



SII RP JavaPOS Driver
Application Programmer's Guide

U00135794202

Seiko Instruments Inc.

U00135794200	December 2015
U00135794201	March 2022
U00135794202	July 2026

Copyright© 2015-2026 by Seiko Instruments Inc. All rights reserved.

Java is a registered trademark of Oracle and/or its affiliates.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

SUSE is a registered trademark of SUSE LLC in the United States and other countries.

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

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

SII  is a trademark of Seiko Instruments Inc.

Introduction

This document describes the specifications, functions, and operating instructions of the SII RP JavaPOS Driver (hereinafter referred to as "the software") provided by Seiko Instruments Inc. (hereinafter referred to as "SII") for the RP series POS printers (hereinafter referred to as "printer") and cash drawers that are connected to the printers.

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

- Operating instructions with a mouse and a keyboard

Operating System Abbreviations

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

- Linux® => Linux

Terms

Definition	Description
UPOS V1.13	"UnifiedPOS Retail Peripheral Architecture Version 1.13 APPENDIX B Java for Retail POS "
POSPrinter	JavaPOS Driver of the POS Printer device class. It controls the printer device.
CashDrawer	JavaPOS Driver of the Cash Drawer device class. It controls the drawer device.
Printer	It means the printer product (physical device of printer).
Drawer	It means the drawer product which is connected to the printer (physical device of drawer).
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	In the software, this means the height of each single-high print line (total value of the printed line height and the whitespace between each pair of lines).

Table of Contents

1	Overview	1-1
1.1	Configuration	1-1
1.1.1	Structural Diagram.....	1-1
1.2	Limitation	1-2
1.2.1	POSPrinter	1-2
1.2.2	CashDrawer	1-2
2	Operating Environment	2-1
2.1	System Requirements	2-1
2.2	Supported Models	2-1
3	Installation	3-1
3.1	Installation	3-1
3.1.1	Installation Steps	3-1
3.2	Uninstallation	3-2
3.2.1	Uninstallation Steps	3-2
3.3	Configuration Program	3-3
3.3.1	Item and Detail of the Configuration Program.....	3-3
3.3.2	Add JposEntry	3-11
3.3.3	Modify JposEntry	3-20
3.3.4	Delete JposEntry	3-21
3.3.5	Log Setting	3-22

4.1	Summary	4-1
4.1.1	Common Properties.....	4-1
4.1.2	Specific Properties.....	4-2
4.1.3	Common Methods	4-9
4.1.4	Specific Methods	4-9
4.1.5	Events.....	4-10
4.2	Data Characters and Escape Sequences	4-11
4.2.1	Escape Sequence Operated When Specified.....	4-11
4.2.2	Escape Sequence Operated During Printing.....	4-13
4.2.3	Escape Sequence Operated When Printing	4-14
4.3	Details	4-16
4.3.1	Common Properties.....	4-16
	CapCompareFirmwareVersion Property.....	4-16
	CapPowerReporting Property	4-16
	CapStatisticsReporting Property	4-16
	CapUpdateFirmware Property	4-17
	CapUpdateStatistics Property	4-17
	CheckHealthText Property.....	4-17
	Claimed Property.....	4-18
	DeviceEnabled Property R/W	4-18
	FreezeEvents Property R/W	4-19
	OutputID Property.....	4-19
	PowerNotify Property R/W	4-20
	PowerState Property.....	4-20
	State Property	4-21
	DeviceControlDescription Property	4-21
	DeviceControlVersion Property	4-21
	DeviceServiceDescription Property.....	4-21
	DeviceServiceVersion Property.....	4-22
	PhysicalDeviceDescription Property.....	4-22
	PhysicalDeviceName Property.....	4-22
4.3.2	Specific Properties.....	4-23
	AsyncMode Property R/W.....	4-23
	CapCharacterSet Property.....	4-23
	CapCoverSensor Property.....	4-23
	CapMapCharacterSet Property	4-24
	CapRec2Color Property.....	4-24
	CapRecBarCode Property	4-24
	CapRecBitmap Property	4-25
	CapRecBold Property	4-25
	CapRecCartridgeSensor Property.....	4-25
	CapRecColor Property.....	4-25
	CapRecDhigh Property	4-26
	CapRecDwide Property	4-26

CapRecDwideDhigh Property	4-26
CapRecEmptySensor Property	4-26
CapRecItalic Property	4-27
CapRecLeft90 Property	4-27
CapRecMarkFeed Property	4-27
CapRecNearEndSensor Property	4-27
CapRecPageMode Property	4-28
CapRecPaperCut Property	4-28
CapRecPresent Property	4-28
CapRecRight90 Property	4-29
CapRecRotate180 Property	4-29
CapRecRuledLine Property	4-29
CapRecStamp Property	4-29
CapRecUnderline Property	4-30
CapTransaction Property	4-30
CartridgeNotify Property R/W	4-30
CharacterSet Property R/W	4-31
CharacterSetList Property	4-31
CoverOpen Property	4-32
ErrorLevel Property	4-32
ErrorStation Property	4-32
ErrorString Property	4-33
FlagWhenIdle Property R/W	4-34
FontTypefaceList Property	4-34
MapCharacterSet Property R/W	4-34
MapMode Property R/W	4-35
PageModeArea Property	4-36
PageModeDescriptor Property	4-36
PageModeHorizontalPosition Property R/W	4-37
PageModePrintArea Property R/W	4-38
PageModePrintDirection Property R/W	4-39
PageModeStation Property R/W	4-41
PageModeVerticalPosition Property R/W	4-41
RecBarCodeRotationList Property	4-42
RecBitmapRotationList Property	4-42
RecCartridgeState Property	4-42
RecCurrentCartridge Property R/W	4-43
RecEmpty Property	4-43
RecLetterQuality Property R/W	4-43
RecLineChars Property R/W	4-44
RecLineCharsList Property	4-45
RecLineHeight Property R/W	4-45
RecLineSpacing Property R/W	4-46
RecLinesToPaperCut Property	4-47
RecLineWidth Property	4-47
RecNearEnd Property	4-48
RecSidewaysMaxChars Property	4-48

	RecSidewaysMaxLines Property.....	4-49
	RotateSpecial Property R/W	4-49
4.3.3	Common Methods	4-50
	checkHealth Method	4-50
	claim Method.....	4-50
	clearOutput Method	4-50
	close Method.....	4-51
	compareFirmwareVersion Method	4-51
	directIO Method.....	4-51
	open Method	4-53
	release Method.....	4-53
	resetStatistics Method	4-53
	retrieveStatistics Method.....	4-54
	updateFirmware Method	4-54
	updateStatistics Method	4-54
4.3.4	Specific Methods	4-55
	beginInsertion Method	4-55
	beginRemoval Method.....	4-55
	changePrintSide Method.....	4-55
	clearPrintArea Method.....	4-55
	cutPaper Method	4-56
	drawRuledLine Method.....	4-57
	endInsertion Method.....	4-57
	endRemoval Method.....	4-57
	markFeed Method	4-57
	pageModePrint Method	4-58
	printBarCode Method.....	4-60
	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.....	4-77
4.3.5	Events	4-78
	DirectIOEvent Event	4-78
	ErrorEvent Event	4-78
	OutputCompleteEvent Event.....	4-79
	StatusUpdateEvent Event.....	4-79

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.....	5-11
	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	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-34

Appendix B Statistics B-1

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

1 Overview

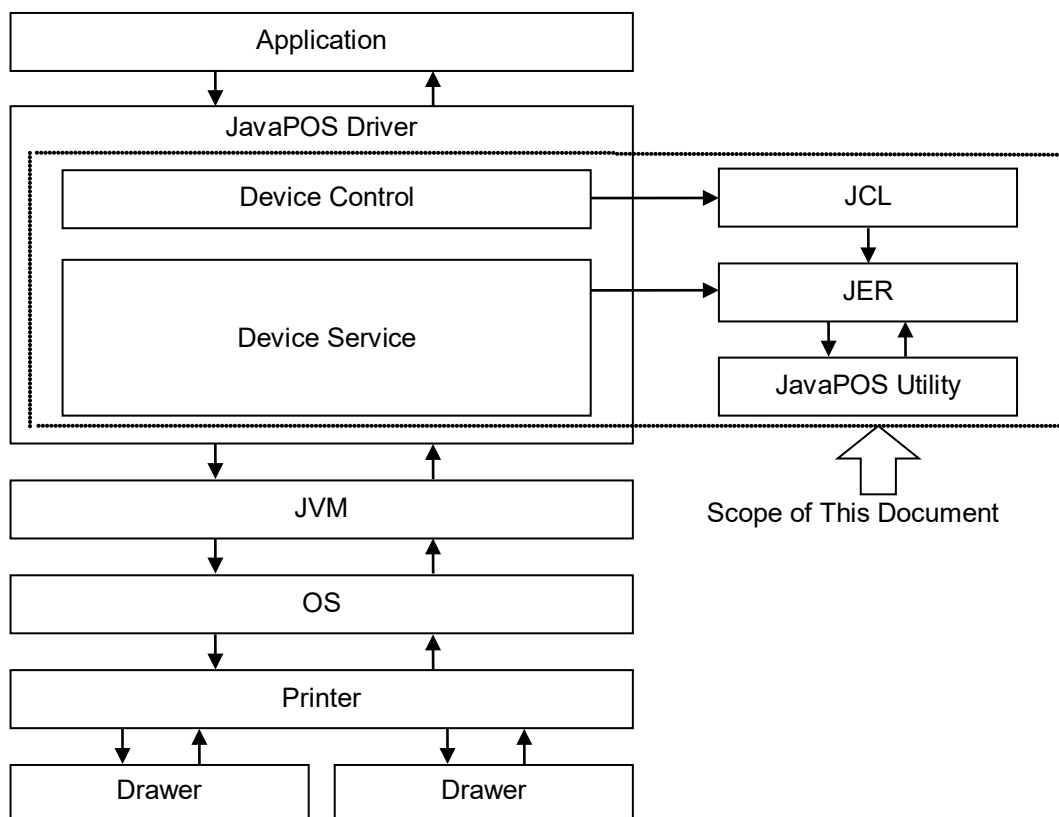
This software is JavaPOS driver which controls the printer and its connected cash drawer which conforms to the JavaPOS Version 1.13.

Refer to the "Appendix B Java for Retail POS" in UPOS V1.13 for using the JavaPOS Device Control.

1.1 Configuration

1.1.1 Structural Diagram

Structure of the software and the scope of this document are as follows.



1.2 Limitation

1.2.1 POSPrinter

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

- (a) The method and property settings related to journal and slip print are not supported.
- (b) Italic, Custom Color, Shading and cartridge function are not supported.
- (c) All the following methods always return JPOS_E_ILLEGAL(106) after they are enabled.
 - **beginInsertion** Method
 - **beginRemoval** Method
 - **changePrintSide** Method
 - **drawRuledLine** Method
 - **endInsertion** Method
 - **endRemoval** Method
 - **markFeed** Method
 - **printTwoNormal** Method
- (d) The **DirectIOEvent** event (device-specific event) is not supported.
- (e) DO NOT disconnect the interface cable during printing or transmission. After the interface cable is connected correctly, the printer may not print correctly.

1.2.2 CashDrawer

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

- (a) All the following methods always return JPOS_E_ILLEGAL(106) after they are opened.
 - **directIO** Method
- (b) The **DirectIOEvent** event (device-specific event) is not supported.

2 Operating Environment

2.1 System Requirements

Item	Specifications
Operating System	SUSE Linux Enterprise 11 (32 bits and 64 bits)* ¹
Java Operation Environment	Java Standard Edition 6 Update 45 or later* ²

*1: Only x86/x64 environments are supported. Linux ARM is not supported.

*2: 64-bit Java is supported on Java Standard Edition 8 or later.

2.2 Supported Models

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

3 Installation

3.1 Installation

Prior to install this software, be sure to install the appropriate Java execution environment (JRE 6 update46 or later) for the operating system to use.

The installation steps for the operating system are shown in the following.

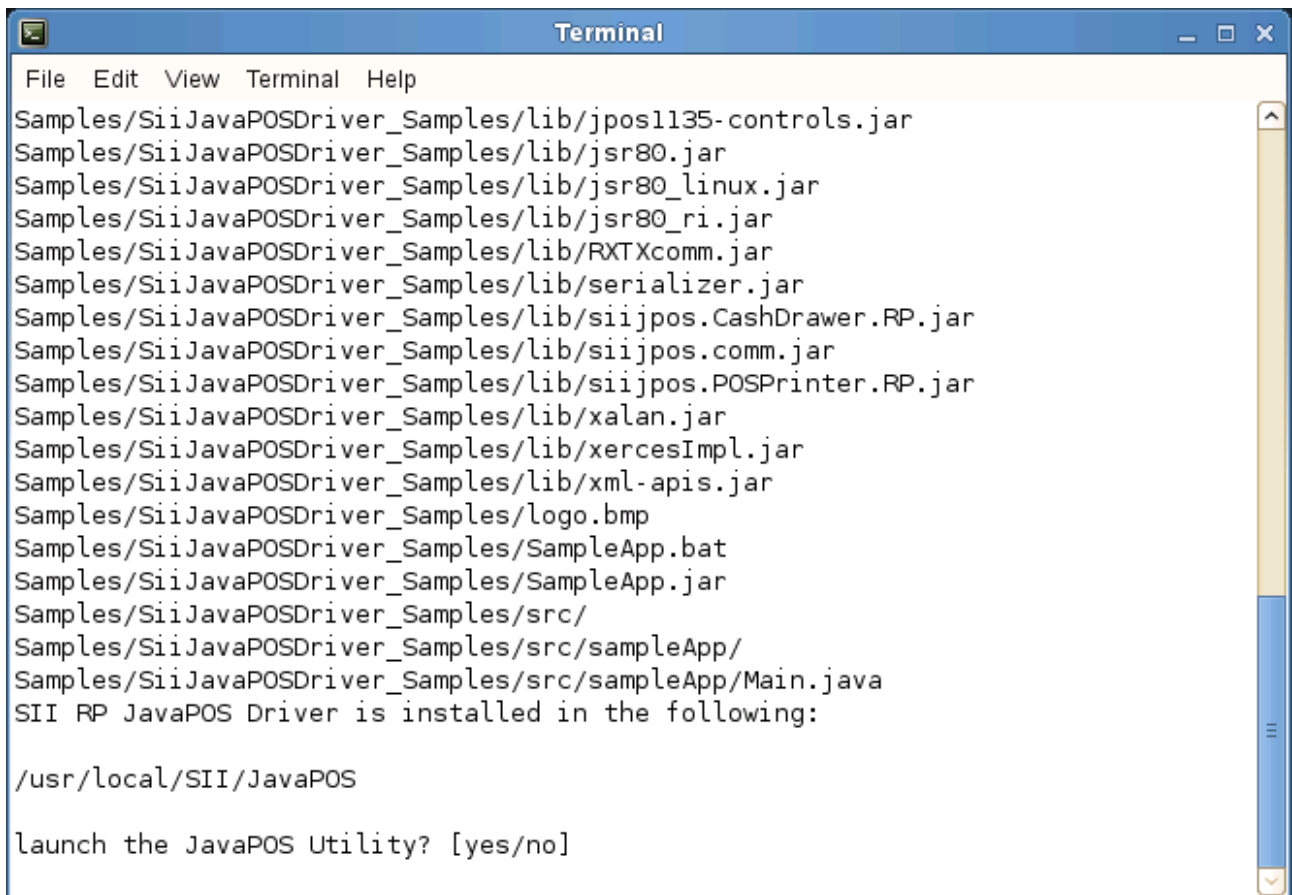
The installer needs to be executed by administrator authority.

3.1.1 Installation Steps

Open the terminal window, go to the directory where the installer exists, and type `"/SII_RP_JavaPOS_1013xxx.bin"`.

When install is succeeded, the installed location is displayed.

By typing "yes", JavaPOS Utility is started.



```
Terminal
File Edit View Terminal Help
Samples/SiiJavaPOSDriver_Samples/lib/jpos1135-controls.jar
Samples/SiiJavaPOSDriver_Samples/lib/jsr80.jar
Samples/SiiJavaPOSDriver_Samples/lib/jsr80_linux.jar
Samples/SiiJavaPOSDriver_Samples/lib/jsr80_rti.jar
Samples/SiiJavaPOSDriver_Samples/lib/RXTXcomm.jar
Samples/SiiJavaPOSDriver_Samples/lib/serializer.jar
Samples/SiiJavaPOSDriver_Samples/lib/siijpos.CashDrawer.RP.jar
Samples/SiiJavaPOSDriver_Samples/lib/siijpos.comm.jar
Samples/SiiJavaPOSDriver_Samples/lib/siijpos.POSPrinter.RP.jar
Samples/SiiJavaPOSDriver_Samples/lib/xalan.jar
Samples/SiiJavaPOSDriver_Samples/lib/xercesImpl.jar
Samples/SiiJavaPOSDriver_Samples/lib/xml-apis.jar
Samples/SiiJavaPOSDriver_Samples/logo.bmp
Samples/SiiJavaPOSDriver_Samples/SampleApp.bat
Samples/SiiJavaPOSDriver_Samples/SampleApp.jar
Samples/SiiJavaPOSDriver_Samples/src/
Samples/SiiJavaPOSDriver_Samples/src/sampleApp/
Samples/SiiJavaPOSDriver_Samples/src/sampleApp/Main.java
SII RP JavaPOS Driver is installed in the following:

/usr/local/SII/JavaPOS

launch the JavaPOS Utility? [yes/no]
```

3.2 Uninstallation

3.2.1 Uninstallation Steps

Delete /usr/local/SII/JavaPOS directory.

3.3 Configuration Program

This section describes the configuration program (SII JavaPOS Utility) provided by the software.

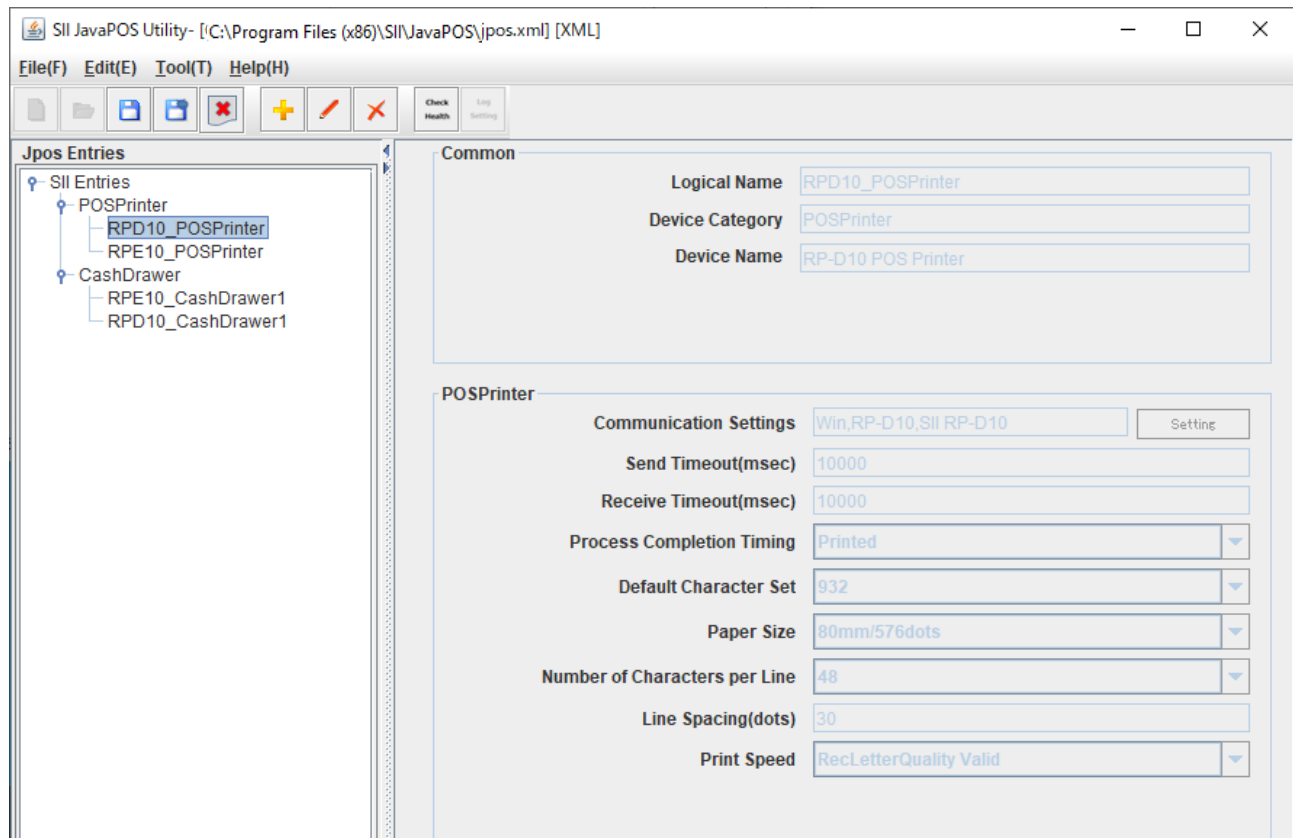
This program provides following functions:

- Add JposEntry to JavaPOS Entry Registry.
- Modify JposEntry in JavaPOS Entry Registry.
- Delete JposEntry from JavaPOS Entry Registry.

3.3.1 Item and Detail of the Configuration Program

(1) POSPrinter

1) User Interface



2) Item and Detail

(a) Logical Name

Set the logical name.

An arbitrary name can be specified by using 1-byte alphanumeric characters.

It is impossible to set the same logical name for more than 1 POS device.

(b) Device Category

Select the type of the device.

To add the POSPrinter device, select [POSPrinter].

This item can be selected only when it is newly added. It is not modifiable.

(c) Device Name

Select the device name to use.

One of the following values can be selected.

This item can be selected only when it is newly added. It is not modifiable.

Dialogue Value Selection	Detail
RP-D10 POS Printer	Use RP-D10.
RP-E10 POS Printer	Use RP-E10.

(d) Communication Settings

Set the communication protocol.

The setting differs depending on the using interface.

Environment	Detail
Linux - USB	Available to specify iSerialNumber. When iSerialNumber is not used, specify "0". When "0" is specified, 1 printer connected to the system can be worked.
Linux - Serial	Available to set the port configuration.
Linux - Ethernet	Available to set IP address.

(e) Send Timeout (msec)

Set the send timeout period (msec).

Available setting range is 3000 to 60000 (msec).

(f) Receive Timeout (msec)

Set the receive timeout period (msec).

Available setting range is 3000 to 60000 (msec).

(g) Process Completion Timing

Select the completion timing of printing method.
One of the following values can be selected.

Dialogue Value Selection	Detail	JposEntry
Data transmitted	When the POS device sends data to printer, the method is completed.	ProcessCompletionTiming="0"
Printed	When the printer prints data, the method is completed.	ProcessCompletionTiming="1"

(h) Character Set

Select the default of **CharacterSet** property.
One of the following values can be selected.

Dialogue Value Selection	Detail	JposEntry
437	Sets Code Page437.	DefaultCharacterSet="437"
850	Sets Code Page850.	DefaultCharacterSet="850"
852	Sets Code Page852.	DefaultCharacterSet="852"
858	Sets Code Page858.	DefaultCharacterSet="858"
860	Sets Code Page860.	DefaultCharacterSet="860"
863	Sets Code Page863.	DefaultCharacterSet="863"
932	Sets Code Page932.	DefaultCharacterSet="932"
999	Sets the Windows ANSI character set (Code Page 1252).	DefaultCharacterSet="999"
1250	Sets Code Page1250.	DefaultCharacterSet="1250"
1251	Sets Code Page1251.	DefaultCharacterSet="1251"
1252	Sets Code Page1252.	DefaultCharacterSet="1252"
1253	Sets Code Page1253.	DefaultCharacterSet="1253"
1254	Sets Code Page1254.	DefaultCharacterSet="1254"

(i) Paper Size

Select the paper width and number of printable dots per 1 line.
One of the following values can be selected.

Dialogue Value Selection	Detail	JposEntry
58mm/360dots	Use the printer with 58 mm paper width and 360 dots of printable dots.	PaperSize="0"
58mm/432dots	Use the printer with 58 mm paper width and 432 dots of printable dots.	PaperSize="1"
80mm/512dots	Use the printer with 80 mm paper width and 512 dots of printable dots.	PaperSize="2"
80mm/576dots	Use the printer with 80 mm paper width and 576 dots of printable dots.	PaperSize="3"

(j) Number of Characters per Line

Select the number of 1-byte character to print in 1 line.

One of the following values can be selected.

The value range differs depending on the paper size.

- When the paper size is 58mm/360dots:

Dialogue Value Selection	Detail	JposEntry
30	Set to 30.	RecLineChars ="30"
40	Set to 40.	RecLineChars ="40"

- When the paper size is 58mm/432dots:

Dialogue Value Selection	Detail	JposEntry
27	Set to 27.	RecLineChars ="27"
30	Set to 30.	RecLineChars ="30"
33	Set to 33.	RecLineChars ="33"
36	Set to 36.	RecLineChars ="36"
43	Set to 43.	RecLineChars ="43"
48	Set to 48.	RecLineChars ="48"
54	Set to 54.	RecLineChars ="54"

- When the paper size is 80mm/512dots:

Dialogue Value Selection	Detail	JposEntry
42	Set to 42.	RecLineChars ="42"
56	Set to 56.	RecLineChars ="56"

- When the paper size is 80mm/576dots:

Dialogue Value Selection	Detail	JposEntry
36	Set to 36.	RecLineChars ="36"
41	Set to 41.	RecLineChars ="41"
44	Set to 44.	RecLineChars ="44"
48	Set to 48.	RecLineChars ="48"
57	Set to 57.	RecLineChars ="57"
64	Set to 64.	RecLineChars ="64"
72	Set to 72.	RecLineChars ="72"

(k) Line Spacing(dots)

Set the line spacing.

Available setting range is 0 to 255.

RecLineSpacing in JposEntry is updated to the set value by this item.

(l) NearEnd Sensor

Select enable/disable for near end sensor of the printer.

One of the following values can be selected.

Only RP-E10 is available to set this item.

Dialogue Value Selection	Detail	JposEntry
Disable	Near end sensor is set to Disable.	NearEndSensor="0"
Enable	Near end sensor is set to Enable.	NearEndSensor="1"

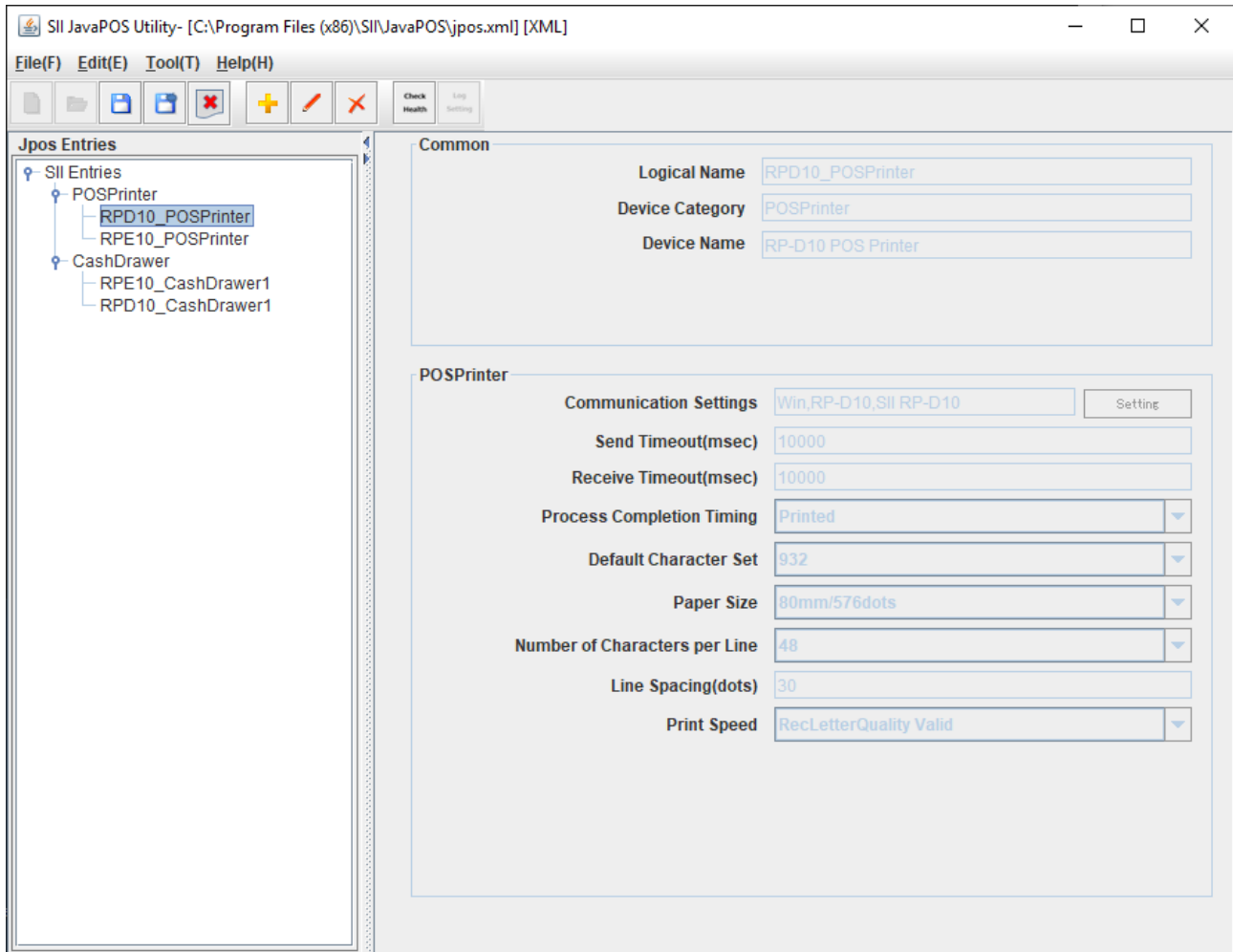
(m) Print Speed

Select the highest print speed.

One of the following values can be selected.

Dialogue Value Selection	Detail	JposEntry
RecLetterQuality Valid	The print speed depends on the RecLetterQuality property.	PrintSpeed="0"
High	The print speed is fixed to High regardless of RecLetterQuality property.	PrintSpeed="1"
Middle(Quality)	The print speed is fixed to High regardless of RecLetterQuality property.	PrintSpeed="2"
Middle(Silent)	The print speed is fixed to High regardless of RecLetterQuality property.	PrintSpeed="3"
Low	The print speed is fixed to High regardless of RecLetterQuality property.	PrintSpeed="4"

- (2) CashDrawer
- 1) User Interface



2) Item and Detail

(a) Logical Name

Set the logical name.

An arbitrary name can be specified by using 1-byte alphanumeric characters.

The same logical name cannot be specified to more than 1 device.

(b) Device Category

Select the type of the device.

To add the CashDrawer device, select [CashDrawer].

This item can be selected only when it is newly added. It is not modifiable.

(c) Device Name

Select the device name to use.

One of the following values can be selected.

This item can be selected only when it is newly added. It is not modifiable.

Dialogue Value Selection	Detail
RP-D10 Cash Drawer	Use the drawer that is connected to RP-D10.
RP-E10 Cash Drawer	Use the drawer that is connected to RP-E10.

(d) POSPrinter Logical Name

Select the logical name of the printer which connects drawer.

POSPrinter devices registered in JavaPOS Entry Registry can be selected.

This item can be selected only when it is newly added. It is not modifiable.

(e) Drawer Number

Select the drawer to open when **OpenDrawer** method is executed.

One of the following values can be selected.

This item can be selected only when it is newly added. It is not modifiable.

Dialogue Value Selection	Detail	JposEntry
1	Opens the drawer connected to drive circuit 1.	DrawerNo ="1"
2	Opens the drawer connected to drive circuit 2.	DrawerNo ="2"

(f) Sensor status when drawer is open

Select the drawer sensor status when the drawer is open.

One of the following values can be selected.

Dialogue Value Selection	Detail	JposEntry
Low	The drawer is being opened when the drawer sensor is Low.	DrawerOpenStatus ="Low"
High	The drawer is being opened when the drawer sensor is High.	DrawerOpenStatus ="High"

(g) Pulse Time (msec)

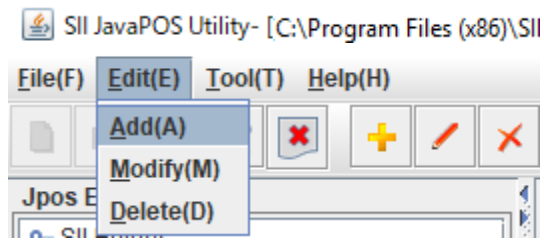
Select the time of the pulse to open the drawer.
One of the following values can be selected.

Dialogue Value Selection	Detail	JposEntry
100	Send 100 msec pulse to drawer.	PulseTime="100"
200	Send 200 msec pulse to drawer.	PulseTime="200"
300	Send 300 msec pulse to drawer.	PulseTime="300"
400	Send 400 msec pulse to drawer.	PulseTime="400"
500	Send 500 msec pulse to drawer.	PulseTime="500"
600	Send 600 msec pulse to drawer.	PulseTime="600"
700	Send 700 msec pulse to drawer.	PulseTime="700"
800	Send 800 msec pulse to drawer.	PulseTime="800"

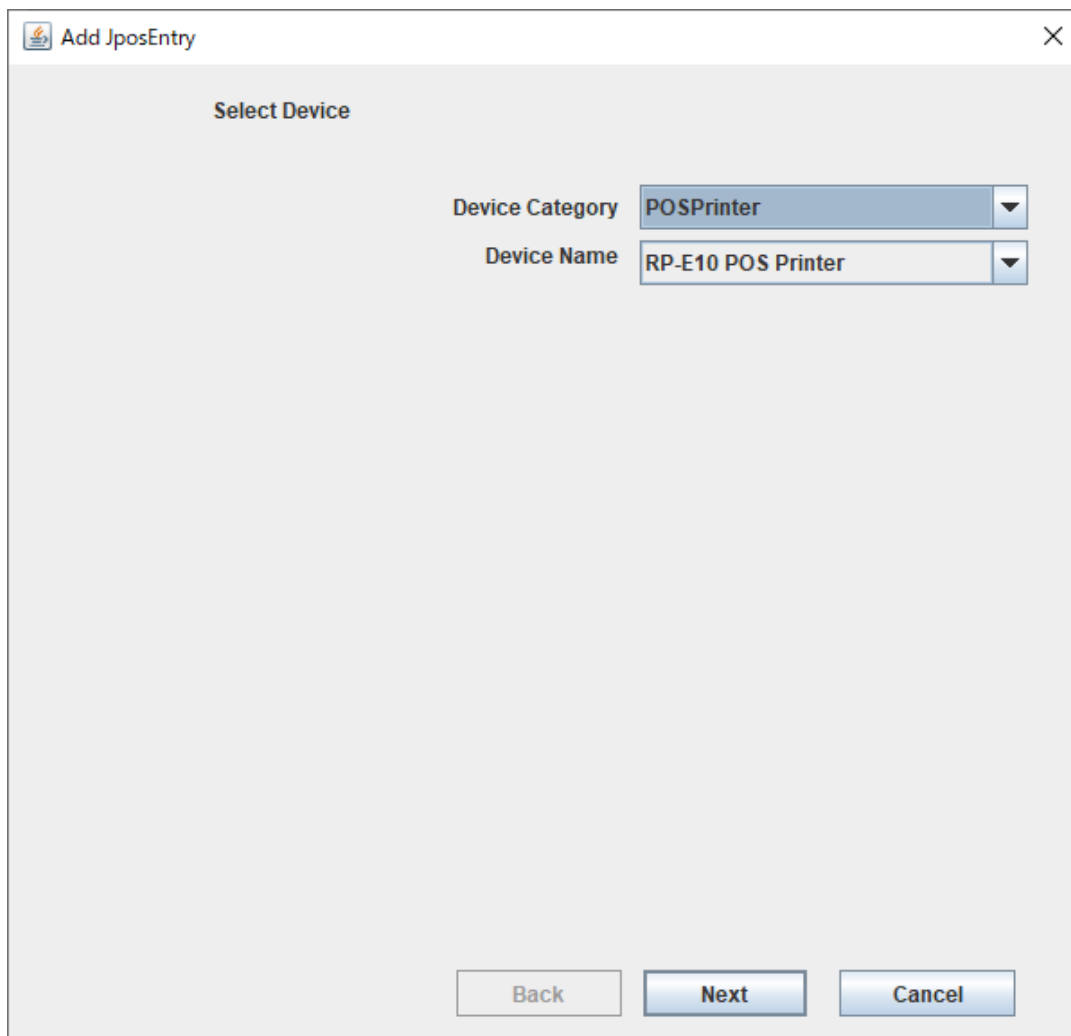
3.3.2 Add JposEntry

(1) Add POSPrinter device

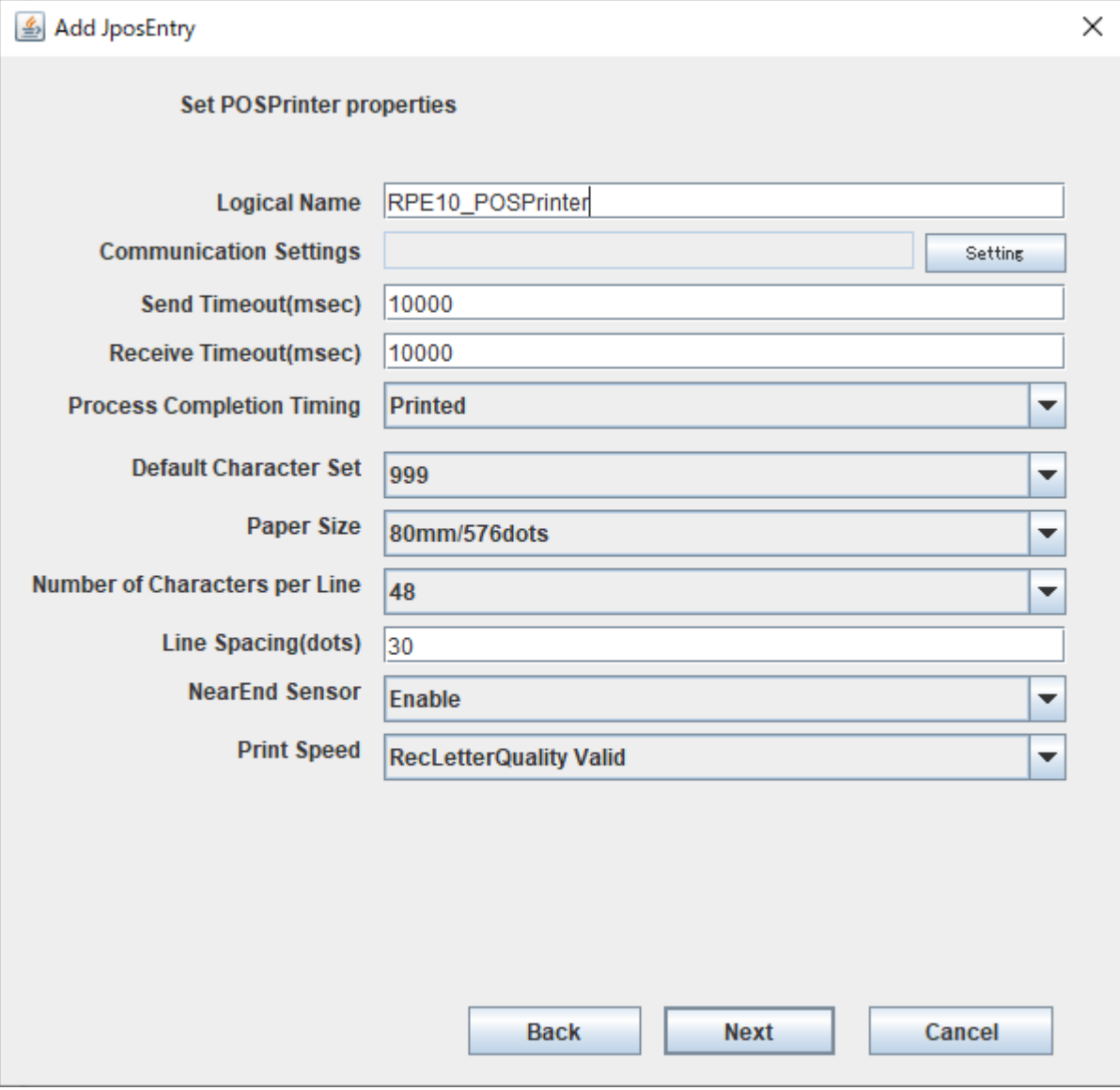
1. Click [Add(A)] in the menu opened by clicking [Edit(E)] menu.



2. [Add JposEntry] wizard is activated.
Select [POSPrinter] for [Device Category].
When using RP-D10, select [RP-D10 POS Printer] for [Device Name] and click [Next].
When using RP-E10, select [RP-E10 POS Printer] for [Device Name] and click [Next].



3. The advanced setting for POSPrinter window is displayed.
Set each item, and click [Next].
For details about the items, please refer to "3.3.1 Item and Detail of the Configuration Program".



The screenshot shows a window titled "Add JposEntry" with a close button (X) in the top right corner. The main area is titled "Set POSPrinter properties". It contains several configuration fields:

- Logical Name:** A text box containing "RPE10_POSPrinter".
- Communication Settings:** A button labeled "Setting".
- Send Timeout(msec):** A text box containing "10000".
- Receive Timeout(msec):** A text box containing "10000".
- Process Completion Timing:** A dropdown menu showing "Printed".
- Default Character Set:** A dropdown menu showing "999".
- Paper Size:** A dropdown menu showing "80mm/576dots".
- Number of Characters per Line:** A dropdown menu showing "48".
- Line Spacing(dots):** A text box containing "30".
- NearEnd Sensor:** A dropdown menu showing "Enable".
- Print Speed:** A dropdown menu showing "RecLetterQuality Valid".

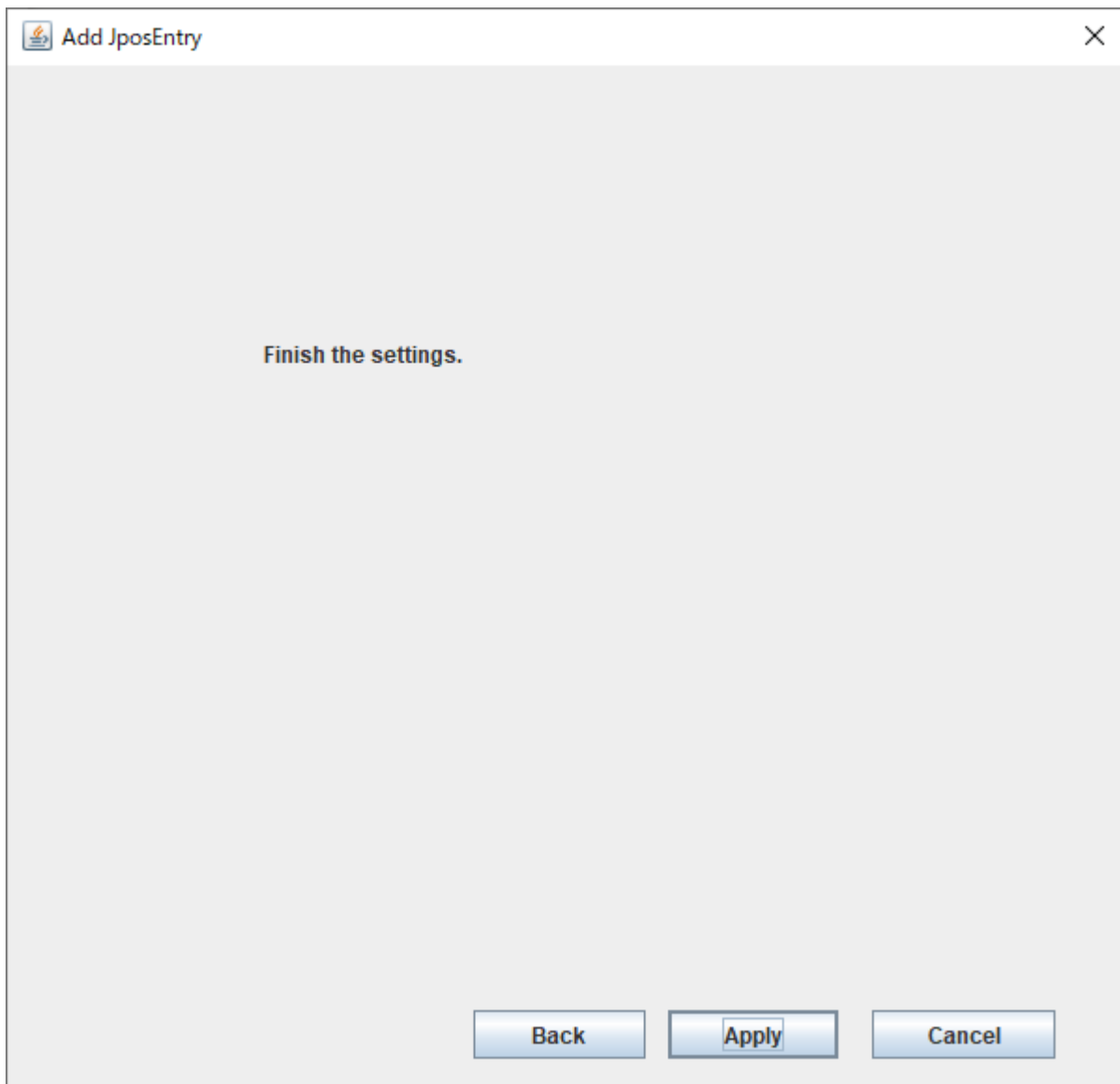
At the bottom of the window, there are three buttons: "Back", "Next", and "Cancel".

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

The screenshot shows a dialog box titled "Add JposEntry" with a close button (X) in the top right corner. The main content area has a heading "Select CashDrawer number". Below this heading is a label "Number of CashDrawer devices" followed by a dropdown menu. The dropdown menu is open, showing three options: "0", "1", and "2". The "0" option is currently selected. At the bottom of the dialog box, there are three buttons: "Back", "Next", and "Cancel".

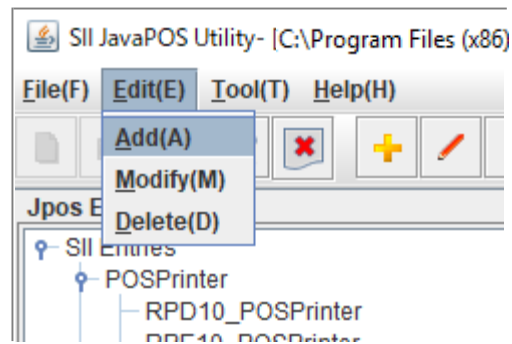
5. Finish adding the device.

Click [Apply] to complete the addition of the device and the wizard.



(2) Add CashDrawer device

1. Click [Add(A)] in the menu opened by clicking [Edit(E)] menu.

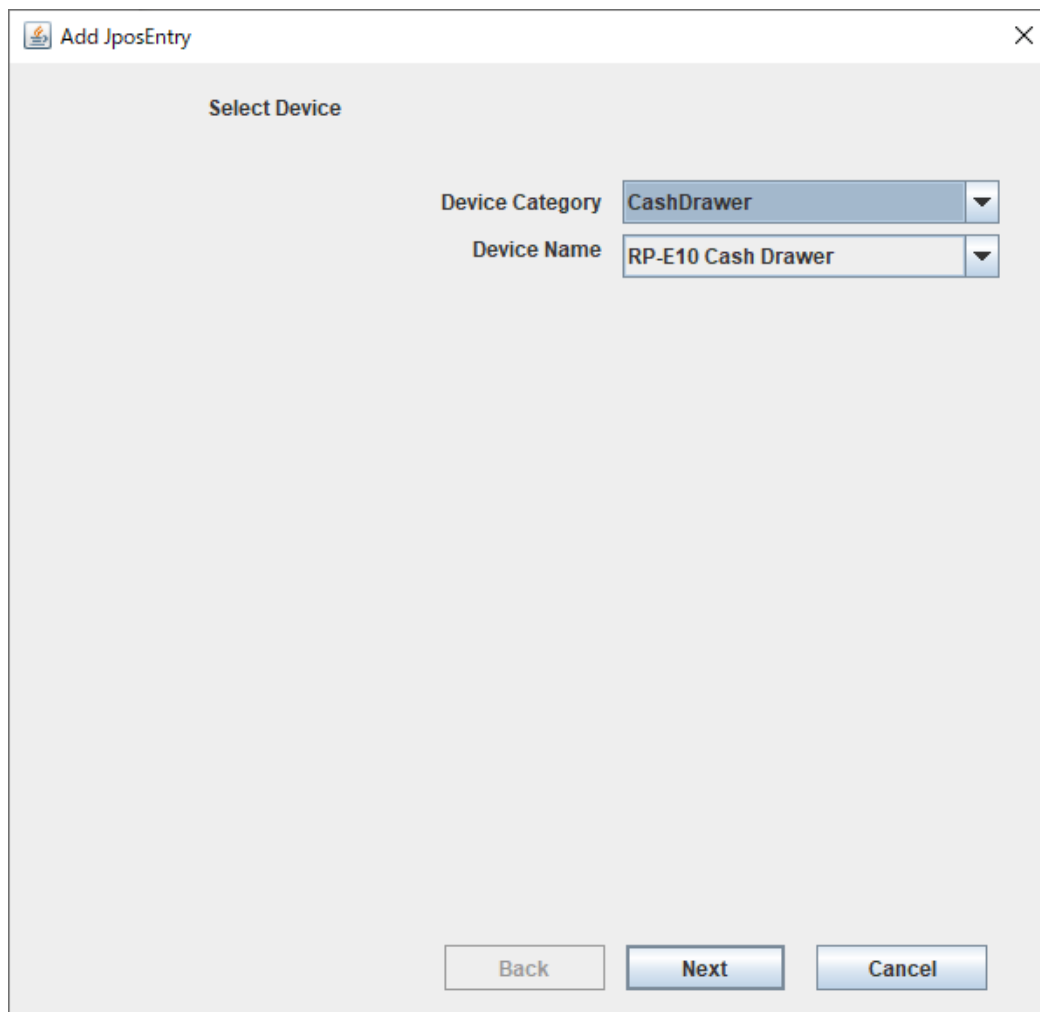


2. [Add JposEntry] wizard is displayed.

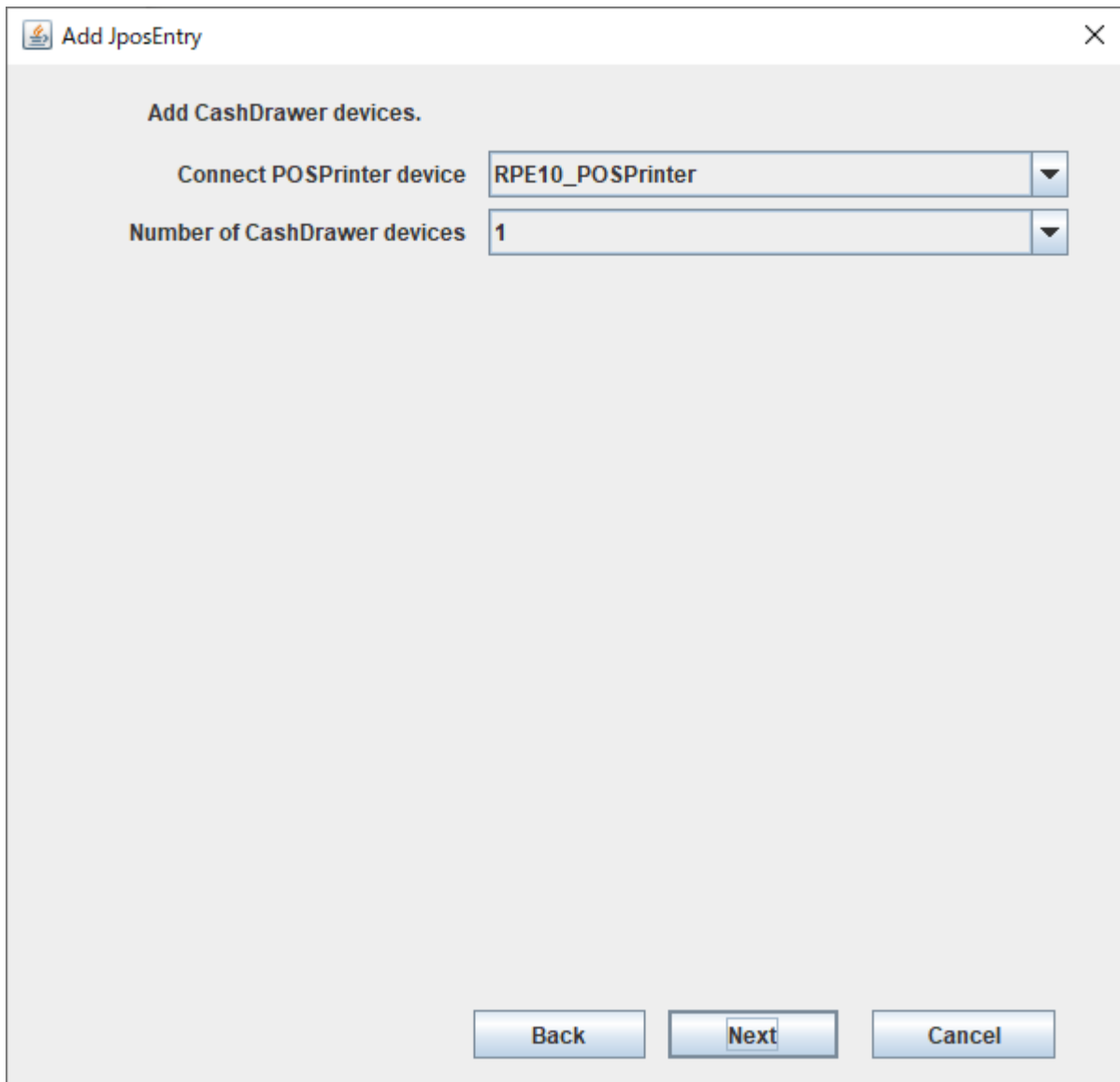
Select [CashDrawer] for [Device Category].

When using the drawer that is connected RP-D10, Select [RP-D10 Cash Drawer] for [Device Name] and click [Next].

When using the drawer that is connected RP-E10, Select [RP-E10 Cash Drawer] for [Device Name] and click [Next].



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 [Next].
Refer to 4 for subsequent settings.
When adding 2 CashDrawer:
select [2] in [Number of CashDrawer devices], and click [Next].
Refer to 5 for subsequent settings.



Add JposEntry

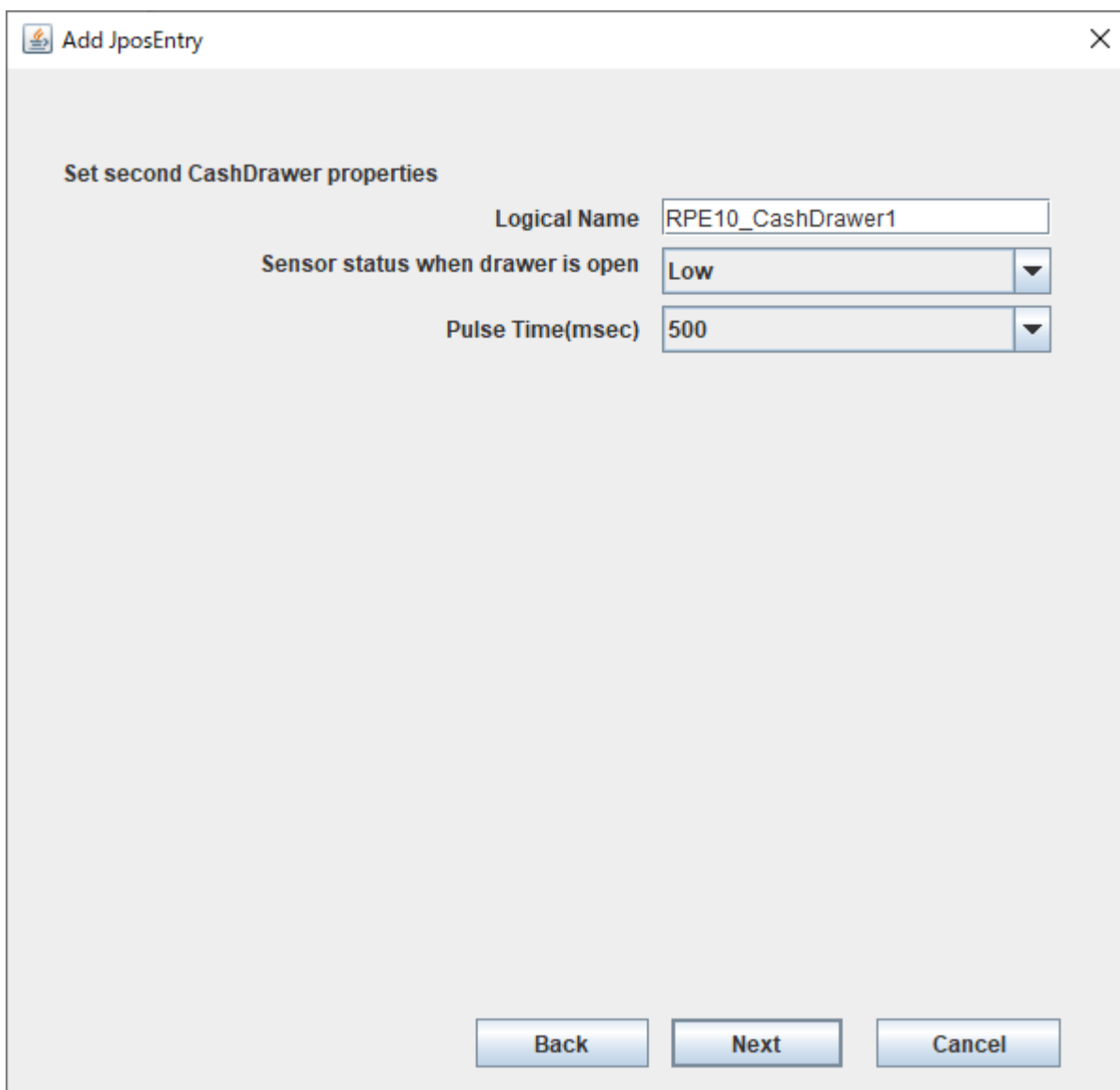
Add CashDrawer devices.

Connect POSPrinter device **RPE10_POSPrinter**

Number of CashDrawer devices **1**

Back **Next** **Cancel**

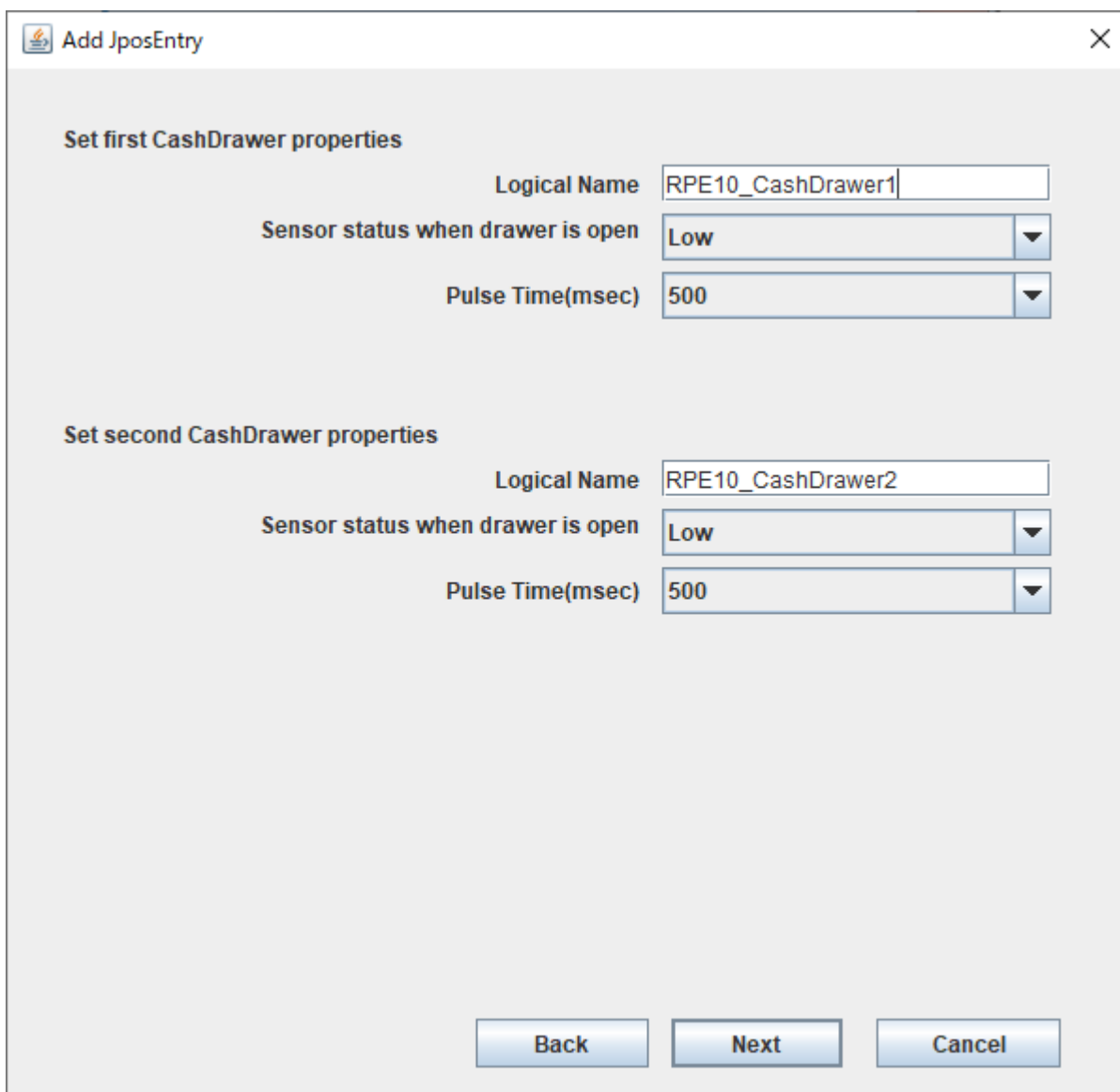
4. The advanced setting window for driving 1 drawer is displayed.
Set each item, and click [Next].
For details about the items, please refer to "3.3.1 Item and Detail of the Configuration Program".



The screenshot shows a window titled "Add JposEntry" with a close button (X) in the top right corner. The window contains the following elements:

- Set second CashDrawer properties**: A section header.
- Logical Name**: A text input field containing "RPE10_CashDrawer1".
- Sensor status when drawer is open**: A dropdown menu with "Low" selected.
- Pulse Time(msec)**: A dropdown menu with "500" selected.
- Buttons**: Three buttons at the bottom: "Back", "Next", and "Cancel".

5. The advanced setting window for driving 2 drawers is displayed.
Set each item, and select [Next].
For details about the items, please refer to "3.3.1 Item and Detail of the Configuration Program".



The screenshot shows a window titled "Add JposEntry" with a close button (X) in the top right corner. The window is divided into two sections for configuring cash drawers.

Set first CashDrawer properties

- Logical Name:** RPE10_CashDrawer1
- Sensor status when drawer is open:** Low (selected from a dropdown menu)
- Pulse Time(msec):** 500 (selected from a dropdown menu)

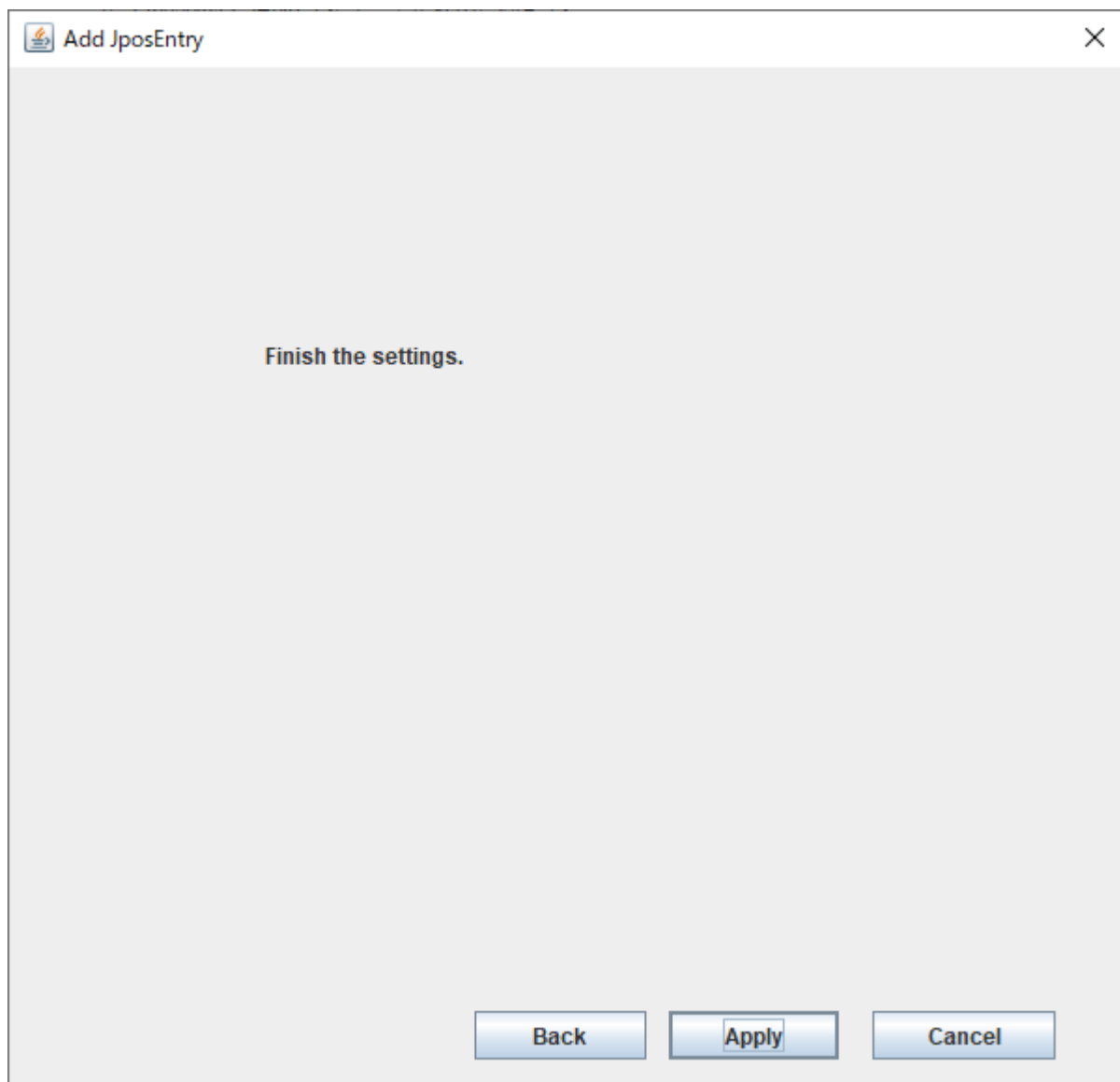
Set second CashDrawer properties

- Logical Name:** RPE10_CashDrawer2
- Sensor status when drawer is open:** Low (selected from a dropdown menu)
- Pulse Time(msec):** 500 (selected from a dropdown menu)

At the bottom of the window, there are three buttons: **Back**, **Next**, and **Cancel**.

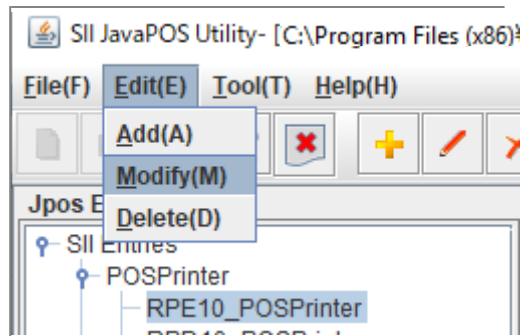
6. Finish adding the device.

Click [Apply] to complete the addition of the device and the wizard.



3.3.3 Modify JposEntry

1. While selecting the device to edit, click [Modify(M)] in the menu opened by clicking [Edit(E)] menu.

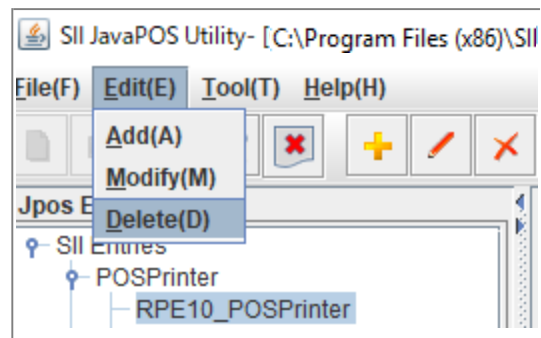


2. Modifying the device setting is available. After modifying the device, select [Modify(M)] again to update the setting change.

A screenshot of the 'SII JavaPOS Utility' window showing the configuration for the 'RPE10_POSPrinter' device. The window title is 'SII JavaPOS Utility- [C:\Program Files (x86)\SII\JavaPOS\jpos.xml] [XML]'. The menu bar includes 'File(F)', 'Edit(E)', 'Tool(T)', and 'Help(H)'. The toolbar has icons for file operations, a 'Check Health' button, and a 'Log Setting' button. The main area is divided into two sections: 'Common' and 'POSPrinter'.
Common Section:
- Logical Name: RPE10_POSPrinter
- Device Category: POSPrinter
- Device Name: RP-E10 POS Printer
POSPrinter Section:
- Communication Settings: Win,RP-E10,SII RP-E10 (with a 'Setting' button)
- Send Timeout(msec): 10000
- Receive Timeout(msec): 10000
- Process Completion Timing: Printed (dropdown)
- Default Character Set: 932 (dropdown)
- Paper Size: 80mm/576dots (dropdown)
- Number of Characters per Line: 48 (dropdown)
- Line Spacing(dots): 30
- NearEnd Sensor: Enable (dropdown)
- Print Speed: RecLetterQuality Valid (dropdown)

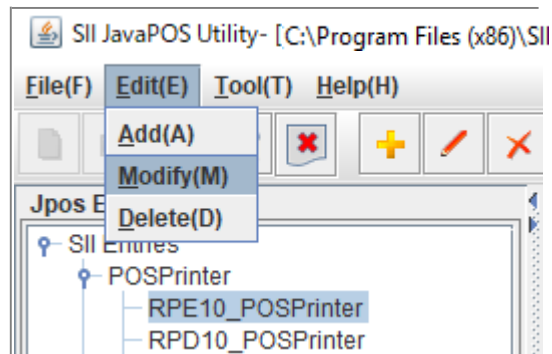
3.3.4 Delete JposEntry

While selecting the device to delete, click [Delete(D)] in the menu opened by clicking [Edit(E)] menu.

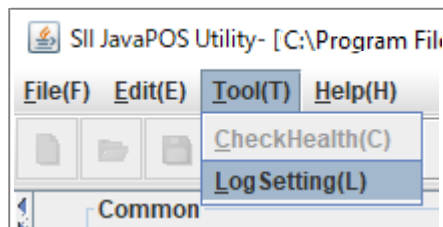


3.3.5 Log Setting

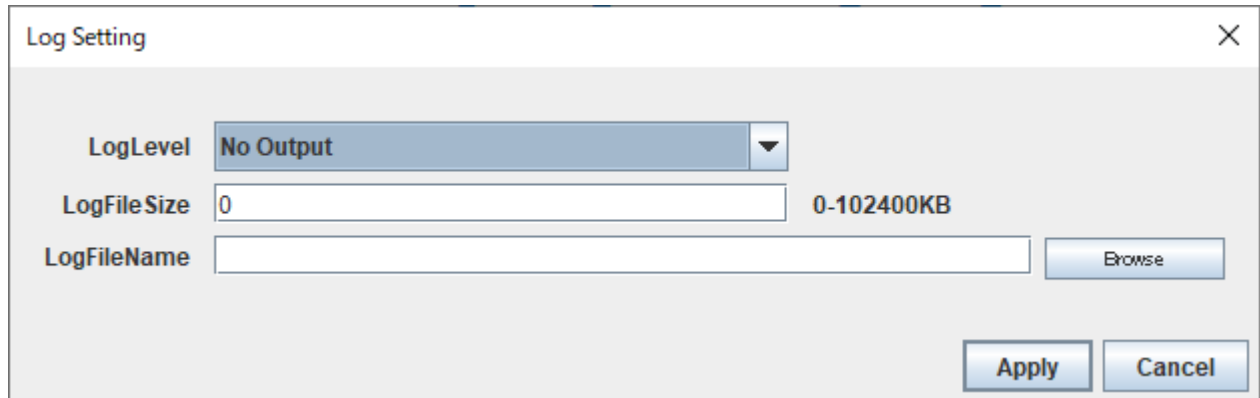
1. While selecting the device to change log setting, click [Modify(M)] in the menu opened by clicking [Edit(E)] menu.



2. Select [LogSetting(L)] in the menu opened by clicking [Tool(T)] menu.



3. [Log Setting Wizard] is displayed.
For details about the setting items, refer to the following table.
Click [Apply] to complete the log setting and the wizard.
Select [Modify(M)] again to update the setting change.



(a) LogLevel

Select the log output level.
One of the following values can be selected.

Dialogue Value Selection	Detail
No output	Log is not output.
Error	Error level log is output.
Trace	Trace level log is output.

(b) LogFileSize

Set the maximum log file size (KB).
Available setting range is 0 to 102400.

(c) LogFileName

Set the full path file name to store the log file.

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 Series JavaPOS POSPrinter Service Driver, Copyright (C) 20xx Seiko Instruments Inc."
DeviceServiceVersion	int	R	open	"1013000"
PhysicalDeviceDescription	String	R	open	"SII RP-D10 POS Printer"*1
				"SII RP-E10 POS Printer"*2
PhysicalDeviceName	String	R	open	"RP-D10 POS Printer"*1
				"RP-E10 POS Printer"*2

*1: In case of RP-D10.

*2: In case of RP-E10.

4.1.2 Specific Properties

Property Name	Type	Access	Availability Condition	Initial Value
AsyncMode	boolean	R/W	open	false
CapCharacterSet	int	R	open	PTR_CCS_KANJI(11)
CapConcurrentJrnRec	boolean	R	open	false
CapConcurrentJrnSlp	boolean	R	open	false
CapConcurrentPageMode	boolean	R	open	false
CapConcurrentRecSlp	boolean	R	open	false
CapCoverSensor	boolean	R	open	true
CapJrn2Color	boolean	R	open	false
CapJrnBold	boolean	R	open	false
CapJrnCartridgeSensor	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
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}
				Depends on JavaPOS Entry Registry ^{*2}
CapRecPageMode	boolean	R	open	true
CapRecPapercut	boolean	R	open	true
CapRecPresent	boolean	R	open	true

Property Name	Type	Access	Availability Condition	Initial Value
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
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
CapTransaction	boolean	R	open	true
CartridgeNotify	int	R/W	open	PTR_CN_DISABLED(0)
CharacterSet	int	R/W	open, claim, & enable	Depends on JavaPOS Entry Registry
CharacterSetList	String	R	open	"437,850,852,858,860,863,865,932,999,1250,1251,1252,1253,1254"
CoverOpen	boolean	R	open, claim, & enable	Depends on printer status
ErrorLevel	int	R	open	PTR_EL_NONE(1)
ErrorStation	int	R	open	0
ErrorString	String	R	open	""
FlagWhenIdle	boolean	R/W	open	false
FontTypefaceList	String	R	open	""
JrnCartridgeState	int	R	open, claim, & enable	PTR_CART_UNKNOWN (0x10000000)

Property Name	Type	Access	Availability Condition	Initial Value
JrnCurrentCartridge	int	R/W	open, claim, & enable	0
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
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	Depends on JavaPOS Entry Registry
RecLineCharsList	String	R	open	Depends on JavaPOS Entry Registry
RecLineHeight	int	R/W	open, claim, & enable	Depends on JavaPOS Entry Registry
RecLineSpacing	int	R/W	open, claim, & enable	Depends on JavaPOS Entry Registry
RecLinesToPaperCut	int	R	open, claim, & enable	Depends on JavaPOS Entry Registry

Property Name	Type	Access	Availability Condition	Initial Value
RecLineWidth	int	R	open, claim, & enable	Depends on JavaPOS Entry Registry
RecNearEnd	boolean	R	open, claim, & enable	false ^{*1}
				Depends on printer status ^{*2}
RecSidewaysMaxChars	int	R	open, claim, & enable	Depends on JavaPOS Entry Registry
RecSidewaysMaxLines	int	R	open, claim, & enable	Depends on JavaPOS Entry Registry
RotateSpecial	int	R/W	open	PTR_RP_NORMAL(1)
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

*1: In case of RP-D10.

*2: In case of RP-E10.

The Properties which depends on JavaPOS Entry Registry.

- CapRecNearEndSensor

This property is initialized by following value.

CapRecNearEndSensor	JavaPOS Entry Registry
true	NearEndSensor="1"
false	NearEndSensor="0"

- CharacterSet

This property is initialized to the value specified by DefaultCharacterSet property.

- RecLineChars, RecLineHeight, RecLineCharsList

This property is initialized by following value.

RecLineChars	RecLineHeight	RecLineCharsList	JavaPOS Entry Registry			
30	24	"30,40"	PaperSize="0"	RecLineChars="30"		
40	16			RecLineChars="40"		
27	24	"27,30,33,36,43,48,54"	PaperSize="1"	RecLineChars="27"		
30				RecLineChars="30"		
33				RecLineChars="33"		
36				RecLineChars="36"		
43	16			RecLineChars="43"		
48				RecLineChars="48"		
54				RecLineChars="54"		
42	24			"42,56"	PaperSize="2"	RecLineChars="42"
56	16					RecLineChars="56"
36	24	"36,41,44,48,57,64,72"	PaperSize="3"	RecLineChars="36"		
41				RecLineChars="41"		
44				RecLineChars="44"		
48				RecLineChars="48"		
57	16			RecLineChars="57"		
64				RecLineChars="64"		
72				RecLineChars="72"		

- RecLineSpacing

This property is initialized to the value specified by **RecLineSpacing** property.

When the specified value in **RecLineSpacing** property is smaller than the one in **RecLineHeight** property, it is automatically set to the value in **RecLineHeight** property.

- **RecLinesToPaperCut**

Depending on the printer or **RecLineSpacing** property, this property is initialized to the value calculated by the following formula (round up digits after the decimal point).

Printer	RecLinesToPaperCut
RP-D10	$\text{RecLinesToPaperCut} = 88 / \text{RecLineSpacing}$
RP-E10	$\text{RecLinesToPaperCut} = 100 / \text{RecLineSpacing}$

- **RecLineWidth**

This property is initialized by the following value.

RecLineWidth	JavaPOS Entry Registry
360	PaperSize="0"
432	PaperSize="1"
512	PaperSize="2"
576	PaperSize="3"

- **RecSidewaysMaxChars**

Depending on the character space determined by **RecLineChars** property or **RecLineHeight** property, this property is initialized to the value calculated by the following formula (round up digits after the decimal point).

$$\text{RecSidewaysMaxChars} = 2400 / (\text{RecLineHeight} / 2 + \text{Character space})$$

- **RecSidewaysMaxLines**

Depending on **RecLineWidth** or **RecLineSpacing**, his property is initialized to the value calculated by the following formula (round up digits after the decimal point).

$$\text{RecSidewaysMaxLines} = ((\text{RecLineWidth} - \text{RecLineHeight}) / \text{RecLineSpacing}) + 1$$

The following specific properties for POSPrinter are not supported.

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

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, a partial cut is executed. If '#' is omitted, a full cut is executed. If '#' is 0, it is ignored. This is ignored during rotated 90° right/left mode by rotatePrint method or during page mode by pageModePrint method.
Feed and Paper cut	ESC [#]fP	Cuts receipt paper, after feeding the paper by the RecLinesToPaperCut lines. The placeholder '#' is defined by "Paper cut" escape sequence. This is ignored during rotated 90° right/left mode by rotatePrint method or during page mode by pageModePrint method.
Feed, Paper cut, and Stamp	ESC [#]sP	Not supported.
Print bitmap	ESC #B	Prints the pre-stored bitmap. The placeholder '#' is replaced by the bitmap number. If the character '#' is omitted, the data is regarded as print data instead of an escape sequence. A value from 1 to 20 can be specified for '#'. If values other than 1 to 20 are specified for '#', they are ignored.
Print top logo	ESC tL	Prints the pre-stored top logo.
Print bottom logo	ESC bL	Prints the pre-stored bottom logo.
Fire stamp	ESC sL	Not supported.
Feed lines*	ESC [#]lF	Feeds the paper forward by lines. The placeholder '#' is replaced by an ASCII decimal string indicating the number of lines to be fed. A value from 0 to 255 can be specified for '#'. If '#' exceeds this range, the maximum supported number of 255 lines are fed. If '#' is omitted, then 1 line is fed. A value from 0 to 255 can be specified for '#'. If '#' exceeds this range, the maximum supported number (255) of lines is fed. This is ignored during rotated 90° right/left mode by rotatePrint method or during page mode by pageModePrint method.
Feed units*	ESC [#]uF	Feeds the paper forward by units in MapMode property. 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 property is set to PTR_MM_DOTS(1), '#' is available from 1 to 255. If '#' exceeds this range, the maximum supported number (255) of units is fed. This is ignored during rotated 90° right/left mode by rotatePrint method or during page mode by pageModePrint method.
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. If '#' is omitted, the data is regarded as print data instead of an escape sequence. A value from 1 to 65535 can be specified for '#'. If '#' exceeds this range, transmission of embedded data is not executed. If the print data 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.) Also, during rotated 90° right/left mode by rotatePrint method, the width cannot be calculated exactly because data string specified by transmission of embedded data is not counted as character string. Therefore, make an appropriate adjustment by inserting blanks.
Print in-line barcode (See 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. Also, during rotated 90° right/left mode by the rotatePrint method, 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.

*: The line spacing is reduced according to the setting value of the paper saving in the printer. Therefore, the following printing processes are changed.

- The amount of feed specified by the "Feed lines" escape sequence (ESC|[#]lF) is smaller than the setting value of the **RecLineSpacing** property.
- The amount of feed specified by the "Feed units" escape sequence (ESC|[#]uF) is processed based on the setting value of the paper saving (When Mode1 is selected, paper is not fed at all).

However, the line spacing from last print line to receipt cut position is not reduced because paper is cut after executing the paper feed for saved dot lines.

Cut operation which executes the paper feed for saved dot lines is executed for "Paper cut" escape sequence (ESC|[#]P), "Feed and Paper cut" escape sequence (ESC|[#]fP), and the **cutPaper** method.

- **In-Line Barcode Printing**

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

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

The following characters are used to indicate the attributes.

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

Attributes must be written in the above order.

Every attribute is mandatory.

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

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 #T	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 omitted, then a thickness of 1 is used for the underline. If '#' is 3 or larger, then a thickness of 2 is used. If '!' is specified, the underline mode is cleared.
Italic	ESC [!] iC	Not supported.
Alternate color (Custom)	ESC [#] rC	Not supported.
Reverse video	ESC [!] rvC	Prints in a reverse video format. If '!' is specified, reverse video is disabled.
Shading	ESC [#] sC	Not supported.
Single high and wide	ESC 1C	Prints normal size.
Double wide	ESC 2C	Prints double-wide characters.
Double high	ESC 3C	Prints double-high characters.
Double high and wide	ESC 4C	Prints double-high/double-wide characters.
Scale horizontally	ESC # hC	A supported value for the placeholder '#' is 1 to 8. If '#' is omitted, the data is regarded as print data instead of an escape sequence. If values less than 1 are specified for '#', print in 1 scale. If values greater than 8 are specified for '#', print in 8 scale.
Scale vertically	ESC # vC	A supported value for the placeholder '#' is 1 to 8. If '#' is omitted, the data is regarded as print data instead of an escape sequence. If values less than 1 are specified for '#', print in 1 scale. If values greater than 8 are specified for '#', print in 8 scale.
RGB Color	ESC [#] fC	Not supported.
Center	ESC cA	Aligns following text in the center. This must be specified at the head of the line. If not, this is invalid. Also, if there is a linefeed on the print data, the center is valid after linefeed. This specification is ignored during rotated 90° right/left mode by rotatePrint method or during page mode by pageModePrint method.
Right justify	ESC rA	Aligns the subsequent texts to the right. This must be specified at the head of the line. If not, this is invalid. Also, if there is a line feed on the print data, the right justify is valid after linefeed. This specification is ignored during rotated 90° right/left mode by rotatePrint method or during page mode by pageModePrint method.
Left justify	ESC lA	Aligns the subsequent texts to the left. This must be specified at the head of the line. If not, this is invalid. Also, if there is a line feed on the print data, the left justify valid after line feed. This specification is ignored during rotated 90° right/left mode by rotatePrint method or during page mode by pageModePrint method.
Normal	ESC N	Restores printer characteristics to normal condition.
SubScript	ESC [!] tbC	Not supported.

Name	Data	Remarks
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	Gets a Boolean value that 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 the **open** method.

CapPowerReporting Property

Type	int
Remarks	Gets the power 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 the **open** method.

CapStatisticsReporting Property

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

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

CapUpdateFirmware Property

Type **boolean**

Remarks Gets a Boolean value that indicates whether a 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 the **open** method.

CapUpdateStatistics Property

Type **boolean**

Remarks Gets a Boolean value that indicates whether some or all the device statistics can be reset to 0 by using the **resetStatistics** methods.
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 the resetStatistics method.

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

CheckHealthText Property

Type **String**

Remarks Holds the results of the most recent call to the **checkHealth** method.
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 the **open** method.

Claimed Property

Type **boolean**

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

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

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

DeviceEnabled Property R/W

Type **boolean**

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

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

The application must set this property to true before using the device.

If the **State** property is other than JPOS_S_IDLE(2), **DeviceEnabled** property cannot be changed from true to false.

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

FreezeEvents Property R/W

Type **boolean**

Remarks Gets or sets a Boolean value that indicates whether the events will be delivered or not. The following table shows the valid property values.

Value	Meaning
false	The application allows events to be delivered. If some events have been held while events were frozen and all other conditions are correct for delivering the events, changing this property to false allows these events to be delivered.
true	Events will not be delivered. Events will be enqueued until this property 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 a print method such as the **printNormal** method is operated under the **AsyncMode** property is true, the **ErrorEvent** event is frozen and the **State** property turns to JPOS_S_BUSY(3). In this case, discard the frozen event by the **clearOutput** method or set the **FreezeEvents** property to false to cause **ErrorEvent**, and then execute the **close** method, since the POSPrinter device cannot be closed under this circumstance.

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

OutputID Property

Type **int**

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

When a method successfully initiates an asynchronous output, the 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.

PowerNotify Property R/W

Type **int**

Remarks Gets or sets 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 the PowerState property may not be set.
JPOS_PN_ENABLED(1)	When DeviceEnabled is set to true, the POSPrinter device will fire the power notification StatusUpdateEvents and update the PowerState property. The level of functionality depends on the value of CapPowerReporting .

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

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

PowerState Property

Type **int**

Remarks Gets 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 due to 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 for use.
JPOS_PS_OFF_OFFLINE(2004)	The device is powered off or offline.

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

State Property

Type **int**

Remarks Gets 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 normal state and is busy executing output.
JPOS_S_ERROR(4)	An error has been reported, and the application must recover the Control to a normal state before normal I/O can resume. This state is only possible inside the ErrorEvent event handler.

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

DeviceControlDescription Property

Type **String**

Remarks Gets a string identifying the Device Control and the company that produced it.
This property is indicated to "JavaPOS POSPrinter Device Control".
This property is always readable.

DeviceControlVersion Property

Type **String**

Remarks Gets the Device Control version number.
This property is indicated to "1013000".
This property is always readable.

DeviceServiceDescription Property

Type **String**

Remarks Gets a string identifying the Device Service and the company that produced it.
This property is initialized to "SII RP Series JavaPOS POSPrinter Service Driver, Copyright (C) 20xx Seiko Instruments Inc." by the **open** method.

DeviceServiceVersion Property

Type **String**

Remarks Gets the Device Service version number.
This property is initialized to "1013000" by the **open** method.

PhysicalDeviceDescription Property

Type **String**

Remarks Gets a string identifying the physical device.
This property depends on the printer.
This property is initialized to one of the following values by the **open** method.

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

PhysicalDeviceName Property

Type **String**

Remarks Gets a short string identifying the physical device.
This property depends on the printer.
This property is initialized to one of the following values by the **open** method.

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

4.3.2 Specific Properties

AsyncMode Property R/W

Type **boolean**

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

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

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

CapCharacterSet Property

Type **int**

Remarks Indicates 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 the **open** method.

CapCoverSensor Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapMapCharacterSet Property

Type **boolean**

Remarks Gets a Boolean value that indicates whether the POSPrinter device is able to map the characters of the application to a character set defined by the **CharacterSetList** property. 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 the CharacterSetList property.

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

CapRec2Color Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecBarCode Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecBitmap Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecBold Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecCartridgeSensor Property

Type **int**

Remarks Gets a value that 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 the **open** method.

CapRecColor Property

Type **int**

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

Value	Meaning
PTR_COLOR_PRIMARY (0x00000001)	Receipt supports primary color.

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

CapRecDhigh Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecDwide Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecDwideDhigh Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecEmptySensor Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecItalic Property

Type **boolean**

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

Value	Meaning
false	The receipt cannot print Italic characters.

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

CapRecLeft90 Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecMarkFeed Property

Type **int**

Remarks Gets a value that holds the type of mark-sensed paper handling available.
The following table shows the valid property values.

Value	Meaning
0	Mark-sensed paper handling is not supported.

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

CapRecNearEndSensor Property

Type **boolean**

Remarks Gets a Boolean value that indicates whether the receipt has a paper-near-end sensor.
The following table shows the valid property values.

Value	Meaning
false	The printer doesn't have a "paper-near-end" sensor.
true	The printer has a "paper-near-end" sensor.

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

In case of RP-D10:

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

In case of RP-E10:

When NearEndSensor property of JposEntry is set to "0", then this property is initialized to false by the **open** method.

When NearEndSensor property of JposEntry is set to "1", then this property is initialized to true by the **open** method.

CapRecPageMode Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecPaperCut Property

Type **boolean**

Remarks Gets a Boolean value that 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 false by the **open** method.

CapRecPresent Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecRight90 Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecRotate180 Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapRecRuledLine Property

Type **int**

Remarks Indicates the printable ruled lines of the receipt.
A logical OR combination of the following values is set.

Value	Meaning
0	Ruled line drawing printing is not supported..

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

CapRecStamp Property

Type **boolean**

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

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

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

CapRecUnderline Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapTransaction Property

Type **boolean**

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

Value	Meaning
true	The receipt station supports printer transactions.

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

CartridgeNotify Property R/W

Type **int**

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

Value	Meaning
PTR_CN_DISABLED(0)	Cartridge-state notifications are not available.

This property cannot be rewritten.

This property is initialized to PTR_CN_DISABLED(0) by the **open** method.

CharacterSet Property R/W

Type **int**

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

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

This property is initialized to the value set to **DefaultCharacterSet** property of JposEntry when the device is first enabled following the **open** method.

CharacterSetList Property

Type **String**

Remarks Gets the list of character set numbers supported for printing.
This property is initialized to "437, 850, 852, 858, 860, 863, 865, 932, 999, 1250, 1251, 1252, 1253, 1254" by the **open** method.

CoverOpen Property

Type **boolean**

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

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

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

ErrorLevel Property

Type **int**

Remarks Gets the severity of the most recent 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 delivering an **ErrorEvent**.

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 the **open** method.

ErrorStation Property

Type **int**

Remarks Holds the station or stations that were 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 delivering an **ErrorEvent**.

When the error is cleared, then the property is changed to 0.

This property is initialized to 0 by the **open** method

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
"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.
"Time out."	ErrorCode=JPOS_E_FAILURE(111)

When multiple errors occur simultaneously, the higher-priority value is set.

This property is set just before delivering an **ErrorEvent**.

When the error is cleared, then the property is changed to empty string.

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

FlagWhenIdle Property R/W

Type **boolean**

Remarks Gets or sets a Boolean value that indicates whether or not to notify that **Status** turns to JPOS_S_IDLE(2).
The following table shows the valid property values.

Value	Meaning
false	StatusUpdateEvent is not notified.
true	StatusUpdateEvent will be sent when the State property 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 the **clearOutput** method or by an **ErrorEvent** handler.

If the **State** property 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 the **open** method.

FontTypefaceList Property

Type **String[]**

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

MapCharacterSet Property R/W

Type **boolean**

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

Value	Meaning
false	No mapping is supported.

It's possible to set only false.

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

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 () indicate the values 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 formulae.

MapMode Property	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 $dot=english * dpi * k$
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 $dot=(metric * dpi * k)/mmpi$

The **MapMode** property only changes the unit of each property for display, and all internal processings are executed in dot regardless of the **MapMode** property.

Therefore, the rounding errors of values do not accumulate.

When converting a dot value to a map mode value, the value is rounded up to an integer.

When converting from a map mode value to a dot value, the decimal part is truncated.

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

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

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

Value	Meaning
"360,2400"	Page mode area when RecLineWidth property is 360.
"432,2400"	Page mode area when RecLineWidth property is 432.
"512,2400"	Page mode area when RecLineWidth property is 512.
"576,2400"	Page mode area when RecLineWidth property is 576.

Specify PTR_S_RECEIPT(2) for the **PageModeStation** property before accessing this property.

When a valid station is not specified, an empty string is returned.

PageModeDescriptor Property

Type **int**

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

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

Specify PTR_S_RECEIPT(2) for the **PageModeStation** property before accessing this property.

This property returns 0 until a valid station is specified.

It is initialized to OR of the above values when PTR_S_RECEIPT(2) is specified.

PageModeHorizontalPosition Property R/W

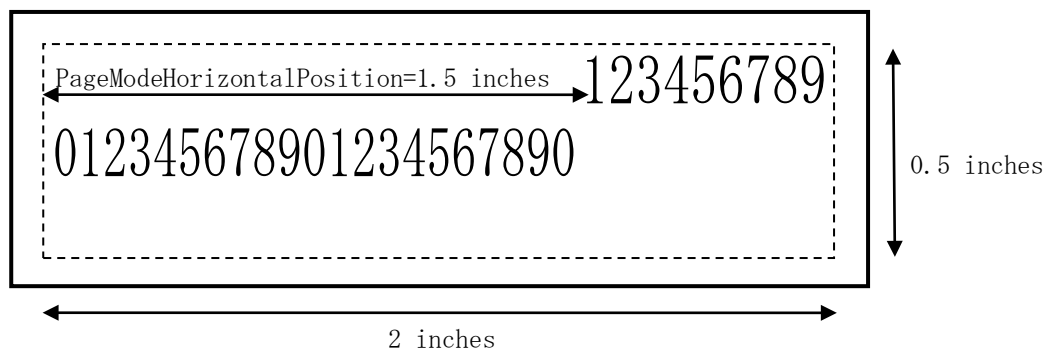
Type **int**

Remarks Gets or sets the horizontal start position offset within the print area for the print station specified by the **PageModeStation** property.
It expressed in the unit of measure specified by the **MapMode** property.
The horizontal direction is the same as the actual **PageModePrintDirection** property.
A read/get on this property will return the horizontal position offset set by the last write/set and not the current position.
The **PageModeStation** property must be set to PTR_S_RECEIPT(2) 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,"123456789012345678901234567890\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** property expressed in the unit specified by **MapMode**.

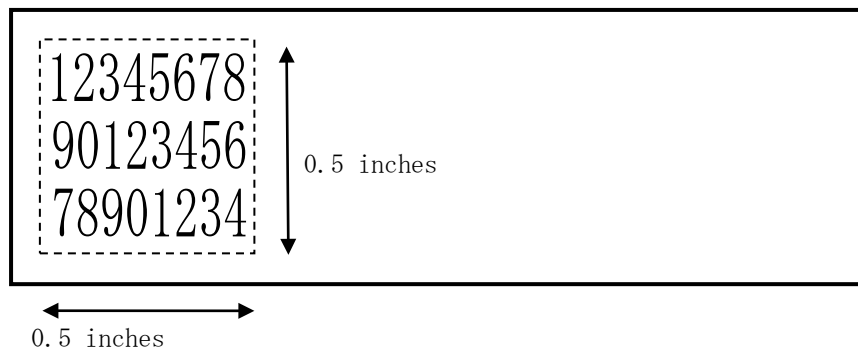
The maximum print area is the page area.

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 the **PageModeStation** property before accessing this property. This property returns an empty string until a valid station is specified. It is initialized to "0,0,0,0" when PTR_S_RECEIPT(2) is specified.

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.

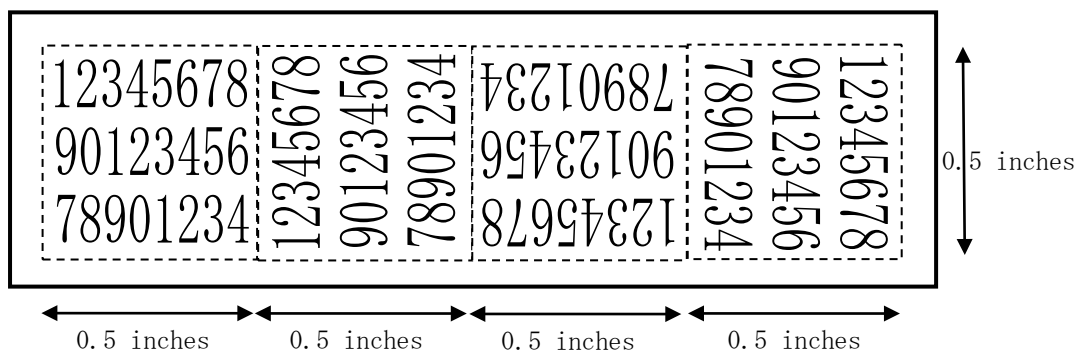
Value	Meaning
PTR_PD_LEFT_TO_RIGHT(1)	Normal direction printing. Print from left to right, starting at the top left corner of the page mode print area.
PTR_PD_BOTTOM_TO_TOP(2)	Rotated left 90° printing. Print from bottom to top, starting at the bottom left corner of the page mode print area.
PTR_PD_RIGHT_TO_LEFT(3)	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 the **PageModeHorizontalPosition** and **PageModeVerticalPosition** properties. Changing this property is only effective for the current print area. By changing the print areas, it is possible to generate a receipt with text printed in multiple rotations.

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);
```

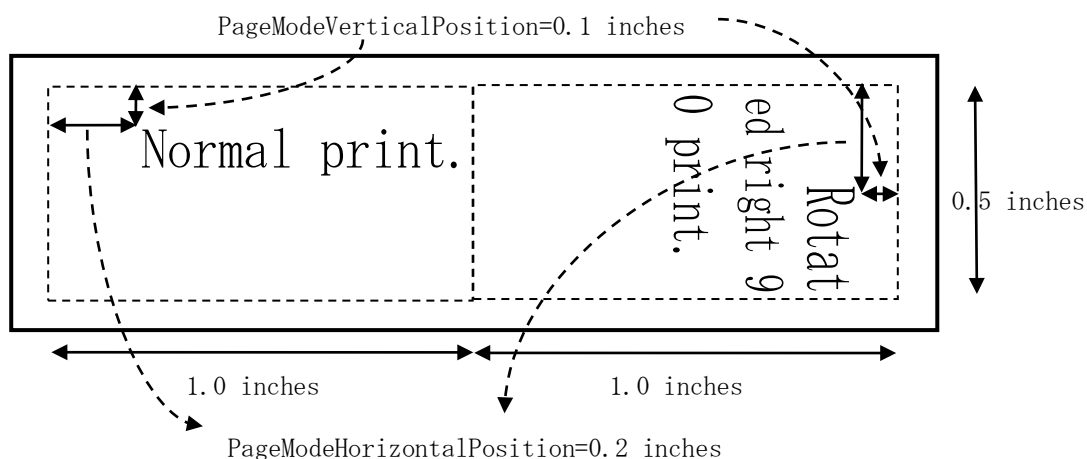
The code sample above will generate the following receipt.



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 the **PageModeStation** property before accessing this property. This property returns 0 until a valid station is specified. It is initialized to PTR_PD_LEFT_TO_RIGHT(1) when PTR_S_RECEIPT(2) is specified.

PageModeStation Property R/W

Type	int
Remarks	Gets or sets the printer station for subsequent page mode properties. The available station is PTR_S_RECEIPT(2) only. This property is initialized to 0 by the open method. Be sure to specify PTR_S_RECEIPT(2) for this property before accessing the property or method of the page mode function.

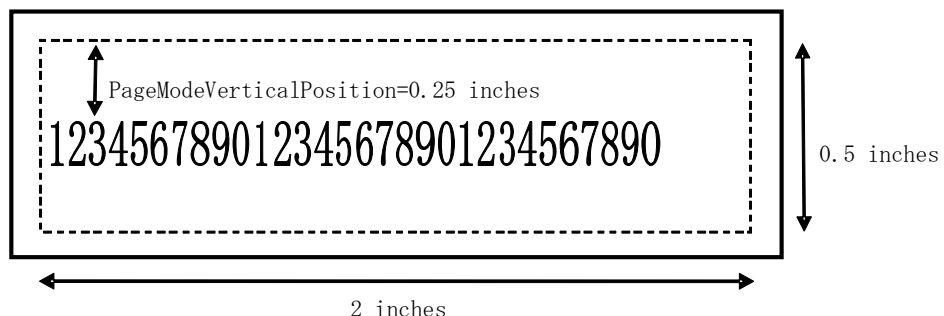
PageModeVerticalPosition Property R/W

Type	int
Remarks	Gets or sets the vertical start position offset within the print area for the print station specified by the PageModeStation property. This is expressed in the unit specified for the MapMode property. The vertical direction is perpendicular to the direction specified in the actual PageModePrintDirection property. 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,"123456789012345678901234567890\n");  
myptr.pageModePrint(PTR_PM_NORMAL);
```

The code sample above will generate the following receipt.



Specify PTR_S_RECEIPT(2) for the **PageModeStation** property before accessing this property. 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 the **open** method. The string consists of rotation strings separated by commas. The legal rotation strings are as follows:

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

RecBitmapRotationList Property

Type **String**

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

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

RecCartridgeState Property

Type **int**

Remarks Gets the status of the 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 following the **open** method call.

RecCurrentCartridge Property R/W

Type **int**

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

Value	Meaning
PTR_COLOR_PRIMARY (0x00000001)	Supports primary color.

It's possible to set only PTR_COLOR_PRIMARY(0x00000001).
This property is initialized to PTR_COLOR_PRIMARY(0x00000001) when the device is first enabled following the **open** method call.

RecEmpty Property

Type **boolean**

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

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

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

RecLetterQuality Property R/W

Type **boolean**

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

Value	Meaning
false	The prints in high speed mode.
true	The prints in high quality mode.

When the PrintSpeed property of JposEntry is set to "0", this property is valid.
Then the speed of printer is changed depending on the print mode.
When the PrintSpeed property 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 following the **open** method.

RecLineChars Property R/W

Type **int**

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

When **RecLineWidth** property is 360:

Value	Print Font (H × W)	Character space	RecLineHeight Property	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** property is 432:

Value	Print Font (H × W)	Character space	RecLineHeight Property	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	3 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** property is 512:

Value	Print Font (H × W)	Character space	RecLineHeight Property	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** property is 576:

Value	Print Font (H × W)	Character space	RecLineHeight Property	Available Range to specify
36	24 × 12 dots	4 dots	24	1 to 36
41	24 × 12 dots	2 dots	24	3 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** property is 360 and "41" is set, and then an exception is thrown.

Setting **RecLineChars** property may also update **RecLineHeight** property, **RecLineSpacing** property, **RecSidewaysMaxChars** property and **RecSidewaysMaxLines** property.

This property is initialized to the value set to **RecLineChar** property of **JposEntry** when the device is first enabled following the **open** method.

RecLineCharsList Property

Type **String**

Remarks Gets a collection of the line widths (characters per line) supported by the receipt station. This property is initialized to one of the following values by the **open** method.

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

RecLineHeight Property R/W

Type **int**

Remarks Gets or sets the receipt print line height.
It expressed in the unit of measure indicated by the **MapMode** property.
This property is not settable.
This property is automatically set by **RecLineWidth** property and **RecLineChars** property.
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** property is **PTR_MM_DOTS(1)**)

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

See the **RecLineChars** property for the relationship **RecLineChars** property and the print font. This property is initialized when the device is first enabled following the **open** method.

RecLineSpacing Property R/W

Type **int**

Remarks Gets or sets the spacing of each single-high print line. This includes both the printed line height and the white space between each pair of lines.
It expressed in the unit of measure indicated by the **MapMode** property setting.
The configurable range differs depending on the setting of **RecLineWidth** property and **RecLineChars** property.
The following table shows the valid configurable ranges. (When **MapMode** property is PTR_MM_DOTS(1))

RecLineWidth Property	RecLineChars Property	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 setting value is not supported, then the exception is thrown.

This property is initialized to the value set to **RecLineSpacing** property of JposEntry when the device is first enabled following the **open** method.

When the setting of Paper Saving is not Disable in case of RP-D10, the value specified in the setting of Paper Saving is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed. Therefore the value specified by the **RecLineSpacing** property is ignored.

See "RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details.

RecLinesToPaperCut Property

Type **int**

Remarks Gets 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** property.

[Calculation formula]

In case of RP-D10

RecLinesToPaperCut property = $88 / \text{RecLineSpacing}$ property

In case of RP-E10

RecLinesToPaperCut property = $100 / \text{RecLineSpacing}$ property

Example:

When the printer is RP-D10 and **RecLineSpacing** property is "31".

(**MapMode**=PTR_MM_DOTS(1))

RecLinesToPaperCut property = $88 / 31 = 2.93... = 3$

(round a number up to the nearest integer)

This property is initialized when the device is first enabled following the **open** method.

This property is initialized to the value based on the above calculation.

RecLineWidth Property

Type **int**

Remarks Gets the width of a line of the printer.
It expressed in the unit of measure indicated by the **MapMode** property setting.
This property is initialized to one of the following values when the device is first enabled following the **open** method call. (When **MapMode** property is PTR_MM_DOTS(1))

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

RecNearEnd Property

Type **boolean**

Remarks Gets a Boolean value that indicates whether the receipt paper is low.
The following table shows the valid property values.

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

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

When the **CapRecNearEndSensor** property is false, the value of this property is always false.

When the **CapRecNearEndSensor** property is true and the **RecEmpty** property is true, **RecNearEnd** property is always true.

RecSidewaysMaxChars Property

Type **int**

Remarks Gets 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 the **PageModeArea** property and print font.

[Calculation formula]

RecSidewaysMaxChars Property

=Maximum height of **PageModeArea** property/((Print font width/2)+Character space)

Example:

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

RecSidewaysMaxChars property=2400/((24/2)+0)=200

(round a number down to the nearest integer)

This property is initialized when the device is first enabled following the **open** method call.

This property is initialized to the value based on the above calculation.

RecSidewaysMaxLines Property

Type **int**

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

This property is determined by the following calculation based on the **RecLineWidth** property, **RecLineSpacing** property and **RecLineHeight** property.

[Calculation formula]

RecSidewaysMaxLines property

$$=(\text{RecLineWidth property}-\text{RecLineHeight property})/\text{RecLineSpacing property}+1$$

Example:

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

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

(round a number down to the nearest integer)

This property is initialized when the device is first enabled following the **open** method call.

This property is initialized to the value based on the above calculation.

RotateSpecial Property R/W

Type **int**

Remarks Holds the rotation orientation for barcodes.

It has one of the following values:

Value	Meaning
PTR_RP_NORMAL(1)	Prints subsequent barcodes 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 the **open** method.

While the *rotation* that is **rotationPrint** method parameter 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* parameter 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. This 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 the **CheckHealthText** property.
This method is always executed synchronous.

claim Method

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

The *timeout* parameter gives the maximum number of milliseconds to wait for exclusive access to be satisfied.

If the parameter is 0, then immediately either returns (if successful) or throws an appropriate exception.

If the parameter is JPOS_FOREVER(-1), the method waits until exclusive access is satisfied.

Remarks Calls this method to request exclusive access to the device.
The POSPrinter device cannot be used until exclusive access is obtained.
When successful, the **Claimed** property 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 Calls this method to clear all buffered output data.
Any output error events that were queued - usually waiting for that **FreezeEvents** to be set to false - are also cleared.

close Method

- Syntax** **void close()** throws **JposException**;
- Remarks** Calls this method to release the device and its resources.
If the **DeviceEnabled** property is true, then the device is disabled.
If the **Claimed** property is true, then exclusive access to the device is released.
Do not execute this 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
<i>command</i>	Command number. Specific values assigned by the Device Service.
<i>data</i>	An array of 1 mutable integer. Specific values or usage vary by <i>command</i> and Device Service.
<i>object</i>	Additional data. Usage vary by <i>command</i> and Device Service.

- 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
<i>command</i>	3
<i>data</i>	OUT Remaining memory
<i>object</i>	null

Issues the remaining memory response command and returns its response as a numeric value. The response data is stored in *data*.

- **Status response**

Parameter	Remarks
<i>command</i>	501
<i>data</i>	OUT Status
<i>object</i>	null

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

Out Status	RP-D10	RP-E10*1
0	Paper ready.	Paper ready / no near-end
1	No paper.	No paper / no paper-near-end
2	-	Paper is ready / paper-near-end
3	-	No paper / paper-near-end

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

- **International character selection**

Parameter	Remarks
<i>command</i>	201
<i>data</i>	International character number n $0 \leq n \leq 12$
<i>object</i>	null

Selects the character set of each country.

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 American

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

open Method

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

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

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

release Method

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

Remarks Calls this method to release exclusive access to the device.
If the **DeviceEnabled** property 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;**

Parameter	Remarks
<i>statisticsBuffer</i>	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 Calls this method to resets the resettable statistics defined in a device.
For the statistics that can be reset, see "Appendix B Statistics".
This method is always executed synchronous.

retrieveStatistics Method

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

Parameter	Remarks
<i>statisticsBuffer</i>	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 Calls this method to retrieves the requested statistics from a device.
When this method is successful, the XML string of the statistics is returned.
For the statistics that can be retrieved, see "Appendix B Statistics".
This method 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 <i>timeout</i>) throws JposException;
Remarks	This method is not supported.

beginRemoval Method

Syntax	void beginRemoval(int <i>timeout</i>) throws JposException;
Remarks	This method is not supported.

changePrintSide Method

Syntax	void changePrintSide(int <i>side</i>) throws JposException;
Remarks	This method is not supported.

clearPrintArea Method

Syntax	void clearPrintArea() throws JposException;
Remarks	Calls this method to clear the print data on the page mode print area defined by the PageModePrintArea property. The entire page may be cleared by setting the PageModePrintArea to be the same as the PageModeArea property and then using clearPrintArea method. The PageModeStation property must be set to PTR_S_RECEIPT(2) to invoking this method.

cutPaper Method

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

Parameter	Remarks
<i>percentage</i>	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

Calls this method to cut the receipt paper.

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

Paper cut can also be executed by using "Paper cut" escape sequence (ESC|[#]P) when calling **printNormal** or **printImmediate** methods.

If printing data remains in the printer buffer, paper cut is executed after all buffered data is printed.

During rotated 90° right/left mode by **rotatePrint** method, or while page mode by **pageModePrint** method is selected, this method is failed and paper cut is not executed.

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

When the setting of paper saving is enabled in RP-D10, the value specified in the setting of paper saving in the printer is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed (The value specified by the **RecLineSpacing** property is ignored).

However, when the "Paper cut" escape sequence (ESC|[#]P), the "Feed and Paper cut" escape sequence (ESC|[#]fP), or the **cutPaper** method is executed after the paper is fed by the carriage return (CR) or line feed (LF), distance from the last print line to the cut position is not reduced because paper is cut after executing the paper feed for saved dot lines.

See "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* parameter are as follows.

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

Remarks Calls this method to enter or exit page mode for the station specified in the **PageModeStation** property.

If *control* is PTR_PM_PAGE_MODE(1), then page mode is entered.

Subsequent calls to **printNormal**, **printBarCode**, **printBitmap**, and **printMemoryBitmap** will buffer the print data until **pageModePrint** is called with the *control* parameter set to PTR_PM_PRINT_SAVE(2), PTR_PM_NORMAL(3), or PTR_PM_CANCEL(4). (In this case, the print methods only validate the method parameters and buffer the data they do not initiate printing. Also, the value of the **AsyncMode** property does not affect their operation. No **OutputID** will be assigned to the request, nor will an **OutputCompleteEvent** be queued.)

If *control* is specified to PTR_PM_PRINT_SAVE(2), then page mode is not exited.

If some data is buffered by calls to the methods **printNormal**, **printBarCode**, **printBitmap**, and **printMemoryBitmap**, then the buffered data is saved and printed. This control is used to print the same page layout with additional print items inside of the page.

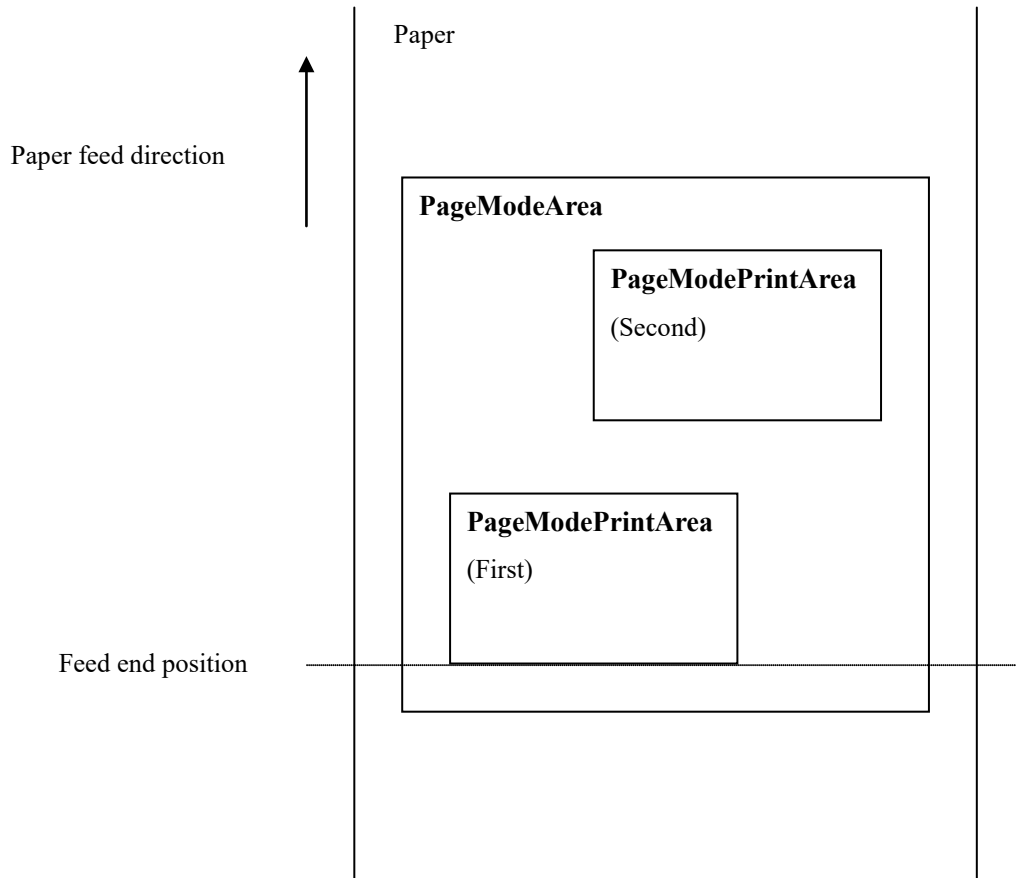
If *control* is PTR_PM_NORMAL(3), then page mode is exited.

If some data is buffered by calls to the methods **printNormal**, **printBarCode**, **printBitmap**, and **printMemoryBitmap**, then the buffered data is printed. The buffered data will not be saved.

If *control* is PTR_PM_CANCEL(4), then page mode is exited.

If some data is buffered by calls to the methods **printNormal**, **printBarCode**, **printBitmap**, and **printMemoryBitmap**, then the buffered data is not printed and is not saved.

Note that when the **pageModePrint** method is called, all of the print data on the page mode print area defined by **PageModePrintArea** will be printed and the paper is fed to the end of the page mode print area. If more than 1 **PageModePrintArea** is defined, then after the **PageModePrint** method is called, all of the data that is to be printed in their respective **PageModePrintArea(s)** will be printed and the paper will be fed to the end of page of the **PageModePrintArea** located the farthest "down" the sheet of paper. (see figure below).



The entire page mode transaction is treated as 1 message.

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

Calling the **clearOutput** method cancels page mode to return to the normal state. Any buffered print lines are also cleared.

page mode can be used within a transaction print, but not within a rotate print.

Specify **PTR_S_RECEIPT(2)** for the **PageModeStation** property before calling this method.

printBarcode Method

Syntax

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

Parameter	Remarks
<i>station</i>	Specifies the station to be used. Available station is only PTR_S_RECEIPT(2).
<i>data</i>	Specifies the character string of the barcode.
<i>symbology</i>	Specifies the barcode type to be used. See the table of <i>symbology</i> below.
<i>height</i>	Specifies the height of the barcode. Expressed in the unit given by MapMode . For PTR_MM_DOTS(1), specify the value between 1 and 255. For QR Code, although <i>height</i> is ignored, specify the value between 1 and 255 as well. During page mode by pageModePrint method, specify the value within the range of print area defined by the PageModePrintArea property and the PageModeVerticalPosition property.
<i>width</i>	Specifies the width of the barcode. Expressed in the unit given by MapMode . The actual printed barcode width is the best size that fits into the width specified by <i>width</i> . For PTR_MM_DOTS(1), specify the value between 1 and RecLineWidth . When the RotateSpecial property and printing barcode in a rotated 90° to right/left mode by rotatePrint method are executed, specify the value not exceeding the maximum value of the connected devices (see the PageModeArea property for the maximum value). During page mode by pageModePrint method, specify the value within the range of print area defined by the PageModePrintArea property and the PageModeHorizontalPosition property.
<i>alignment</i>	Specifies the position of the barcode. See the table of <i>alignment</i> below.
<i>textPosition</i>	Specifies the position of the text printed in the barcode. See the table of <i>textPosition</i> below.

- Values of *symbology* parameter

Value	Meaning
PTR_BCS_UPCA(101)	UPC-A
PTR_BCS_UPCE(102)	UPC-E
PTR_BCS_EAN8(103)	EAN8 (JAN8)
PTR_BCS_JAN8(103)	JAN8 (EAN8)
PTR_BCS_EAN13(104)	EAN13 (JAN13)
PTR_BCS_JAN13(104)	JAN13 (EAN13)
PTR_BCS_EAN13_S(119)	EAN13 (JAN13) with supplemental barcode
PTR_BCS_ITF(106)	Interleaved 2 of 5
PTR_BCS_Codabar(107)	Codabar (NW-7)
PTR_BCS_Code39(108)	Code39
PTR_BCS_Code93(109)	Code93
PTR_BCS_Code128(110)	Code128
PTR_BCS_PDF417(201)	PDF417
PTR_BCS_QRCODE(204)	QR Code (Mixed mode)

- Values of *alignment* parameter

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

When rotation 90° right/left is specified by **RotateSpecial** property, **rotatePrint** method, and during page mode by **pageModePrint** method, the setting of *alignment* 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 to print the barcode on the specified printer.
 This method is executed synchronously if **AsyncMode** property is false, and asynchronously if **AsyncMode** property is true.
 If the **RotateSpecial** property indicates that the barcode is rotated, the barcode is printed in rotated mode. The *height*, *width*, and *textPosition* parameters are applied to the barcode before it is rotated. For example, if PTR_BC_TEXT_BELOW(-13) is specified, the barcode is rotated to the left, and then text is printed to the right of the barcode.

Hereinafter, the limitations for each parameter of the barcode are described when the **MapMode** property is PTR_MM_DOTS(1).

[PTR_BCS_UPCA(101)]

Parameter	Limitation
<i>data</i>	Specify 11 or 12 letters consisting of '0' to '9'. The 12th letter does not affect the barcode printing data.
<i>width</i>	$width = 133 \times X$ $226 \leq width \leq 678$ X: fine element width $2 \leq X \leq 6$ X is automatically set according to <i>width</i> .

[PTR_BCS_UPCE(102)]

Parameter	Limitation
<i>data</i>	Specify 11 or 12 letters consisting of '0' to '9'. The 12th letter does not affect the barcode printing data.
<i>width</i>	$width = 65 \times X$ $130 \leq width \leq 390$ X: fine element width $2 \leq X \leq 6$ X is automatically set according to <i>width</i> .

Additionally, the allowable character must follow the rules below.

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

[PTR_BCS_EAN8(103)]

Parameter	Limitation
<i>data</i>	Specify 7 or 8 letters consisting of '0' to '9'. The 8th letter does not affect the barcode printing data.
<i>width</i>	$width=81 \times X$ $162 \leq width \leq 486$ X: fine element width $2 \leq X \leq 6$ X is automatically set according to <i>width</i> .

[PTR_BCS_JAN13(104)]

Parameter	Limitation
<i>data</i>	Specify 12 or 13 letters consisting of '0' to '9'. The 13th letter does not affect the barcode printing data.
<i>width</i>	$width=113 \times X$ $226 \leq width \leq 678$ X: fine element width $2 \leq X \leq 6$ X is automatically set according to <i>width</i> .

[PTR_BCS_EAN13_S(119)]

Parameter	Limitation
<i>data</i>	Specify 14, 15, 17, or 18 letters consisting of '0' to '9'. When 15 letters or 18 letters are inputted, the 13th character does not affect the printing data.
<i>width</i>	- When 14 or 15 letters are specified $width=138 \times X$ $276 \leq width \leq 828$ - When 17 or 18 letters are specified $width=165 \times X$ $330 \leq width \leq 990$ X: fine element width $2 \leq X \leq 6$ X is automatically set according to <i>width</i> .

[PTR_BCS_ITF(106)]

Parameter	Limitation
<i>data</i>	Specify any value consisting of '0' to '9'. Note that the number of specified letters must be an even number except for 0.
<i>width</i>	$width = ((D \times 2 + 1) \times X \times N + (D \times 3) + 6 + (10 \times 2)) \times X$ $(14 \times D + 56) \leq width \leq (54 \times D + 174)$ D: the number of barcode character X: fine element width $2 \leq X \leq 6$ 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 <i>width</i> .

[PTR_BCS_Codabar(107)]

Parameter	Limitation
<i>data</i>	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 '/'.
<i>width</i>	$width = ((6 \times X + 2 \times X \times N) \times D) + ((X \times N - X) \times D') + (X \times (10 \times 2 - 1))$ $(20 \times D + 2 \times D' + 38) \leq width \leq (72 \times D + 12 \times D' + 114)$ D: the number of barcode character D': the number of data character (the number of 'A' to 'D', '+', ':', '/', '.' included in barcode data) X: fine element width $2 \leq X \leq 6$ 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 <i>width</i> .

[PTR_BCS_Code39(108)]

Parameter	Limitation
<i>data</i>	At least one of '0' to '9', 'A' to 'Z', '-', '\$', '%', '+', '-', '.', '/' must be specified.
<i>width</i>	$width = (((X \times 7) + (X \times N \times 3)) \times (D + 2)) + ((10 \times 2 - 1) \times X)$ $(26 \times D + 90) \leq width \leq (96 \times D + 306)$ D: the number of barcode character X: fine element width $2 \leq X \leq 6$ 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 <i>width</i> .

[PTR_BCS_Code93(109)]

Parameter	Limitation
<i>data</i>	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.
<i>width</i>	$width = X \times ((10 \times 2) + ((D + 2) \times 9) + 1)$ $(18 \times D + 114) \leq width \leq (54 \times D + 342)$ D: the number of barcode character X: fine element width $2 \leq X \leq 6$ X is automatically set according to <i>width</i> .

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		

[PTR_BCS_Code128(110)]

Parameter	Limitation
<i>data</i>	Specify any value consisting of decimal numbers from 0 to 105. Each numeric value is treated as the corresponding character shown in the table below. The first letter of the first line must be a decimal number 103, 104, or 105, and at least 1 letter must follow it.
<i>width</i>	$width = X \times ((10 \times 2) + ((D + 2) \times 11) + 2)$ $(22 \times D + 88) \leq width \leq (66 \times D + 264)$ D: the number of barcode character (including start code) X: fine element width $2 \leq X \leq 6$ X is automatically set according to <i>width</i> .

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)		

[PTR_BCS_PDF417(201)]

Parameter	Limitation
<i>data</i>	00H to 7FH must follow the ASCII code and 80H to FFH must follow the extended character set of PC437 English list.
<i>width</i>	$width = (17 \times C + 69) \times X + (X \times 4)$ $(180 \leq width)$ $Height = R \times Y + (X \times 4)$ $(14 \leq Height \leq 255)$ X: module width (2 to 4) Y: module height (2 to 127) C: the number of vertical column (1 to 30) R: the number of row (3 to 90) For the number of row and the number of vertical column, the smallest value that input data can be converted to barcode is selected. For module width and module height, the maximum size that does not exceed the <i>width</i> and <i>height</i> parameters is selected after the number of row and the number of vertical column are determined.

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

[PTR_BCS_QRCODE (204)]

Parameter	Limitation
<i>data</i>	A character string combining arbitrary modes, including numeric mode, alphanumeric mode, 8 bits byte mode, and kanji mode. Each mode can handle the following characters: ·Numeric mode: 10 numbers ('0' to '9') ·Alphanumeric mode: 10 numbers ('0' to '9'), 26 alphabets ('A' to 'Z'), and 9 symbols (Space, '\$', '%', '*', '+', '-', '.', ':', '/') ·8 bits byte mode: 8 bits Latin/katakana characters based on JIS X 0201 (ASCII 00H to FFH) ·Kanji mode: Shift JIS code based on JIS X 0208
<i>width</i>	$width = (4V + 17) \times M + (4M \times 2)$ $(58 \leq width)$ V: Version of QR Code (1 to 40) M: Module size (2 to 11) For version, the smallest value that input data can be converted to barcode is selected. For module size, the maximum size that does not exceed the <i>width</i> parameter is selected after the version is determined.

QR Code model is fixed at 2 and an error correction level is fixed at M. Printing size is based on *width*, and *height* is ignored since QR Code is a square.

If data other than the printable characters is specified, JPOS_E_ILLEGAL(106) is thrown.

printBitmap Method

Syntax

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

Parameter	Remarks
<i>station</i>	Specifies the station to be used. Available station is only PTR_S_RECEIPT(2).
<i>fileName</i>	Specifies the name of bitmap file. or the supported image file, see below.
<i>width</i>	Specifies the print width of bitmap. See values below.
<i>alignment</i>	Specifies the print position of bitmap. See values below.

· Supported bitmap file

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

- Values of *width* parameter

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 defined by MapMode . If MapMode is PTR_MM_DOTS(1), specify between 1 and RecLineWidth property. When printing bitmap in a rotated 90° to right/left mode by RotatePrint method is executed, specify the value not exceeding the maximum value of the connected devices (see the PageModeArea property for the maximum value). During page mode by PageModePrint method, specify the value within the range of print area defined by the PageModePrintArea property and the PageModeHorizontalPosition property.

- Values of the *alignment* parameter

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

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

Remarks

Calls this method to print bitmap on the specified station.

The highest performance cannot be achieved since the bitmap data is transferred to the printer after **printBitmap** is called. It is recommended to print the bitmap data using **setBitmap** and escape sequence.

If any character data is already sent but not yet printed, that character data is printed first, a linefeed is automatically added, and then the bitmap is printed on the next print line.

Any character data sent after **printBitmap** is printed on the print line next to the bitmap.

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

The *width* parameter controls the transformation of bitmap data. If *width* is PTR_BM_ASIS(-11), then no transformation is executed. The bitmap is printed with 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
<i>station</i>	Specifies the station to be used. Available station is only PTR_S_RECEIPT(2).
<i>data</i>	Specifies the 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* on the device immediately.
The print data that exceeds the maximum number of characters per line is printed on the next print line. This method tries to print its data immediately without buffering by **pageModePrint** method, **rotatePrint** method or **transactionPrint** method.
If printing 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 feed to the next line.
CR	Replaceable with the same operation as line feed (LF).
LF & CR	Carriage return (CR) is replaceable with the same operation as line feed (LF). Therefore, operation of line feed (LF) is executed twice.
CR & LF	Carriage return (CR) is ignored. Operation of line feed (LF) is executed once.

When the setting of Paper Saving is not Disable in case of RP-D10, the value specified in the setting of Paper Saving is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed (The value specified by the **RecLineSpacing** property is ignored).

However, when any of the "Paper cut" escape sequence (ESC|[#]P), the "Feed and Paper cut" escape sequence (ESC|[#]fP), or the **cutPaper** method is executed after the paper is fed by the carriage return (CR) or line feed (LF), distance from the last print line to the cut position is not reduced because paper is cut after executing the paper feed for saved dot lines.

See "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
<i>station</i>	Specifies the station to be used. Available station is only PTR_S_RECEIPT(2).
<i>data</i>	Byte array holding the bitmap data. For the supported image file, see the printBitmap method.
<i>type</i>	Specifies the bitmap format. Only PTR_BMT_BMP (1) is supported.
<i>width</i>	Specifies the print width of bitmap. see the printBitmap method.
<i>alignment</i>	Specifies the print position of bitmap. see the printBitmap method.

Remarks Calls this method to print the bitmap held on the memory on the specified printer.
For the operation specifications, see the **printBitmap** method.
This method is executed synchronously if **AsyncMode** property is false, and asynchronously if **AsyncMode** property is true.

printNormal Method

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

Parameter	Remarks
<i>station</i>	Specifies the station to be used. Available station is only PTR_S_RECEIPT(2).
<i>data</i>	Specifies the 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* on the device.
The print data that exceeds the maximum number of characters per line is printed on the next print line.
If printing data remains in the printer buffer, printing is executed after all the buffered data is printed.
This method is executed synchronously if **AsyncMode** property is false, and asynchronously if **AsyncMode** property is true.

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

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

When the setting of Paper Saving is not Disable in case of RP-D10, the value specified in the setting of Paper Saving is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed (The value specified by the **RecLineSpacing** property is ignored).

However, when any of the "Paper cut" escape sequence (ESC[*#*]P), the "Feed and Paper cut" escape sequence (ESC[*#*]fP), or the **cutPaper** method is executed after the paper is fed by the carriage return (CR) or line feed (LF), distance from the last print line to the cut position is not reduced because paper is cut after executing the paper feed for saved dot lines.

See "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
<i>station</i>	Specifies the station to be used. Available station is only PTR_S_RECEIPT(2).
<i>rotation</i>	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.

Value	Meaning
PTR_RP_BITMAP(8192)	Starts rotated bitmap printing. This value is ORed with one of the above start rotated print values. Rotates the bitmap printed by printBitmap .

Remarks

Calls this method to executes the rotated printing.

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

If *rotation* is PTR_RP_ROTATE180(259), then upside-down print mode starts.

Subsequent calls to **printNormal** or **printImmediate** will print the data upside-down until **rotatePrint** is called with the *rotation* parameter set to PTR_RP_NORMAL(1). Lines are printed in the order that they are sent, with the start of each line justified at the right margin of the printer.

If PTR_RP_BARCODE(4096) is set as OR of PTR_RP_ROTATE180(259) in *rotation*, the barcode printing by **printBarCode** method makes upside-down barcode.

Also, if PTR_RP_BITMAP(8192) is set as OR of PTR_RP_ROTATE180(259) in *rotation*, the bitmap printing by **printBitmap** method makes upside-down bitmap.

If *rotation* is PTR_RP_RIGHT90(257) or PTR_RP_LEFT90(258), the horizontal writing mode starts. Until **RotatePrint** is called by setting *rotation* parameter to PTR_RP_NORMAL(1), the data called by **printNormal** method is buffered. (In this case, the data of the above method is only buffered and not printed. Also, the value of **AsyncMode** property does not affect its operation. In other words, no **OutputID** is assigned and no **OutputCompleteEvent** is informed.)

If PTR_RP_BARCODE (4096) is set in *rotation*, barcode printing by **printBarCode** method is buffered.

Also, if PTR_RP_BITMAP(8192) is set in *rotation*, bitmap printing by the **printBitmap** method is buffered.

If *Rotation* is PTR_RP_NORMAL(1), rotated print mode is end. If some print data is buffered during the sideways rotated print is valid, the buffered data is printed.

When *rotation* contains PTR_RP_BITMAP(8192), the **RotateSpecial** setting is ignored.

Calling the **clearOutput** method cancels the rotated print mode.

Any buffered lines of sideways rotated print are also cleared.

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

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** method 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* parameter with OR.

setBitmap Method

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

Parameter	Remarks
<i>bitmapNumber</i>	Specifies the number to be assigned to this bitmap. Valid values are 1 to 20.
<i>station</i>	Specifies the station to be used. Available station is only PTR_S_RECEIPT(2).
<i>fileName</i>	Specifies the name of bitmap file. If "(empty string)" is set, the bitmap setting is canceled. For the supported image file, see printBitmap .
<i>width</i>	Specifies the print width of bitmap. See values below.
<i>alignment</i>	Specifies the print position of bitmap. See printBitmap for values.

• Values of *width* parameter

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 defined by MapMode . If MapMode is PTR_MM_DOTS(1), specify between 1 and RecSidewaysMaxLines property.

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 bitmap may then be printed by calling the **printNormal** or **printImmediate** methods 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 printing with downloading bitmap data in the downloaded bit image area of the printer and NV graphics area of the printer. When bitmap print is specified by escape sequence, only command which conducts printing is transmitted to provide better performance.

setLogo Method

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

Parameter	Remarks
<i>location</i>	Specifies the logo to be set. Valid values are PTR_L_TOP(1) or PTR_L_BOTTOM(2).
<i>data</i>	Specifies the 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 feed to the next line.
CR	Replaceable with the same operation as line feed (LF).
LF & CR	Carriage return (CR) is replaceable with the same operation as line feed (LF). Therefore, operation of line feed (LF) is executed twice.
CR & LF	Carriage return (CR) is ignored. Operation of line feed (LF) is executed once.

When the setting of Paper Saving is not Disable in case of RP-D10, the value specified in the setting of Paper Saving is applied to the line spacing when the carriage return (CR) or line feed (LF) is executed.

(The value specified by the **RecLineSpacing** property is ignored).

However, when any of the "Paper cut" escape sequence (ESC[*#*]P), the "Feed and Paper cut" escape sequence (ESC[*#*]fP), or the **cutPaper** method is executed after the paper is fed by the carriage return (CR) or line feed (LF), distance from the last print line to the cut position is not reduced because paper is cut after executing the paper feed for saved dot lines.

See "RP-D10 SERIES THERMAL PRINTER TECHNICAL REFERENCE" for details.

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

The *data* registered by this method is kept by the character of **CharacterSet** property at the time when the method is executed. If the data including kanji character(s) is registered by **setLogo** method when the **CharacterSet** property is set to 932, the kanji character(s) will be printed even if the **CharacterSet** property 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
<i>station</i>	Specifies the station to be used. Available station is only PTR_S_RECEIPT(2).
<i>control</i>	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**, and **printBitmap** methods will buffer the print data at the POSPrinter device until **transactionPrint** is called with the *control* parameter set to PTR_TP_NORMAL(12). (In this case, the print methods only validate the method parameters and buffer the data – they do not initiate printing. Also, the value of **AsyncMode** property does not affect its operation. In other words, no **OutputID** property is assigned and no **OutputCompleteEvent** is informed.)

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

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

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

validateData Method

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

Parameter	Remarks
<i>station</i>	Specifies the station to be used. Available station is only PTR_S_RECEIPT(2).
<i>data</i>	Specifies the data to be validated. Consists of printable characters, escape sequences, carriage returns (CR), and line feeds (LF).

Remarks Before calling the **printImmediate** or **printNormal** methods, this method is called when determining whether a data sequence, which possibly including 1 or more escape sequences, is valid for the specified printer.

This method does not cause any printing but is used to determine the capability of the printer.

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 <i>e</i>);
Remarks	This event is not supported.

ErrorEvent Event

Interface	jpos.events.ErrorListener
Syntax	errorOccurred(ErrorEvent <i>e</i>);

This event contains the following properties:

Property	Type	Remarks
<i>ErrorCode</i>	int	Error Code causing the error event.
<i>ErrorCodeExtended</i>	int	Extended Error code causing the error event.
<i>ErrorLocus</i>	int	Location of the error.
<i>ErrorResponse</i>	int	Error response, whose default value may be overridden by the application.

The *ErrorCode* property has 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 <i>ErrorCodeExtended</i> is occurred.

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

Value	Meaning
EPTR_COVER_OPEN(201)	The printer cover is open status.
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	Unrecoverable error has occurred.

The *ErrorLocus* property has one of the following values:

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

The *ErrorResponse* property 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);**

This event contains the following property:

Property	Type	Remarks
<i>OutputID</i>	int	The ID number of the asynchronous output request that is complete.

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

StatusUpdateEvent Event

Interface **jpos.events.StatusUpdateListener**

Syntax **statusUpdateOccurred(StatusUpdateEvent e);**

This event contains the following property:

Property	Type	Remarks
<i>Status</i>	int	The state of printer.

Status property has one of the following values:

Value	Meaning
PTR_SUE_COVER_OPEN(11)	The printer cover is open.
PTR_SUE_COVER_OK(12)	The printer cover is closed.
PTR_SUE_REC_EMPTY(24)	The receipt is out of paper.
PTR_SUE_REC_NEAREMPTY(25) ^{*1}	The receipt 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, the State property is always JPOS_S_IDLE(2). The FlagWhenIdle property must be set to true for this event to be notified. The POSPrinter device sets the FlagWhenIdle property to false and notify the event.
JPOS_SUE_POWER_ONLINE(2001)* ²	The device is powered on and ready.
JPOS_SUE_POWER_OFF_OFFLINE(2004)* ²	The device is powered off or offline.

*1: Only RP-E10 is supported.

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

Remarks

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

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	"1013000"
PhysicalDeviceDescription	String	R	open	"SII RP-D10 Cash Drawer"*1
				"SII RP-E10 Cash Drawer"*2
PhysicalDeviceName	String	R	open	"RP-D10 Cash Drawer" *1
				"RP-E10 Cash Drawer" *2

*1: In case of RP-D10

*2: In case of RP-E10

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 Gets a Boolean value that 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 the **open** method.

CapPowerReporting Property

Type **int**

Remarks Gets the power 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 the **open** method.

CapStatisticsReporting Property

Type **boolean**

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

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

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

CapUpdateFirmware Property

Type **boolean**

Remarks Gets a Boolean value that indicates whether a 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 the **open** method.

CapUpdateStatistics Property

Type **boolean**

Remarks Gets a Boolean value that indicates whether some or all the device statistics can be reset to 0 by using the **resetStatistics** methods.
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 the **open** method.

CheckHealthText Property

Type **String**

Remarks Holds the results of the most recent call to the **checkHealth** method.
The following examples show the results of diagnosis.

Value	Meaning
"Internal HCheck: Successful"	Succeeded in the health check without using the device.
"Internal HCheck: Failure"	Failed in the communication confirmation of the cash drawer.
"External HCheck: Successful"	Succeeded in opening the drawer using the device.
"External HCheck: Failure"	Failed in opening the drawer using the device.
"Interactive HCheck: Successful"	Succeeded in the interactive test of the device.
"Interactive HCheck: Failure"	Failed in the interactive test of the device.
"Interactive HCheck: Canceled"	For the interactive test of the device, the dialog is closed without testing.

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

Claimed Property

Type **boolean**

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

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

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

DeviceEnabled Property R/W

Type **boolean**

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

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

The application must set this property to true before using the device.

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

FreezeEvents Property R/W

Type **boolean**

Remarks Gets or sets a Boolean value that indicates whether the events will be delivered or not. The following table shows the valid property values.

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

PowerNotify Property R/W

Type **int**

Remarks Gets or sets 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 the PowerState property may not be set.
JPOS_PN_ENABLED(1)	When DeviceEnabled is set to true, the CashDrawer class will fire the power notification StatusUpdateEvents and update the PowerState property. The level of functionality depends on the value of CapPowerReporting property.

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

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

PowerState Property

Type **int**

Remarks Gets 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 due to one of the following reasons. • PowerNotify property is JPOS_PN_DISABLED(0). • DeviceEnabled property is false.
JPOS_PS_ONLINE(2001)	The device is powered on and ready for use.
JPOS_PS_OFF_OFFLINE(2004)	The device is powered off or offline.

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

State Property

Type **int**

Remarks Gets 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 the **open** method is successfully completed.

DeviceControlDescription Property

Type **String**

Remarks Gets a string identifying the Device Control and the company that produced it.
This property is indicated to "JavaPOS CashDrawer Device Control".
This property is always readable.

DeviceControlVersion Property

Type **String**

Remarks Gets the Device Control version number.
This property is indicated to "1013000".
This property is always readable.

DeviceServiceDescription Property

Type **String**

Remarks Gets a string identifying the Device Service and the company that produced it.
This property is initialized to "SII RP Series JavaPOS CashDrawer Service Driver,
Copyright (C) 20xx Seiko Instruments Inc." by the **open** method.

DeviceServiceVersion Property

Type **String**

Remarks Gets the Device Service version number.
This property is initialized to "1013000" by the **open** method.

PhysicalDeviceDescription Property

Type **String**

Remarks Gets a string identifying the physical device.
This property depends on the printer.
This property is initialized to one of the following values by the **open** method.

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

PhysicalDeviceName Property

Type **String**

Remarks Gets a short string identifying the physical device.
This property depends on the printer.
This property is initialized to one of the following values by the **open** method.

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

5.2.2 Specific Properties

CapStatus Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

CapStatusMultiDrawerDetect Property

Type **boolean**

Remarks Gets a Boolean value that 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 the **open** method.

DrawerOpened Property

Type **boolean**

Remarks Gets the Boolean value that 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 the **DrawerOpenStatus** property of JposEntry.

- If **DrawerOpenStatus** property of JposEntry is set to "Low":
When the sensor status is "Low", this property indicates true.
- If **DrawerOpenStatus** property 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* parameter 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. This method fails when another application has exclusive access to the device.

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

claim Method

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

The *timeout* parameter gives the maximum number of milliseconds to wait for exclusive access to be satisfied.

If the parameter is 0, then immediately either returns (if successful) or throws an appropriate exception.

If the parameter is JPOS_FOREVER(-1), the method waits until exclusive access is satisfied.

Remarks Calls this method to request exclusive access to the device.
Acquisition of exclusive access is not essential since the cash drawer device is a sharable device.
When successful, the **Claimed** property 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	Calls this method to release the device and its resources. If the DeviceEnabled property is true, then the device is disabled. If the Claimed property is true, then exclusive access to the device is released. Do not execute this while the event is in progress (or in the event handler).

compareFirmwareVersion Method

Syntax	void compareFirmwareVersion(String <i>firmwareFileName</i>) throws JposException ;
Remarks	This method is not supported.

directIO Method

Syntax	void directIO(int <i>command</i>, int[] <i>data</i>, Object <i>object</i>) throws JposException ;
Remarks	This method is not supported.

open Method

Syntax	void open(String <i>logicalDeviceName</i>) throws JposException ;
	The <i>logicalDeviceName</i> parameter specifies the device name to open. Specify the registered logical name of the CashDrawer device to execute this method.
Remarks	Calls this method to open the device. When this method is successful, the common property and other class-specific properties are initialized.

release Method

Syntax	void release() throws JposException ;
Remarks	Calls this method to release exclusive access to the device. Do not execute this while the event is in progress (or in the event handler).

resetStatistics Method

Syntax	void resetStatistics(String <i>statisticsBuffer</i>) throws JposException ;
Remarks	This method is not supported.

retrieveStatistics Method

Syntax	void retrieveStatistics(String[1] <i>statisticsBuffer</i>) 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 <i>beepTimeout</i>, int <i>beepFrequency</i>, int <i>beepDuration</i>, int <i>beepDelay</i>) throws JposException ;
Remarks	Waits for notification that the drawer is closed. No control is returned until notification that the drawer is closed 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:

Property	Type	Description
<i>Status</i>	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.

*: This is notified when **PowerNotify** = JPOS_PN_ENABLED(1).

Remarks This event is notified when the open/close state of the drawer is changed.
The CashDrawer class notifies the first **StatusUpdateEvent** when the device is enabled.

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 the PageModeStation property.

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 the PageModeStation property.
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 the PageModeStation property.
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 the PageModeStation property.
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 <i>level</i> 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 <i>timeout</i> 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 <i>Timeout</i> (in millisecond) has elapsed before the device is released. Or, the device is not available before the <i>Timeout</i> (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 <i>logicalDeviceName</i> 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 the PageModeStation property. - This method can be executed while in the page mode by pageModePrint method.

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 <i>percentage</i> parameter is less than 0 or more than 100. - This method is not supported while in the rotated 90°right/left mode by rotatePrint method, or page mode by pageModePrint method.
	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 <i>control</i> parameter is not PTR_PM_PAGE_MODE(1), PTR_PM_PAGE_SAVE(2), PTR_PM_NORMAL(3), or PTR_PM_CANCEL(4). • The <i>control</i> parameter is specified PTR_PM_PAGE_SAVE(2), PTR_PM_NORMAL(3), or PTR_PM_CANCEL(4), before the <i>control</i> parameter is specified PTR_PM_PAGE_MODE(1). • Valid station is not specified for the pageModeStation property. • This method is not supported while in the rotated 90°right/left mode by rotatePrint method.
	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 <i>station</i> parameter is not PTR_S_RECEIPT(2). • Invalid value is specified for the <i>data</i> parameter. • Invalid value is specified for the <i>symbology</i> parameter. • The <i>height</i> parameter is less than 1 or more than 255. • Invalid value is specified for the <i>width</i> parameter. • Invalid value is specified for the <i>alignment</i> parameter. • The <i>textPosition</i> 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 <i>station</i> parameter is not PTR_S_RECEIPT(2). • Invalid value is specified for the <i>width</i> parameter. • Invalid value is specified for the <i>alignment</i> 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 <i>fileName</i> 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 <i>station</i> parameter is not PTR_S_RECEIPT(2). • Invalid value is specified for the <i>data</i> 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 <i>station</i> parameter is not PTR_S_RECEIPT(2). • Invalid value is specified for the <i>width</i> parameter. • Invalid value is specified for the <i>alignment</i> 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 <i>station</i> parameter is not PTR_S_RECEIPT(2). • Invalid value is specified for the <i>data</i> 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 <i>station</i> parameter is not PTR_S_RECEIPT(2). • Invalid value is specified for the <i>rotation</i> parameter. • This method is not supported while in the page mode by pageModePrint method.
	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 <i>bitmapNumber</i> parameter is less than 1 or more than 20. • The <i>station</i> parameter is not PTR_S_RECEIPT(2). • Invalid value is specified for the <i>width</i> parameter. • Invalid value is specified for the <i>alignment</i> 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 <i>fileName</i> parameter does not exist.
	JPOS_E_FAILURE(111)	Communication error has occurred.
	JPOS_E_BUSY(113)	Asynchronous output is in progress.
setLogo	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.
	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 <i>location</i> parameter is not neither PTR_L_TOP(1), nor PTR_L_BOTTOM(2). • The <i>data</i> 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 <i>station</i> parameter is not PTR_S_RECEIPT(2). • The <i>control</i> 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 method, or page mode by pageModePrint method.
	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 below for detail.
	JPOS_E_FAILURE(111)	See below for detail.

A.1.5 Escape Sequence

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 method, or page mode by pageModePrint method.
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 method, or page mode by pageModePrint method.
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 method, or page mode by pageModePrint method.
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 the MapMode property. • The feed unit number '#' is not correct. • This escape sequence is not supported while in the rotated 90°right/left mode by rotatePrint method, or page mode by pageModePrint method.
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 method, or page mode by pageModePrint method.
Center	This escape sequence is not supported while in the rotated 90°right/left mode by rotatePrint method, or page mode by pageModePrint method.
Right justify	This escape sequence is not supported while in the rotated 90°right/left mode by rotatePrint method, or page mode by pageModePrint method.

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 <i>level</i> 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 <i>timeout</i> 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 <i>timeout</i> (in millisecond) has elapsed before the device is released. Or, the device is not available before the <i>timeout</i> (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 <i>logicalDeviceName</i> 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-D10" ¹	-
	"RP-E10" ²	
SerialNumber	"Unknown"	-
ManufactureDate	"Unknown"	-
MechanicalRevision	"1C" ¹	-
	"1A" ²	
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	-
ReceiptLinePrintedCount	0	-

XML Definition Name	Response	Resettable
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-D10.

*2: In case of RP-E10.

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)	-