters *width* = 122 × X 244 ≤ *width* ≤ 732 • For 17 or 18 letters *width* = 149 × X 298 ≤ *width* ≤ 894 RP-E10, RP-D10: • 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] PTR_BCS_ITF(106)

Parameter	Limitation
data	Specify any value consisting of '0' to '9'. Note that the number of specified letters must be an even number except for 0.
width	RP-F10: *width* = ((D × 2 + 1) × X × N) + ((D × 3 + 6) × X) 14 × D + 16 ≤ *width* ≤ 54 × D + 54 RP-E10, RP-D10: *width* = ((D × 2 + 1) × X × N) + (((D × 3) + 6 + (10 × 2)) × X) (14 × D + 56) ≤ *width* ≤ (54 × D + 174) D: the number of barcode characters X: fine element width 2 ≤ X ≤ 6 N: ratio of wide element width to fine element width (Set to 2, 2.5, or 3) X and N are automatically set according to *width*.

[Codabar (NW-7)] PTR_BCS_Codabar(107)

Parameter	Limitation
data	The head and end of line must be one of 'A' to 'D'. Other data must be at least one of '0' to '9', '$', '+', ':', '-', '.', and '/'.
width	RP-F10: *width* = (((6 × X) + (2 × X × N)) × D) + ((X × N - X) × D') + (-1 × X) 20 × D + 2 × D' – 2 ≤ *width* ≤ 72 × D + 12 × D' – 6 RP-E10, RP-D10: *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 characters D': the number of data characters (the number of 'A' to 'D', '+', '.', '/', ':' included in barcode data) X: fine element width 2 ≤ X ≤ 6 N: ratio of wide element width to fine element width (Set to 2, 2.5, or 3) X and N are automatically set according to *width*.

[Code39] PTR_BCS_Code39(108)

Parameter	Limitation
data	At least one of '0' to '9', 'A' to 'Z', space, '$', '%', '+', '-', '.', '/' must be specified.
width	RP-F10: *width* = (((X × 7) + (X × N × 3)) × (D + 2)) + (-1 × X) 26 × D + 50 ≤ *width* ≤ 96 × D + 186) RP-E10, RP-D10: *width* = (((X × 7) + (X × N × 3)) × (D + 2)) + ((10 × 2 - 1) × X) (26 × D + 90) ≤ *width* ≤ (96 × D + 306) D: the number of barcode characters X: fine element width 2 ≤ X ≤ 6 N: ratio of wide element width to fine element width (Set to 2, 2.5, or 3) X and N are automatically set according to *width*.

[Code93] PTR_BCS_Code93(109)

Parameter	Limitation
data	Specify any value consisting of decimal numbers from 0 to 46. Each numeric value is treated as the corresponding character shown in the table below.
width	RP-F10: *width* = X × ((D + 2 + 2) × 9) + 1) 18 × D + 74 ≤ *width* ≤ 54 × D + 222 RP-E10, RP-D10: *width* = X × ((10 × 2) + ((D + 2 + 2) × 9) + 1) 18 × D + 114 ≤ *width* ≤ 54 × D + 342 D: the number of barcode characters X: fine element width 2≤X≤6 X is automatically set according to *width*.

・Character set of Code93

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

[Code128] PTR_BCS_Code128(110)

Parameter	Limitation
data	Specify any value consisting of decimal numbers from 0 to 105. Each numeric value is treated as the corresponding character shown in the table below. When the first letter is not a decimal number 103, 104, or 105, 105 is added to the first letter.
width	RP-F10: *width* = X ×(((D + 2) × 11) + 2) 22 × D + 48 ≤ *width* ≤ 66 × D + 144 RP-E10, RP-D10: *width* = X × ((10 × 2) + ((D + 2) × 11) + 2) (22 × D + 88) ≤ *width* ≤ (66 × D + 264) D: the number of barcode characters (including start code) X: fine element width 2 ≤ X ≤ 6 When the value of *width* is between 2 and 6, it is the narrow element width. Otherwise, X is automatically set according to the *width*.

・Character set of Code128

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
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

Number	Character			Number	Character		
	CODE A	CODE B	CODE C		CODE A	CODE B	CODE C
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)		

[Code128 Parsed] PTR_BCS_Code128_Parsed(123)

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

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

- Code128 Special Code Table

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

Input example of *data*

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

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

Selecting Code A : *data*="{A0123"

Selecting Code B : *data*="{B0123"

Selecting Code C : *data*="{C0123" or *data*="{C123"

[GS1 Databar/GS1 Databar Omni-directional] PTR_BCS_GS1DATABAR(131)

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

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

[GS1 Databar Expanded] PTR_BCS_GS1DATABAR_E(132)

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

[GS1 Databar Stacked Omni-directional] PTR_BCS_GS1DATABAR_S(133)

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

Module width is fixed at 2.

[GS1 Databar Expanded Stacked] PTR_BCS_GS1DATABAR_E_S(134)

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

Module width is fixed at 2.

[PDF417] PTR_BCS_PDF417(201)

Parameter	Limitation
data	00H to 7FH must follow the ASCII code and 80H to FFH must follow the extended character set of PC437 English list.
width	RP-F10: *width* = (17C + 69) × X *height* = R × Y RP-E10, RP-D10: *width* = (17C + 69) × X + (2X × 2) *height* = RY + (2X × 2) X: module width (2 to 4) Y: module height (2 to 127) C: the number of vertical columns (1 to 30) R: the number of rows (3 to 90) For the number of rows and the number of vertical columns, the smallest value that input data can be converted to barcode is selected. For module width and module height, the maximum size that does not exceed *width* and *height* is selected after the number of rows and the number of vertical columns are determined.

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

[QR Code (Mixed mode)] PTR_BCS_QRCODE (204)

Parameter	Limitation
data	Specify characters of the following range: ・ASCII characters ・8 bits Latin/Katakana characters based on JIS X 0201 ・Shift-JIS code based on JIS X 0208
width	RP-F10: *width* = (4V + 17) × X 42 ≤ *width* RP-E10, RP-D10 : *width* = (4V + 17) × X + (4X × 2) 42 ≤ *width* V: Version of QR Code (RP-F10: 1 to 18) (RP-E10, RP-D10: 1 to 40) X: Module size (RP-F10: 2 to 16) (RP-E10, RP-D10: 2 to 11) For version, the smallest value that input data can be converted to barcode is selected. When the value of *width* is between 2 and 16, it is the module size. Otherwise, the module size is selected as the maximum size that does not exceed the *width* after the version is determined.

QR Code model is fixed at 2 and the error correction level is fixed at M. Printing size is based on *width*, and *height* is ignored since QR Code is a square.
If data other than the printable characters is specified, JPOS_E_ILLEGAL(106) is thrown.

[GS1 Databar Limited] PTR_BCS_OTHER+6(507)

Parameter	Limitation
data	Specify 13 letters consisting of '0' to '9'.
width	The effective range of *height* differs by *width*. *width* = 79 × X 1[*1] ≤ *width* ≤ 474 X: module width 2 ≤ X ≤ 6 X is automatically set according to *width.*

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

[GS1 Databar Stacked] PTR_BCS_OTHER+7(508)

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

Module width is fixed at 2.

[GS1 Databar Truncated] PTR_BCS_OTHER+10(511)

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

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

printBitmap Method

Syntax **void printBitmap(int** *station*, **String** *fileName*, **int** *width*, **int** *alignment*) throws **JposException**;

Parameter	Remarks
station	Station to be used. Specify PTR_S_RECEIPT(2).
fileName	Name of bitmap file. For the supported image file, see below.
width	Print width of bitmap. See values below.
alignment	Print position of bitmap. See values below.

・ Supported bitmap file

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

・ Values of *width*

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

・ Values of the *alignment*

Value	Meaning
PTR_BM_LEFT(-1)	Left justify
PTR_BM_CENTER(-2)	Center
PTR_BM_RIGHT(-3)	Right justify
Other values	Printed with the left margin of the specified value. Expressed in the unit specified for **MapMode**.

During rotated 90° right/left mode by **rotatePrint** or during Page Mode by **pageModePrint**, the setting of *alignment* is invalid and the data is always aligned with left justify.

Remarks

Calls this method when printing a bitmap on the specified station.
The highest performance cannot be achieved since the bitmap data is sent to the printer after **printBitmap** is called. It is recommended to print the bitmap data using **setBitmap** and escape sequence.
If any character data is already sent but not yet printed, that character data is printed first, a linefeed is automatically added, and then the bitmap is printed on the next print line. Any character data sent after **printBitmap** is printed on the print line next to the bitmap.
This method is executed synchronously if **AsyncMode** is false, and asynchronously if **AsyncMode** is true.
width controls the transformation of bitmap data. If *width* is PTR_BM_ASIS(-11), then no transformation is executed. The bitmap is printed with 1 pixel per dot of the 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.

printImmediate Method

Syntax **void printImmediate(int** *station*, **String** *data*) throws **JposException**;

Parameter	Remarks
station	Station to be used. Specify PTR_S_RECEIPT(2).
data	Characters to be printed. Consists of printable characters, escape sequences, carriage returns (CR), and line feeds (LF).

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

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

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

In RP-F10 and RP-D10, when the printer function setting "Paper Saving" is enabled, the value specified in the printer function setting "Paper Saving" is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed (The value specified by **RecLineSpacing** is ignored).
However, when any of the "Paper cut" escape sequence (ESC|[#]P), the "Feed and Paper cut" escape sequence (ESC|[#]fP), or **cutPaper** 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.
See "RP-F10 SERIES THERMAL PRINTER TECHNICAL REFERENCE", or "RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details.

printMemoryBitmap Method

Syntax **void printMemoryBitmap(int** *station*, **byte[]** *data*, **int** *type*, **int** *width*, **int** *alignment*) throws **JposException**;

Parameter	Remarks
station	Station to be used. Specify PTR_S_RECEIPT(2).
data	Byte array holding the bitmap data. For the supported image file, see **printBitmap**.
type	Bitmap format. Only PTR_BMT_BMP (1) is supported.
width	Print width of bitmap. See **printBitmap**.
alignment	Print position of bitmap. See **printBitmap**.

Remarks Calls this method when printing the bitmap on the specified station.
For the operation specifications, see **printBitmap**.
This method is executed synchronously if **AsyncMode** is false, and asynchronously if **AsyncMode** is true.

printNormal Method

Syntax **void printNormal(int** *station*, **String** *data*) throws **JposException**;

Parameter	Remarks
station	Station to be used. Specify PTR_S_RECEIPT(2).
data	Characters to be printed. Consists of printable characters, escape sequences, carriage returns (CR), and line feeds (LF).

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

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

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

In RP-F10 and RP-D10, when the printer function setting "Paper Saving" is enabled, the value specified in the printer function setting "Paper Saving" is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed (The value specified by **RecLineSpacing** is ignored).
However, when any of the "Paper cut" escape sequence (ESC|[#]P), the "Feed and Paper cut" escape sequence (ESC|[#]fP), or **cutPaper** 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.
See "RP-F10 SERIES THERMAL PRINTER TECHNICAL REFERENCE", or "RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details.

printTwoNormal Method

Syntax **void printTwoNormal(int** *station*, **String** *data1*, **String** *data2*) throws **JposException**;

Remarks This method is not supported.

rotatePrint Method

Syntax **void rotatePrint(int** *station*, **int** *rotation*);

Parameter	Remarks
station	Station to be used. Specify PTR_S_RECEIPT(2).
rotation	Specifies the rotation direction. See values below.

The values of *rotation* are as follows.

Value	Meaning
PTR_RP_NORMAL(1)	End of rotated printing.
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.

Remarks Calls this method to execute the rotated printing.
This method is executed synchronously if **AsyncMode** is false, and asynchronously if **AsyncMode** is true.

If *rotation* is PTR_RP_ROTATE180(259), then upside-down print mode starts. Subsequent calls to **printNormal** or **printImmediate** will print the data upside-down until **rotatePrint** is called with the *rotation* set to PTR_RP_NORMAL(1). Lines are printed in the order that they are sent to the printer, with the start of each line justified at the right margin of the printer.
If PTR_RP_BARCODE(4096) is set as OR of PTR_RP_ROTATE180(259) in *rotation*, the barcode printing by **printBarCode** makes upside-down barcode.
Also, if PTR_RP_BITMAP(8192) is set as OR of PTR_RP_ROTATE180(259) in *rotation*, the bitmap printing by **printBitmap** makes upside-down bitmap.

If *rotation* is PTR_RP_RIGHT90(257) or PTR_RP_LEFT90(258), the sideways print mode starts. Subsequent calls to **printNormal** will buffer the print data until **rotatePrint** is called with *rotation* set to PTR_RP_NORMAL(1). Also, the value of **AsyncMode** does not affect its operation. In other words, no **OutputID** will be assigned to the request, nor will **OutputCompleteEvent** be notified.
Each print line is rotated by 90°. If all lines are not the same length, the start positions of the lines are aligned.
If PTR_RP_BARCODE (4096) is set in *rotation* as the OR of PTR_RP_ROTATE180(259), barcode printing by **printBarCode** is buffered.
Also, if PTR_RP_BITMAP(8192) is set in *rotation* as the OR of PTR_RP_ROTATE180(259), bitmap printing by the **printBitmap** method is buffered.
If *rotation* is PTR_RP_NORMAL(1), rotated print mode is exited. If some print data is buffered while the sideways rotated print is in effect, the buffered data is printed.

When *rotation* contains PTR_RP_BARCODE(4096), the **RotateSpecial** setting is ignored.

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

The maximum width is 2400 dots in the sideways print mode. If the print data per line exceeds this range, the width is 2400 dots and non-printed data is printed by feeding to the next print line.

If the bitmap print and barcode print by the "Print in-line barcode" escape sequence (ESC|#R) or the "Print bitmap" escape sequence (ESC|#B) is specified on the **printNormal** during the rotation mode, the print data rotates regardless of whether or not PTR_RP_BARCODE (4096) and PTR_RP_BITMAP(8192) is specified in *rotation* with OR.

setBitmap Method

Syntax

void setBitmap(int *bitmapNumber*, **int** *station*, **String** *fileName*, **int** *width*, **int** *alignment*) throws **JposException**;

Parameter	Remarks
bitmapNumber	The number to be assigned to this bitmap. Valid values are 1 to 20.
station	Station to be used. Specify PTR_S_RECEIPT(2).
fileName	Name of bitmap file. If an empty string is set, the bitmap setting is canceled. For the supported image file, see **printBitmap**.
width	Print width of bitmap. See the following values.
alignment	Print position of bitmap. See **printBitmap** for values.

・ Values of *width*

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

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

Remarks

Calls this method to save the information about the bitmap to be printed.
The saved content is not initialized by **close**. It is initialized when the class is generated.
When using multiple printers, execute **setBitmap** for each printer.
The bitmap may then be printed by calling **printNormal** or **printImmediate** with the "Print Bitmap" escape sequence (ESC|#B) in the print data. The "Print Bitmap" escape sequence (ESC|#B) usually contains the character strings for printing the start and end process headers.
If any character data was sent before the "Print Bitmap" escape sequence (ESC|#B) and has not been printed, that character data is printed first, a linefeed is automatically placed, and then the bitmap is printed. Any character data sent after the "Print Bitmap" escape sequence (ESC|#B) is printed on the line next to the bitmap.

POSPrinter device prepares for printing with downloading bitmap data to the NV graphics area of the printer. When bitmap print is specified by escape sequence, only command which conducts printing is transmitted to provide better performance.

setLogo Method

Syntax **void setLogo(int** *location*, **String** *data*) throws **JposException**;

Parameter	Remarks
location	Logo to be set. Specify PTR_L_TOP(1) or PTR_L_BOTTOM(2).
data	Characters that produce the logo. Consists of printable characters, escape sequences, carriage returns (CR), and line feeds (LF).

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

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

In RP-F10 and RP-D10, when the printer function setting "Paper Saving" is enabled, the value specified in the printer function setting "Paper Saving" is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed. (The value specified by **RecLineSpacing** is ignored).
However, when any of the "Paper cut" escape sequence (ESC|[#]P), the "Feed and Paper cut" escape sequence (ESC|[#]fP), or **cutPaper** 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.
See "RP-F10 SERIES THERMAL PRINTER TECHNICAL REFERENCE", or "RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details.

Remarks Calls this method to save a data string as the top or bottom logo.
The logo can be printed by calling **printNormal** or **printImmediate** with "Print Top Logo" escape sequence (ESC|tL) or "Print Bottom Logo" escape sequence (ESC|bL) in the print data.

data registered by this method holds the data in the character set of **CharacterSet** at the time when the method is executed. Therefore, execute **setLogo** with **CharacterSet** 932 to register the data including kanji character(s). After that, kanji character(s) will be printed even if **CharacterSet** is set to 999 when the data is printed by escape sequence.

transactionPrint Method

Syntax **void transactionPrint(int** *station*, **int** *control*) throws **JposException**;

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

・ values of *control* parameter

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**, or **pageModePrint** will buffer the print data at the Service Object until **transactionPrint** is called with the *control* set to PTR_TP_NORMAL(12). In this case, the print methods only validate the method parameters and buffer the data, and they do not initiate printing. Also, the value of **AsyncMode** does not affect its operation. In other words, no **OutputID** is assigned to the request and no **OutputCompleteEvent** is informed.

If *control* is PTR_TP_NORMAL(12), then transaction mode is exited. If some data was buffered by calls to **printNormal**, **cutPaper**, **rotatePrint**, **printBarCode**, **printBitmap**, and **pageModePrint**, then the buffered data is printed.

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

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

validateData Method

Syntax **void validateData(int** *station*, **String** *data*) throws **JposException**;

Parameter	Remarks
station	Station to be used. Specify PTR_S_RECEIPT(2).
data	Data to be validated. Consists of printable characters, escape sequences, carriage returns (CR), and line feeds (LF).

Remarks Before calling **printImmediate** or **printNormal**, calls this method when determining whether a data sequence, possibly including 1 or more escape sequences, is valid for the specified station.
This method does not cause any printing but is used to determine the capability of the printer.
When at least one of the specified escape sequences is not valid, the exception is thrown.

4.3.5 Events

DirectIOEvent Event

Interface **jpos.events.DirectIOListener**

Syntax **directIOOccurred(DirectIOEvent** *e*);

Remarks This event is not supported.

ErrorEvent Event

Interface **jpos.events.ErrorListener**

Syntax **errorOccurred(ErrorEvent** *e*);

ErrorEvent contains the following properties:

Parameter	Type	Remarks
ErrorCode	**int**	Factor code causing the error event.
ErrorCodeExtended	**int**	Extended code of the factor causing the error event.
ErrorLocus	**int**	Location of the error.
ErrorResponse	**int**	Pointer for error event response.

ErrorCode is set to one of the following values:

Value	Meaning
JPOS_E_NOHARDWARE(107)	The printer is powered off or the cable is not connected.
JPOS_E_FAILURE(111)	Communication with the printer failed.
JPOS_E_EXTENDED(114)	The error defined by the *ErrorCodeExtended* is occurred.

If *ErrorCode* is JPOS_E_EXTENDED(114), then the *ErrorCodeExtended* is set to one of the following values:

Value	Meaning
EPTR_COVER_OPEN(201)	The printer cover is open.
EPTR_REC_EMPTY(203)	The receipt is out of paper.
1001	Vp voltage error has occurred.
1002	Autocutter error has occurred.
1005	Head-temperature error has occurred.
1010	A non-recoverable error has occurred.

ErrorLocus is set to the following value:

Value	Meaning
JPOS_EL_OUTPUT(1)	Error occurred while processing asynchronous output.

ErrorResponse can be set to one of the following values by the application. The default value is JPOS_ER_RETRY(11):

Value	Meaning
JPOS_EL_RETRY(11)	Exits the error state and retries the asynchronous output.
JPOS_EL_CLEAR(12)	Exits the error state and clears the asynchronous output.

Remarks This event is enqueued when the error is detected.

OutputCompleteEvent Event

Interface **jpos.events.OutputCompleteListener**

Syntax **outputCompleteOccurred(OutputCompleteEvent** *e*);

OutputCompleteEvent contains the following property:

Parameter	Type	Remarks
OutputID	**int**	The ID number of the asynchronous output request that is complete.

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

StatusUpdateEvent Event

Interface **jpos.events.StatusUpdateListener**

Syntax **statusUpdateOccurred(StatusUpdateEvent** *e*);

StatusUpdateEvent contains the following property:

Parameter	Type	Remarks
Status	**int**	The state of printer.

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)	The receipt paper is low.
PTR_SUE_REC_PAPEROK(26)	The receipt paper is ready.

Value	Meaning
PTR_SUE_IDLE(1001)	All the asynchronous outputs finished either successfully or by cleared. When this event is enqueued, **State** is always JPOS_S_IDLE(2). **FlagWhenIdle** must be set to true for this event to be notified. The POSPrinter device sets **FlagWhenIdle** to false and notify the event.
JPOS_SUE_POWER_ONLINE(2001)*1	The device is powered on and ready.
JPOS_SUE_POWER_OFF_OFFLINE(2004)*1	The device is powered off or offline.

*1: It is notified when **PowerNotify** is JPOS_PN_ENABLED(1).

Remarks This event is notified when an important state change has occurred in the printer device. The POSPrinter device notifies the first **StatusUpdateEvent** when the device is change to enable.

Chapter 5 CashDrawer

5.1 Summery

5.1.1 Common Properties

Property Name	Type	Access	Availability Condition	Initial Value
CapCompareFirmwareVersion	**boolean**	R	open	false
CapPowerReporting	**int**	R	open	JPOS_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
DeviceEnabled	**boolean**	R/W	open	false
FreezeEvents	**boolean**	R/W	open	false
PowerNotify	**int**	R/W	open	JPOS_PN_DISABLED(0)
PowerState	**int**	R	open	JPOS_PS_UNKNOWN(2000)
State	**int**	R	open	JPOS_S_CLOSED(1)
DeviceControlDescription	**String**	R	-	"JavaPOS CashDrawer Device Control"
DeviceControlVersion	**int**	R	-	"1013000"
DeviceServiceDescription	**String**	R	open	"SII RP Series JavaPOS Cash Drawer Service Driver, Copyright (C) 20xx Seiko Instruments Inc."
DeviceServiceVersion	**int**	R	open	"1013XXX"
PhysicalDeviceDescription	**String**	R	open	"SII RP-F10 Cash Drawer"*1
				"SII RP-E10 Cash Drawer"*2
				"SII RP-D10 Cash Drawer"*3
PhysicalDeviceName	**String**	R	open	"RP-F10 Cash Drawer"*1
				"RP-E10 Cash Drawer"*2
				"RP-D10 Cash Drawer"*3

*1: In case of RP-F10
*2: In case of RP-E10
*3: In case of RP-D10

5.1.2 Specific Properties

Property Name	Type	Access	Availability Condition	Initial Value
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
claim	open
close	open
compareFirmwareVersion	open, claim, & enable
directIO	open
open	-
release	open & claim
resetStatistics	open, claim, & enable
retrieveStatistics	open, claim, & enable
updateFirmware	open, claim, & enable
updateStatistics	open, claim, & enable

5.1.4 Specific Methods

Method Name	Availability Condition
openDrawer	open & enable
waitForDrawerClose	open & enable

5.1.5 Events

Event Name	Availability Condition
DirectIOEvent	open & enable
StatusUpdateEvent	open & enable

5.2 Detail

This section describes the details of the CashDrawer.
For details of the thrown exception errors, see "Appendix A Exceptions".

5.2.1 Common Properties

CapCompareFirmwareVersion Property

Type **boolean**

Remarks Indicates whether a firmware file's version can be compared against the firmware version of the device.
The following table shows the valid property values.

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

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

CapPowerReporting Property

Type **int**

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

Value	Meaning
JPOS_PR_STANDARD(1)	2 types of power states, JPOS_PS_OFF_OFFLINE(2004) (power off or offline) and JPOS_PS_ONLINE(2001), can be determined and reported.

This property is initialized to JPOS_PR_STANDARD(1) by **open**.

CapStatisticsReporting Property

Type **boolean**

Remarks Indicates the statistics accumulation function of the device.
The following table shows the valid property values.

Value	Meaning
false	No statistical data regarding the device is available.

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

CapUpdateFirmware Property

Type **boolean**

Remarks Indicates whether the device supports firmware updating.
The following table shows the valid property values.

Value	Meaning
false	Firmware update is not supported.

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

CapUpdateStatistics Property

Type **boolean**

Remarks Indicates the function that some or all device statistics can be reset.
The following table shows the valid property values.

Value	Meaning
false	None of the statistical data can be reset/updated.

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

CheckHealthText Property

Type **String**

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

Value	Meaning
"Internal HCheck: Successful"	Confirmed that communication with the drawer is available.
"Internal HCheck: Failure"	Unable to confirm that communication with the drawer is available.
"External HCheck: Successful"	Succeeded in confirming communication with and opening the drawer.
"External HCheck: Failure"	Failed in confirming communication with and opening the drawer.
"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 property is initialized to empty string by **open**.

Claimed Property

Type **boolean**

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

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

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

DeviceEnabled Property R/W

Type **boolean**

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

Value	Meaning
false	The device has been disabled. If changed to false, then the device is physically disabled.
true	The device has been in an operational state. If changed to true, then the device is brought to an operational state.

The application must set this property to true before using the device.
If **State** is other than JPOS_S_IDLE(2), **DeviceEnabled** cannot be changed from true to false.

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

FreezeEvents Property R/W

Type **boolean**

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

Value	Meaning
false	The application allows events to be delivered. If some events have been held while events were frozen and all other conditions are correct for delivering the events, then changing **FreezeEvents** to false will cause these events to be delivered.
true	Events will not be delivered. Events will be enqueued until **FreezeEvents** is set to false.

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 **open**.

PowerNotify Property R/W

Type **int**

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

Value	Meaning
JPOS_PN_DISABLED(0)	The CashDrawer class will not provide any power notifications to the application. No power notification **StatusUpdateEvents** will be fired, and **PowerState** is not be set.
JPOS_PN_ENABLED(1)	The CashDrawer class will fire the power notification **StatusUpdateEvents** and update **PowerState**, when **DeviceEnabled** is set to true. The level of functionality depends upon **CapPowerReporting**.

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

This property is initialized to JPOS_PN_DISABLED(0) by **open**.

PowerState Property

Type **int**

Remarks Indicates the current power condition.
The following table shows the valid property values.

Value	Meaning
JPOS_PS_UNKNOWN(2000)	Cannot determine the device's power state, for one of the following reasons. • **PowerNotify** is JPOS_PN_DISABLED(0). • **DeviceEnabled** is false.
JPOS_PS_ONLINE(2001)	The device is powered on and ready.
JPOS_PS_OFF_OFFLINE(2004)	The device is powered off or offline.

This property is initialized to JPOS_PS_UNKNOWN(2000) by **open**.

State Property

Type **int**

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

Value	Meaning
JPOS_S_CLOSED(1)	The device is closed.
JPOS_S_IDLE(2)	The device is in a good state and is not busy.

This property is always readable.
This property returns JPOS _S_CLOSED(1) until **open** is successfully completed.

DeviceControlDescription Property

Type **String**

Remarks "JavaPOS CashDrawer Device Control" is set.
Holds an identifier for the Device Control.
This property is always readable.

DeviceControlVersion Property

Type **String**

Remarks "1013000" is set.
Holds the Device Control version number.
This property is always readable.

DeviceServiceDescription Property

Type **String**

Remarks Holds an identifier for the Device Service.
This property is initialized to "SII RP Series JavaPOS Cash Drawer Service Driver, Copyright (C) 20xx Seiko Instruments Inc." by **open**.

DeviceServiceVersion Property

Type **String**

Remarks Holds the Device Service version number.
This property is initialized to "1013XXX" by **open**.

PhysicalDeviceDescription Property

Type **String**

Remarks Holds an identifier for the physical device.
This property depends on the printer.
One of the following values is set to this property by **open**.

Printer	Value
RP-F10	"SII RP-F10 Cash Drawer"
RP-E10	"SII RP-E10 Cash Drawer"
RP-D10	"SII RP-D10 Cash Drawer"

PhysicalDeviceName Property

Type **String**

Remarks Holds a short name identifying the physical device.
This property depends on the printer.
One of the following values is set to this property by **open**.

Printer	Value
RP-F10	"RP-F10 Cash Drawer"
RP-E10	"RP-E10 Cash Drawer"
RP-D10	"RP-D10 Cash Drawer"

5.2.2 Specific Properties

CapStatus Property

Type **boolean**

Remarks Indicates whether the drawer can report its status or not.
The following table shows the valid property values.

Value	Meaning
true	The drawer can report its status.

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

CapStatusMultiDrawerDetect Property

Type **boolean**

Remarks Indicates whether the status unique to each drawer in multiple drawer configurations can be reported or not.
The following table shows the valid property values.

Value	Meaning
false	The status unique to each drawer in multiple drawer configurations cannot be reported.

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

DrawerOpened Property

Type **boolean**

Remarks Indicates whether the drawer is opened or closed.
The following table shows the valid property values.

Value	Meaning
false	The cash drawer is closed.
true	The cash drawer is opened.

This property depends on **DrawerOpenStatus** of JposEntry.

- If **DrawerOpenStatus** of JposEntry is set to "Low":
  When the sensor status is "Low", this property indicates true.

- If **DrawerOpenStatus** of JposEntry is set to "High":
  When the sensor status is "High", this property indicates true.

This property is initialized and kept current while the device is enabled.

5.2.3 Common Methods

checkHealth Method

Syntax **void checkHealth**(**int** *level*) throws **JposException**;

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

Value	Meaning
JPOS_CH_INTERNAL(1)	Executes a health check without using the device physically.
JPOS_CH_EXTERNAL(2)	Opens the drawer, if possible, after confirming the communication with the drawer using the device. This method fails when another application has exclusive access to the device.
JPOS_CH_INTERACTIVE(3)	Executes an interactive test of the device. The Service Object displays the modal dialog and opens the cash drawer, if possible.

Remarks Calls this method to test the state of a device.
A text description of the results of this method is placed in **CheckHealthText**.
This method fails when another application has exclusive access to the device.

claim Method

Syntax **void claim**(**int** *timeout*) throws **JposException**;

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

Remarks Requests exclusive access to the device.
Acquisition of exclusive access is not essential since the cash drawer device is a sharable device.
When successful, **Claimed** is changed to true.
When the power is OFF or the cable is not connected, this method is failed.

close Method

Syntax **void close**() throws **JposException**;

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

compareFirmwareVersion Method

Syntax **void compareFirmwareVersion**(**String** *firmwareFileName*) throws **JposException**;

Remarks This method is not supported.

directIO Method

Syntax **void directIO**(**int** *command*, **int**[] *data*, **Object** *object*) throws **JposException**;

Remarks This method is not supported.

open Method

Syntax **void open**(**String** *logicalDeviceName*) throws **JposException**;

logicalDeviceName specifies the device name to open.
Specify the registered logical name of the CashDrawer device to execute this method.

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

release Method

Syntax **void release**() throws **JposException**;

Remarks Releases exclusive access to the device.
If **DeviceEnabled** is true, then the device is also disabled.
Do not execute this while the event is in progress (or in the event handler).

resetStatistics Method

Syntax **void resetStatistics**(**String** *statisticsBuffer*) throws **JposException**;

Remarks This method is not supported.

retrieveStatistics Method

Syntax **void retrieveStatistics(String**[1] *statisticsBuffer*) throws **JposException**;

Remarks This method is not supported.

updateFirmware Method

Syntax **void updateFirmware(String** *firmwareFileName*) throws **JposException**;

Remarks This method is not supported.

updateStatistics Method

Syntax **void updateStatistics(String** *statisticsBuffer*) throws **JposException**;

Remarks This method is not supported.

5.2.4 Specific Methods

openDrawer Method

Syntax **void openDrawer**() throws **JposException**;

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

waitForDrawerClose Method

Syntax **void waitForDrawerClose**(**int** *beepTimeout*, **int** *beepFrequency*, **int** *beepDuration*, **int** *beepDelay*) throws **JposException**;

Remarks Waits for notification that the drawer is closed.
No control is returned until notification that **DrawerOpened** is false or printer power is off is detected.
The beep sound is not supported.
This method fails when another application has exclusive access to the device.

5.2.5 Events

DirectIOEvent Event

Interface **jpos.events.DirectIOListener**

Syntax **directIOOccurred(DirectIOEvent** *e*);

Remarks This event is not supported.

StatusUpdateEvent Event

Interface **jpos.events.StatusUpdateListener**

Syntax **statusUpdateOccurred(StatusUpdateEvent** *e*);

This event contains following property:

Parameter	Type	Description
Status	**int**	The drawer status.

Status is set to one of the following values.

Value	Meaning
StatusClosed(0)	The drawer is closed.
StatusOpen(1)	The drawer is open.
StatusPowerOnline(2001)*	The device is powered on and ready.
StatusPowerOffOffline(2004)*	The device is powered off or offline.

*: It is notified when **PowerNotify** is JPOS_PN_ENABLED(1).

Remarks This event is notified when an important state change has occurred in the drawer device. The CashDrawer class notifies the first **StatusUpdateEvent** when the device is changed to enable.

Appendix A Exceptions

A.1 POSPrinter

A.1.1 Common Properties

Property	ErrorCode	Meaning
CapCompareFirmwareVersion	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapPowerReporting	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapStatisticsReporting	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapUpdateFirmware	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapUpdateStatistics	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CheckHealthText	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
Claimed	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
DeviceEnabled	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_BUSY(113)	Asynchronous output is in progress.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=1010 A non-recoverable error has occurred.
FreezeEvents	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
OutputID	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.

Property	ErrorCode	Meaning
PowerNotify	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・ The setting range of property is improper. ・ **DeviceEnabled** is set to true.
PowerState	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
DeviceServiceDescription	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
DeviceServiceVersion	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
PhysicalDeviceDescription	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
PhysicalDeviceName	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.

A.1.2 Specific Properties

Property	ErrorCode	Meaning
AsyncMode	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapCharacterSet	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapConcurrentJrnRec	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapConcurrentJrnSlp	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapConcurrentPageMode	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapConcurrentRecSlp	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapCoverSensor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrn2Color	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrnBold	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrnCartridgeSensor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrnColor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrnDhigh	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrnDwide	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrnDwideDhigh	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrnEmptySensor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrnItalic	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrnNearEndSensor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrnPresent	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapJrnUnderline	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapMapCharacterSet	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRec2Color	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecBarCode	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecBitmap	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.

Property	ErrorCode	Meaning
CapRecBold	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecCartridgeSensor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecColor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecDhigh	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecDwide	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecDwideDhigh	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecEmptySensor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecItalic	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecLeft90	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecMarkFeed	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecNearEndSensor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecPageMode	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecPapercut	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecPresent	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecRight90	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecRotate180	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecRuledLine	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecStamp	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapRecUnderline	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlp2Color	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpBarCode	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpBitmap	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpBold	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpBothSidesPrint	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.

Property	ErrorCode	Meaning
CapSlpCartridgeSensor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpColor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpDhigh	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpDwide	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpDwideDhigh	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpEmptySensor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpFullslip	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpItalic	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpLeft90	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpNearEndSensor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpPageMode	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpPresent	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpRight90	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpRotate180	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpRuledLine	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapSlpUnderline	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapTransaction	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CartridgeNotify	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	An attempt was made to perform an unsupported operation with the Device.
CharacterSet	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Invalid parameter value was used.
CharacterSetList	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.

Property	ErrorCode	Meaning
CoverOpen	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
ErrorLevel	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
ErrorStation	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
ErrorString	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
FlagWhenIdle	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
FontTypefaceList	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
JrnCartridgeState	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
JrnCurrentCartridge	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Journal station is not supported.
JrnEmpty	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
JrnLetterQuality	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Journal station is not supported.
JrnLineChars	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Journal station is not supported.
JrnLineCharsList	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.

Property	ErrorCode	Meaning
JrnLineHeight	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Journal station is not supported.
JrnLineSpacing	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Journal station is not supported.
JrnLineWidth	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
JrnNearEnd	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
MapCharacterSet	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	Invalid parameter value was used.
MapMode	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	An attempt was made to perform an illegal operation with the Device.
PageModeArea	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
PageModeDescriptor	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
PageModeHorizontalPosition	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・ Invalid parameter value was used. ・ Valid station is not specified for **PageModeStation**.

Property	ErrorCode	Meaning
PageModePrintArea	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・ Invalid parameter value was used. ・ Valid station is not specified for **PageModeStation**.
PageModePrintDirection	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・ Invalid parameter value was used. ・ Valid station is not specified for **PageModeStation**.
PageModeStation	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	An attempt was made to perform an unsupported operation with the Device.
PageModeVerticalPosition	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・ Invalid parameter value was used. ・ Valid station is not specified for **PageModeStation**.
RecBarCodeRotationList	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
RecBitmapRotationList	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
RecCartridgeState	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
RecCurrentCartridge	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	An attempt was made to perform an unsupported operation with the Device.
RecEmpty	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
RecLetterQuality	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.

Property	ErrorCode	Meaning
RecLineChars	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Invalid parameter value was used.
RecLineCharsList	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
RecLineHeight	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Invalid parameter value was used.
RecLineSpacing	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Invalid parameter value was used.
RecLinesToPaperCut	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
RecLineWidth	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
RecNearEnd	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
RecSidewaysMaxChars	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.

Property	ErrorCode	Meaning
RecSidewaysMaxLines	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
RotateSpecial	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	An attempt was made to perform an illegal operation with the Device.
SlpBarCodeRotationList	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
SlpBitmapRotationList	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
SlpCartridgeState	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
SlpCurrentCartridge	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Slip station is not supported.
SlpEmpty	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
SlpLetterQuality	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Slip station is not supported.
SlpLineChars	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Slip station is not supported.
SlpLineCharsList	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.

Property	ErrorCode	Meaning
SlpLineHeight	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Slip station is not supported.
SlpLinesNearEndToEnd	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
SlpLineSpacing	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Slip station is not supported.
SlpLineWidth	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
SlpMaxLines	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
SlpNearEnd	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
SlpPrintSide	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.

Property	ErrorCode	Meaning
SlpSidewaysMaxChars	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
SlpSidewaysMaxLines	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.

A.1.3 Common Methods

Method	ErrorCode	Meaning
checkHealth	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	The *level* parameter is not JPOS_CH_INTERNAL(1), JPOS_CH_EXTERNAL(2), or JPOS_CH_INTERACTIVE(3).
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_BUSY(113)	Asynchronous output is in progress.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=201 The cover is open. ErrorCodeExtended=203 The receipt is out of paper. ErrorCodeExtended=1001 Vp voltage error has occurred. ErrorCodeExtended=1002 Autocutter error has occurred. ErrorCodeExtended=1005 Head-temperature error has occurred. ErrorCodeExtended=1010 A non-recoverable error has occurred.
claim	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	The *timeout* parameter is -2 or smaller.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_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.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=1010 A non-recoverable error has occurred.

Method	ErrorCode	Meaning
clearOutput	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=1010 A non-recoverable error has occurred.
close	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_BUSY(113)	Asynchronous output is in progress.
compareFirmwareVersion	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.
directIO	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	Invalid parameter value was used.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_BUSY(113)	Asynchronous output is in progress.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=1010 A non-recoverable error has occurred.
open	JPOS_E_NOSERVICE(104)	The Control cannot communicate with the Service Object.
	JPOS_E_ILLEGAL(106)	The Control is already open.
	JPOS_E_NOEXIST(109)	The specified *logicalDeviceName* is not found.
release	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	The application does not have exclusive access to the device.
	JPOS_E_SY(113)	Asynchronous output is in progress.

Method	ErrorCode	Meaning
resetStatistics	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	One of the specified statistics is not defined or resettable.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_BUSY(113)	Asynchronous output is in progress.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=280 At least one of the specified statistics could not be reset. ErrorCodeExtended=1010 A non-recoverable error has occurred.
retrieveStatistics	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	One of the specified statistics is not defined.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_BUSY(113)	Asynchronous output is in progress.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=280 At least one of the specified statistics could not be retrieved. ErrorCodeExtended=1010 A non-recoverable error has occurred.
updateFirmware	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.

Method	ErrorCode	Meaning
updateStatistics	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.

A.1.4 Specific Methods

Method	ErrorCode	Meaning
beginInsertion	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.
beginRemoval	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.
changePrintSide	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.
clearPrintArea	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・ Valid station is not specified for **PageModeStation**. ・ This method can be executed while in the page mode by **pageModePrint**.

Method	ErrorCode	Meaning
cutPaper	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・ The *percentage* parameter is less than 0 or more than 100. ・ This method is not supported while in the rotated 90°right/left mode by **rotatePrint**, or page mode by **pageModePrint**.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=201 The cover is open. ErrorCodeExtended=203 The receipt is out of paper. ErrorCodeExtended=1001 Vp voltage error has occurred. ErrorCodeExtended=1002 Autocutter error has occurred. ErrorCodeExtended=1005 Head-temperature error has occurred. ErrorCodeExtended=1010 A non-recoverable error has occurred.
drawRuledLine	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.
endInsertion	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.
endRemoval	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.
markFeed	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.

Method	ErrorCode	Meaning
pageModePrint	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・ The *control* parameter is not PTR_PM_PAGE_MODE(1), PTR_PM_PAGE_SAVE(2), PTR_PM_NORMAL(3), or PTR_PM_CANCEL(4). ・ The *control* parameter is specified PTR_PM_PAGE_SAVE(2), PTR_PM_NORMAL(3), or PTR_PM_CANCEL(4), before the *control* parameter is specified PTR_PM_PAGE_MODE(1). ・ Valid station is not specified for **pageModeStation**. ・ This method is not supported while in the rotated 90°right/left mode by **rotatePrint**.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=201 The cover is open. ErrorCodeExtended=203 The receipt is out of paper. ErrorCodeExtended=1001 Vp voltage error has occurred. ErrorCodeExtended=1002 Autocutter error has occurred. ErrorCodeExtended=1005 Head-temperature error has occurred. ErrorCodeExtended=1010 A non-recoverable error has occurred.

Method	ErrorCode	Meaning
printBarCode	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・The *station* parameter is not PTR_S_RECEIPT(2). ・Invalid value is specified for the *data* parameter. ・Invalid value is specified for the *symbology* parameter. ・The *height* parameter is less than 1 or more than 255. ・Invalid value is specified for the *width* parameter. ・Invalid value is specified for the *alignment* parameter. ・The *textPosition* parameter is not PTR_BC_TEXT_NONE(-11), PTR_BC_TEXT_ABOVE(-12), or PTR_BC_TEXT_BELOW(-13).
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=201 The cover is open. ErrorCodeExtended=203 The receipt is out of paper. ErrorCodeExtended=1001 Vp voltage error has occurred. ErrorCodeExtended=1002 Autocutter error has occurred. ErrorCodeExtended=1005 Head-temperature error has occurred. ErrorCodeExtended=1010 A non-recoverable error has occurred.

Method	ErrorCode	Meaning
printBitmap	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・The *station* parameter is not PTR_S_RECEIPT(2). ・Invalid value is specified for the *width* parameter. ・Invalid value is specified for the *alignment* parameter. ・The height of the specified bitmap is too big. ・The size of the specified bitmap is too big.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_NOEXIST(109)	The *fileName* parameter does not exist.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=201 The cover is open. ErrorCodeExtended=203 The receipt is out of paper. ErrorCodeExtended =206 The bitmap is either too wide to print without transformation, or it is too big to transform. ErrorCodeExtended =207 The specified file is either not a bitmap file, or it is in an unsupported format. ErrorCodeExtended=1001 Vp voltage error has occurred. ErrorCodeExtended=1002 Autocutter error has occurred. ErrorCodeExtended=1005 Head-temperature error has occurred. ErrorCodeExtended=1010 A non-recoverable error has occurred.

Method	ErrorCode	Meaning
printImmediate	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・The *station* parameter is not PTR_S_RECEIPT(2). ・Invalid value is specified for the *data* parameter.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_BUSY(113)	Asynchronous output is in progress.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=201 The cover is open. ErrorCodeExtended=203 The receipt is out of paper. ErrorCodeExtended=1001 Vp voltage error has occurred. ErrorCodeExtended=1002 Autocutter error has occurred. ErrorCodeExtended=1005 Head-temperature error has occurred. ErrorCodeExtended=1010 A non-recoverable error has occurred.

Method	ErrorCode	Meaning
printMemoryBitmap	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・The *station* parameter is not PTR_S_RECEIPT(2). ・Invalid value is specified for the *width* parameter. ・Invalid value is specified for the *alignment* parameter. ・The height of the specified bitmap is too big. ・The size of the specified bitmap is too big.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=201 The cover is open. ErrorCodeExtended=203 The receipt is out of paper. ErrorCodeExtended =206 The bitmap is either too wide to print without transformation, or it is too big to transform. ErrorCodeExtended =207 The specified file is either not a bitmap file, or it is in an unsupported format. ErrorCodeExtended=1001 Vp voltage error has occurred. ErrorCodeExtended=1002 Autocutter error has occurred. ErrorCodeExtended=1005 Head-temperature error has occurred. ErrorCodeExtended=1010 A non-recoverable error has occurred.

Method	ErrorCode	Meaning
printNormal	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・The *station* parameter is not PTR_S_RECEIPT(2). ・Invalid value is specified for the *data* parameter.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=201 The cover is open. ErrorCodeExtended=203 The receipt is out of paper. ErrorCodeExtended=1001 Vp voltage error has occurred. ErrorCodeExtended=1002 Autocutter error has occurred. ErrorCodeExtended=1005 Head-temperature error has occurred. ErrorCodeExtended=1010 A non-recoverable error has occurred.
printTwoNormal	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.

Method	ErrorCode	Meaning
rotatePrint	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・The *station* parameter is not PTR_S_RECEIPT(2). ・Invalid value is specified for the *rotation* parameter. ・This method is not supported while in the page mode by **pageModePrint**.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=201 The cover is open. ErrorCodeExtended=203 The receipt is out of paper. ErrorCodeExtended=1001 Vp voltage error has occurred. ErrorCodeExtended=1002 Autocutter error has occurred. ErrorCodeExtended=1005 Head-temperature error has occurred. ErrorCodeExtended=1010 A non-recoverable error has occurred.

Method	ErrorCode	Meaning
setBitmap	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・The *bitmapNumber* parameter is less than 1 or more than 20. ・The *station* parameter is not PTR_S_RECEIPT(2). ・Invalid value is specified for the *width* parameter. ・Invalid value is specified for the *alignment* parameter. ・The height of the specified bitmap is too big. ・The size of the specified bitmap is too big.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_NOEXIST(109)	The *fileName* parameter does not exist.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_BUSY(113)	Asynchronous output is in progress.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=201 The cover is open. ErrorCodeExtended=203 The receipt is out of paper. ErrorCodeExtended =206 The bitmap is either too wide to print without transformation, or it is too big to transform. ErrorCodeExtended =207 The specified file is either not a bitmap file, or it is in an unsupported format. ErrorCodeExtended=1001 Vp voltage error has occurred. ErrorCodeExtended=1002 Autocutter error has occurred. ErrorCodeExtended=1005 Head-temperature error has occurred. ErrorCodeExtended=1009 No memory is available to register the image to the user area of the printer. ErrorCodeExtended=1010 A non-recoverable error has occurred.
setLogo	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・The *location* parameter is not neither PTR_L_TOP(1), nor PTR_L_BOTTOM(2). ・The *data* parameter includes "ESC|tL" or "ESC|bL".

Method	ErrorCode	Meaning
transactionPrint	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・The *station* parameter is not PTR_S_RECEIPT(2). ・The *control* parameter is not neither PTR_TP_TRANSACTION(11), nor PTR_TP_NORMAL(12). ・This method is not supported while in the rotated 90°right/left mode by **rotatePrint**, or page mode by **pageModePrint**.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=201 The cover is open. ErrorCodeExtended=203 The receipt is out of paper. ErrorCodeExtended=1001 Vp voltage error has occurred. ErrorCodeExtended=1002 Autocutter error has occurred. ErrorCodeExtended=1005 Head-temperature error has occurred. ErrorCodeExtended=1010 A non-recoverable error has occurred.
validateData	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	See "A.1.5(1)" for detail.
	JPOS_E_FAILURE(111)	See "A.1.5(2)" for detail.

A.1.5 Escape Sequence

(1) JPOS_E_ILLEGAL(106) is thrown for **validateData** in the following cases:

Escape Sequence	Meaning
Paper cut	It is in one of the following states. ・Percentage '#' is not precisely supported. ・This escape sequence is not supported while in the rotated 90°right/left mode by **RotatePrint**, or page mode by **pageModePrint**.
Feed and Paper cut	It is in one of the following states. ・Percentage '#' is not precisely supported. ・This escape sequence is not supported while in the rotated 90°right/left mode by **rotatePrint**, or page mode by **pageModePrint**.
Feed lines	It is in one of the following states. ・The line number '#' is not correct. ・This escape sequence is not supported while in the rotated 90°right/left mode by **rotatePrint**, or page mode by **pageModePrint**.
Feed units	It is in one of the following states. ・Feed unit number '#' is not precisely supported due to occurrence of rounding error of one dot depending on the setting of **MapMode**. ・The feed unit number '#' is not correct. ・This escape sequence is not supported while in the rotated 90°right/left mode by **rotatePrint**, or page mode by **pageModePrint**.
Pass through embedded data	The number of bytes of embedded data '#' is not correct.
Print in-line barcode	It is in one of the following states. ・The number of bytes of barcode data '#' is not precisely supported. ・The character string following ESC|#R is not correct.
Underline	The thickness '#' is not correct.
Scale vertically	The scale factor '#' is not correct.
Scale horizontally	The scale factor '#' is not correct.
Left justify	This escape sequence is not supported while in the rotated 90°right/left mode by **rotatePrint**, or page mode by **pageModePrint**.
Center	This escape sequence is not supported while in the rotated 90°right/left mode by **rotatePrint**, or page mode by **pageModePrint**.
Right justify	This escape sequence is not supported while in the rotated 90°right/left mode by **rotatePrint**, or page mode by **pageModePrint**.

(2) JPOS_E_ FAILURE(111) is thrown for **validateData** in the following cases:

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

A.2 CashDrawer

A.2.1 Common Properties

Property	ErrorCode	Meaning
CapCompareFirmwareVersion	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapPowerReporting	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapStatisticsReporting	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapUpdateFirmware	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapUpdateStatistics	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CheckHealthText	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
Claimed	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
DeviceEnabled	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_CLAIMED(102)	Attempt was made to access a device that is exclusively
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=1010 A non-recoverable error has occurred.
FreezeEvents	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
PowerNotify	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	It is in one of the following states. ・ The setting range of property is improper. ・ **DeviceEnabled** is set to true.
PowerState	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
DeviceServiceDescription	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
DeviceServiceVersion	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
PhysicalDeviceDescription	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
PhysicalDeviceName	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.

A.2.2 Specific Properties

Property	ErrorCode	Meaning
CapStatus	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
CapStatusMultiDrawerDetect	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
DrawerOpened	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.

A.2.3 Common Methods

Method	ErrorCode	Meaning
checkHealth	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_CLAIMED(102)	Attempt was made to access a device that is exclusively
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	The *level* parameter is not JPOS_CH_INTERNAL(1), JPOS_CH_EXTERNAL(2), or JPOS_CH_INTERACTIVE(3).
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=1010 A non-recoverable error has occurred.
claim	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	The *timeout* parameter is less than -2.
	JPOS_E_NOHARDWARE(107)	The device is not connected to the system or is not powered on.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_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.
	JPOS_E_EXTENDED(114)	ErrorCodeExtended=1010 A non-recoverable error has occurred.
close	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
compareFirmwareVersion	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.
directIO	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_ILLEGAL(106)	This method is not supported.
open	JPOS_E_NOSERVICE(104)	The Control cannot communicate with the Service Object.
	JPOS_E_ILLEGAL(106)	The Control is already open.
	JPOS_E_NOEXIST(109)	The specified *logicalDeviceName* is not found.

Method	ErrorCode	Meaning
release	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_ILLEGAL(106)	The application does not have exclusive access to the device.
resetStatistics	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.
retrieveStatistics	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.
updateFirmware	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.
updateStatistics	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_NOTCLAIMED(103)	Exclusive access is not available. Call **claim** to gain exclusive access.
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
	JPOS_E_ILLEGAL(106)	This method is not supported.

A.2.4 Specific Methods

Method	ErrorCode	Meaning
openDrawer	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_CLAIMED(102)	Attempt was made to access a device that is exclusively
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.
waitForDrawerClose	JPOS_E_CLOSED(101)	Attempt was made to access a closed device.
	JPOS_E_CLAIMED(102)	Attempt was made to access a device that is exclusively
	JPOS_E_DISABLED(105)	Cannot execute operation while device is disabled.

Appendix B Statistics

B.1 Statistics Defined by UnifiedPOS

XML Definition Name	Response	Resettable
UnifiedPOSVersion	1.13	-
DeviceCategory	"POSPrinter"	-
ManufacturerName	"Seiko Instruments Inc."	-
ModelName	"RP-F10"*1	-
	"RP-E10"*2	
	"RP-D10"*3	
SerialNumber	"Unknown"	-
ManufactureDate	"Unknown"	-
MechanicalRevision	"25"*1	-
	"1A"*2	
	"1C"*3	
FirmwareRevision	Firmware version	-
Interface	"Unknown"	-
InstallationDate	"Unknown"	-
HoursPoweredCount	Drive time of printer unit (unit: hours)	✓
CommunicationErrorCount	0	-
BarcodePrintedCount	0	-
FormInsertionCount	0	-
HomeErrorCount	0	-
JournalCharacterPrintedCount	0	-
JournalLinePrintedCount	0	-
MaximumTempReachedCount	0	-
NVRAMWriteCount	0	-
PaperCutCount	Number of autocutter drive	✓
FailedPaperCutCount	0	-
PrinterFaultCount	0	-
PrintSideChangeCount	0	-
FailedPrintSideChangeCount	0	-
ReceiptCharacterPrintedCount	0	-

XML Definition Name	Response	Resettable
ReceiptLinePrintedCount	0	-
ReceiptLineFeedCount	Number of line feed (unit: 100 dot-lines)	✓
ReceiptCoverOpenCount	0	-
SlipCharacterPrintedCount	0	-
SlipLinePrintedCount	0	-
SlipLineFeedCount	0	-
SlipCoverOpenCount	0	-
StampFiredCount	0	-

*1: In case of RP-F10
*2: In case of RP-E10
*3: In case of RP-D10

B.2 Statistics Defined by Manufacturer

XML Definition Name	Response	Resettable
HoursPoweredCount_Accumulated	Drive time of printer unit (unit: hours) (accumulated)	-
PaperCutCount_Accumulated	Number of autocutter drive (accumulated)	-
ReceiptLineFeedCount_Accumulated	Number of line feed (unit: 100 dot-lines) (accumulated)	-