

FP-2200 / FP-2100 / FP-2000

POSPrinter, CashDrawer

Application Programmer's Guide

of

Java for Retail POS Driver

for Serial/ USB Interface

Version 1.2

FUJITSU ISOTEC LIMITED.

Table of Contents

Preface.....	1
1. Outline.....	3
1.1. Subject Scope of this document.....	3
1.2. JavaPOS Driver Outline.....	4
1.3. Restrictions	6
1.4. Connection Way to POS Printer.....	8
1.5. About install.....	10
1.6. Setting Program Usage	11
2. Using JavaPOS Driver	15
2.1. Common.....	15
2.2. POS Printer	15
2.3. Drawer	15
2.4. Notes	15
3. JavaPOS Interface Specifications (Printer).....	16
3.1. List.....	16
3.2. Print Data and Escape Sequences	24
3.3. Common Properties	30
CapCompareFirmwareVersion Property	30
CapPowerReporting Property	30
CapStatisticsReporting Property	31
CapUpdateFirmware Property	31
CapUpdateStatistics Property.....	31
CheckHealthText Property	32
Claimed Property	32
DeviceControlDescription Property	32
DeviceControlVersion Property	33
PhysicalDeviceDescription	33
DeviceEnabled Property R/W	34
PhysicalDeviceName Property.....	34
FreezeEvents Property R/W	35
OutputID Property.....	35
PowerNotify Property R/W	36
PowerState Property.....	37
DeviceServiceDescription Property	37
DeviceServiceVersion Property	38
State Property.....	38
3.4. Common Method	39
checkHealth Method	39
claim Method	40

clearOutput Method	41
close Method	41
compareFirmwareVersion Method.....	42
directIO Method	44
open Method.....	46
release Method	46
resetStatistics Method	47
retrieveStatistics Method.....	47
updateFirmwareMethod	48
updateStatistics Method	49
3.5. Specific Properties	50
AsyncMode Property R/W	50
CapCharacterSet Property	50
CapCoverSensor Property	51
CapMapCharacterSet Property.....	51
CapRec2Color Property	51
CapRecBarCode Property	52
CapRecBitmap Property.....	52
CapRecBold Property.....	52
CapRecCartridgeSensor Property	53
CapRecColor Property	53
CapRecDhigh Property	53
CapRecDwide Property.....	54
CapRecDwideDhigh Property.....	54
CapRecEmptySensor Property	54
CapRecItalic Property	55
CapRecLeft90 Property.....	55
CapRecMarkFeed Property	55
CapRecNearEndSensor Property	56
CapRecPageMode Property	56
CapRecPapercut Property	56
CapRecRight90 Property	57
CapRecRotate180 Property	57
CapRecStamp Property	57
CapRecUnderline Property	58
CapTransaction Property	58
CartridgeNotify Property R/W	58
CharacterSet Property R/W	59
CharacterSetList Property	60
CoverOpen Property.....	61
ErrorLevel Property	62

ErrorStation Property	62
ErrorString Property	63
FlagWhenIdle Property R/W	63
FontTypefaceList Property	64
MapCharacterSet Property	64
MapMode Property R/W	65
PageModeArea Property	65
PageModeDescriptor Property	66
PageModeHorizontalPosition Property	66
PageModePrintArea Property	66
PageModePrintDirection Property	67
PageModeStation Property	67
PageModeVerticalPosition Property	67
RecBarcodeRotationList Property	68
RecBitmapRotationList Property	68
RecCartridgeState Property	69
RecCurrentCartridge Property R/W	69
RecEmpty Property	69
RecLetterQuality Property R/W	70
RecLineChars Property R/W	71
RecLineCharsList Property	73
RecLineHeight Property R/W	74
RecLineSpacing Property R/W	75
RecLinesToPaperCut Property	75
RecLineWidth Property	77
RecNearEnd Property	77
RecSidewaysMaxChars Property	78
RecSidewaysMaxLines Property	79
RotateSpecial Property R/W	80
3.6. Specific Methods	82
beginInsertion Method	82
beginRemoval Method	82
changePrintSide Method	83
clearPrintArea Method	83
cutPaper Method	84
endInsertion Method	85
endRemoval Method	85
markFeed Method	86
pageModePrint Method	86
printBarcode Method	87
printBitmap Method	102

printImmediate Method.....	105
printMemoryBitmap Method.....	107
printNormal Method.....	110
printTwoNormal Method.....	112
rotatePrint Method.....	113
setBitmap Method.....	116
setLogo Method.....	118
transactionPrint Method.....	119
validateData Method.....	121
3.7. Event.....	123
DirectIOEvent Event.....	123
ErrorEvent Event.....	124
OutputCompleteEvent Event.....	125
StatusUpdateEvent Event.....	126
4. JavaPOS Interface Specifications (Drawer).....	127
4.1. List.....	127
4.2. Common Properties.....	130
CapCompareFirmwareVersion Property.....	130
CapPowerReporting Property.....	130
CapStatisticsReporting Property.....	131
CapUpdateFirmware Property.....	131
CapUpdateStatistics Property.....	131
CheckHealthText Property.....	132
Claimed Property.....	132
DeviceControlDescription Property.....	132
DeviceControlVersion Property.....	133
PhysicalDeviceDescription Property.....	133
DeviceEnabled Property R/W.....	134
PhysicalDeviceName Property.....	134
FreezeEvents Property R/W.....	135
PowerNotify Property R/W.....	135
PowerState Property.....	136
DeviceServiceDescription Property.....	136
DeviceServiceVersion Property.....	136
State Property.....	137
4.3. Common Methods.....	138
checkHealth Method.....	138
claim Method.....	139
close Method.....	139
compareFirmwareVersion Method.....	140
directIO Method.....	140

open Method.....	141
release Method	141
resetStatistics Method	143
retrieveStatistics Method.....	143
updateFirmwareMethod	143
updateStatistics Method	144
4.4. Specific Properties	145
CapStatus Property.....	145
CapStatusMultiDrawerDetect Property.....	145
DrawerOpened Property.....	146
4.5. Specific Methods	147
openDrawer Method.....	147
waitForDrawerClose Method.....	147
4.6. Event	148
DirectIOEvent Event.....	148
StatusUpdateEvent Event.....	148
5. Xml file Configuration.....	149
5.1. The explanation of XML items (POS Printer)	149
5.2. The explanation of XML items (Drawer)	153
6. Log Files	155
7. Using Multiple Printers.....	157
8. Replacement of printer.....	158

Preface

(1) OS being targeted by this driver

This JavaPOS driver targets the following operating systems.

- Linux SUSE11.4 and later (x86/x64)
- Linux CentOS 6.4 and later (x86/x64)
- Microsoft(R) Windows Server(TM) 2008
- Microsoft(R) Windows Server(TM) 2008 x64 Edition
- Microsoft(R) Windows Vista(TM)
- Microsoft(R) Windows Vista(TM) x64 Edition
- Microsoft(R) Windows(R) Embedded POS Ready 2009 Edition
- Microsoft(R) Windows Server(TM) 2008R2 x64 Edition
- Microsoft(R) Windows 7
- Microsoft(R) Windows 7 x64 Edition
- Microsoft(R) Windows(R) Embedded POS Ready 7 Edition
- Microsoft(R) Windows(R) Embedded POS Ready 7 x64 Edition
- Microsoft(R) Windows Server(TM) 2012 x64 Edition
- Microsoft(R) Windows 8
- Microsoft(R) Windows 8 x64 Edition
- Microsoft(R) Windows Server(TM) 2012R2 x64 Edition
- Microsoft(R) Windows 8.1
- Microsoft(R) Windows 8.1 x64 Edition
- Microsoft(R) Windows 10(x86/x64)
- Microsoft(R) Windows Server(TM) 2016 x64 Edition
- Microsoft(R) Windows Server(TM) 2019 x64 Edition

(2) Interface being supported by this driver

This JavaPOS driver is supporting the printer connection by the following interfaces.

- USB
- RS232C

(3) About the description of this document

In this book, the expression that changes depending on OS is written in case of Linux SUSE11.4.

(When OS is Windows, the screen such as setup tool is different from an actual display.)

Only when the explanation of Windows is specially necessary, the explanation is written with the comment of "Windows".

(4) Supplementation concerning '5. XML file Configuration '

An initial value of the serial and USB interface of JavaPOS driver configuration file 'jpos.xml' is different in Windows and Linux.

e.g.) For FP-2000

- Linux

```
logicalName = "FP2000SERPRT"
portName = "/dev/ttyS0"
LogFile = "/usr/local/FP/log/fp2000serprt%g.log"
logicalName = "FP2000SER2PRT"
portName = "/dev/ttyS1"
LogFile = "/usr/local/FP/log/fp2000ser2prt%g.log"
logicalName = "FP2000USBPRT"
LogFile = "/usr/local/FP/log/fp2000usbprt%g.log"
logicalName = "FP2000USB2PRT"
LogFile = "/usr/local/FP/log/fp2000usb2prt%g.log"
```

- Windows

```
logicalName = "FP2000SERPRT"
portName = "COM1"
LogFile = "*/FIT/FP/log/fp2000serprt%g.log"
logicalName = "FP2000SER2PRT"
portName = "COM2"
LogFile = "*/FIT/FP/log/fp2000ser2prt%g.log"
logicalName = "FP2000USBPRT"
LogFile = "*/FIT/FP/log/fp2000usbprt%g.log"
logicalName = "FP2000USB2PRT"
LogFile = "*/FIT/FP/log/fp2000usb2prt%g.log"
```

(5) About the balloon message when the Java application is started on Windows

When you start the Java application when the operating system is Windows Vista and later, and the Aero function is effective, balloon message "The color scheme has been changed" might be displayed. There is no influence in the operation of the application though the Aero function temporarily becomes invalid when this message is displayed. Please click the balloon, and check "Don't show me this again." when you do not want to make this message be displayed from next time.

1. Outline

FP-2000 / FP-2100 / FP-2200 POS Printer JavaPOS Driver and Drawer JavaPOS Driver that control Fujitsu Isotec Limited FP-2000 Series POS Printer ("FP Printer") and Drawer connected to the printer are JavaPOS Driver conforming to JavaPOS Driver 1.13 POS Printer Device and Drawer Device. When using this JavaPOS Driver, refer to "UnifiedPOS Specification Version 1.13 Version", as well.

1.1. Subject Scope of this document

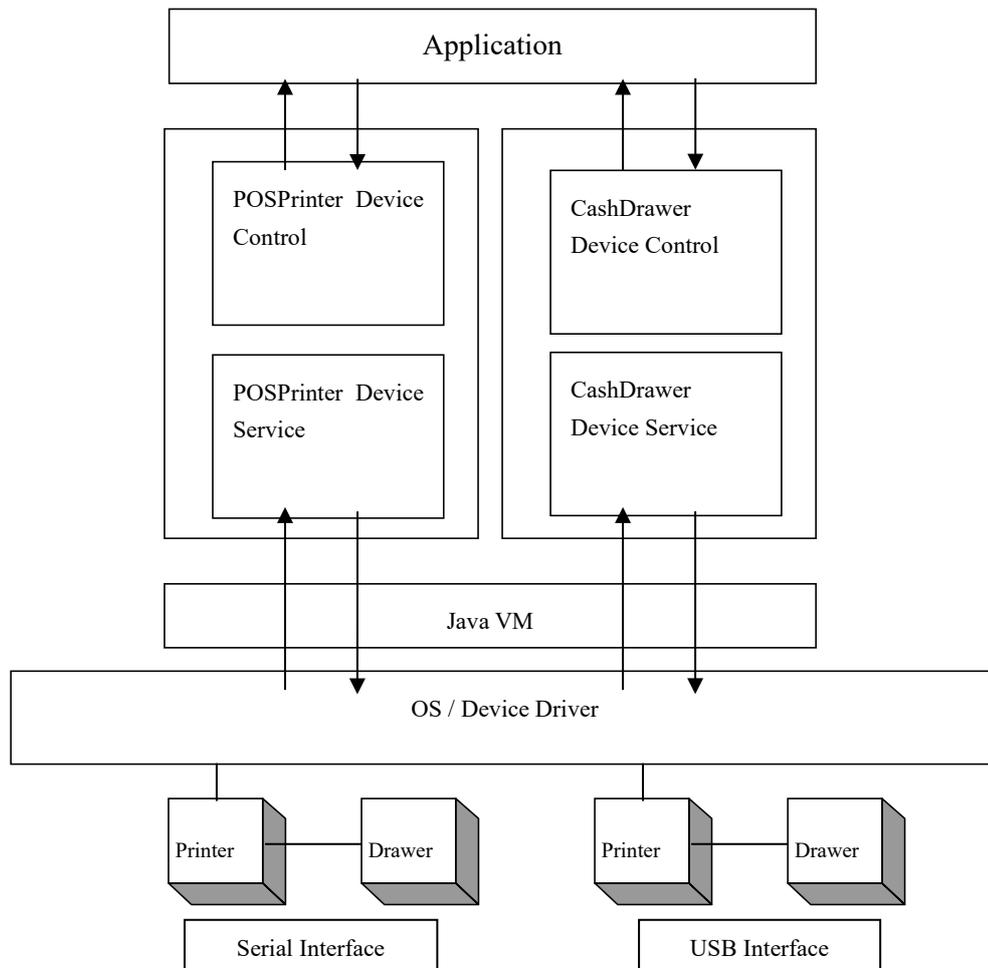
These instructions (Interface Instructions) aim for the main reference of programmers who develop the application for the use of this JavaPOS Driver, and describe the following contents necessary for that.

- Installation way of this JavaPOS Driver
Please refer to the installation text being bundled by the driver for the installation method.
Linux : "fpinstall_en_linux.txt"
Windows : "fpinstall_en_win.txt"
- Restrictions of this JavaPOS Driver
- Usage of this JavaPOS printer setup tool
- Usage of this JavaPOS Driver
- Restrictions of this JavaPOS Driver
- Interface (Property/Method/Event) Remarks of this JavaPOS Driver
- Item Setting Remarks of this JavaPOS Driver

1.2. JavaPOS Driver Outline

1) JavaPOS Driver Configuration Drawing

JavaPOS Driver provides the properties, methods and events to the application. The driver is invisible on UI during application execution. Only the application, which uses it, requests to process through the method and property. The application receives the processing result through the parameter, property, event and error.



*This JavaPOS Driver supports to control the Serial and USB Interface Printer and of at most two drawers each, connected to one printer.

*Multiple numbers of interface and printers can be set to the driver. For details, refer to Chapter 7 "Using Multiple Printers."

2) Terminology

a. Device Control (DC)

According to each device class, it provides application with the set of the properties, methods, and events. This Document explains these API.

b. Device Service (Device Service; DS)

It executes the function which is called from Device Control and which is prescribed by JavaPOS for each device.

1.3. Restrictions

The following restrictions are applied:

1) POS Printer

[Restrictions on JavaPOS specifications]

1. All interface of the JavaPOS POS Printer Device are provided, but there are the following restrictions:
 - a. It does not support property setting concerning journal printing and journal.
 - b. It does not support property setting concerning slip printing and slip.
 - c. It does not support functions of Italic, custom color, shading printing, and cartridge.
 - d. It does not support change of receipt printing character font. (Printing font change)
 - e. The following methods always return JPOS_E_ILLEGAL(106) after enabling.
 - printTwoNormal** Method
 - beginInsertion** Method
 - endInsertion** Method
 - beginRemoval** Method
 - endRemoval** Method
 - changePrintSide** Method
 - markFeed** Method
 - resetStatistics** Method
 - retrieveStatistics** Method
 - updateStatistics** Method
 - drawRuledLine** Method
2. For USB Interface connection, there are the following restrictions:
 - The behavior to set the DeviceEnabled property = true to the same printer from the application running on other VM is not supported.
3. The behavior after recovering from Suspend/Stand by is not supported.
The Suspend/Stand by mode should not be used.

2) Drawer

[Restrictions of JPOS specifications JavaPOS]

1. All the interfaces of JPOS Drawer Device are provided, but there are the following restrictions.
 - a. **PowerNotify** Property (Power source notifying function setting)
Setting is only for JPOS_PN_DISABLED(0) (Impossible to notify)and unchangeable.
 - b. **PowerState** Property (Power source state)
Only JPOS_PS_UNKNOWN(2000)(Unclear) is set.
 - c. **DirectIO** Method (Particular-to-Device function)
It is not supported. After enabling, it always returns JPOS_E_ILLEGAL(106).
 - d. **WaitForDrawerClose** Method (Waiting for the drawer to close)
It is not supported. After enabling, it always returns JPOS_E_ILLEGAL(106).

- e. **DirectIOEvent** Event (Particular- to-Device event)
It is not supported.
- f. **DrawerOpened** Property, **StatusUpdateEvent** Event
Status notification of the Drawer is available only when **CapStatus** is **true** and the driver is enabled (**DeviceEnabled=true**) for the printer connected to the drawer. In case these conditions are not met, the status of the drawer is not notified.

- 2. For USB Interface connection, there are the following restrictions:
 - The behavior to set the DeviceEnabled property = true to the same printer from the application running on other VM is not supported.
- 3. The behavior after recovering from Suspend/Stand by is not supported.
The Suspend/Stand by mode should not be used.

[Restriction of Drawer Hardware Specifications]

It does not support the function to notify the drawer power source condition.

3) Restriction when Windows driver and JavaPOS driver are installed in the same system

Problems such as failure to print correctly from the JavaPOS driver may occur if the Windows driver and JavaPOS driver are both installed in the same system.

In this case, it is recommended that you uninstall the driver that is not being used.

4) This JavaPOS driver does not support the following printer functions.

- "Retry at Error"
- "Plug and Play"

* The above settings are changed to "Disable" when setting with Setup Tool.

1.4. Connection Way to POS Printer

Set the POS Printer to the following settings (in gray highlight). Rest of the values can be set in the setting file attached with jpos.xml or the installer.

Memory Swith 1

No.	Setting Item	Setting Contents
1	Power On Status	*Set form the jpos.xml
2	Receive Buffer	64 KB
3	Busy Condition	Bufferfull
4	Receive Error	? Print
5	Auto LF	Disable
6	DSR (#6) RESET	Disable
7	INIT (#25) RESET	Disable
8	USB Soft Reset	Enable

Memory Swith 2

No.	Setting Item	Setting Contents
1	Cover Open Error	Auto Recovery
2	Error	Recovery by CMND
3	Batch (COM IF)	Disable
4	Batch (Other IF)	Disable
5	Serial Number	Enable
6	ASB	Enable
7	Font-B	Model

Print

No.	Setting Item	Setting Contents
1	DPI	*Set from the setting program
2	Paper Width	*Set from the setting program
3	Max Speed	*Set from the setting program
4	Print Density	*Set from the setting program
5	Retry at Error	Disable
6	Language Selection	*Set from the setting program

Hardware

No.	Setting Item	Setting Contents
1	User NV Memory	192KB
2	Graphic Memory	FP-2000 : 384KB FP-2100 / FP-2200 : 896KB
3	Cut at CoverClose	*Set from the setting program
4	Cutter Mode	*Set from the setting program (only FP-2200)
5	PNE Detect	*Set from the setting program

Hardware

No.	Setting Item	Setting Contents
1	Error Alert	*Set from the setting program
2	Buzzer Interval	*Set from the setting program
3	Buzzer Repetition	*Set from the setting program
7	Buzzer after Cut	*Set from the setting program

Interface

No.	Setting Item	Setting Contents
1	USB	Printer
2	Protocol	XON/XOFF
3	Plug and Play	Disable

1.5. About install

For Linux:

See the "fpinstall_en_linux.txt" and install the JavaPOS according to it.

For Windows:

See the "fpinstall_en_win.txt" and install the JavaPOS according to it.

1.6. Setting Program Usage

Operation Conditions

This Java-POS must be installed.

Screen and function

Setting Program is executed as the following procedure.

1. Compile

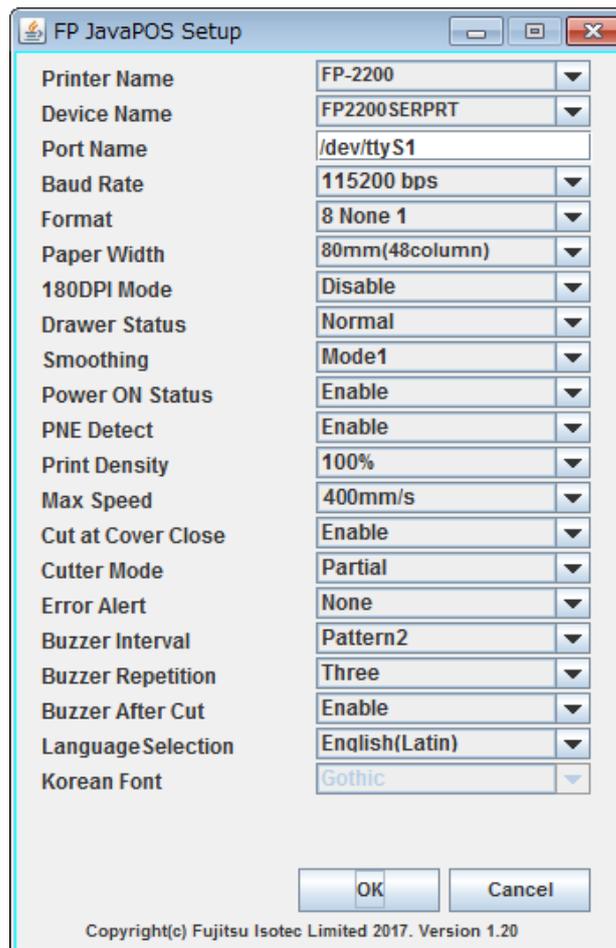
```
>javac -encoding UTF-8 Setup.java
```

2. Execution

```
>java Setup
```

The following screen is displayed.

<< Serial interface setting screen >>

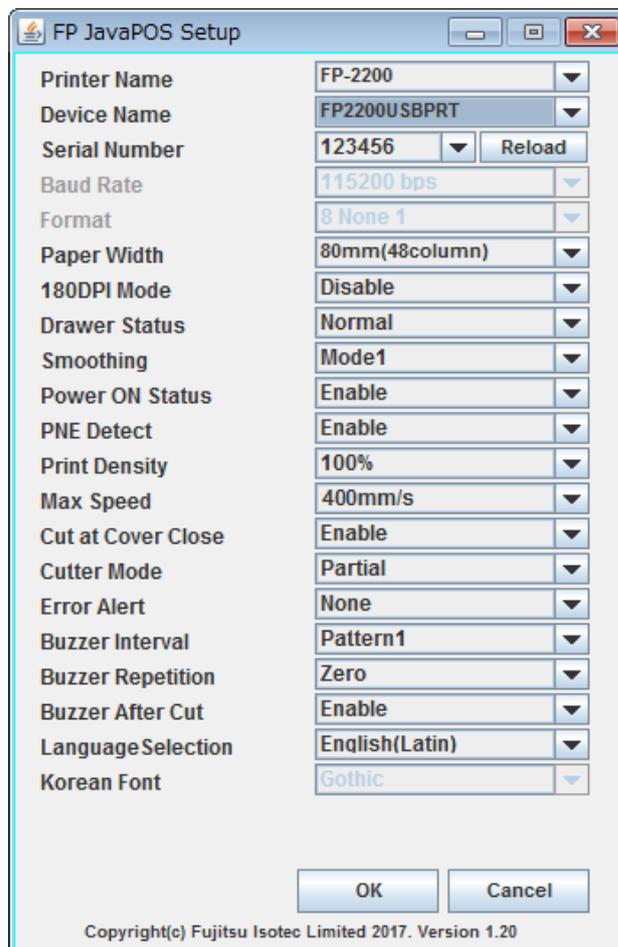


The screenshot shows a window titled "FP JavaPOS Setup" with a list of settings for a printer. Each setting is a label followed by a dropdown menu. The settings are:

Printer Name	FP-2200
Device Name	FP2200SERPRT
Port Name	/dev/ttyS1
Baud Rate	115200 bps
Format	8 None 1
Paper Width	80mm(48column)
180DPI Mode	Disable
Drawer Status	Normal
Smoothing	Mode1
Power ON Status	Enable
PNE Detect	Enable
Print Density	100%
Max Speed	400mm/s
Cut at Cover Close	Enable
Cutter Mode	Partial
Error Alert	None
Buzzer Interval	Pattern2
Buzzer Repetition	Three
Buzzer After Cut	Enable
Language Selection	English(Latin)
Korean Font	Gothic

At the bottom of the window are "OK" and "Cancel" buttons. Below the buttons is the copyright notice: "Copyright(c) Fujitsu Isotec Limited 2017. Version 1.20".

<< USB interface setting screen >>



Select serial number of printer from "Serial Number".

(Refer to "9. Serial number confirm method of printer" for serial number of printer.)

OK button becomes effective by selecting serial number.

When the Reload button is pushed, the serial number of the connected printer is reread, and displayed in "Serial Number".

"Korean Font" is valid only when Korean language is selected .

When an error occurred with OK button pushed, the following messages are displayed.

Confirm an error factor, and setup again.

<An error factor>

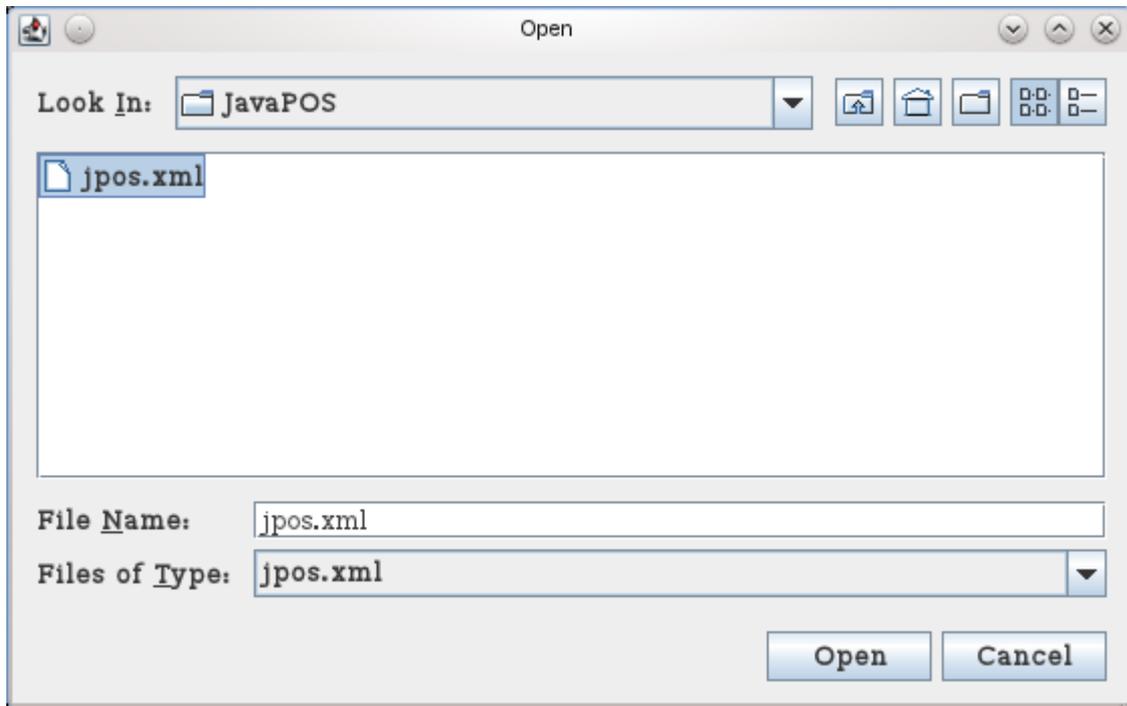
- A cable is not connected.
- The printer is not switched on.
- A cover opens.
- There is not paper.
- A port is already used in others.
- A communication condition does not accord with a printer. (Serial connection)
- Serial number input is wrong. (USB connection)



When the printer status is "error" and this setting program is executed, the setting is not reflected in the printer.

After confirming the setting, execute the setting program, and then follow the procedure again.

In case the "jpos.xml" file cannot be located when the setting program is executed, the following dialog is displayed. Select the "jpos.xml" file from this dialog.



2. Using JavaPOS Driver

2.1. Common

The application uses the OPOS control in the steps as follows:

1. **open** method: Called to link the control object to the service object.
2. **claim** method: Called to enable exclusive access to the device. For the device of exclusive use, this method is required, and for the device of sharable use, it is optional.
3. **DeviceEnabled** property: Set to **true** to operate the device.
4. Use the device. (Each property, method, event)
5. **DeviceEnabled** property: Set to **false** to disable the device.
6. **release** method: Called to clear exclusive access to the device.
7. **close** method: Called to release the service object from the control object.

2.2. POS Printer

The POS printer supports only "Receipt." For the methods and properties of other than that (Journal or Slip), interface is supplied but behavior is not supported.

According to the general output model, synchronous and asynchronous output is available for the POS printer.

The POS printer is the device to be used exclusively.

2.3. Drawer

The Drawer can be used in the same way as the POS printer, but all features are executable without executing the **claim** method. However, when exclusive permission is acquired for particular application by the **claim** method, the **openDevice** method cannot be executed by the application enabled with the same name. If there is no application with exclusive permission, this is not the case.

2.4. Notes

- When the application using this JavaPOS driver is started, you should set the New area of the memory to 10MByte or more. (java command option `-Xmn10m`). When the New area is a little, the memory utilization might increase because a large amount of object generated with the communication is not liberated for a long term.
- If the `JPOS_E_TIMEOUT` is returned when you enable the application, increase and adjust the value of **SendTimeout** of `jpos.xml` to adjust it. If you can not solve the problem even if you do this, also increase the value of **ResetTimeout** of `jpos.xml`. (Refer to Chapter 5.3).

3. JavaPOS Interface Specifications (Printer)

3.1. List

Properties

Common	Type	Access	May Use After	Initial Value, Conditions
CapCompareFirmwareVersion	boolean	R	open	true
CapPowerReporting	int	R	open	JPOS_PR_STANDARD (1)
CapStatisticsReporting	boolean	R	open	false
CapUpdateFirmware	boolean	R	open	true
CapUpdateStatistics	boolean	R	open	false
CheckHealthText	String	R	open	""
Claimed	boolean	R	open	false
DeviceEnabled	boolean	R/W	open & claim	false
FreezeEvents	boolean	R/W	false open	false Made writable after open.
OutputID	int	R	open	1
PowerNotify	int	R/W	open	JPOS_PN_DISABLED (0) Made writable after open, and unwritable after enabled.
PowerState	int	R	open	JPOS_PS_UNKNOWN (2000)
State	int	R	---	1
DeviceControlDescription	String	R	---	"JavaPOS POSPrinter Device Control"
DeviceControlVersion	int	R	---	1013XXX
DeviceServiceDescription	String	R	open	"FP POS Printer Device Service, (C) 20xx- Fujitsu Isotec"
DeviceServiceVersion	int	R	open	1013XXX
PhysicalDeviceDescription	String	R	open	"FP 1 Station Thermal POSPrinter (C) 20xx- Fujitsu Isote"
PhysicalDeviceName	String	R	open	"FP 1 Station Thermal POSPrinter "

Specific	Type	Access	May Use After	Initial Value, Conditions
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
CapMapCharacterSet	boolean	R	open	true
CapTransaction	boolean	R	open	true
CapJrnPresent	boolean	R	open	false
CapJrn2Color	boolean	R	open	false
CapJrnBold	boolean	R	open	false
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
CapJrnUnderline	boolean	R	open	false
CapJrnCartridgeSensor	int	R	open	0
CapJrnColor	int	R	open	0
CapRecPresent	boolean	R	open	true
CapRec2Color	boolean	R	open	The initial value may vary according to the contents of jpos.xml.
CapRecBarCode	boolean	R	open	true
CapRecBitmap	boolean	R	open	true
CapRecBold	boolean	R	open	true
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	The initial value may vary according to the contents of jpos.xml.
CapRecPapercut	boolean	R	open	true
CapRecRight90	boolean	R	open	true

Specific	Type	Access	May Use After	Initial Value, Conditions
CapRecRotate180	boolean	R	open	true
CapRecStamp	boolean	R	open	false
CapRecUnderline	boolean	R	open	true
CapRecCartridgeSensor	int	R	open	0
CapRecColor	int	R	open	0
CapRecMarkFeed	int	R	open	0
CapRecPageMode	boolean	R	open	false
CapSlpPresent	boolean	R	open	false
CapSlpFullslip	boolean	R	open	false
CapSlp2Color	boolean	R	open	false
CapSlpBarCode	boolean	R	open	false
CapSlpBitmap	boolean	R	open	false
CapSlpBold	boolean	R	open	false
CapSlpDhigh	boolean	R	open	false
CapSlpDwide	boolean	R	open	false
CapSlpDwideDhigh	boolean	R	open	false
CapSlpEmptySensor	boolean	R	open	false
CapSlpItalic	boolean	R	open	false
CapSlpLeft90	boolean	R	open	false
CapSlpNearEndSensor	boolean	R	open	false
CapSlpRight90	boolean	R	open	false
CapSlpRotate180	boolean	R	open	false
CapSlpUnderline	boolean	R	open	false
CapSlpBothSidesPrint	boolean	R	open	false
CapSlpCartridgeSensor	int	R	open	0
CapSlpColor	int	R	open	0
CapSlpPageMode	boolean	R	open	false
AsyncMode	boolean	R/Wopen	false open	false Made writable after enabled
CartridgeNotify	int	R/W	open	PTR_CN_DISABLED (0) Unwritable
CharacterSet	int	R/W	open, claim & Enable	The initial value may vary according to the contents of jpos.xml. Made writable after enabled
CharacterSetList	String	R	open	The initial value may vary according to the contents of jpos.xml.
CoverOpen	boolean	R	open, claim & Enable	false

Specific	Type	Access	May Use After	Initial Value, Conditions
ErrorLevel	int	R	open	1
ErrorStation	int	R	open	0
ErrorString	String	R	open	""
FontTypefaceList	String	R	open	""
FlagWhenIdle	boolean	R/W	open	false Made writable after enabled
MapCharacterSet	boolean	R/W	open	true
MapMode	int	R/W	open	PTR_MM_DOTS (1) Made writable after open
PageModeArea	String	R	open	""
PageModeDescriptor	int	R	open	0
PageModeHorizontalPosition	int	R	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
RotateSpecial	int	R/W	open	PTR_RP_NORMAL (1) Made writable after open
JrnLineChars	int	R/W	open, claim & Enable	0 Unwritable
JrnLineCharsList	String	R	open	""
JrnLineHeight	int	R	open, claim & Enable	0 Unwritable
JrnLineSpacing	int	R/W	open, claim & Enable	0 Unwritable
JrnLineWidth	int	R	open, claim & Enable	0
JrnLetterQuality	boolean	R/W	open, claim & Enable	false Unwritable
JrnEmpty	boolean	R	open, claim & Enable	false
JrnNearEnd	boolean	R	open, claim & Enable	false
JrnCartridgeState	int	R	open, claim & Enable	0
JrnCurrentCartridge	int	R/W	open, claim & Enable	0 Unwritable
RecLineChars	int	R/W	open, claim & Enable	The initial value may vary according to the jpos.xml contents. Made writable after

Specific	Type	Access	May Use After	Initial Value, Conditions
				open.
RecLineCharsList	String	R	open	The initial value may vary according to the jpos.xml contents.
RecLineHeight	int	R/W	open, claim & Enable	The initial value may vary according to the jpos.xml contents. Unwritable
RecLineSpacing	int	R/W	open, claim & Enable	The initial value may vary according to the jpos.xml contents. Made writable after open.
RecLineWidth	int	R	open, claim & Enable	The initial value may vary according to the jpos.xml contents.
RecLetterQuality	boolean	R/W	open, claim & Enable	true Made writable after open.
RecEmpty	boolean	R	open, claim & Enable	false
RecNearEnd	boolean	R	open, claim & Enable	false
RecSidewaysMaxLines	int	R	open, claim & Enable	The initial value may vary according to the jpos.xml contents.
RecSidewaysMaxChars	int	R	open, claim & Enable	The initial value may vary according to the jpos.xml contents.
RecLinesToPaperCut	int	R	open, claim & Enable	The initial value may vary according to the jpos.xml contents.
RecBarcodeRotationList	String	R	open	"0,R90,L90,180"
RecCartridgeState	int	R	open	268435456
RecCurrentCartridge	int	R/W	open	0 Unwritable
RecBitmapRotationList	String	R	open	"0"
SlpLineChars	int	R/W	open	0 Unwritable
SlpLineCharsList	String	R	open, claim & Enable	""
SlpLineHeight	int	R/W	open, claim & Enable	0 Unwritable
SlpLineSpacing	int	R/Wopen	open, claim & Enable	0 Unwritable

Specific	Type	Access	May Use After	Initial Value, Conditions
SlpLineWidth	int	R	open, claim & Enable	0
SlpLetterQuality	boolean	R/W	open, claim & Enable	false Unwritable
SlpEmpty	boolean	R	open, claim & Enable	false
SlpNearEnd	boolean	R	open, claim & Enable	false
SlpSidewaysMaxLines	int	R	open, claim & Enable	0
SlpSidewaysMaxChars	int	R	open, claim & Enable	0
SlpMaxLines	int	R	open, claim & Enable	0
SlpLinesNearEndToEnd	int	R	open, claim & Enable	0
SlpBarcodeRotationList	String	R	open	""
SlpPrintSide	int	R	open, claim & Enable	0
SlpCartridgeState	int	R	open, claim & Enable	PTR_CART_UNKNOWN (268435456)
SlpCurrentCartridge	int	R/W	open, claim & Enable	0 Unwritable
SlpBitmapRotationList	String	R	open	""

* In the Access column, R indicates Read-Only, R/W indicates Read/Write. The item in May Use After is the method and property required for initialization, open indicates the open method, claim indicates the claim method and Enable indicates setting the DeviceEnabled property to true. If required procedure is not executed, JposException may be notified. For the property with open & claim or open, claim & Enable in May Use After, it is available for acquisition after the open method is executed, but the value may not be initialized until all open, claim & Enable are executed. To acquire such property, access it after the conditions are met.

Methods

Common	Initialization
open	none
close	open
claim	open
release	open & claim
clearOutput	open, claim & Enable
checkHealth	open, claim & Enable
compareFirmwareVersion	open, claim & Enable
directIO	open, claim & Enable
resetStatistics	open, claim & Enable
retrieveStatistics	open, claim & Enable
updateFirmware	open, claim & Enable
updateStatistics	open, claim & Enable

Specific	Initialization
printNormal	open, claim & Enable
printTwoNormal	open, claim & Enable
printImmediate	open, claim & Enable
beginInsertion	open, claim & Enable
endInsertion	open, claim & Enable
beginRemoval	open, claim & Enable
endRemoval	open, claim & Enable
cutPaper	open, claim & Enable
rotatePrint	open, claim & Enable
printBarcode	open, claim & Enable
printBitmap	open, claim & Enable
transactionPrint	open, claim & Enable
validateData	open, claim & Enable
setBitmap	open, claim & Enable
setLogo	open, claim & Enable
changePrintSide	open, claim & Enable
markFeed	open, claim & Enable
clearPrintArea	open, claim & Enable
pageModePrint	open, claim & Enable
printMemoryBitmap	open, claim & Enable

Events

Event	Initialization
DirectIOEvent	open, claim & Enable
ErrorEvent	open, claim & Enable
OutputCompleteEvent	open, claim & Enable
StatusUpdateEvent	open, claim & Enable

3.2. Print Data and Escape Sequences

This POS printer supports the following escape sequences.

1) Escape Sequence which operates only when assigned time.

Name	Data	Remarks
Paper cut	ESC #P	<p>Cuts receipt paper. The character '#' is replaced by the character string of ASCII decimal string telling the percentage of required cutting. It is possible to omit '#'. When the value is between '1' to '99', partial cutting is performed. When the value is '100' or omitted, full cutting is performed. When the value is other than any value between '1' to '100', it is ignored. However, if data has been buffered at the POS Printer, that is, the print request is enqueued, but not printed on the POS Printer, a papercut is not allowed. A papercut is performed at the beginning of the line.</p> <p>It is unavailable in 90 degrees rotating to the left or to the right by RotatePrint Method, and back in operation after clearing 90 degrees rotating to the left or to the right.</p>
Feed and Paper cut	ESC #fP	<p>Cuts receipt paper, after feeding the paper by the ReCLinesToPaperCut lines. The character '#' is defined by the "Paper cut" escape sequence. However, if data has been buffered at the POS Printer, that is, print request is enqueued, but not printed on the POS Printer, a papercut is not allowed. A papercut is performed at the beginning of the line.</p> <p>It is unavailable in 90 degrees rotating to the left or to the right by RotatePrint Method, and back in operation after clearing 90 degrees rotating to the left or to the right.</p>
Feed, Paper cut, and Stamp	ESC sP	Not supported.
Print bitmap	ESC #B	<p>Prints the bitmap stored with the setBitmap method. '#' is the Bitmap number and supports 20 bitmap printing '1' to '20'. It is possible to change printing quality by changing RecLetterQuality property value in printing. As for handling printing quality, it is same as PrintBitmap Method. When '#' is omitted, it is handled as character string.</p> <p>When '#' is omitted, it is regarded character string data starting with the character "B".</p> <p>When the number that is not stored in the SetBitmap method, the print command is issued to the printer, but printing is not performed.</p>
Print top logo	ESC tL	Prints the top logo stored by the setLogo method.
Print bottom logo	ESC bL	Prints the bottom logo stored by the setLogo method.
Print stamp	ESC sL	Not supported.

Name	Data	Remarks
Feed lines	ESC #lF	<p>Feeds the paper forward by lines. The character '#' is replaced by an ASCII decimal string telling the number of lines to be fed. If '#' is omitted, then one line is fed. '#' supports the values from '1' to '255'.</p> <p>If print data is not presence, line feed operation is executed according to the amount of line feed, and if print data is presence, the height of the print data is fed. If the value specified for "#" exceeds 35.4 in (approx. 900 mm), the command is executed feeding the paper by 35.4 in (approx. 900 mm)</p> <p>In 90 degrees rotating to the left or to the right by RotatePrint Method, it prints next printing location after Returns of feed-assigned lines.</p>
Feed units	ESC #uF	<p>Feeds the paper forward by the units defined with MapMode. The character '#' is replaced by an ASCII decimal string telling the number of units to be fed. If '#' is omitted, then one unit is fed.</p> <p>MapMode = PTR_MM_DOTS(1) '#' supports the values from '1' to '127'. (If the value is smaller than '1', the command is not executed, and if the value is larger than '127', the command is executed regarding that '127' is assigned.)</p> <p>MapMode = PTR_MM_TWIPS(2) '#' supports the values from '1' to '903'. (If the value is smaller than '1', the command is not executed, and if the value is larger than '903', the command is executed regarding that '903' is assigned.)</p> <p>MapMode = PTR_MM_ENGLISH(3) '#' supports the values from '1' to '627'. (If the value is smaller than '1', the command is not executed, and if the value is larger than '627', the command is executed regarding that '627' is assigned.)</p> <p>MapMode = PTR_MM_METRIC(4) '#' supports the values from '1' to '1594'. (If the value is smaller than '1', the command is not executed, and if the value is larger than '1594', the command is executed regarding that '1594' is assigned.)</p> <p>The line feed setting of the printer does not affect the amount of line feed.</p> <p>It is executed on the halfway of the line, and when the specified amount of line feed is less than 1 line, then 1 line is fed.</p> <p>In 90 degrees rotating to the left or to the right by RotatePrint Method, it prints next printing location after Returns of feed-assigned lines.</p>
Feed reverse	ESC #rF	Not supported.

Name	Data	Remarks
Send embedded data	ESC #E	<p>The successive character string of "#E" is passed to the device without any change. The character '#' is replaced by an ASCII decimal string specifying the number of bytes following the escape sequence to be passed directly to the device. If '#' is omitted, it is not regarded as the escape sequence, and handled as print data.</p> <p>When the print data specified by '#' is not set after the escape sequence is specified, available print data is sent. (Example: When ESC 2E"a" is specified, only "a" is sent because the character string is set only for one byte.)</p> <p>In rotate printing 90 degrees to the left or to the right by the RotatePrint method, the data column specified by Send embedded data is not counted as the character string, the width cannot be calculated. Adjust the printing width by inserting empty space and so on.</p> <p>The maximum length of the character string that is supported is the maximum value of int (2147483647)</p> <p>The codes "80H" to "FFH" cannot be printed as properly.</p>
Barcode printing (Refer to the next page)	ESC #R	<p>Prints the barcode. The character '#' is replaced by an ASCII decimal string and the number of characters following the R to use in the definition of the characteristics of the barcode to be printed. See details below.</p> <p>The barcode may be printed during rotate printing 90 degrees to the left or to the right by the RotatePrint method, but printing may not be performed normally because the print area is not calculated by the specified barcode width. When the other character string data specified exceeds the barcode width, printing is executed.</p> <p>The available width that can be set by the parameter is up to the value of the RecLineWidth property and is not affected by the RotateSpecial property.</p>

The application can use the ESC|#R escape sequence to print barcodes. The character '#' is the number of characters following the R to use in the definition of the characteristics of the barcode to be printed.

With the character string following the R, the lower case letters and numbers are used to specify the characteristics of the barcode. As the value, the constant defined for the printBarCode method can be used.

The attribute symbols are defined as follows:

- s symbology (type of the barcode)
- h height (the height of the barcode)
- w width (the width of the barcode)
- a alignment (the position of the barcode)
- t human readable text position (the position of the HRI character string)
- d start of data

e end of data

The attributes must appear in the order specified in the above list. (It cannot be omitted)

Using a basic UPCA, center aligned, with bottom text, 200 dots height and 400 dots wide, the command is as follows:

ESC|33Rs101h200w400a-2t-13d123456789012e

Data can not contain non-ASCII characters. In that case please use the printBarCode method.

The followings are excerption of the definitions of the constants used in the example above from the header file.

```
public static final int PTR_BCS_UPCA = 101; // Digits
public static final int PTR_BC_CENTER = -2;
public static final int PTR_BC_TEXT_BELOW = -13;
```

In addition, the threshold of each parameter is as follows. When the threshold exceeded it, the barcode is not printed.

Barcode	Width(dot)	Height(dot)	Alignment
Except two dimension code	The most narrow width of the individual barcod – RecLineWidth	1 - 255	All values defined by PrintBarcode method are available.
PDF417	172 - RecLineWidth	12 - 831	All values defined by PrintBarcode method are available.
PDF417 During right and left 90 degrees turn by RotatePrint method.	172 - 831	12 – RecLineWidth	As well as an appointed value, all values become the left hotchpotch
QR	21 - RecLineWidth	1 – 16 (*1) (as width of the module)	All values defined by PrintBarcode method are available.
QR During right and left 90 degrees turn by RotatePrint method.	21 - RecLineWidth	1 – 16 (*1) (as width of the module)	As well as an appointed value, all values become the left hotchpotch
MicroQR	11 - RecLineWidth	1 – 16 (*1) (as width of the module)	All values defined by PrintBarcode method are available.
MicroQR During right and left 90 degrees turn by RotatePrint method.	11 - RecLineWidth	1 – 16 (*1) (as width of the module)	As well as an appointed value, all values become the left hotchpotch

*1 : When height parameter of QR is 17 or more, it is printed with module width 3.

2) Escape Sequence which operates during printing

It has characteristics that are remembered until explicitly changed

Name	Data	Contents
Font typeface selection	ESC #fT	Not supported.

3) Escape Sequence which operates at the time of printing

It has the characteristics that are reset at the end of each print method or by a "Normal" sequence."

Name	Data	Remarks
Bold	ESC (1)bC	Prints in bold. When "!" is specified, the bold is not valid.
Underline	ESC #uC	Prints with underline. The character '#' is replaced by an ASCII decimal string telling the thickness of the underline in printer dot units. Underlines in 1 dot and 2 dots are supported. If '#' is omitted, underline in 1 dot will be printed.
Italic	ESC (1)iC	Not supported.
Alternate color (Custom)	ESC #rC	Not supported.
Red	ESC rC	Prints with the second color of the receipt. Printing is possible only when "Printing Color" of Printer Setting is "Two Colors". When Color = mono is set in jpos.xml, printing is not affected by specifying this escape sequence.
Invert	ESC (1)rvC	Prints light and shade inverted. When "!" is specified, the bold is not valid.
Shading	ESC #sC	Not supported.
Strike lines	ESC stC	Strike lines are added and printed. When "!" is specified, the strike line is not valid.
Single high & wide	ESC 1C	Prints normal size.
Double wide	ESC 2C	Prints double-wide characters.
Double high	ESC 3C	Prints double-high characters.
Double high & wide	ESC 4C	Prints double-high/double-wide characters.
Scale horizontally	ESC #hC	Prints with the width scaled '#' times the normal size, where '#' is replaced by an ASCII decimal string. 1 to 8 times are supported. Prints in same size when '#' is omitted.
Scale vertically	ESC #vC	Prints with the height scaled '#' times the normal size, where '#' is replaced by an ASCII decimal string. 1 to 8 times are supported. Prints in same size when '#' is omitted.
RBG Color	ESC #fC	Not supported.

Name	Data	Remarks
Center	ESC cA	Aligns following text in the center. It is available only when it is specified at the beginning of the line. It is unavailable in 90 degrees rotating to the left or to the right by the RotatePrint method.
Right justify	ESC rA	Aligns following text at the right. It is available only when it is specified at the beginning of the line. It is unavailable in 90 degrees rotating to the left or to the right by the RotatePrint method.
Normal	ESC N	Restores printer characteristics to normal condition. Centering or Align Right cannot be cleared unless it is specified at the beginning of the line.
SubScript	ESC (l)tbC	Not supported.
SuperScript	ESC (l)tpC	Not supported.

3.3. Common Properties

The following sections describe the properties provided commonly to the POS printer.

There are two kinds of properties: Read-Only and Read/Write. For the property that is writable, R/W is added next to the property name.

Only when exception's *errorCode* has the special meaning, the description is provided.

CapCompareFirmwareVersion Property

Type	boolean
Remarks	If TRUE, then the Service/device supports comparing the version of the firmware in the physical device against that of a firmware file. This property is initialized to true by the open method.
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value:
Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapPowerReporting Property

Type	int
Remarks	Identifies the reporting capabilities of the device. It has one of the following values:
Value	Meaning
JPOS_PR_STANDARD (1)	The Device Service can determine and report two of the power states – OFF_OFFLINE (that is, off or offline)
	This property is initialized by the open method.
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value:
Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapStatisticsReporting Property

Type

boolean

Remarks

This property is initialized to **false** by the **open** method. Statistics reporting is not supported.

Errors

This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapUpdateFirmware Property

Type

boolean

Remarks

This property is initialized to **true** by the **open** method. The device's firmware can be updated.

Errors

This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapUpdateStatistics Property

Type

boolean

Remarks

This property is initialized to **false** by the **open** method. Statistics reporting is not supported.

Errors

This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CheckHealthText Property

Type	String
Remarks	<p>Holds the results of the most recent call to the checkHealth method. The following examples illustrate some possible diagnoses:</p> <ul style="list-style-type: none">- In case of Internal "Internal HCheck: Successful", "Internal HCheck: OFF/OFFLINE"- In case of External "External HCheck: Not Supported"- In case of Interactive "Interactive HCheck: Not Supported" <p>This property is initialized to an empty string ("") before the first call to the checkHealth method.</p>
Errors	<p>This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value:</p>

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

Claimed Property

Type	boolean
Remarks	<p>true: The device is claimed for exclusive access.</p> <p>false: The device is released for sharing with other applications.</p> <p>The value of the Claimed property is initialized to false by the open method.</p>
Errors	<p>This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value:</p>

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

DeviceControlDescription Property

Type	String
Remarks	<p>"JavaPOS POSPrinter Device Control" is set. (The value may vary depending on the DC in use.) Identifies the Device Control. It is a character string identifying the Device Control and the company that produced it and always readable.</p>
Errors	None.

DeviceControlVersion Property

Type	int								
Remarks	"1013XXX" is set. Holds the Control Object version number. Three version levels are specified, as follows: <table><thead><tr><th>Version Level</th><th>Description</th></tr></thead><tbody><tr><td>Major</td><td>The "millions" place. Holds the OPOS major version level.</td></tr><tr><td>Minor</td><td>The "thousands" place. Holds the OPOS minor version level. This is always set to 13 since this OPOS control conforms to OPOS version 1.13.</td></tr><tr><td>Build</td><td>The "units" place. Updated when corrections are made to the Control Object.</td></tr></tbody></table> This property is always readable. (XXX varies depending on the time the Control Object is distributed.)	Version Level	Description	Major	The "millions" place. Holds the OPOS major version level.	Minor	The "thousands" place. Holds the OPOS minor version level. This is always set to 13 since this OPOS control conforms to OPOS version 1.13.	Build	The "units" place. Updated when corrections are made to the Control Object.
Version Level	Description								
Major	The "millions" place. Holds the OPOS major version level.								
Minor	The "thousands" place. Holds the OPOS minor version level. This is always set to 13 since this OPOS control conforms to OPOS version 1.13.								
Build	The "units" place. Updated when corrections are made to the Control Object.								
Errors	None.								

PhysicalDeviceDescription

Type	String				
Remarks	"FP 1 Station Thermal POS Printer (C) 20xx Fujitsu Isotec." is set. It is a character string identifying the device and holds the device name and related information. This property is initialized by the open method.				
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value: <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>JPOS_E_CLOSED (101)</td><td>An attempt was made to access a closed device.</td></tr></tbody></table>	Value	Meaning	JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
Value	Meaning				
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.				

DeviceEnabled Property R/W

Type	boolean
Remarks	<p>true: The device is enabled and in an operational state. If changed to true, then the device is brought to an operational state.</p> <p>false: The device has been disabled. If it is changed to false, then the device is physically disabled. Before the device is used, application must set this property true.</p> <p>Also, while DeviceEnabled is true, Device Connection State (PowerReporting) is reported. This property is initialized to false by the open method.</p>
Errors	<p>This property throws a JposException.</p> <p>The exception's <i>ErrorCode</i> property will be one of the following values:</p>

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_NOHARDWARE (107)	POS Printer is OFF or OFFLINE or the cable is not connected. Clear the problem, and then execute the property again.
JPOS_E_FAILURE (111)	The connection to the device is failed. There is the possibility that the port specified does not exist.
JPOS_E_TIMEOUT (112)	Connection to the POS Printer could not be established. Or the POS Printer could not be replaced. There is the possibility of cover open or running out of paper.
JPOS_E_BUSY (113)	An error occurred during setting the property because the processing is in progress. Set the property after the processing is completed.

PhysicalDeviceName Property

Type	String
Remarks	<p>"FP 1 Station Thermal POSPrinter" is set.</p> <p>Holds the device name and related information.. This is a short version of PhysicalDeviceDescription and should be limited to 30 characters.</p> <p>This property is initialized by the open method.</p>
Errors	<p>This property throws a JposException.</p> <p>The exception's <i>ErrorCode</i> property will be the following value:</p>

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

FreezeEvents Property R/W

Type **boolean**

Remarks If **true**, events will not be delivered. Events will be enqueued until this property is set to **false**.
If **false**, the application allows events to be delivered. If some events have been held while events were frozen during **true** and all other conditions are correct for delivering the events, then changing this property to **false** will allow these events to be delivered. An application may choose to freeze events for a specific sequence of code where interruption by an event is not desirable.
This property is initialized to **false** by the **open** method.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

OutputID Property

Type **int**

Remarks Holds the identifier to identify the asynchronous request (call to the method which corresponds asynchronously when the **AsyncMode** property is set to **true**) uniquely.
When an output method successfully starts asynchronous or synchronous output, the JavaPOS Driver assigns an identifier to the request. When the asynchronous output completes, an **OutputCompleteEvent** will be enqueued with this output ID as a parameter.
The **OutputID** property is assigned to a value based on a circular numbering system among the values from 1 through 65535.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

PowerNotify Property R/W

Type **int**

Remarks Contains the type of power notification selection made by the application.
The value to indicate the power notification feature is one of the followings:

<u>Value</u>	<u>Meaning</u>
JPOS_PN_DISABLED (0)	The Driver will not provide any power notifications to the application. No power notification StatusUpdateEvents will be fired, and PowerState may not be set. (Default value)
JPOS_PN_ENABLED (1)	The Driver will fire power notification StatusUpdateEvents and update PowerState , when DeviceEnabled is set to true .

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

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

Errors This property throws a **JposException**.

The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	Either of the followings occurred. - The device was already enabled. - The setting value of the property was illegal.

PowerState Property

Type **int**

Remarks Specifies the current power condition of the device while PowerNotify is **JPOS_PN_DISABLED (0)**. The value to indicate the power condition is one of the followings:

<u>Value</u>	<u>Meaning</u>
--------------	----------------

JPOS_PS_UNKNOWN (2000)	The device's power state cannot be determined for one of the following reasons (Default value): PowerNotify = JPOS_PN_DISABLED (0) ; power notifications are disabled. DeviceEnabled = false ; Power state monitoring does not occur until the device is enabled.
-------------------------------	---

JPOS_PS_ONLINE (2001)	The device is powered on and ready.
------------------------------	-------------------------------------

JPOS_PS_OFF_OFFLINE (2004)	The device is not powered on or not connected to the system.
-----------------------------------	--

This property is initialized to **JPOS_PS_UNKNOWN (2000)** by the **open** method. When **PowerNotify** is set to **JPOS_PS_ENABLED (1)** and **DeviceEnabled** is **true**, then this property is updated as the power condition change is detected.

Errors This property throws a *JposException*.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
--------------	----------------

JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
----------------------------	--

DeviceServiceDescription Property

Type **string**

Remarks "FP POS Printer Device Service, (C) 20xx Fujitsu Isotec" is set.
This property is initialized by the **open** method.

Errors This property throws a *JposException*.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
--------------	----------------

JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
----------------------------	--

DeviceServiceVersion Property

Type **string**

Remarks "1013XXX" is set. Holds the Device Service version number. (XXX varies depending on the time the Device Service is distributed)
This property is initialized by the **open** method.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

State Property

Type **int**

Remarks Holds the current state of the driver.

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed driver. (Default)
JPOS_S_IDLE (2)	The driver is in a good state and is not busy.
JPOS_S_BUSY (3)	The driver is in a good state and is busy performing output.
JPOS_S_ERROR (4)	An error has been reported, and the application must recover the driver to a good state before normal I/O can resume.

This property is always readable.

Errors None.

3.4. Common Method

The following sections describe the methods provided commonly to the POS printer.

checkHealth Method

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

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

Value	Meaning
JPOS_CH_INTERNAL (1)	Performs an online check. The result will be set to the CheckHealth property as follows: - If the POS printer is connected to POS and power is on, "Internal HCheck: Successful" is specified. - If the POS printer is not connected to POS or power is not on, "Internal HCheck: OFF/OFFLINE" is specified.
JPOS_CH_EXTERNAL (2)	This parameter is not supported. "External HCheck: Not Supported" is specified for the CheckHealthText property.
JPOS_CH_INTERACTIVE (3)	This parameter is not supported. "Interactive HCheck: Not Supported" is specified for the CheckHealthText property.

Remarks This method is called to test the state of the device. The result of the method is stored in the **CheckHealthText** property. The **CheckHealthText** property is always synchronous.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed for exclusive access before it can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The specified <i>level</i> parameter is not supported.
JPOS_E_BUSY (113)	This operation cannot be performed while processing is in progress.

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 it is zero (0), then the method immediately returns the result even if acquisition of exclusive access fails/

If **JPOS_FOREVER (-1)** is set, the method waits as long as needed until exclusive access is satisfied.

Remarks

This method is called to request exclusive access to the device.
The POS printer cannot be used unless exclusive access is obtained.
When successful, the **Claimed** property is changed to **true**.

When the claim method is executed, the connection with the POS printer device is established, and it is confirmed to process it. If it is possible to process it, fixed data is requested, and the claim method finishes normally. Exclusive control by the claim method is effective only between application programs on same VM.

Errors

This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	An invalid <i>timeout</i> parameter was specified.
JPOS_E_TIMEOUT (112)	Another application has exclusive access to the device, and did not relinquish control before <i>timeout</i> milliseconds expired.

clearOutput Method

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

Remarks This method is called to clear all device output that is buffered by issuing the **printNormal**, **cutPaper**, **rotatePrint**, **printBarCode**, **printBitmap**, **transactionPrintdata** method asynchronously. Released the rotate mode or batch transaction print mode by the rotatePrint method or the transactionPrint method.
Any output error events that are enqueued – usually waiting for **FreezeEvents** to be set to **false** – are also cleared.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_FAILURE (111)	The device is accessed exclusively by other process.

close Method

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

Remarks Releases 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.
This method should not be excuted while the event is being processed (in the event handler).

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_BUSY (113)	Asynchronous processing is in progress.

compareFirmwareVersion Method

Syntax `void compareFirmwareVersion(String firmwareFileName, int[] result) throws JposException`

Parameter	Description
<i>FirmWareFileName</i>	Holds the name of the firmware file whose version is to be compared against the firmware version of the device.
<i>result</i>	Returns the result of the comparison. JPOS_CFV_FIRMWARE_OLDER (1) Indicates that the version of one or more of the firmware files is older than the firmware in the device. JPOS_CFV_FIRMWARE_SAME(2) Indicates that the versions of all of the firmware files are the same as the firmware in the device. JPOS_CFV_FIRMWARE_NEWER(3) Indicates that the version of one or more of the firmware files is newer than the firmware in the device. JPOS_CFV_FIRMWARE_UNKNOWN(5) Indicates that a relationship between the two firmware versions could not be determined

Remarks This method determines whether the version of the firmware contained in the specified file is newer than, older than, or the same as the version of the firmware in the POS printer.

The version of the firmware is represented by the value of first six successive numbers found in the FirmWareFileName parameter. For example, when "01rm012345.hex" is set in the FirmWareFileName parameter, "012345" is recognized as the version of the firmware file to be compared. The version of the POS printer can be acquired by sending the command to the POS printer every time this method is executed.

In the first step of comparison, first four places in the version acquired from the POS printer and first four places in the version of the firmware file are compared as the character strings. When those are different, JPOS_CFV_FIRMWARE_UNKNOWN(5) is set in the result parameter.

When the first four places are equal, then last two places are compared as the values. When the version of the firmware file is smaller, JPOS_CFV_FIRMWARE_OLDER (1) is set. When the version of the POS printer is smaller, JPOS_CFV_FIRMWARE_NEWER(3) is set. When those are equal, JPOS_CFV_FIRMWARE_SAME(2) is set.

Errors This method throws a JposException.
 The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL(106)	Specified argument is invalid as the firmware file name.
JPOS_E_FAILURE (111)	Comparison of the version was failed.

JPOS_E_BUSY (113)

This operation cannot be performed while processing is in progress.

directIO Method

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

Parameter	Description
<i>command</i>	Command number.
<i>data</i>	An array of one modifiable integer.
<i>object</i>	Additional data.

Remarks Function varies depending on *command*.

<i>command</i>	<i>Meaning</i>
JPOS_FIT_DIO_BIN(0)	Send to the binary data. Sends the data in specified bytes asynchronously or synchronously when <i>command</i> = 0 (JPOS_FIT_DIO_BIN), and <i>object</i> is set to java.io.ByteArrayOutputStream . <i>data</i> is unused.

JPOS_FIT_DIO_SET_QRERRORLV(20)
Specifies error correction level of QR code. Specify the value in the table below for *data [0]*. *object* is unused.

Value	Error correction level
0	Level L(7%)
1	Level M(15%)
2	Level Q(25%)
3	Level H(30%)

JPOS_FIT_DIO_GET_QRERRORLV(21)
The error correction level of the QR code is returned. Error correction level (0-3) of the QR code is set to *data[0]*. *object* is unused.

JPOS_FIT_DIO_SET_MICROQRERRORLV(22)
Specifies error correction level of Micro QR code. Specify the value in the table below for *data [0]*. *object* is unused.

Value	Error correction level
0	Level L(7%)
1	Level M(15%)
2	Level Q(25%)

JPOS_FIT_DIO_GET_MICROQRERRORLV(23)
The error correction level of the Micro QR code is returned. Error correction level (0-2) of the Micro QR code is set to *data[0]*. *object* is unused.

Errors

This method throws a `JposException`.

The exception's `ErrorCode` property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	This method is not supported.
JPOS_E_NOHARDWARE (107)	The POS printer is not connected to the system or is not powered on.
JPOS_E_BUSY (113)	This operation cannot be performed during the error (State = <code>JPOS_S_ERROR(4)</code>). This operation cannot be performed while processing is in progress.
JPOS_E_EXTENDED (114)	Extended error codes: <i>ErrorCodeExtended</i> = <code>JPOS_EPTR_COVER_OPEN (201)</code> : The POS printer cover is open. (Can be returned only if AsyncMode is false .) <i>ErrorCodeExtended</i> = <code>JPOS_EPTR_REC_EMPTY (203)</code> : The receipt station is out of paper. (Can be returned only if AsyncMode is false .) <i>ErrorCodeExtended</i> = <code>JPOS_FIT_EPTR_FATAL (20003)</code> : A non-recoverable error occurred. (Can be returned only if AsyncMode is false .)

ErrorCodeExtended =
 JPOS_FIT_EPTR_OVERHEAT (20006):
 The print head is overheated.
 (Can be returned only if **AsyncMode** is **false**.)

ErrorCodeExtended =
 JPOS_FIT_EPTR_CUTTERJAM (20008):
 A cutter jam error occurred.
 (Can be returned only if **AsyncMode** is **false**.)

open Method

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

The *logicalDeviceName* parameter specifies the device name to open.

Specify one of the following values according to interface of the POS printer to connect:

- FP-2000 Serial Interface	"FP2000SERPRT","FP2000SER2PRT"
- FP-2000 USB Interface	"FP2000USBPRT","FP2000USB2PRT"
- FP-2100 Serial Interface	"FP2100SERPRT","FP2100SER2PRT"
- FP-2100 USB Interface	"FP2100USBPRT","FP2100USB2PRT"
- FP-2200 Serial Interface	"FP2200SERPRT","FP2200SER2PRT"
- FP-2200 USB Interface	"FP2200USBPRT","FP2200USB2PRT"

Remarks This method is called to open a device.
 When the **open** method is successful, the **Claimed**, and **DeviceEnabled** properties and other properties are initialized.

Errors This method throws a JposException.
 The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_NOSERVICE (104)	A connection to the corresponding Device Service could not be established.
JPOS_E_ILLEGAL (106)	The Device Driver is already open.

release Method

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

Remarks This method is called to release exclusive access to the device.
 If the **DeviceEnabled** property is **true** and the device is an exclusive-use device, then the device is also disabled.
 This method should not be executed while the event is being processed (in the event handler)..

Errors This method throws a JposException.
 The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

JPOS_E_ILLEGAL (106)	Exclusive access to the device is not allowed for the application.
JPOS_E_BUSY (113)	This operation cannot be performed while processing is in progress.

resetStatistics Method

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

Remarks This method is not supported.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	This method is not supported.

retrieveStatistics Method

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

Remarks This method is not supported.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	This method is not supported.

updateFirmwareMethod

Syntax `void updateFirmware(String firmwareFileName) throws JposException`

<u>Parameter</u>	<u>Description</u>
<i>FirmwareFileName</i>	Specifies either the name of the file containing the firmware or a file containing a set of firmware files that are to be downloaded into the device..

Remarks This method updates the firmware of a device with the version of the firmware contained or defined in the file specified by the *FirmwareFileName* parameter regardless of whether that firmware's version is newer than, older than, or the same as the version of the firmware already in the device. When this method is invoked, the Service Object should check that the specified firmware file exists. If so, this method should return immediately and the remainder of the update firmware process should continue asynchronously. The Service Object should notify the application of the status of the update firmware process by firing *StatusUpdateEvents* with values of *JPOS_SUE_UF_PROGRESS*(2100) + an integer between 1 and 100 indicating the completion percentage of the update firmware process. For application convenience, the *StatusUpdateEvent* value *JPOS_SUE_UF_COMPLETE*(2200) is defined to be the same value as *JPOS_SUE_UF_PROGRESS*(2100) + 100. If an error is detected during the asynchronous portion of an update firmware process, one of the following *StatusUpdateEvents* will be fired: After downloading the firmware to the POS printer, when the firmware version acquired from the file name and the version acquired from the POS printer are compared (same processing as the *CompareFirmware* method). If inconsistency is found, *JPOS_SUE_UF_FAILED_DEV_OK*(2201) is notified instead of *JPOS_SUE_UF_COMPLETE*(2200).

<u>Value</u>	<u>Meaning</u>
<i>JPOS_SUE_UF_FAILED_DEV_OK</i> (2201)	The update firmware process failed but the device is still operational.

Errors This method throws a *JposException*.
The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
<i>JPOS_E_CLOSED</i> (101)	An attempt was made to access a closed device.
<i>JPOS_E_NOTCLAIMED</i> (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
<i>JPOS_E_DISABLED</i> (105)	This operation cannot be performed while the device is disabled.
<i>JPOS_E_NOEXIST</i>(109)	The file specified by <i>FirmwareFileName</i> does not exist
<i>JPOS_E_BUSY</i> (113)	This operation cannot be performed during the error (State = <i>JPOS_S_ERROR</i> (4)). This operation cannot be performed while processing is in progress.
<i>JPOS_E_EXTENDED</i>(114)	<i>ResultCodeExtended</i> = <i>POS_EFIRMWARE_BAD_FILE</i> (281):

The specified firmware file or files exist, but one or more are either not in the correct format or are corrupt.
(When the extension is other than ".hxl", this error occurs)

updateStatistics Method

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

Remarks This method is not supported.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	This method is not supported.

3.5. Specific Properties

AsyncMode Property R/W

Type **boolean**

Remarks **true:** The print methods **printNormal**, **cutPaper**, **rotatePrint**, **printBarCode**, **printBitmap**, **transactionPrint** and **directIO** will be performed asynchronously.

false: Those methods will be performed synchronously.

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

Errors This property throws a JposException.

 The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapCharacterSet Property

Type **int**

Remarks It shows printable character setting of the PS printer.

 This property has one of the following values

<u>Value</u>	<u>Meaning</u>
PTR_CCS_KANJI (11)	Character setting supports code page 932. It supports single-byte katakanas from 0xA1 to 0xDF, and all the ASCII characters from 0x20 to 0x7F. Also, it supports the Sift JIS Code characters which are defined by the first JIS standard level and the third JIS standard level.

For languages other than Japanese, the following code characters are supported.

 Korean : KS code characters

 Thai : TIS code characters

 Simplified Chinese : GB2312 code characters

 Traditional Chinese : BIG5 code characters

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

Errors This property throws a JposException.

 The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapCoverSensor Property

Type **boolean**

Remarks **true:** The POS printer has a "cover open" sensor.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapMapCharacterSet Property

Type **boolean**

Remarks **true:** The Device Service is able to map the characters to the character sets defined in the **CharacterSetList** property.

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRec2Color Property

Type **boolean**

Remarks **true:** The receipt can print dark plus an alternate color.

false: The receipt does not support printing dark plus an alternate color.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecBarCode Property

Type **boolean**

Remarks **true:** The receipt can print bar code.
This property is initialized by the **open** method.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecBitmap Property

Type **boolean**

Remarks **true:** The receipt can print bitmaps.
This property is initialized by the **open** method.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecBold Property

Type **boolean**

Remarks **true:** The receipt can print bold characters.
This property is initialized by the **open** method.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecCartridgeSensor Property

Type **int**

Remarks This property is "0". Receipt Cartridge monitoring sensors are not loaded.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecColor Property

Type **int**

Remarks This property is "0". The receipt does not support the capability to print in color.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecDhigh Property

Type **boolean**

Remarks **true**: The receipt can print double high characters.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecDwide Property

Type **boolean**

Remarks **true:** The receipt can print double wide characters.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecDwideDhigh Property

Type **boolean**

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

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecEmptySensor Property

Type **boolean**

Remarks **true:** The receipt has an out-of-paper sensor.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecItalic Property

Type **boolean**

Remarks **false:** The receipt cannot print italic characters.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecLeft90 Property

Type **boolean**

Remarks **true:** The receipt can print in a rotated 90 degree left mode.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecMarkFeed Property

Type **int**

Remarks **0:** The receipt does not have the mark sensed paper handling capability.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecNearEndSensor Property

Type	boolean				
Remarks	true: The receipt has a low paper sensor. false: The low paper sensor does not work. This property is initialized by the open method and it is set to either true or false according to the if prop name = PNEsense setting in in the XML file.				
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value: <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>JPOS_E_CLOSED (101)</td><td>An attempt was made to access a closed device.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
<u>Value</u>	<u>Meaning</u>				
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.				

CapRecPageMode Property

Type	boolean				
Remarks	false: The printer is not capable of supporting Page Mode for the receipt station. This property is initialized by the open method				
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value: <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>JPOS_E_CLOSED (101)</td><td>An attempt was made to access a closed device.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
<u>Value</u>	<u>Meaning</u>				
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.				

CapRecPapercut Property

Type	boolean				
Remarks	true: The receipt can perform paper cuts. This property is initialized by the open method.				
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value: <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>JPOS_E_CLOSED (101)</td><td>An attempt was made to access a closed device.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
<u>Value</u>	<u>Meaning</u>				
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.				

CapRecRight90 Property

Type **boolean**

Remarks **true:** The receipt can print in a rotated 90 degree right mode.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecRotate180 Property

Type **boolean**

Remarks **true:** The receipt can print in a rotated upside-down mode.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecStamp Property

Type **boolean**

Remarks **false:** The receipt does not have a stamp capability.

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

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapRecUnderline Property

Type **boolean**

Remarks **true:** The receipt can print underlined characters.
This property is initialized by the **open** method.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapTransaction Property

Type **boolean**

Remarks **true:** Batch processing of the POS Printer is valid.
This property is initialized by the **open** method.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CartridgeNotify Property R/W

Type **int**

Remarks Contains whether cartridge state notification is available.
This property is specified by the application.

<u>Value</u>	<u>Meaning</u>
PTR_CN_DISABLED(0)	The device will not provide any cartridge state notifications to the application. No cartridge state notification StatusUpdateEvents will be fired, and JrnCartridgeState , RecCartridgeState , and SlpCartridgeState may not be set.

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

Errors This property throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_ILLEGAL (106)	This property cannot be set.

CharacterSet Property R/W

Type int

Remarks It sets up the characters for printing.
This property is initialized when Device is first enabled after the **open** method.
One of the following values is set up in this property.

< **When the Language is English(Latin) or Japanese** >

Value	Meaning
101	Selects MIK character set.
102	Selects PC866 (Cyrillic #2) character set.
103	Selects Thai code 18.
437	Selects PC437 (USA: Standard Europe) Character Set.
850	Selects PC850 (Multilingual) Character Set.
851	Selects PC851 (Greece - obsolete) Character Set.
852	Selects PC852 (Latin2) Character Set.
857	Selects PC857 (Turkish) Character Set.
858	Selects PC858 (Euro) Character Set.
860	Selects PC860 (Portuguese) Character Set.
863	Selects PC863 (Canadian-French) Character Set.
864	Selects PC864 (Arabic without BOX DRAWINGS below 20) Character Set.
865	Selects PC865 (Nordic) Character Set.
866	Selects PC866 (Cyrillic #2) Character Set.
869	Selects PC869 (Greece) Character Set.
932	Selects Japanese Version Shift-JIS (Katakana) character set.
998	Sets up ASCII Character. It supports ASCII Characters from 0x20 to 0x7F.
1250	Selects WPC1250 Character Set.
1251	Selects WPC1251 Character Set.
1252	Selects WPC1252 Character Set.
1258	Selects Vietnam CharacterSet. (TCVN5712:1993VN1)
2859	Selects ISO8859-2 (1999 Latin Alphabet No.2) Character Set.
28597	Selects ISO8859-7 (1987 LatinGreek Alphabet) Character Set.

< **When the Language is not English(Latin) or Japanese** >

Value	Meaning
437	Selects PC437 (USA: Standard Europe) Character Set.
850	Selects PC850 (Multilingual) Character Set.
858	Selects PC858 (Euro) Character Set.
860	Selects PC860 (Portuguese) Character Set.
863	Selects PC863 (Canadian-French) Character Set.
865	Selects PC865 (Nordic) Character Set.
874	Thailand Windows Code Page; TISCODE.
936	China Windows Code Page; GB2312
949	Korea Windows Code Page; KSCODE.
950	Taiwan Windows Code Page; Big5.

998 Sets up ASCII Character. It supports ASCII Characters from 0x20 to 0x7F.

Errors This property throws a JposException.

The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_ILLEGAL (106)	The value other than above is specified.

CharacterSetList Property

Type String

Remarks Holds the character string of the character set number.
 In English(Latin) language, "101,102,103,437,850,851,852,857,858,860,863,864,865,866,869,932,998,1250,1251,1252,1258,28592,28597" are set up.
 In Korean language, "437,850,858,860,863,865,932,949,998" are set up.
 In Thai language, "437,850,858,860,863,865,874,932,998" are set up.
 In Simplified Chinese language, "437,850,858,860,863,865,932,936,998" are set up.
 In Traditional Chinese language, "437,850,858,860,863,865,932,950,998" are set up.
 One of these values is set to the CharacterSet property by Installer.
 This property is initialized by the **open** method.

Errors This property throws a JposException.

The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CoverOpen Property

Type **boolean**

Remarks **true:** The printer cover is open.

false: The printer cover is closed.

This property is initialized when the device is enabled and updated to the current value while the device is enabled.

Errors This property throws a JposException.

The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

ErrorLevel Property

Type **int**

Remarks Holds the severity of the error condition. It has one of the following values:

Value	Meaning
PTR_EL_NONE (1)	No error condition is present.
PTR_EL_RECOVERABLE (2)	A recoverable error has occurred (at the time of Cover Open Error, Receipt End Error, Head Hot Error, Cutter Jam Error or Power Discontinuity Error).
PTR_EL_FATAL (3)	A non-recoverable error has occurred. (only in case of fatal error).

This property is set just before delivering an **ErrorEvent** for asynchronous printing. When the error is cleared, then the property is changed to **PTR_EL_NONE (1)**.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

ErrorStation Property

Type **int**

Remarks Holds the POS printer (PTR_S_RECEIPT(2)) in printing when an error is detected.
This property is set up before ErrorEvent is notified.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

ErrorString Property

Type String

Remarks Holds the vender specific description of the current error.
This property is set by the driver just before delivering an **ErrorEvent** for asynchronous printing. If this description is not used, an empty string is set up in property. When the error is cleared, this property is changed to an empty string.
The following wordings are set up by the POS Printer

<u>Value</u>	<u>Meaning</u>
"Cover Open"	The printer cover is open.
"Paper End"	The station is out of paper
"Head Hot"	The print head is overheated.
"Fatal Error"	A non-recoverable error occurred.
"Cutter Jam Error"	A cutter jam error occurred.
"Power Off or Offline "	The power is off (offline).

Errors This property throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

FlagWhenIdle Property R/W

Type boolean

Remarks **true**: A **StatusUpdateEvent** will be enqueued when the device is in the idle state.
false: This event is not notified. If this status event is notified, **FlagWhenIdle** is automatically reset to **false**.

By utilizing Status Event with this property, Application can know the end of all the asynchronous output. When output ends successfully, or when output is deleted by the event handler which receives ErrorEvent, the event is notified.

If the **State** property is already JPOS_S_IDLE(2) when the **FlagWhenIdle** property is set to **true**, **StatusUpdateEvent** is immediately notified. Thus, the application can use this event without worrying about the time difference between asynchronous output end and setting up of this flag.

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

Errors This property throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

FontTypefaceList Property

Type	String
Remarks	An empty string is set. It indicates that only the default typeface is supported. This property is initialized by the open method.
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

MapCharacterSet Property

Type	BOOL MapCharacterSet;
Remarks	Indicates whether the device service map the character passed by the application for printing to the character set selected in the CharacterSet property. The driver map the character regardless whether true or false . This method is initialized to true by the open method.
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

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 height and line spacing.
It supports the following map modes. The values inside the parentheses are the values calculated in dots per each unit:

Value	Meaning
PTR_MM_DOTS (1)	The printer's dot width. (1 dot)
PTR_MM_TWIPS (2)	1/1440 of an inch. (7.0866 dots)
PTR_MM_ENGLISH (3)	0.001 inch. (4.921 dots)
PTR_MM_METRIC (4)	0.01 millimeter. (12.5 dots)

Setting **MapMode** may also change **RecLineHeight**, **RecLineSpacing** and **RecLineWidth**.

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

Errors This property throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_ILLEGAL (106)	An invalid mapping mode was specified.

PageModeArea Property

Type **String**

Remarks It is initialized to "" (empty string) when the **open** method is executed. The **pageModePrint** is not supported.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

PageModeDescriptor Property

Type **int**

Remarks It is initialized to 0 (zero) when the **open** method is executed. The **pageModePrint** is not supported.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

PageModeHorizontalPosition Property

Type **int**

Remarks It is initialized to 0 (zero) when the **open** method is executed. The **pageModePrint** is not supported.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_ILLEGAL (106)	An invalid value was used.

PageModePrintArea Property

Type **String**

Remarks It is initialized to "" (empty string) when the **open** method is executed. The **pageModePrint** is not supported.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_ILLEGAL (106)	An invalid value was specified.

PageModePrintDirection Property

Type	int
Remarks	It is initialized to 0 (zero) when the open method is executed. The pageModePrint is not supported.
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_ILLEGAL (106)	An invalid value was specified.

PageModeStation Property

Type	int
Remarks	It is initialized to 0 (zero) when the open method is executed. The pageModePrint is not supported.
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_ILLEGAL (106)	An invalid value was specified.

PageModeVerticalPosition Property

Type	int
Remarks	It is initialized to 0 (zero) when the open method is executed. The pageModePrint is not supported.
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_ILLEGAL (106)	An invalid value was specified.

RecBarcodeRotationList Property

Type String

Remarks This character string shows the direction in which a receipt bar-code may be rotated. "0, R90, L90, 180" can be set.

This property is initialized by the **open** method. The character strings consist of groups of character strings separated by commas, and indicating rotation direction. The following show character strings which indicate rotation direction.

<u>Value</u>	<u>Meaning</u>
0	Barcode may be printed in the normal orientation.
R90	Barcode may be rotated 90° to the right.
L90	Bar code may be rotated 90° to the left.
180	Barcode may be rotated 180° - upside down.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

RecBitmapRotationList Property

Type String

Remarks This character string shows the directions in which a receipt bitmap may be rotated. "0, R90, L90, 180" can be set.

This property is initialized by the **open** method. The string consists of rotation strings separated by commas. The following show character strings which indicate rotation direction:

<u>Value</u>	<u>Meaning</u>
0	Bitmap may be printed in the normal orientation.
R90	Bitmap may be rotated 90° to the right. (Not supported)
L90	Bitmap may be rotated 90° to the left. (Not supported)
180	Bitmap may be rotated 180° - upside down. (Not supported)

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

RecCartridgeState Property

Type **int**

Remarks Indicates the status of the currently selected Receipt cartridge (ink, ribbon or toner).
Since the POS printer is the thermal printer, this is fixed to the following value..

<u>Value</u>	<u>Meaning</u>
--------------	----------------

PTR_CART_UNKNOWN (268435456)	
-------------------------------------	--

The device does not support the feature of notifying the cartridge state.

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

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
--------------	----------------

JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
----------------------------	--

RecCurrentCartridge Property R/W

Type **int**

Remarks Selection of the receipt cartridge is not supported. It is initialized to 0.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
--------------	----------------

JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
----------------------------	--

JPOS_E_ILLEGAL(106)	Specifying cartridge state is invalid.
----------------------------	--

RecEmpty Property

Type **boolean**

Remarks **true:** The receipt is out of paper.
false: The receipt paper is present.

This property is initialized when the device is enabled and updated to the current value while the device is enabled.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
--------------	----------------

JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
----------------------------	--

RecLetterQuality Property R/W

Type	boolean
Remarks	true: Prints in normal printing mode. false: Prints in high speed printing mode.

This property is initialized to **true** when the device is first enabled after the **open** method.

High quality printing mode affects the built-in characters and down-load characters. And at the same time, in case of double-width and double-height characters, it can print with smoothing processing, but it prints a little bit slower.

In case of normal printing mode, bitmap is printed in 1/3 resolution. (The inputted size is same as that of high quality printing mode but its resolution is 1/3. Also, in case of printing of double-width-and-double-height built-in characters and larger than that, smoothing processing is not done.

When the bitmap is registered by **SetBitmap**, it is not affected by the **RecLetterQuality** at that time. (If the bitmap is registered by **SetBitmap**, printing the bitmap centered or aligned right in the normal print mode results printing position incorrect. In such case, it is recommended to print in high-quality print mode.)

In case of bitmap printing in Escape Sequence, in high quality printing mode, printing is in normal resolution, and in normal printing mode, printing is in 1/2 resolution. (double-width and double-height) The method follows **RecLetterQuality** in the same way.

(*When Smoothing setting is off in the jpos.xml settings, it does not do smoothing processing even when it is set to **true**.)

Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value:
---------------	---

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

RecLineChars Property R/W

Type **boolean**

Remarks It is the number of the half-size characters, per line of receipts. This property is initialized to one of the following values by the open method according to the setting.
Printing is done in the following font, according to the assigned number of half-size characters per line of receipt.

When 180 dpi mode is disabled, it is as follows.

<u>Characters per Line</u>	<u>Printing Font (WidthxHeight)</u>
-----------------------------------	--

Printer Setting: Paper width 80 mm (576 dots)

48 (double-width 24)	12x24 dots (font A)
57 (double-width 28)	10x24 dots (font B)
72 (double-width 36)	8x16 dots (font C) * Only English(Latin) language.

Printer Setting: Paper width 80 mm (512 dots)

42 (double-width 21)	12x24 dots (font A)
51 (double-width 25)	10x24 dots (font B)
64 (double-width 32)	8x16 dots (font C) * Only English(Latin) language.

Printer Setting: Paper width 58 mm (420 dots)

35 (double-width 17)	12x24 dots (font A)
42 (double-width 21)	10x24 dots (font B)
52 (double-width 26)	8x16 dots (font C) * Only English(Latin) language.

Printer Setting: Paper width 58 mm (384 dots)

32 (double-width 16)	12x24 dots (font A)
38 (double-width 19)	10x24 dots (font B)
48 (double-width 24)	8x16 dots (font C) * Only English(Latin) language.

Printer Setting: Paper width 50 mm (360 dots)

30 (double-width 15)	12x24 dots (font A)
36 (double-width 18)	10x24 dots (font B)
45 (double-width 22)	8x16 dots (font C) * Only English(Latin) language.

When 180 dpi mode is enabled, it is as follows.

Characters per Line	Printing Font (WidthxHeight)
Printer Setting: Paper width 80 mm (512 dots)	
42 (double-width 21)	12x24 dots (font A)
51 (double-width 25)	10x24 dots (font B)
64 (double-width 32)	8x16 dots (font C) * Only English(Latin) language.

Printer Setting: Paper width 58 mm (360 dots)	
30 (double-width 15)	12x24 dots (font A)
36 (double-width 18)	10x24 dots (font B)
45 (double-width 22)	8x16 dots (font C) * Only English(Latin) language.

If this value is changed to supported line character width, the character width is set up to the assigned value. If it cannot support exact width, it is set up to the nearest value in supported line width and at the same time larger value than supported line width. (For example, when to set up paper width to 83 mm and to set 40 for Printer, Device Service will select character size of "53".) If it cannot support character width, Error will return.

Setting **RecLineChars** may also update the **RecLineHeight**, **RecLineSpacing**, **RecSideWayMaxChars** and **RecSidewaysMaxlinesa** properties.

Errors

This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL(106)	Illegal line character width is specified.

RecLineCharsList Property

Type **String**

Remarks Holds the character string including the line character widths supported by the receipt station.
This property is initialized to one of the following values by the open method according to the paper width and the setting.

When 180 dpi mode is disabled, it is as follows.

Printer Paper Width	Value
Paper Width 80mm (576 dots)	"48,57,72(*)"
Paper Width 80mm (512 dots)	"42,51,64(*)"
Paper Width 58mm (420 dots)	"35,42,52(*)"
Paper Width 58mm (384 dots)	"32,38,48(*)"
Paper Width 50mm (360 dots)	"30,36,45(*)"

When 180 dpi mode is enabled, it is as follows.

Printer Paper Width	Value
Paper Width 80mm (512 dots)	"42,51,64(*)"
Paper Width 58mm (360 dots)	"30,36,45(*)"

* Only English(Latin) language.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

RecLineHeight Property R/W

Type **int**

Remarks Holds the receipt print line height, expressed in the unit of measure given by **MapMode**.
When **RecLineChars** is changed, **RecLineHeight** is updated to the default line height for the selected width.
The value of **RecLineHeight** is initialized by the open method to the default line height for the POS Printer. The following are the applicable values, when 180 dpi mode is disabled. (the value of the property is when the MapMode property is PTR_MM_DOTS(1)):

<u>Characters per Line</u>	<u>Value of the RecLineHeight Property</u>
Printer Setting: Paper width 80 mm (576 dots)	
48	24
57	24
72	16 * Only English(Latin) language.
Printer Setting: Paper width 80 mm (512 dots)	
42	24
51	24
64	16 * Only English(Latin) language.
Printer Setting: Paper width 58 mm (420 dots)	
35	24
42	24
52	16 * Only English(Latin) language.
Printer Setting: Paper width 58 mm (384 dots)	
32	24
38	24
48	16 * Only English(Latin) language.
Printer Setting: Paper width 50 mm (360 dots)	
30	24
36	24
45	16 * Only English(Latin) language.

When 180 dpi mode is enabled, it is as follows.

Characters per Line	Value of the RecLineHeight Property
Printer Setting: Paper width 80 mm (512 dots)	
42	24
51	24
64	16 * Only English(Latin) language.
Printer Setting: Paper width 58 mm (360 dots)	
30	24
36	24
45	16 * Only English(Latin) language.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	This property cannot be accessed. It can be only acquired

RecLineSpacing Property R/W

Type int

Remarks Holds the spacing of each single-high print line, including both the printed lineheight plus the whitespace between each pair of lines. This property is expressed in the unit of measure given by **MapMod**.
If the value of **RecLineHeight** is larger than the value specified for **RecLineSpacing** after **RecLineChars** is changed, this property is set to the same value as **RecLineHeight**.
RecLineSpacing is initialized to the default spacing of the POS Printer after the **open** method.
The available range of the value is from 16 (dot) through 127 (dot).

Errors JposException may be thrown.
ErrorCode for the exception will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	The range set for the property is invalid.

RecLinesToPaperCut Property

Type int

Remarks It holds the number of lines that must be advanced before the receipt paper is cut.
This is the line count before reaching the paper cut mechanism.

Changing the properties **RecLineChars**, **RecLineHeight**, and **RecLineSpacing** may cause this property to change.

Errors

JposException may be thrown.

ErrorCode for the exception will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

RecLineWidth Property

Type **int**

Remarks Holds the width of a line of **RecLineChars** characters. This property is expressed in the unit of measure given by **MapMode**.

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

It has one of the following values according to the paper width:

When 180 dpi mode is disabled, it is as follows.

Paper Width	Value
Paper width 80 mm	576
Paper width 80 mm	512
Paper width 58 mm	420
Paper width 58 mm	384
Paper width 50 mm	360

When 180 dpi mode is enabled, it is as follows.

Paper Width	Value
Paper width 80 mm	512
Paper width 58 mm	360

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

RecNearEnd Property

Type **boolean**

Remarks **true:** The receipt paper is low.

false: The receipt paper is not low.

This property is initialized when device enabled, and the current value is kept while it enabled.

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

RecSidewaysMaxChars Property

Type **int**

Remarks Holds the maximum number of half-width characters that may be printed on each line in sideways mode. It has one of the following values. Since the width of 90 degrees rotating to the left or to the right is declined to half when the **CapRec2Color** property is **true** (2 colors), the number of printable characters is half of the value.

<u>Characters per Line</u>	<u>Characters per Line in Sideways Mode</u>	<u>2 Colors</u>
Printer Setting: Paper width 80 mm (576 dots)		
48 (double-width 24)	138	69
57 (double-width 28)	166	83
72 (double-width 36) *	207	103
Printer Setting: Paper width 80 mm (512 dots)		
42 (double-width 21)	138	69
51 (double-width 25)	166	83
64 (double-width 32) *	207	103
Printer Setting: Paper width 58 mm (420 dots)		
35 (double-width 17)	138	69
42 (double-width 21)	166	83
52 (double-width 26) *	207	103
Printer Setting: Paper width 58 mm (384 dots)		
32 (double-width 16)	138	69
38 (double-width 19)	166	83
48 (double-width 24) *	207	103
Printer Setting: Paper width 50 mm (360 dots)		
30 (double-width 15)	138	69
36 (double-width 18)	166	83
45 (double-width 22) *	207	103

* Only English(Latin) language.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

RecSidewaysMaxLines Property

Type **int**

Remarks Holds the maximum number of lines that may be printed on each line in sideways mode.

The value of this property is obtained when the value of **RecLineWidth** is divided by the value of **RecLineSpacing**. If the remainder is equal to or greater than the value of **RecLineHeight** (font height), the value of this property is the sum of the remainder and one (1). Thus, changing **RecLineSpacing** may cause this property to change.

If the printing font is font C (refer to the **RecLineChars** property), then **RecLineWidth** is 7 (dot) to calculate the value of this property according to the conditions above.

This property is initialized when Device is enabled for the first time after Open Method.

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

RotateSpecial Property R/W

Type **int**

Remarks It shows the rotation orientation for bar codes.
This property is initialized to PTR_RP_NORMAL(1) by the **open** method.
This property has one of the following values:

Value	Meaning
PTR_RP_NORMAL (1)	Prints subsequent bar codes in normal orientation.
PTR_RP_RIGHT90 (2)	Rotate printing 90 degree to the right.
PTR_RP_LEFT90 (3)	Rotate printing 90 degree to the left.
PTR_RP_ROTATE180 (259)	Rotate printing 180 degree, that is, print upside-down.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	An invalid parameter value was used.

*The following specific POS printer properties are not supported.

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

3.6. Specific Methods

beginInsertion Method

- Syntax** `void beginInsertion (int timeout) throws JposException;`
- Remarks** Because this method is only applicable for the Slip Printers, this is not supported by this driver.
- Errors** This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The POS printer does not have the slip

beginRemoval Method

- Syntax** `void beginRemoval (int timeout) throws JposException;`
- Remarks** Because this method is only applicable for the Slip Printers, this is not supported by this driver.
- Errors** This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The POS printer does not have the slip

changePrintSide Method

- Syntax** `void changePrintSide (int side) throws JposException;`
- Remarks** Because this method is only applicable for the Slip Printers, this is not supported by this driver.
- Errors** This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The POS printer does not have the slip

clearPrintArea Method

- Syntax** `void clearPrintArea() throws JposException`
- Remarks** This method is not supported by this driver.
- Errors** This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The POS printer does not have the slip

cutPaper Method

- Syntax** **void cutPaper (int *percentage*) throws JposException;**
- Percentage Parameter indicates the percentage of the paper to be cut. When the value is between '1' to '99', partial cutting is performed. When the value is '100', full cutting is performed.
- When the value is other than '1' to '100', JPOS_E_ILLEGAL(106) is returned.
- Remarks** This method is called when to cut receipt paper.
- This method is executed synchronously if **AsyncMode** is **false** and asynchronously if **AsyncMode** is **true**. When the **printNormal** method or **printImmediate** method is called, paper cutting can be done using Escape Sequence of paper cutting, too. In addition to that, if POS Printer has buffered data (even though printing is requested, POS Printer does not print), it cannot cut paper. In order to cut receipt paper, it must be at the head of each line.
- Errors** This method throws a JposException.
- The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	Illegal percentage is assigned.
JPOS_E_NOHARDWARE (107)	The POS printer is not connected to the system or is not powered on.
JPOS_E_FAILURE(111)	The POS printer is in error state. Execute the method after the error state is cleared.
JPOS_E_TIMEOUT(112)	Sending data to the POS Printer was timed out, or completion of printing could not be confirmed after time out.
JPOS_E_BUSY (113)	This operation cannot be performed while output is in progress.
JPOS_E_EXTENDED (114)	Extended error codes: <code>ErrorCodeExtended =</code> JPOS_EPTR_COVER_OPEN (201): The POS Printer cover is open. (Can be returned only if AsyncMode is false .) <code>ErrorCodeExtended =</code> JPOS_EPTR_REC_EMPTY (203): The receipt station is out of paper. (Can be returned only if AsyncMode is false .) <code>ErrorCodeExtended =</code> JPOS_FIT_EPTR_FATAL (20003): A non-recoverable error occurred. (Can be returned only if AsyncMode is false .)

ErrorCodeExtended =

JPOS_FIT_EPTR_OVERHEAT (20006):

The print head is overheated.

(Can be returned only if **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_FIT_EPTR_CUTTERJAM (20008):

A cutter jam error occurred.

(Can be returned only if **AsyncMode** is **false**.)

endInsertion Method

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

Remarks Because this method is only applicable for the Slip Printers, this is not supported by this driver.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The POS printer does not have the slip

endRemoval Method

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

Remarks Because this method is only applicable for the Slip Printers, this is not supported by this driver.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The POS printer does not have the slip

markFeed Method

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

Remarks The *type* parameter indicates the type of mark sensed paper handling.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The receipt print station does not support the capability of mark sensed paper handling. (Refer to the CapRecMarkFeed property)

pageModePrint Method

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

The control parameter has one of the following values:

Value	Description
PTR_PM_PAGE_MODE(1)	Starts the page mode.
PTR_PM_NORMAL(3)	Prints printing data within the print area in the page mode, deletes printing data, and then exits the page mode.
PTR_PM_CANCEL(4)	Deletes printing data within the print area in the page mode, and exits the page mode without printing.

Remarks It is not supported by this driver.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The POS printer does not have the slip

printBarcode Method

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

Parameter	Description
<i>station</i>	PTR_S_RECEIPT(2) is assigned.
<i>data</i>	The character string of the bar code.
<i>symbology</i>	Bar code symbol type to use. (Refer to the following values.)
<i>height</i>	Bar code height. Expressed in the unit of measure given by MapMode . Possible values are 1 to 255 dots. In case of PDF417 printing with upright/upside-down, it is possible to set from 1 to 831(dot). Or, in case of 90 degree rotation to the right or to the left, it is possible to set from 12 to the value of RecLineWidth Property (dot). Specify the range of 1-16 for width of the module for the QR code and micro QR code of two dimension bar code. OPOS_E_ILLGAL(106) is notified when other values are specified.
<i>width</i>	Bar code width. Expressed in the unit of measure given by MapMode . For normal orientation and upside-down printing, the value can be set to the value set by RecLineWidth .
<i>alignment</i>	Placement of the bar code. Refer to the values below.
<i>textPosition</i>	Placement of the readable character string. Refer to the values below.

The *symbology* parameter in this release has one of the following values:

Value	Label Type
PTR_BCS_UPCA(101)	UPC-A
PTR_BCS_UPCE(102)	UPC-E
PTR_BCS_EAN8(103)	EAN 8 (= JAN 8)
PTR_BCS_JAN8(103)	JAN 8 (= EAN 8)
PTR_BCS_EAN13(104)	EAN 13 (= JAN 13)
PTR_BCS_JAN13(104)	JAN 13 (= EAN 13)
PTR_BCS_ITF(106)	Interleaved 2 of 5
PTR_BCS_Codabar(107)	Codabar(NW-7)
PTR_BCS_Code39(108)	Code 39
PTR_BCS_Code93(109)	Code 93
PTR_BCS_Code128(110)	Code 128
PTR_BCS_PDF417(201)	PDF417 (two dimension barcode)
PTR_BCS_QRCODE(204)	QR code (two dimension barcode)
PTR_BCS_UQRCODE(205)	micro QR code (two dimension barcode)

The *alignment* parameter has one of the following values:

Value	Meaning
PTR_BC_LEFT (-1)	Aligns with the left-most print column. (Since alignment is applied to the print data before the rotation, for upside-down printing, the bar code is aligned right with print direction of the POS Printer.)
PTR_BC_CENTER (-2)	Aligns in the center of the station. In the case of two dimension barcode, this parameter is not supported in 90 degrees rotating to the left or to the right. it works as PTR_BC_LEFT(-1).
PTR_BC_RIGHT (-3)	Aligns with the right-most print column. (Since alignment is applied to the print data before the rotation, for upside-down printing, the bar code is aligned left with print direction of the POS Printer.) In the case of two dimension barcode, this parameter is not supported in 90 degrees rotating to the left or to the right. It works as PTR_BC_LEFT(-1).

Other Values

Distance from the left-most print column to the start of the bar code. Expressed in the unit of measure given by **MapMode**. If the sum of the actual width of bar code that is calculated from the specified bar code width by the *width* parameter and the distance from the left-most print column exceeds the value of **RecLineWidth**, then the application returns JPOS_E_ILLEGAL (106). In this case, if **RotateSpecial** is PTR_RP_RIGHT90 (257) or PTR_RP_LEFT90 (258), the application regards that this parameter is set to PTR_BC_LEFT (-1) and performs printing.

The *textPosition* parameter has one of the following values:

Value	Meaning
PTR_BC_TEXT_NONE (-11)	No text is printed. Only prints the bar code.
PTR_BC_TEXT_ABOVE (-12)	Prints the text above the bar code.
PTR_BC_TEXT_BELOW (-13)	Prints the text below the bar code.

Remarks

This method is called when to print bar codes with the assigned POS Printer.
This method is executed synchronously if **AsyncMode** is **false**, and asynchronously if **AsyncMode** is **true**.

Following are printable bar code conditions per each *symbology*. On rotation printing, the range of available values differs depending on whether 2 colors printing (**CapRec2Color** = TRUE) is available.

Symbology	Each printable character kind	Upright/Upside-down mode		90 degree rotation to the left/right	
		Character string length	Width (dots)	Character string length	Width (dots)
PTR_BCS_UPCA	10 kinds ('0'-'9')	11-12	95- RecLineWidth Value	11-12	95-1662
PTR_BCS_UPCE		11-12	51- RecLineWidth Value	11-12	51-1662
PTR_BCS_JAN8		7-8	67- RecLineWidth Value	7-8	67-1662
PTR_BCS_JAN13		12-13	95- RecLineWidth Value	12-13	95-1662
PTR_BCS_CODE39	43 kinds ('0'-'9', 'A'-'Z', space, '\$', '%', '+', '-', ':', '/') (Start/Stop character of '*' is automatically added.)	1-34	47- RecLineWidth Value	1.131	47-1662
PTR_BCS_ITF	10 kinds ('0'-'9')	2-62	27- RecLineWidth Value	2-182	27-1662
PTR_BCS_CODABE R	20 kinds ('0'-'9', 'A'-'D', '\$', '+', '-', ':', '/', ':')	3-47	41- RecLineWidth Value	3-138	41-1662
PTR_BCS_CODE93	128 kinds (0x00-0x7F) (Lower stage is for two characters)	1-59	46- RecLineWidth Value	1-88	46-1662
		1-29		1-44	
PTR_BCS_CODE128	Code Set A: 0x00 - 0x5F Code Set B 0x20 - 0x7F Code Set C 0x00 - 0x63 However the characters including "{" are exception. For details refer to later.	3-51	46- RecLineWidth Value	3-74	46-1662

Symbology	Each printable character kind	Upright/Upside-down mode		90 degree rotation to the left/right	
		Character string length	Width (dots)	Character string length	Width (dots)
PTR_BCS_PDF417	256 kinds including 0x00 to 0xFF. The character strings 0x00 to 0x7F conform to the ASCII code, and 0x80 to 0xFF conform to the extended character sets in the English table of PC437 (USA: Standard Europe)	1-1069	172-RecLineWidth Value	1-1069	172-831
PTR_BCS_QRCODE	Figure ('0'~'9'), Capital letter ('A' ~ 'Z'), Special sign (space, '\$', '%', '*', '+', '-', '.', '/', ':'), Binary (0x00~0xFF)	1-7089	21-RecLineWidth Value	1-7089	21-RecLineWidth Value
PTR_BCS_QRCODE	Figure ('0'~'9'), Capital letter ('A'~'Z'), Special sign (space, '\$', '%', '*', '+', '-', '.', '/', ':'), Binary (0x00~0xFF)	1-35	11-RecLineWidth Value	1-35	11-RecLineWidth Value

Following is printing width decision algorithm for each bar code. As for final printing width (dot), printing is done with nearest value not exceeding the value assigned by Width Parameter of PrintBarcode, in changing parameters.

symbology	Formula for calculating printing width
PTR_BCS_UPCA	Bar code width = 95 * dotNarrow
PTR_BCS_UPCE	Bar code width = 51 * dotNarrow
PTR_BCS_JAN8	Bar code width = 67 * dotNarrow
PTR_BCS_JAN13	Bar code width = 95 * dotNarrow
PTR_BCS_CODE39	Bar code width = $6 * \text{dotNarrow} + 3 * \text{dotWide} + 1 * \text{dotNarrow} +$ $(6 * \text{dotNarrow} + 3 * \text{dotWide} + 1 * \text{dotNarrow}) * \text{Length} +$ $6 * \text{dotNarrow} + 3 * \text{dotWide}$ (Length = Number of characters printed)

symbology	Formula for calculating printing width
PTR_BCS_UQRCODE	<p>Bar code width = Number of cells of vertical directions * Module width (1 – 16)</p> <p>Bar code height = Horizontal number of cells * Module width (1 – 16)</p> <p>* Width and height of the bar code are set to the maximum size that doesn't exceed the value of Width.</p>

*Relation between dotNarrow and dotWide

dotNarrow	1	2	3	4	5	6
dotWide	3	5	9	11	14	18

Notes for Bar Code Printing

1. When to print CODE39, "*" (Start/Stop Character) is automatically added. So, there is no need of setting up in Character.
2. When to assign ITF, even-number character must be assigned. If odd-number is assigned, JPOS_E_ILLEGAL(106) will return.
3. When to assign CODABER, the head and the tail of the characters must be among "A" - "D". Accordingly, three or more than three characters (the head character plus any characters plus the tail character) must be assigned. In the other cases, JPOS_E_ILLEGAL(106) returns.
4. When to assign UPC-E, development is done according to the following list. UPC-A Left Code shows top characters (2-6), UPC-A Right Code shows 7th-11th characters. The shortened code is actually printed as UPC-E. If the UPC-A top character assigned is except 0 or, characters not based on the following list is assigned, JPOS_E_ILLEGAL(106) returns

Example 05820000226 -> Converted to c58226.
 09859363583 -> JPOS_E_ILLEGAL returns.

Manufacturer Code Left Code for UPC-A					Item Code Right Code for UPC-A					Shortened 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

5. When to print CODE128, set up characters as followed.
 1. One of "{A", "{B", "{C" must be assigned as the head of the bar code. Following that, each of CODE A, CODE B, CODE C must be set up.
 2. When to assign Function Code, assign "{1", "{2", "{3", or "{4". Each is to assign FNC1, FNC2, FNC3, or FNC4. For further information, in CODE C, only FUNC1 is available. If you assign except FUNC1 in CODE C, JPOS_E_ILLEGAL(106) returns.
 3. When to print "{" in CODE B, assign "{{".
 4. When to set up SHIFT, assign "{S". After that, code set of one character sifts like CODE A <- -> CODE B. If you assign in CODE C, JPOS_E_ILLEGAL(106) returns.

Following are printable character in CODE A, CODE B, CODE C.

[Code128]

Character to Print			Character to Print		
CODE-A	CODE-B	CODE-C	CODE-A	CODE-B	CODE-C
SPACE	SPACE	00(00H)	U	U	53(35H)
!	!	01(01H)	V	V	54(36H)
"	"	02(02H)	W	W	55(37H)
#	#	03(03H)	X	X	56(38H)
\$	\$	04(04H)	Y	Y	57(39H)
%	%	05(05H)	Z	Z	58(3AH)
&	&	06(06H)	[[59(3BH)
'	'	07(07H)	/	/	60(3CH)
((08(08H)]]	61(3DH)
))	09(09H)	^	^	62(3EH)
*	*	10(0AH)	_	_	63(3FH)
+	+	11(0BH)	NULL(00H)	`	64(40H)
,	,	12(0CH)	SOH(01H)	a	65(41H)
-	-	13(0DH)	STX(02H)	b	66(42H)
.	.	14(0EH)	ETX(03H)	c	67(43H)
/	/	15(0FH)	EOT(04H)	d	68(44H)
0	0	16(10H)	ENG(05H)	e	69(45H)
1	1	17(11H)	ACK(06H)	f	70(46H)
2	2	18(12H)	BEL(07H)	g	71(47H)
3	3	19(13H)	BS(08H)	h	72(48H)

Character to Print			Character to Print		
CODE-A	CODE-B	CODE-C	CODE-A	CODE-B	CODE-C
4	4	20(14H)	HT(09H)	i	73(49H)
5	5	21(15H)	LF(0AH)	j	74(4AH)
6	6	22(16H)	VT(0BH)	k	75(4BH)
7	7	23(17H)	FF(0CH)	l	76(4CH)
8	8	24(18H)	CR(0DH)	m	77(4DH)
9	9	25(19H)	SO(0EH)	n	78(4EH)
:	:	26(1AH)	SI(0FH)	o	79(4FH)
;	;	27(1BH)	DLE(10H)	p	80(50H)
<	<	28(1CH)	DC1(11H)	q	81(51H)
=	=	29(1DH)	DC2(12H)	r	82(52H)
>	>	30(1EH)	DC3(13H)	s	83(53H)
?	?	31(1FH)	DC4(14H)	t	84(54H)
@	@	32(20H)	NAK(15H)	u	85(55H)
A	A	33(21H)	SYN(16H)	v	86(56H)
B	B	34(22H)	ETB(17H)	w	87(57H)
C	C	35(23H)	CAN(18H)	x	88(58H)
D	D	36(24H)	EM(19H)	y	89(59H)
E	E	37(25H)	SUB(1AH)	z	90(5AH)
F	F	38(26H)	ESC(1BH)	{ "{"	91(5BH)
G	G	39(27H)	FS(1CH)		92(5CH)
H	H	40(28H)	GS(1DH)	}	93(5DH)
I	I	41(29H)	RS(1EH)	~	94(5EH)
J	J	42(2AH)	US(1FH)	DEL	95(5FH)
K	K	43(2BH)			96(60H)
L	L	44(2CH)			97(61H)
M	M	45(2DH)			98(62H)
N	N	46(2EH)			99(63H)
O	O	47(2FH)	Following are used by assigning "{"		
P	P	48(30H)	FNC 3 "3"	FNC 3 "3"	
Q	Q	49(31H)	FNC 2 "2"	FNC 2 "2"	

Character to Print			Character to Print		
CODE-A	CODE-B	CODE-C	CODE-A	CODE-B	CODE-C
R	R	50(32H)	SHIFT "{S"	SHIFT "{S"	
S	S	51(33H)	CODE C "{C"	CODE C "{C"	
T	T	52(34H)	CODE B "{B"	CODE A "{A"	CODE B "{B"
			FNC 4 "{4"	FNC 4 "{4"	CODE A "{A"
			FNC 1 "{1"	FNC 1 "{1"	FNC 1 "{1"

6. Following are TextPosition assignment, and bar code printing possibility condition according to Width. As for Width Parameter, if without special description, they mean that it is possible to print bar codes independently on TextPosition, within the printable area. As for the following list, it is prerequisite that Width Parameter is in units of dots and that it is within the printable area

<i>symbology</i>	<i>textPosition</i> PTR_BC_TEXT_NONE	<i>extPosition</i> PTR_BC_TEXT_ABOVE PTR_BC_TEXT_BELOW
PTR_BCS_UPCA(101)	Printable	Width=95 - 189 JPOS_E_ILLEGAL
PTR_BCS_UPCE(102)	Printable	Width=51.131 JPOS_E_ILLEGAL
PTR_BCS_JAN8(103)	Printable	Width=67 - 133 JPOS_E_ILLEGAL
PTR_BCS_JAN13(104)	Printable	Width=95 - 189 JPOS_E_ILLEGAL
PTR_BCS_CODE39(108)	Printable	Printable
PTR_BCS_ITF(106)	Printable	Printable
PTR_BCS_Codabar(107)	Printable	Printable
PTR_BCS_CODE93(109)	* 1	* 1
PTR_BCS_CODE128(1.13)	* 2	* 2

*1: If *width* and number of characters are within the range as follows, JPOS_E_ILLEGAL will be returned:

$$37 + 9 * wlen + 18(len - wlen) \leq width < 74 + 18 * wlen + 36(len - wlen)$$

- wlen = the number of '0' - '9', 'A' - 'Z', ',', '\$', '%', '+', '-', '.', '/' within character

- len = Character length

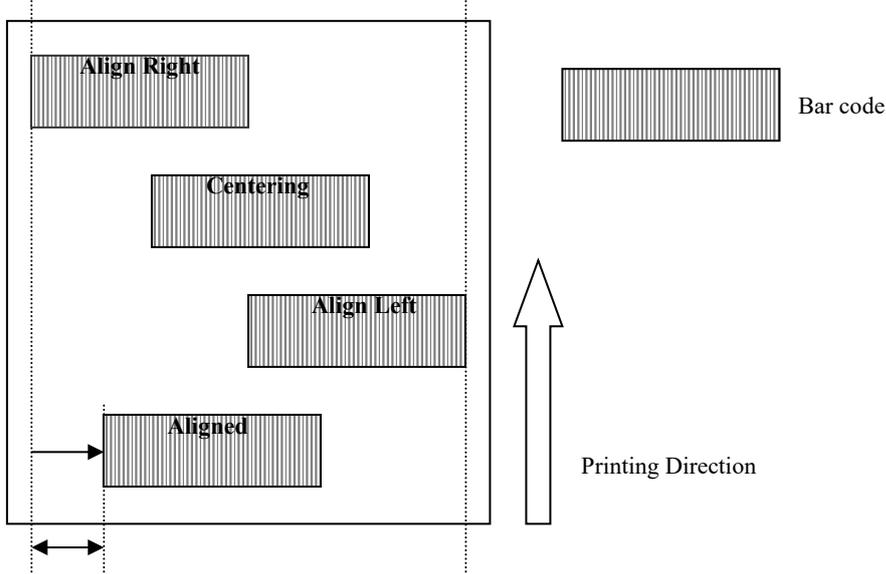
*2: If *width* and number of characters are within the range as follows, JPOS_E_ILLEGAL will be returned.

$$24 + 11 * len \leq width < 48 + 22 * len$$

- len = The gained value by subtracting the number of "{A", "{B", "{C", "{1", "{2", "{3", "{4", "{S", "{" (which are included in the character length) from the character length.

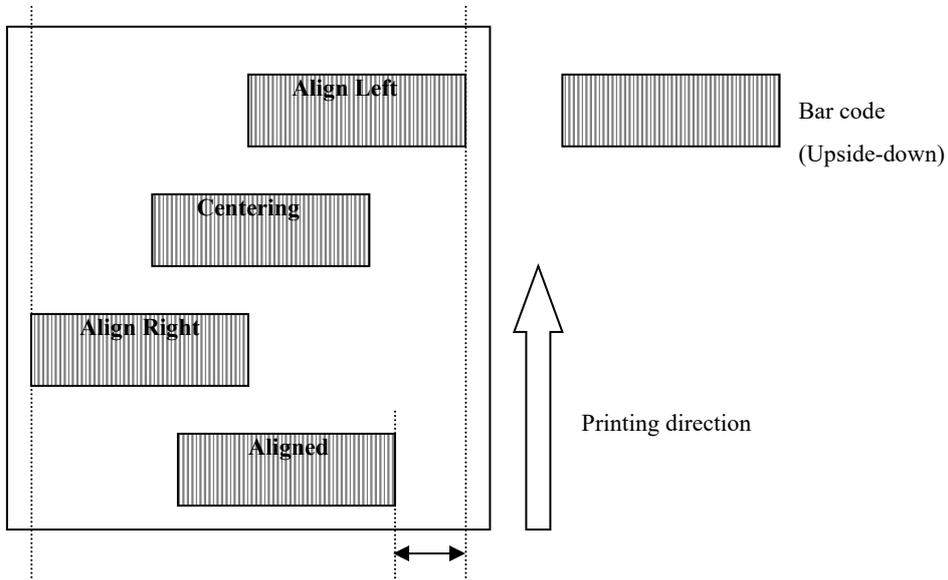
Rotating Bar Code Printing by the RotateSpecial property

Printing positions are changed as follows by the *alignment* parameters in normal orientation.



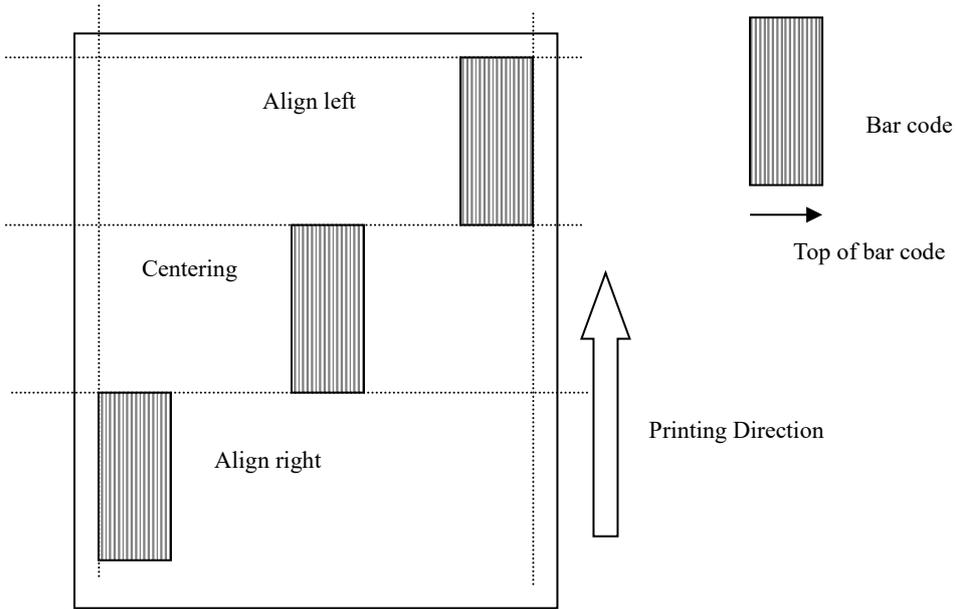
Distance from the left most print column set by *alignment*

Printing positions are changed as follows by the *alignment* parameters in upside-down orientation.

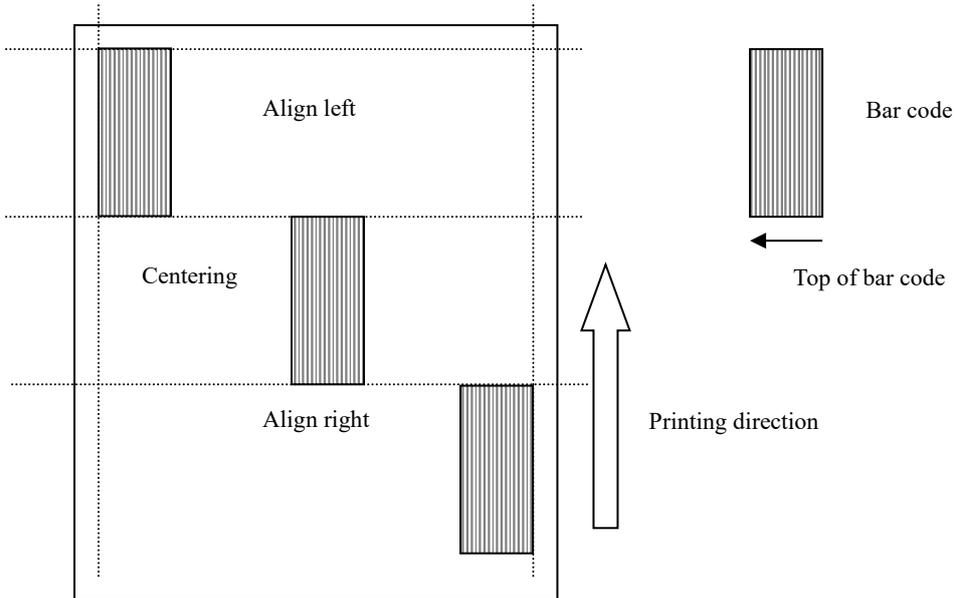


Distance from the left most print column set by *alignment*

Printing positions are changed as follows by the *alignment* parameters in 90-degree-to-the-right orientation.



Printing positions are changed as follows by the *alignment* parameters in 90-degree-to-the-left orientation.



Errors

This method throws a `JposException`.

The exception's `ErrorCode` property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	One of the following conditions has occurred: <ul style="list-style-type: none"> - <i>station</i> does not exist. - <i>station</i> does not support bar code printing. - <i>height</i> or <i>width</i> are zero or too big. - <i>symbology</i> is not supported. - <i>symbology</i> contains unsupported character. - <i>alignment</i> is invalid or too big (If <i>alignment</i> is specified to an absolute position, the sum of the value of alignment and the actual printing width of bar code - the value calculated by the closest value of width - exceeds the available width.) - <i>textPosition</i> is invalid .
JPOS_E_NOHARDWARE (107)	The POS printer is not connected to the system or is not powered on..
JPOS_E_FAILURE(111)	The POS printer is in error state. Execute the method after the error state is cleared.
JPOS_E_TIMEOUT(112)	Sending data to the POS Printer was timed out, or completion of printing could not be confirmed after time out.
JPOS_E_BUSY (113)	This operation cannot be performed because processing is in progress.
JPOS_E_EXTENDED (114)	Extended error codes: <ul style="list-style-type: none"> <i>ErrorCodeExtended</i> = JPOS_EPTR_COVER_OPEN (201): The POS Printer cover is open. (Can be returned only if AsyncMode is false.) <i>ErrorCodeExtended</i> = JPOS_EPTR_REC_EMPTY (203): The receipt station is out of paper. (Can be returned only if AsyncMode is false.) <i>ErrorCodeExtended</i> = JPOS_FIT_EPTR_FATAL (20003): A non-recoverable error occurred. (Can be returned only if AsyncMode is false.)

ErrorCodeExtended =

JPOS_FIT_EPTR_OVERHEAT (20006):

The print head is overheated.

(Can be returned only if **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_FIT_EPTR_CUTTERJAM (20008):

A cutter jam error occurred.

(Can be returned only if **AsyncMode** is **false**.)

printBitmap Method

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

Parameter	Description
<i>station</i>	PTR_S_RECEIPT(2) is assigned.
<i>fileName</i>	File name of bitmap file The file must in the compressed .format. (Specify full path or relative path.)
<i>width</i>	Printed width of the bitmap to be performed. See values below.
<i>alignment</i>	Placement of the bitmap. See values below.

The *width* parameter has one of the following values:

Value	Meaning
PTR_BM_ASIS (-11)	Prints the bitmap with one bitmap pixel per printer dot.
<i>Other Values</i>	Bitmap width expressed in the unit of measure given by MapMode . Valid values are 1 to the value of the RecLineWidth property.

The *alignment* parameter has one of the following values:

Value	Meaning
PTR_BM_LEFT (-1)	Align with the left-most print column.
PTR_BM_CENTER (-2)	Align in the center of the station.
PTR_BM_RIGHT (-3)	Align with the right-most print column.
<i>Other Values</i>	Distance from the left-most print column to the start of the bitmap. Expressed in the unit of measure given by MapMode . The sum of this value and <i>width</i> should not exceed the limitation of the <i>width</i> parameter.

Remarks This method is called to print a bitmap on the specified printer. The bitmap is converted to monochrome or 2 colors and printed.

When 2-color printing is set, black is printed as the first color and red is printed as the second color.

The size of the bitmap that can be registered is the horizontal size (*Width*) that is the dots of **RecLineWidth** or less (when *Alignment* is absolute position specified, $Width + Alignment \leq RecLineWidth$) and the vertical size that is 1662 dot for single color data and 831 dot or less for two color data with 2 color setting.

Because **PrintBitmap** sends bitmap data to the printer at the time of being called, the performance is not high. It is recommended to print the bitmap with **SetBitmap** and the Escape Sequence.

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

The *width* parameter controls transformation of the bitmap. If *width* is PTR_BM_ASIS, then no transformation is performed. The bitmap is printed with one bitmap pixel per one POS Printer dot.

If *width* is not 0, then the bitmap will be transformed by stretching or compressing the bitmap such that its width is the specified width and the aspect ratio is unchanged.

* When the specified bitmap data is in monochrome, monochrome bitmap is set in the printer. For data other than in monochrome, when the **CapRec2Color** property is **true**, 2-color bitmap printing is performed. When the property is **false**, it is printed as monochrome data.

Errors

This method throws a JposException.

The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	One of the following conditions has occurred: <ul style="list-style-type: none"> - <i>station</i> does not exist. - <i>station</i> does not support bitmap printing. - <i>width</i> is too big. - <i>alignment</i> is invalid or too big.
JPOS_E_NOHARDWARE (107)	The POS printer is not connected to the system or is not powered on.
JPOS_E_NOEXIST (109)	The file specified by <i>fileName</i> was not found.
JPOS_E_FAILURE(111)	The POS printer is in error state. Execute the method after the error state is cleared.
JPOS_E_TIMEOUT(112)	Sending data to the POS Printer was timed out, or completion of printing could not be confirmed after time out.
JPOS_E_BUSY (113)	This operation cannot be performed while output is in progress. (Returned only when AsyncMode is false .)
JPOS_E_EXTENDED (114)	Extended error codes: <ul style="list-style-type: none"> <i>ErrorCodeExtended</i> = <ul style="list-style-type: none"> JPOS_EPTR_COVER_OPEN (201): The POS Printer cover is open. (Returned only when AsyncMode is false.) <i>ErrorCodeExtended</i> = <ul style="list-style-type: none"> JPOS_EPTR_REC_EMPTY (203): The receipt station is out of paper. (Returned only when AsyncMode is false.) <i>ErrorCodeExtended</i> = <ul style="list-style-type: none"> JPOS_EPTR_TOOBIG (206): The bitmap width is too big.

Printable bitmap size is: Width (number of dot of RecLineWidth Property) Height (monochrome: 1662 dot, 2-color setting: 831 dot).

ErrorCodeExtended =

JPOS_EPTR_BADFORMAT (207):

This is returned when the specified file is not the bitmap file. This error is delivered when data is 24 bit bitmap in 2 colors setting as well.

(For monochrome printing, 24 bit bitmap can be printed.)

ErrorCodeExtended = JPOS_FIT_EPTR_FATAL (20003):

A non-recoverable error occurred.

(Returned only when **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_FIT_EPTR_OVERHEAT (20006):

The print head is overheated.

(Returned only when **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_FIT_EPTR_CUTTERJAM (20008):

A cutter jam error occurred.

(Can be returned only if **AsyncMode** is **false**.)

printImmediate Method

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

Parameter	Description
<i>station</i>	PTR_S_RECEIPT(2) is assigned.
<i>data</i>	The characters to be printed. May consist of printable characters, escape sequences, carriage returns (13 decimal), and newline / line feed (10 decimal).

Remarks This method is called to print *data* to the POS Printer During asynchronous printing (State=JPOS_S_BUSY(3)), JPOS_E_BUSY(113) is returned. During the error event (State=JPOS_S_ERROR(4)), JPOS_E_FAILURE(111) is returned. It performs a reverse line feed in the case that characters per line of the text exceed maximum-characters-per-line.

Special character values within *data* are:

Value	Meaning
Newline / Line Feed (10 decimal)	Print any data in the line buffer, and feed to the next print line. (A Carriage Return is not required in order to cause the line to be printed.).
Carriage Return (13 decimal)	If a Carriage Return immediately precedes a Line Feed, or if the line buffer is empty, then it is ignored. Carriage Return acts like a Line Feed. The validateData method may be used to determine whether a Carriage Return without Line Feed is possible, and whether a reverse line feed is required to support it.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The POS Printer specified (other than the receipt station) does not exist.
JPOS_E_FAILURE(111)	The POS printer is in error state. Execute the method after the error state is cleared.
JPOS_E_TIMEOUT(112)	Sending data to the POS Printer was timed out, or completion of printing could not be confirmed after time out.
JPOS_E_BUSY (113)	This operation cannot be performed while output is in progress.
JPOS_E_EXTENDED (114)	Extended error codes:

ErrorCodeExtended =
JPOS_EPTR_COVER_OPEN (201):
The POS Printer cover is open.

ErrorCodeExtended =
JPOS_EPTR_REC_EMPTY (203):
The receipt station is out of paper.

ErrorCodeExtended =
JPOS_FIT_EPTR_FATAL (20003):
A non-recoverable error occurred.

ErrorCodeExtended =
JPOS_FIT_EPTR_OVERHEAT (20006):
The print head is overheated.

ErrorCodeExtended =
JPOS_FIT_EPTR_CUTTERJAM (20008):
A cutter jam error occurred.

printMemoryBitmap Method

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

Parameter	Description
Station	It assigns PTR_S_RECEIPT(2)
Data	Pointer to the byte array that holds the bitmap data.
Type	PTR_BMT_BMP is specified.
Width	Printed width of the bitmap to be performed. See values below.
Alignment	Placement of the bitmap. See values below.

The *Width* parameter has one of the following values:

Value	Meaning
PTR_BM_ASIS(-11)	Prints the bitmap with one bitmap pixel per POS Printer dot.
Other Values	Bitmap width. Expressed in the unit of measure given by MapMode .

The *Alignment* parameter has one of the following values:

Value	Meaning
PTR_BM_LEFT(-1)	Align left
PTR_BM_CENTER(-2)	Centering
PTR_BM_RIGHT(-3)	Align right
The others	Distance from the left-most print column to the start of the bitmap. Expressed in the unit of measure given by MapMode .

Remarks This method is called to print a memory-stored bitmap on the specified printer station.. The bitmap passed as the pointer to the byte array is converted to monochrome or 2 colors and printed.

When 2 color printing is set, black is printed as the first color and red is as the second color.

The size of the bitmap that can be registered is the horizontal size (Width) that is the dots of **RecLineWidth** or less (when Alignment is absolute position specified, Width + Alignment <= **RecLineWidth**) and the vertical size that is 1662 dot for single color data and 831 dot or less for two color data with 2 color setting.

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

The Width parameter controls transformation of the bitmap. If width is PTR_BM_ASIS, then no transformation is performed. The bitmap is printed with one bitmap pixel per POS printer dot. Advantages of this option are that it:

- Provides the highest performance bitmap printing.
- Works well for bitmaps tuned for a specific printer's aspect ratio between horizontal dots and vertical dots.

If Width is not 0, then the will be transformed by stretching or compressing the bitmap such that its width is the specified width and the aspect ratio is unchanged.

Because it is not buffered to **TransactionPrint**, data can be sent to the printer in the middle of buffering.

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

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL(106)	One of the following errors occurred. <ul style="list-style-type: none"> - No <i>Station</i>. - <i>Station</i> does not support bitmap printing. - <i>Width</i> is too large. - <i>Alignment</i> is illegal value or too large.
JPOS_E_NOHARDWARE(107)	POS Printer is OFF or OFFLINE.
JPOS_E_FAILURE(111)	The POS printer is in error state. Execute the method after the error state is cleared.
JPOS_E_TIMEOUT(112)	Sending data to the POS Printer was timed out, or completion of printing could not be confirmed after time out.
JPOS_E_BUSY(113)	It cannot perform because it outputting.
JPOS_E_EXTENDED(114)	<p>ResultCodeExtended = JPOS_EPTR_COVER_OPEN(201): POS Printer cover is open. (Only when AsyncMode is FALSE, it is returned.) ResultCodeExtended = JPOS_EPTR_REC_EMPTY(203): It runs out of paper. (Only when AsyncMode is FALSE, it is returned.) ResultCodeExtended = JPOS_EPTR_TOOBIG (206): Assigned bitmap is too large. Printable bitmap size is: Width (number of dot of RecLineWidth Property) Height (monochrome: 1662 dot, 2-clor setting: 831 dot) ResultCodeExtended = JPOS_EPTR_BADSYNTAX(207): Bitmap format is different from the assigned one. The assigned file is not bitmap file. When 2-color printing is set and data is the 24 bit bitmap. (When monochrome is specified, printing 24 bit bitmap is available.)</p>

ResultCodeExtended = JPOS_FIT_EPTR_FATAL(20003):
Fatal error occurs. (Only when **AsyncMode** is **FALSE**, it is returned.)

ResultCodeExtended = JPOS_FIT_EPTR_OVERHEAT
(20006):
Head overheat occurs. (Only when **AsyncMode** is **FALSE**, it is returned.)

ResultCodeExtended = JPOS_FIT_EPTR_CUTTERJAM
(20008):
A cutter jam error occurred. (Only when **AsyncMode** is **FALSE**, it is returned.)

printNormal Method

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

Parameter	Description
<i>station</i>	PTR_S_RECEIPT(2) is assigned.
<i>data</i>	The characters to be printed. May consist of printable characters, escape sequences, carriage returns (13 decimal), and Newline / line feed (10 decimal).

Remarks This method is called to print *data* on the receipt printer station. It performs a reverse line feed in the case that characters per line of the text exceed maximum-characters-per-line
 This method is performed synchronously if **AsyncMode** is **false**, and asynchronously if **AsyncMode** is **true**.

Special character values within *data* are:

Value	Meaning
Newline / Line Feed (10 decimal)	Print any data in the line buffer, and feed to the next print line. (A Carriage Return is not required in order to cause the line to be printed.).
Carriage Return (13 decimal)	If a Carriage Return immediately precedes a Line Feed, or if the line buffer is empty, then it is ignored. Carriage Return acts like a Line Feed. The validateData method may be used to determine whether a Carriage Return without Line Feed is possible, and whether a reverse line feed is required to support it.

Errors This method throws a JposException.
 The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL(106)	The POS Printer specified (other than the receipt station) does not exist.
JPOS_E_FAILURE(111)	The POS printer is in error state. Execute the method after the error state is cleared.
JPOS_E_TIMEOUT(112)	Sending data to the POS Printer was timed out, or completion of printing could not be confirmed after time out.
JPOS_E_BUSY (113)	This operation cannot be performed while output is in progress.
JPOS_E_EXTENDED (114)	Extended error codes: <i>ErrorCodeExtended</i> =

JPOS_EPTR_COVER_OPEN (201):
The POS Printer cover is open.
(Returned only when **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_EPTR_REC_EMPTY (203):
The receipt station is out of paper.
(Returned only when **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_FIT_EPTR_FATAL (20003):
A non-recoverable error occurred.
(Returned only when **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_FIT_EPTR_OVERHEAT (20006):
The print head is overheated.
(Returned only when **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_FIT_EPTR_CUTTERJAM (20008):
A cutter jam error occurred.
(Can be returned only if **AsyncMode** is **false**.)

printTwoNormal Method

Syntax `void printTwoNormal (int stations, String data1, String data2) throws JposException;`

Parameter	Description
<i>station</i>	POS Printer station to be used.
<i>data1</i>	Characters to be printed on the first station.
<i>data2</i>	Characters to be printed on the second station.

Remarks This method is called to print two strings on two print stations simultaneously.
Because this method is only applicable for the Slip Printers, this is not supported by this driver.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The POS printer specified (other than the receipt station) does not exist.

rotatePrint Method

Syntax `void rotatePrint (int station, int rotation) throws JposException;`

Parameter	Description
<i>station</i>	PTR_S_RECEIPT(2) is assigned.
<i>rotation</i>	Direction of rotation. See values below.
Value	Meaning
PTR_RP_RIGHT90 (257)	Rotate printing 90 degree to the right (clockwise).
PTR_RP_LEFT90 (258)	Rotate printing 90 degree to the left (counterclockwise).
PTR_RP_ROTATE180 (259)	Rotate printing 180 degree, that is, print upside-down.
PTR_RP_NORMAL (1)	End rotated printing.

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

If *rotation* is PTR_RP_ROTATE180, then upside-down print mode is entered. Subsequent calls to **printNormal** or **printImmediate** will print the data upsidedown until **rotatePrint** is called with the rotation parameter set to PTR_RP_NORMAL. Lines are printed in the order that they are sent to the POS Printer driver, with the start of each line justified at the right margin of the printer station. Only print methods **printNormal**, and **printImmediate** may be used while in upside-down print mode.

If *rotation* is PTR_RP_RIGHT90 or PTR_RP_LEFT90, then sideways print mode is entered. Subsequent calls to **printNormal** will buffer the print data until **rotatePrint** is called with the *rotation* parameter set to PTR_RP_NORMAL. (In this case, the method only buffers the data – it does not initiate printing. Also, the value of the **AsyncMode** property does not affect its operation: No **OutputID** will be assigned to the request, nor will an **OutputCompleteEvent** be enqueued. For example, even when the POS printer is not powered on, call to any methods will not return an error while rotatePrint is buffering the print data.)

During sideways print mode, the horizontal size will be specified automatically within the range from 0 to 1662 dots for monochrome and from 0 to 831 dots for 2-Color according to the character data that call to the **printNormal** method buffers. The JavaPOS driver will analyze the character data being buffered, then determine horizontal size according to the maximum value of the width among the lines (see the table below). If the width of total character data exceeds 1662 dots (831 dots for 2-color), horizontal size is set to 1662 dots (831 dots for 2-color). Excess data is wrapped into inside the page and printed. If the horizontal size of the character data is double wide or more according to Escape Sequence, the horizontal size will be determined using the value multiplied by the ratio. (For example, if a Font A ANK character is specified in double-wide, it is regarded as 24 dots.)

If no data has been buffered (**printMethod** is not called yet), nothing is printed.

Horizontal size of per character (dots)

Font (Refer to the RecLineChars property)	ANK	kanji
Font A	12 dot	24 dot

Font B	10 dot	20 dot
Font C	8 dot	16 dot

When **PrintBitmap** and **PrintMemoryBitmap** are issued in upside-down printing mode, bitmap is printed upside-down.

For the bitmap performed **SetBitmap** in upside-down printing mode, it is registered without upside-down.

If *rotation* is PTR_RP_NORMAL, then rotated print mode is terminated. If sideways rotated print mode was in effect and some data was buffered by calls to the **printNormal** method, then the buffered data is printed. The entire rotated block of lines is treated as one message.

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

When the vertical length is specified by Escape Sequence with "n" times and print rotated, print may be overlapped or exceed the paper. In this case, input "LF" code before Escape Sequence to specify vertical length. (Ex. To print the data 3 times length vertically, input "LF" code twice. To print the data "n" times length vertically, input "LF" code with the number of "n-1" times.)

However, if bitmap printing or bar code printing is specified by the Escape Sequence while rotated 90 degree left/right is selected, the vertical width will not be calculated. Then the bitmap or bar code exceeding the vertical width cannot be printed normally since they are calculated by other character data.

In addition, if the bar code is printed using the Escape Sequence ESC|#R, the bar code exceeding the print area cannot be printed. In this case, the position to start the print data need to be adjusted to match the bar code by LF and other measures since such positions in the Page Mode may be different between the one-dimensional bar code and two-dimensional bar code.

Errors

This method throws a JposException.

The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL(106)	The POS Printer specified (other than the receipt station) does not exist.

Or *station* does not support the specified rotation.
In different rotation mode, assign PTR_RP_NORMAL(1) or re-execute after clearing rotation printing with **clearOutput**.

- JPOS_E_NOHARDWARE (107)** The POS printer is not connected to the system or is not powered on..
- JPOS_E_FAILURE(111)** The POS printer is in error state. Execute the method after the error state is cleared.
- JPOS_E_TIMEOUT(112)** Sending data to the POS Printer was timed out, or completion of printing could not be confirmed after time out.
- JPOS_E_BUSY (113)** This operation cannot be performed while output is in progress. (Returned only when **AsyncMode** is **false**.)
- JPOS_E_EXTENDED (114)** Extended error codes:

ErrorCodeExtended =

JPOS_EPTR_COVER_OPEN (201):
The POS Printer cover is open.
(Returned only if **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_EPTR_REC_EMPTY (203):
The receipt station is out of paper.
(Returned only if **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_FIT_EPTR_FATAL (20003):
A non-recoverable error occurred.
(Returned only if **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_FIT_EPTR_OVERHEAT (20006):
The print head is overheated.
(Returned only if **AsyncMode** is **false**.)

ErrorCodeExtended =

JPOS_FIT_EPTR_CUTTERJAM (20008):
A cutter jam error occurred.
(Can be returned only if **AsyncMode** is **false**.)

setBitmap Method

Syntax **void setBitmap (int bitmapstationNumber, int station, String filename, int width, int alignment) throws JposException;**

Parameter	Description
<i>bitmapNumber</i>	The number to be assigned to this bitmap. The values from 1 to 20 are valid.
<i>station</i>	PTR_S_RECEIPT(2) is assigned.
<i>fileName</i>	File name of bitmap file The file must in the compressed .format. (Specify full path or relative path.) If empty string is set, the bitmap assigned with the specified <i>bitmapNumber</i> will be deleted from the POS Printer.
<i>width</i>	Printed width of the bitmap to be performed. For the value, refer to PrintBitmap .
<i>alignment</i>	Placement of the bitmap. For the value, refer to PrintBitmap .

Remarks It is called when to save the information concerning bitmap soon to be printed.
Bitmap is printed by calling **PrintNormal** or **PrintImmediate** which has bitmap printing Escape Sequence inside printing data.

When 2-color printing is set, black is stored as the first color and red is stored as the second color

The bitmap that can be registered must be **RecLineWidth** dot (Width) or less (if Alignment is set to absolute position, it is $\text{Width} + \text{Alignment} \leq \text{RecLineWidth}$), and must be vertical size of 2304dot or less and the data size is 384 KB or less after dithering (after converting the data into interpretable bitmap data for POS Printer). When these conditions are not met, OPOS_EPTR_TOOBIG(206) is issued. In addition, when there is no free space on the nonvolatile memory, OPOS_EPTR_TOOBIG is issued. In such case, set empty space in the FileName parameter to secure free space by deleting the bitmap data from the POS printer, then execute again.

* In this driver, when **SetBitmap** is executed, the bitmap that is set is effective even after executing Release Device by driver, because the bitmap is written on nonvolatile memory in the POS Printer. In other words, once setting is complete, bitmap printing is effective with Escape Sequence.

* When the specified bitmap data is in monochrome, monochrome bitmap is set in the printer. For data other than in monochrome, when the **CapRec2Color** property is TRUE, 2-color bitmap printing is performed. When the property is FALSE, it is printed as monochrome data.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.

JPOS_E_ILLEGAL (106)	<p>An invalid parameter value was used:</p> <ul style="list-style-type: none"> - <i>bitmapNumber</i> is invalid. - The POS Printer (other than the receipt station) does not exist - <i>station</i> does not support bitmap printing. - <i>width</i> is too big. - <i>alignment</i> is invalid or too big.
JPOS_E_NOEXIST (109)	<i>fileName</i> was not found.
JPOS_E_FAILURE (111)	Sending the bitmap data to the POS printer failed. The printer cover is open, the receipt station is out of paper, or the printer is powered off.
JPOS_E_TIMEOUT(112)	Sending data to the POS Printer was timed out, or completion of printing could not be confirmed after time out.
JPOS_E_BUSY (113)	This operation cannot be performed while output is in progress.
JPOS_E_EXTENDED (114)	<p>Extended error codes:</p> <p><i>ErrorCodeExtended</i> =</p> <p>JPOS_EPTR_TOOBIG (206):</p> <p>The bitmap width is too big to print without conversion, or is too big to convert.</p> <p><i>ErrorCodeExtended</i> =</p> <p>JPOS_EPTR_BADFORMAT (207):</p> <p>The specified file is not the bitmap file, or in the format not supported.</p> <p>For 2-Color print, if the data is 24 bit bitmap, this error is delivered as well.</p> <p>(For monochrome print, 24 bit bitmap can be printed.).</p>

setLogo Method

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

<u>Parameter</u>	<u>Description</u>
<i>location</i>	The logo to be set. Set PTR_L_TOP (1) for the top logo, and PTR_L_BOTTOM (2) for the bottom logo.
<i>data</i>	The characters that produce the logo. May consist of printable characters, escape sequences (except logos), carriage returns (13 decimal), and line feeds (10 decimal).

Remarks It is called to save a data string as the top or bottom logo.
A logo may then be printed by calling the **printNormal** or **printImmediate** method with the print top logo or print bottom logo escape sequence in the print data.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	An invalid <i>location</i> was specified.
JPOS_E_BUSY (113)	This operation cannot be performed while output is in progress.

transactionPrint Method

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

Parameter	Description
<i>station</i>	PTR_S_RECEIPT(2) is assigned.
<i>control</i>	Batch processing. Refer to the following values:
Value	Meaning
PTR_TP_TRANSACTION (11)	Start of batch processing.
PTR_TP_NORMAL (12)	Ends a transaction by printing the buffered data.

Remarks This method is called when to enter/leave batch processing.

If *control* is PTR_TP_TRANSACTION (11), then transaction mode is entered. Subsequent calls to **printNormal**, **cutPaper**, **rotatePrint**, **printBarCode**, **printBitmap**, and **directIO** will buffer the print data (either at the printer or the Device Service, depending on the printer capabilities) 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 the **AsyncMode** property does not affect their operation: No **OutputID** will be assigned to the request, nor will an **OutputCompleteEvent** be enqueued. In this case, each method is completed successfully. For example, if the POS printer is not powered on, call to each method will not return an error while **transactionPrint** is buffering the print data.)

If *control* is PTR_TP_NORMAL (12), then transaction mode is exited. If some data was buffered by calls to the methods **printNormal**, **cutPaper**, **rotatePrint**, **printBarCode**, **printBitmap**, and **directIO** then the buffered data is printed. The entire transaction is treated as one message. This method is performed synchronously if **AsyncMode** is **false**, and asynchronously if **AsyncMode** is **true**.

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

Attention should be paid when executing the **rotatePrint** method. Once the **transactionPrint** method is executed, the data buffered by calls to the **rotatePrint** method with PTR_RP_RIGHT90 (257), the **printNormal** method and the **rotatePrint** method with PTR_RP_NORMAL (1) will not be printed until transaction mode is exited. If the **rotatePrint** method is called with PTR_RP_RIGHT90 (257) and the **transactionPrint** method is called with PTR_TP_TRANSACTION (11), the buffered data will not be rotated and cannot be printed correctly because buffering by the **transactionPrint** method has priority over the **rotatePrint** method. The **rotatePrint** method must be executed after the **transactionPrint** method and transaction mode is exited.

Errors This method throws a JposException.

The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.

JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The POS Printer specified (other than the receipt station) does not exist.
JPOS_E_NOHARDWARE (107)	The POS printer is not connected to the system or is not powered on.
JPOS_E_FAILURE(111)	The POS printer is in error state. Execute the method after the error state is cleared.
JPOS_E_TIMEOUT(112)	Sending data to the POS Printer was timed out, or completion of printing could not be confirmed after time out.
JPOS_E_BUSY (113)	This operation cannot be performed while output is in progress. (Returned only if AsyncMode is false and the <i>control</i> parameter is PTR_TP_NORMAL (12).)
JPOS_E_EXTENDED (114)	Extended error codes: <i>ErrorCodeExtended =</i> JPOS_EPTR_COVER_OPEN (201): The POS Printer cover is open. (Can be returned only if AsyncMode is false .) <i>ErrorCodeExtended =</i> JPOS_EPTR_REC_EMPTY (203): The receipt station is out of paper. (Can be returned only if AsyncMode is false .) <i>ErrorCodeExtended =</i> JPOS_FIT_EPTR_FATAL (20003): A non-recoverable error occurred. (Can be returned only if AsyncMode is false .) <i>ErrorCodeExtended =</i> JPOS_FIT_EPTR_OVERHEAT (20006): The print head is overheated. (Can be returned only if AsyncMode is false .) <i>ErrorCodeExtended =</i> JPOS_FIT_EPTR_CUTTERJAM (20008): A cutter jam error occurred. (Can be returned only if AsyncMode is false .)

validateData Method

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

<u>Parameter</u>	<u>Description</u>
<i>station</i>	PTR_S_RECEIPT(2) is assigned.
<i>data</i>	Data to be validated. It includes printable data and escape sequence.

Remarks Determines whether a data sequence, possibly including one or more escape sequences, is valid for the specified station, before calling the **printImmediate**, or **printNormal** methods.
This method does not cause any printing, but is used to determine the capabilities of the station.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOTCLAIMED (103)	An attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL(106)	One or more-than-one escape sequences are out of the range, but Control can select valid alternatives. Or, subject station is not yet supported.
JPOS_E_FAILURE(111)	One or more-than-one escape sequences are not supported. There is no alternative to select.

Cases which cause *ErrorCode* of **JPOS_E_ILLEGAL (106)**:

For the Escape Sequence not listed below, exception will not be delivered.

<u>Escape Sequence</u>	<u>Condition</u>
Paper cut (ESC #P)	The percentage '#' is not precisely supported. (Only 1 to 100 are valid.)
Feed and Paper cut (ESC #fP)	The percentage '#' is not precisely supported. (Only 1 to 100 are valid.)
Underline (ESC #uC)	The thickness '#' is not supported. (Only 1 and 2 are valid.)
Scale horizontally (ESC #vC)	The scaling factor '#' is not supported. (Only the values 1 through 8 are valid.)
Scale vertically (ESC #hC)	The scaling factor '#' is not supported. (Only the values 1 through 8 are valid.)

Cases which cause *ErrorCode* of **JPOS_E_FAILURE (111)**:

<u>Escape Sequence</u>	<u>Condition</u>
------------------------	------------------

Feed, Paper cut, and Stamp (ESC #sP)	Not supported.
Stamp (ESC sL)	Not supported.
Print bitmap (ESC #B)	Bitmap number '#' is out of range. (Only 1 and 2 are valid.)
Feed reverse (ESC #rF)	Not supported.
Font typeface selection (ESC #fT)	Not supported.
Italic (ESC (!)iC)	Not supported.
Alternate color (Custom) (ESC #rC)	Not supported.
Printing red (ESC rC)	When the CapRec2Color property is false , it is not supported.
Shading (ESC #C)	Not supported.
RGB Color (ESC #rC)	Not supported.
Subscript/Superscript (ESC (!)tpC)	Not supported.

3.7. Event

DirectIOEvent Event

Interface **jpos.events.DirectIOListener**

Method **directIOOccurred (DirectIOEvent e);**

Description Notifies the application that data that is other than ordinary notification of printer status is received from the printer (including the command status sent by the **directIO** method). Notification will be given by single byte with the following *eventNumber*.

Properties This event contains the following properties:

Property	Type	Description
<i>eventNumber</i>	<i>int</i>	The ID number of the asynchronous output request that is completed.
<i>pData</i>	<i>int</i>	The value of received bytes converted to the int type. (e.g. 0x10 -> 16, 0xFF -> 255)
<i>pString</i>	Object	null

ErrorEvent Event

Interface `jpos.events.ErrorListener`

Method `errorOccurred (ErrorEvent e);`

Description Notifies the application that a printer error has been detected and a suitable response by the application is necessary to process the error condition. This event is delivered only when an error occurs during the asynchronous output.

Properties This event contains the following properties:

<u>Property</u>	<u>Type</u>	<u>Description</u>
<code>errorCode</code>	<code>int</code>	Error Code causing the error event..
<code>errorCodeExtended</code>	<code>int</code>	Extended Error Code causing the error event.
<code>errorLocus</code>	<code>int</code>	Location of the error.
<code>errorResponse</code>	<code>int</code>	Error response.

If `ErrorCode` is `JPOS_E_EXTENDED`, then `ErrorCodeExtended` has one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_EPTR_COVER_OPEN (201)	The printer cover is open.
JPOS_EPTR_REC_EMPTY (203)	The receipt station is out of paper.
JPOS_FIT_EPTR_FATAL (20003)	A non-recoverable error has occurred.
JPOS_FIT_EPTR_OVERHEAT (20006)	Head overheat occurred in the printer.
JPOS_FIT_EPTR_CUTTERJAM(20008)	A cutter jam error occurred.

* The latter three errors among the aforementioned errors are defined by the `com.fujitsu.fit.jpos.IFP510Const` interface.

The application's error event listener may change `ErrorResponse` to one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_ER_RETRY (11)	Retry the asynchronous output. The error state is exited. The default.
JPOS_ER_CLEAR (12)	Clear the asynchronous output or buffered output data. The error state is exited. (It has same effect as the ClearOutput method)

OutputCompleteEvent Event

Interface `jpos.events.OutputCompleteListener`

Method `outputCompleteOccurred (OutputCompleteEvent e);`

Description Notifies the application that the queued output request associated with the OutputID property has been completed successfully.

Properties This event contains the following property:

<u>Property</u>	<u>Type</u>	<u>Description</u>
<i>OutputID</i>	<i>int</i>	The ID number of the asynchronous output request that is completed.

StatusUpdateEvent Event

Interface	jpos.events.StatusUpdateListener
Method	statusUpdateOccurred (StatusUpdateEvent e);
Description	Notifies the application that a printer has had an operation status change.
Properties	This event contains the following property:

Property	Type	Description
<i>Status</i>	<i>int</i>	Indicates the status change.
Value	Meaning	
PTR_SUE_COVER_OPEN (11)	Printer cover is open.	
PTR_SUE_COVER_OK (12)	Printer cover is closed.	
PTR_SUE_REC_EMPTY (24)	No receipt paper.	
PTR_SUE_REC_NEAREMPTY (25)	Receipt paper is low.	
PTR_SUE_REC_PAPEROK(26)	Receipt paper is ready.	
PTR_SUE_REC_COVER_OPEN(62)	The cutter jam error occurred.	
PTR_SUE_REC_COVER_OK(63)	The cutter jam error is recovered.	
PTR_SUE_IDLE (1001)	All the asynchronous output succeeds, or ends by being deleted. POS Printer's State is OPOS_S_IDLE(2) now. FlagWhenIdle Property must be true in order to be notified by this event. And POS Printer Control automatically sets the property to false before the event notifies it.	
JPOS_SUE_POWER_ONLINE (2001)	The device is powered on and ready. (It notifies at the time of PowerNotify = OPOS_PN_ENABLED(1))	
JPOS_SUE_POWER_OFF_OFFLINE (2004)	The device is not powered on or not connected to the system. (It notifies at the time of PowerNotify = OPOS_PN_ENABLED(1))	
OPOS_SUE_UF_PROGRESS(2100) + 1 to 100	(1 to 100 indicate the completion percentage) Specifies the completion percentage of the firmware.	
OPOS_SUE_UF_COMPLETE(2200)	The firmware is updated successfully.	
OPOS_SUE_UF_FAILED_DEV_OK(2201)	The update firmware process failed but the device is still operational.	

4. JavaPOS Interface Specifications (Drawer)

4.1. List

Properties

Common	Type	Access	May Use After	Initial Value, Condition
CapCompareFirmwareVersion	boolean	R	open	false
CapPowerReporting	int	R	open	JPOS_PR_NONE(0)
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 Made writable after open
FreezeEvents	boolean	R/Wopen	false open	false Made writable after open.
PowerNotify	int	R/W	open	JPOS_PN_DISABLED(0) Unwritable
PowerState	int	R	open	JPOS_PS_UNKNOWN(2000)
State	int	R	---	1
DeviceControlDescription	String	R	---	"JavaPOS CashDrawer Device Control"
DeviceControlVersion	int	R	---	1010XXX
DeviceServiceDescription	String	R	open	"FP CashDrawer Device Service, (C) 20xx Fujitsu Isotec"
DeviceServiceVersion	int	R	open	1010XXX
PhysicalDeviceDescription	String	R	open	"FP CashDrawer (C) 20xx Fujitsu Isotec"
PhysicalDeviceName	String	R	open	The device name set in the parameter at the open method.

Specific	Type	Access	May Use After	Initial Value, Condition
CapStatus	boolean	R	open	The initial value is the value of "CapStatus" in jpos.xml.
CapStatusMultiDrawerDetect	boolean	R	open	false
DrawerOpened	boolean	R	Open & Enable	false

* In the Access column, R indicates Read-Only, R/W indicates Read/Write. The item in May Use After is the method and property required for initialization, open indicates the open method, claim indicates the claim method and Enable indicates setting the DeviceEnabled property to true. If required procedure is not executed, JposException may be delivered. When May Use After is open & claim or open, claim & Enable, the property is available for acquisition after the open method is executed, but the value may not be initialized until all open, claim & Enable are executed. To acquire such property, access it after the conditions are met.

Methods

Common	Initialization
open	none
close	open
claim	open
release	open & claim
checkHealth	open & Enable
compareFirmwareVersion	open & Enable
directIO	open
resetStatistics	open & Enable
retrieveStatistics	open & Enable
updateFirmware	open & Enable
updateStatistics	open & Enable

Specific	Initialization
openDrawer	open & Enable
waitForDrawerClose	open & Enable

Events

Event	Initialization
DirectIOEvent	open & Enable
StatusUpdateEvent	open & Enable

4.2. Common Properties

The following sections describe the properties provided commonly to the Drawer.

There are two kinds of properties: Read-Only and Read/Write. For the property that is writable, R/W is added next to the property name.

Only when exception's *errorCode* has the special meaning, the description is provided.

CapCompareFirmwareVersion Property

Type	boolean
Remarks	If true , then the device service supports comparing the version of the firmware in the physical device against that of a firmware file. This property is initialized by the open method.
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value:
Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapPowerReporting Property

Type	int
Remarks	Identifies the reporting capabilities of the device. It has the following value:
Value	Meaning
JPOS_PR_NONE (0)	The device service cannot determine the power.
	This property is initialized by the open method.
Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value:
Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapStatisticsReporting Property

Type **boolean**

Remarks This property is initialized to **false** by the **open** method. Statistics reporting is not supported.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapUpdateFirmware Property

Type **boolean**

Remarks This property is initialized to **false** by the **open** method. Statistics reporting is not supported.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapUpdateStatistics Property

Type **boolean**

Remarks This property is initialized to **false** by the **open** method. Statistics reporting is not supported.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CheckHealthText Property

Type	String
Remarks	<p>Holds the results of the most recent call to the <code>checkHealth</code> method. The following examples illustrate some possible diagnoses:</p> <ul style="list-style-type: none">- "Internal HCheck: Successful" (It succeeded in Internal check.)- "External HCheck: Successful" (It succeeded in External check.)- "External HCheck: Failure" (It failed in External check.)- "InteractiveHCheck: Not Supported" (It is not supported.) <p>This property is not initialized before the first call to the checkHealth method (empty string ("")).</p>
Errors	<p>This property throws a <code>JposException</code>. The exception's <i>ErrorCode</i> property will be the following value:</p>

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

Claimed Property

Type	boolean
Remarks	<p>true: The device is claimed for exclusive access.</p> <p>false: The device is released for sharing with other applications.</p> <p>Many devices must be claimed exclusively before the Control will allow access to many of its methods and properties, and before it will deliver events to the application.</p> <p>The Claimed property is initialized to false by the open method.</p>
Errors	<p>This property throws a <code>JposException</code>. The exception's <i>ErrorCode</i> property will be the following value:</p>

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

DeviceControlDescription Property

Type	String
Remarks	<p>"JavaPOS CashDrawer Device Control" is set. (The value may vary depending on the DC in use.) Identifies the Control Object. It is a character string identifying the Control Object and the company that produced it and always readable.</p>
Errors	None.

DeviceControlVersion Property

Type	int
Remarks	Holds the Control Object version number. "1013XXX" is set. (XXX indicates the version number.) This property is always readable. This property displays the version of DeviceControl implemented in the jcl.jar file. It may differ depending on the environment in use.
Errors	None.

PhysicalDeviceDescription Property

Type	String
Remarks	"FP CashDrawer (C) 20xx Fujitsu Isotec" is set. It is a character string identifying the device and holds the device name and related information. This property is initialized by the open method.

Errors	This property throws a JposException. The exception's <i>ErrorCode</i> property will be the following value:
---------------	---

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

DeviceEnabled Property R/W

Type **boolean**

Remarks **true:** The device is enabled and in an operational state. If changed to **true**, then the device is brought to an operational state.

false: The device has been disabled. If it is changed to **false**, then the device is physically disabled. Subsequent input is discarded and output operation cannot be executed. Before the device is used, application must set this property **true**.

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

Errors This property throws a JposException.
 The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_NOHARDWARE (107)	POS Printer is OFF or OFFLINE or the cable is not connected. Clear the problem, and then execute the property again.
JPOS_E_FAILURE (111)	The connection to the device is failed. There is the possibility that the port specified does not exist.
JPOS_E_TIMEOUT (112)	The POS Printer could not be replaced. There is the possibility of cover open or running out of paper.

PhysicalDeviceName Property

Type **String**

Remarks Set the Device Name : e.g. "FP2000SERDR1".

 This property holds a short name identifying the physical device. This is a short version of **PhysicalDeviceDescription** and should be limited to 30 characters. This property is initialized by the **open** method.

Errors This property throws a JposException.
 The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

FreezeEvents Property R/W

Type	boolean				
Remarks	<p>If true, events will not be delivered. Events will be enqueued until this property is set to false.</p> <p>If false, the application allows events to be delivered. If some events have been held while events were frozen during true and all other conditions are correct for delivering the events, then changing this property to false will allow these events to be delivered. An application may choose to freeze events for a specific sequence of code where interruption by an event is not desirable. This property is initialized to false by the open method.</p>				
Errors	<p>This property throws a JposException.</p> <p>The exception's <i>ErrorCode</i> property will be the following value:</p> <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>JPOS_E_CLOSED (101)</td><td>An attempt was made to access a closed device.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
<u>Value</u>	<u>Meaning</u>				
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.				

PowerNotify Property R/W

Type	int						
Remarks	<p>Contains the type power notification selection made by the Application.</p> <p>The power notification values are:</p> <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>JPOS_PN_DISABLED (0)</td><td>The driver will not provide any power notifications to the application. No power notification StatusUpdateEvents will be fired, and PowerState may not be set</td></tr></tbody></table> <p>This property is initialized to JPOS_PN_DISABLED(0) by the open method</p>	<u>Value</u>	<u>Meaning</u>	JPOS_PN_DISABLED (0)	The driver will not provide any power notifications to the application. No power notification StatusUpdateEvents will be fired, and PowerState may not be set		
<u>Value</u>	<u>Meaning</u>						
JPOS_PN_DISABLED (0)	The driver will not provide any power notifications to the application. No power notification StatusUpdateEvents will be fired, and PowerState may not be set						
Errors	<p>This property throws a JposException.</p> <p>The exception's <i>ErrorCode</i> property will be one of the following values:</p> <table><thead><tr><th><u>Value</u></th><th><u>Meaning</u></th></tr></thead><tbody><tr><td>JPOS_E_CLOSED (101)</td><td>An attempt was made to access a closed device.</td></tr><tr><td>JPOS_E_ILLEGAL (106)</td><td>This property cannot be changed.</td></tr></tbody></table>	<u>Value</u>	<u>Meaning</u>	JPOS_E_CLOSED (101)	An attempt was made to access a closed device.	JPOS_E_ILLEGAL (106)	This property cannot be changed.
<u>Value</u>	<u>Meaning</u>						
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.						
JPOS_E_ILLEGAL (106)	This property cannot be changed.						

PowerState Property

Type **int**

Remarks Identifies the current power condition of the device, if it can be determined.

The power reporting values are:

<u>Value</u>	<u>Meaning</u>
JPOS_PS_UNKNOWN (2000)	The device's power state cannot be determined.

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

DeviceServiceDescription Property

Type **String**

Remarks Holds the Device Service name and the company that produced it.
It is set to "FP CashDrawer Device Service, (C) 20xx Fujitsu Isotec."
This property is initialized by the **open** method.

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

DeviceServiceVersion Property

Type **int**

Remarks "1013XXX" is set. Holds the Device Service version number. (XXX indicates the version number.) This property is initialized by the **open** method.

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

4.3. Common Methods

checkHealth Method

Syntax `void checkHealth(int level) throws JposException`

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

<u>Value</u>	<u>Meaning</u>
JPOS_CH_INTERNAL(1)	Perform a health check that does not physically change the device. It is always successful and "Internal Hcheck:Successful" is set in the CheckHealthText property.
JPOS_CH_EXTERNAL(2)	Perform a more thorough test that may change the device. If possible, it opens the drawer. If drawer is open , "External HCheck:Successful" is set in the CheckHealthText property. This method fails when it is accessed exclusively by other application, the power of the printer is off or the printer. is disconnected. If so, "External HCheck:Failure" is set in the CheckHealthText property
JPOS_CH_INTERACTIVE(3)	Perform an interactive test of the device. This is not supported.T In this case, "Interactive HCheck:Not Supported" is set in the CheckHealthText property and the exception with error code, JPOS_E_ILLEGAL(106) will be delivered.

Remarks It is called to test the state of a device. The result of this method is stored in the **CheckHealthText** property. The **CheckHealth** method is always synchronous

When CapStatus property is set TRUE, CheckHealthMethod waits until Drawer Open status is detected.

Errors This method throws a JposException.

The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_CLAIMED (102)	An attempt was made to access a device that is claimed exclusive-use by another application.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	The specified <i>level</i> parameter is not supported.
JPOS_E_NOHARDWARE(107)	The POS Printer to which the Drawer is connected is OFF or OFFLINE. It is stored only when JPOS_CH_EXTERNAL(2) is set.
JPOS_E_TIMEOUT(112)	Connection is succeeded with printer connected to Drawer, but Drawer Open cannot be detected after timeout period. Only when JPOS_CH_EXTERNAL(2) is set and CapStatus proper is set TRUE, it will be stored.

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 it is zero (0), then the method immediately either returns (if successful) or throws an appropriate exception. If **JPOS_FOREVER (-1)** is set, the method waits as long as needed until exclusive access is satisfied.

Remarks Requires the application to claim for exclusive access to the device before the device can be used. When successful, the **Claimed** property is changed to **true**.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	An invalid <i>timeout</i> parameter was specified.
JPOS_E_TIMEOUT (112)	Another application has exclusive access to the device, and did not relinquish control before <i>timeout</i> milliseconds expired. Or, the state of the Device did not become processible after timeout time has passed.

close Method

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

Remarks Releases 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.
This method should not be executed during event processing (in the Event Handler).

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

compareFirmwareVersion Method

Syntax **void compareFirmwareVersion(String *FirmWareFileName*, int *result*) throws JposException**

Remarks This method is not supported.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	This method is not supported.

directIO Method

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

Remarks This method is called to communicate directly with the Device Service.
This method is not supported.

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	This method is not valid.

open Method

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

The *logicalDeviceName* parameter specifies the device name to open.

The device names of the driver are as follows:

- FP-2000 Serial Interface:	"FP2000SERDR1", "FP2000SERDR2", "FP2000SER2DR1", "FP2000SER2DR2"
- FP-2000 USB Interface:	"FP2000USBDR1", "FP2000USBDR2", "FP2000USB2DR1", "FP2000USB2DR2"
- FP-2100 Serial Interface:	"FP2100SERDR1", "FP2100SERDR2", "FP2100SER2DR1", "FP2100SER2DR2"
- FP-2100 USB Interface:	"FP2100USBDR1", "FP2100USBDR2", "FP2100USB2DR1", "FP2100USB2DR2"
- FP-2200 Serial Interface:	"FP2200SERDR1", "FP2200SERDR2", "FP2200SER2DR1", "FP2200SER2DR2"
- FP-2200 USB Interface:	"FP2200USBDR1", "FP2200USBDR2", "FP2200USB2DR1", "FP2200USB2DR2"

Remarks This method is called to open a device.

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

Errors This method throws a JposException.

The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_NOSERVICE (104)	A connection to the corresponding Device Service could not be established.
JPOS_E_ILLEGAL (106)	The Device Control is already open.
JPOS_E_NOEXIST (109)	The corresponding Device Service does not exist.

release Method

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

Remarks This method is called to release exclusive access to the device.

If the **DeviceEnabled** property is **true** and the device is an exclusive-use device, then the device is also disabled. This method should not be executed during event processing (in the Event Handler).

Errors This method throws a JposException.

The exception's *ErrorCode* property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	Exclusive access to the device is not allowed for the application.
JPOS_E_BUSY (113)	This operation cannot be performed while processing is in

progress.

resetStatistics Method

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

Remarks This method is not supported..

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	This method is not supported..

retrieveStatistics Method

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

Remarks This method is not supported..

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	This method is not supported..

updateFirmwareMethod

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

Remarks This method is not supported..

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	This method is not supported..

updateStatistics Method

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

Remarks This method is not supported..

Errors This method throws a JposException.
The exception's *ErrorCode* property will be one of the following values:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_ILLEGAL (106)	This method is not supported..

4.4. Specific Properties

CapStatus Property

Type **Boolean**

Remarks **true:** The drawer can report the open/closed status.
false: The drawer is not able to determine whether cash drawer is open or closed.

The property of the first Drawer define as "FPxxx[Device Name]DR1" is set to **true** and The property of the second Drawer define as "FPxxx[Device Name]DR2" is set to **false**. The device name contains either "SER (Serial Interface)" or "USB (USB Interface)".

However, evenif the property is **true**, unless the driver is in the enabled state (**DeviceEnabled=true**) for the printer connected to the Drawer, capability of reporting the status of the Drawer is not supported by this property

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

* Use the second drawer property defined by "FPxxx[Device Name]DR2" with FALSE.

 The 2nd drawer does not support the status notification of drawer open/close.

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

CapStatusMultiDrawerDetect Property

Type **boolean**

Remarks **false:** The open/closed status unique to each drawer in a multiple cash drawer configuration can be reported limitedly. The **DrawerOpened** property can determin the following status.

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

Errors This property throws a JposException.
The exception's *ErrorCode* property will be the following value:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

DrawerOpened Property

Type **Boolean**

Remarks **true**: The drawer is open.
false: The drawer is closed.

If the capability **CapStatus** is **false**, then the device does not support status reporting, and the **DrawerOpened** property is always **false**.

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

* Unless the **CapStatus** property is **true**, and the driver is in the enabled state (**DeviceEnabled=true**) for the printer connected to the Drawer, capability of reporting the status of the Drawer is not supported by this property.

* The open/closed status of second Drawer defined as "FPxxx[Device Name]DR2" cannot be reported. The device name is either "SER (Serial Interface)" or "USB (USB Interface)".

Errors This property throws a JposException.

The exception's *ErrorCode* property will be the following value:

Value	Meaning
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.

4.5. Specific Methods

openDrawer Method

Syntax	void openDrawer() throws JposException;
Remarks	Opens the drawer. When CapStatus property is set as TRUE, OpenDrawerMethod waits until the status of Drawer open is detected. When it is being accessed exclusively by other application, this method fails.
Errors	This method throws a JposException. The exception's <i>ErrorCode</i> property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_CLAIMED (102)	An attempt was made to access a device that is claimed exclusive-use by another application.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	Communication to the device was failed.
JPOS_E_FAILURE (111)	The connection to the device is failed. The power of the printer is off or the communication cable is not connected.
JPOS_E_TIMEOUT(112)	Connection is succeeded with printer connected to Drawer, but Drawer Open cannot be detected after timeout period. When CapStatus proper is set TRUE, it will be stored.
JPOS_E_BUSY (113)	This operation cannot be performed while processing is in progress.

waitForDrawerClose Method

Syntax	void waitForDrawerClose(int beepTimeout, int beepFrequency, int beepDuration, int beepDelay) throws JposException;
Remarks	This method is not supported.
Errors	This method throws a JposException. The exception's <i>ErrorCode</i> property will be one of the following values:

<u>Value</u>	<u>Meaning</u>
JPOS_E_CLOSED (101)	An attempt was made to access a closed device.
JPOS_E_DISABLED (105)	This operation cannot be performed while the device is disabled.
JPOS_E_ILLEGAL (106)	This method is not supported by the Device Service.

4.6. Event

DirectIOEvent Event

Interface	<code>jpos.events.DirectIOListener</code>
Method	<code>directIOOccurred(DirectIOEvent e);</code>
Description	This event is not reported.

StatusUpdateEvent Event

Interface	<code>jpos.events.StatusUpdateListener</code>
Method	<code>statusUpdateOccurred(StatusUpdateEvent e);</code>
Description	Notifies the application that a cash drawer has had an operation status change.
Properties	This event contains the following property:

<u>Property</u>	<u>Type</u>	<u>Description</u>
<i>Status</i>	<i>int</i>	Indicates the status of the cash drawer.

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

<u>Value</u>	<u>Meaning</u>
<code>CASH_SUE_DRAWERCLOSED (0)</code>	The drawer is closed.
<code>CASH_SUE_DRAWEROPEN (1)</code>	The drawer is open.

* Unless the `CapStatus` property is `true`, and the Printer OCX is in the enabled (`DeviceEnabled=true`) for the printer connected to the Drawer, Status notification of the Drawer is not supported by this event.

* Nothing is notified when the device is enabled and Drawer is closed.

“CASH_SUE_DRAWEROPEN(1)” is notified when the device is enabled and Drawer is opened.

5. Xml file Configuration

The Device Control reads the following configuration from the jpos.xml file, and starts the Device Service.

Up to the second printer in both Serial and USB can be set to the XML file.

5.1. The explanation of XML items (POS Printer)

Items	Remarks
jposEntry logicalName	The name of the logical device by open. Printer connected to Serial I/F No.1: “[Printer name]SERPRT” Printer connected to Serial I/F No.2: “[Printer name]SER2PRT” Printer connected to USB I/F No.1: “[Printer name]USBPRT” Printer connected to USB I/F No.2: “[Printer name]USB2PRT”
creation factoryClass	The factory class of the Device Service.
creation serviceClass	The class of the Device Service.
vendor name	The name of vendor.
vendor url	The URL of vendor.
jpos category	The device category of JavaPOS.
jpos version	The version number of JavaPOS.
product description	The description of the product.
product name	The name of the product.
product url	The URL of the product.
prop name="portName"	The port name.
prop name="baudRate"	The baud rate.
prop name="parity"	The parity.
prop name="dataBits"	The baud rate.
prop name="stopBits"	The stop bit.
prop name="flowControl"	The flow control. It is fixed to Xon/Xoff.
prop name="vendorID"	The vendor ID set in the Printer. For FP-2000 Series Printer, 1221 is used.
prop name="productID"	The product ID set in the Printer. For FP-2000 Series Printer, 4718 is used.
prop name="serialNumber"	The serial number when the serial number in the printer setting is set to "EXTENDED". If "" (empty string) is set, the printer set to "STANDARD" for the serial number in the printer setting.
prop name="SendTimeout"	The timeout to send (ms). The period of timeout until JPOS_E_TIMEOUT(112) is delivered after the data cannot be sent to the printer while the data is being sent.

Items	Remarks
prop name="ResetTimeout"	The timeout to reset (ms). The period of timeout from when the printer start to reboot until when the Power ON/OFF is detected when the updateFirmware method is executed. (Only for USB).
prop name="InvertDrawerStatus"	Connection drawer invert flag. Due to the machine characteristics of the drawer, the POS printer may notify the open/close status reversed. Set '0' for normal operation and set '1' to reverse the status.
prop name="Smoothing"	Specifies whether to perform smoothing process in the POS printer. For '1', when the RecLetterQuality property is set to true , smoothing process is applied to the font that is equal to or larger than double-width. For '0', regardless the value of the RecLetterQuality property, smoothing process is not applied. (The default value is "1".)
prop name="RecLineChars"	Characters to be printed in a line. Any value of RecLineCharsList is set. (The default value is "48".)
prop name="RecLineCharsList"	The list of the number of the characters that can be printed in a line by the POS printer. According to the conditions for printing, it is set as follows: Receipt is 80 mm and printing characters are 48 per line: "48,57,72" Receipt is 80 mm and printing characters are 42 per line: "42,51,64" Receipt is 58 mm and printing characters are 35 per line: "35,42,52" Receipt is 58 mm and printing characters are 32 per line: "32,42,52"
prop name="RecLineHeight"	The height of the character in the Printer in dot. (Default value is "24")
prop name="RecLineSpacing"	The spacing between print lines in the POS printer in dot. RecLineHeight to 127 dot can be set. (Default value is "30")
prop name="RecLineWidth"	Paper width of the POS printer in dot. According to the conditions for printing, it is set as follows: Receipt is 80 mm and printing characters are 48 per line: "576" Receipt is 80 mm and printing characters are 42 per line: "512" Receipt is 58 mm and printing characters are 35 per line: "420" Receipt is 58 mm and printing characters are 32 per line: "384"
prop name="CapRec2Color"	Specifies whether 2-color printing is available. When 2-color printing is invalid in the Printer, set to "F". (Default) When 2-color printing is valid, set to "T". In case of FP-2000 Series Printer, make this value "F".
prop name="LogLevel"	Indicates the output level of the log. Refer to the Chapter 6 "Log File" for the details.
prop name="LogFile"	Indicates the output destination of the log file.
prop name="LogSize"	Indicates the maximum size of the log file.
prop name="LogRotation"	Indicates the number of the log file to be created when the log file reaches the maximum size..
prop name=" PowerOnNotify "	Indicates the capability of the Power On report of the Printer. (Regardless the value of the PowerNotify property, it specifies whether the printer send the message when the power is turned on.) When "0" is set, it is invalid, and when "1", it is valid. Because this value is not referred from the driver, the behavior is not affected.
prop name="CodePage"	This is the value set in the CharacterSetList property of the POSPrinter.

Items	Remarks
prop name="DefaultCodePage"	This is the value set in the CharacterSet property of the POSPrinter
prop name="DataCompress"	Specifies whether to send data after compression on the PrintBitMap property execution. For " true ", it is compressed, and for " false " it is not compressed. As the default, " false " is set.
prop name="ErrorRecoveryMode"	Specifies how the error is recovered. "0": when error occurs, data that is not sent is sent again, "1": data that is not sent is discarded and error state is recovered by the command. As the default, "1" is set.
prop name="PrintLevel"	Specifies the printing thickness. The default's printing level of black is taken as 100 %. The percentage can be set from 70 to 130 by 10 %.
prop name="PrintSpeed"	Specifies the printing speed. It can be set to the numbers form 100mm/s to 180mm/s. The default is set to "180mm/s " that is highest printing speed.
prop name="BatchPrint"	Specifies the batch printing feature. When "D" (Disable) is set, the batch feature is disabled, and when "E" (Enabled) is set, the batch feature is enabled. "D" (Disabled) is set as the default setting.
prop name="PNESense"	Specifies whether to notify low paper. When it is set to "E" (enabled), notification is executed. When it is set to "D" (disabled), notification is not executed."E" (enabled) is set as the default setting.
prop name="SerialNo"	Specifies the response contents of the serial number. When "N" (Standard) is set, it respond with 30H, and when "E" (Extended) is set, it respond with the serial number written inWhen multiple printers are connected to USE parallel interface, set to "E" (Extended) to distinguish them."N" (Standard) is set as the default setting.
prop name="ID"	The identifier to share the port. This setting should not be changed.
prop name="Device"	Specifies the communication method between the device and the host. "1" is for the serial interface connection, and "3" is or the USB interface connection. This setting should not be changed
prop name="Func2DBarcode"	2D Barcode function. This is a peculiar setting of each printer. Please do not change. (1:Enable, 0:Disable)
prop name="FuncPNESense"	PNE Sense function. This is a peculiar setting of each printer. Please do not change. (1:Enable, 0:Disable))
prop name="FuncConvertKanji"	This setting is whether to convert Kanji character code in JIS at CharacterSet=932. This is a peculiar setting of each printer. Please do not change. (1:JIS Kanji installed, 0:Uninstalled)
prop name="DefaultFont"	Default font setting. This is a peculiar setting of each printer. Please do not change. ("A":Font A, "B":Font B)
prop name=" FuncFontC"	This setting is whether to install font C. This is a peculiar setting of each printer. Please do not change. (1:Font C installed, 0:Uninstalled)
prop name=" CutAtCoverClose"	Specifies the cut operation when cover is closed. When "E" (Enable) is set, cut operation is executed when cover is closed. When "D"(Disable) is set, cut operation is not executed when cover is closed.

Items	Remarks
prop name=" ErrorAlert"	<p>Specifies the warning method with the buzzer when the error occurs.</p> <p>When "N" is set, the buzzer doesn't beep when the printer error occurs.</p> <p>When "O" is set, the buzzer beeps only once when the printer error occurs.</p> <p>When "C" is set, the buzzer keeps beeping from the occurrence of the error to release.</p>
prop name=" BuzzerInterval"	<p>Specifies the beeping interval of the buzzer.</p> <p>The value shows the pattern number. The value can be set from 1(Pattern 1) to 5(Pattern 5)</p>
prop name=" BuzzerRepetition"	<p>Specifies the repetition frequency of the buzzer.</p> <p>The repetition frequency of the buzzer can be set from 0 (None) to 5 (Five).</p>
prop name=" FP1000"	<p>If this device is "FP-1000 Printer", set a value in "true".</p> <p>If this device isn't "FP-1000 Printer", set a value in "false".</p>
prop name=" FP510II"	<p>If this device is "FP-510II Printer", set a value in "true".</p> <p>If this device isn't "FP-510II Printer", set a value in "false".</p>
prop name=" FP2000"	<p>If this device is "FP-2000 Printer", set a value in "true".</p> <p>If this device isn't "FP-2000 Printer", set a value in "false".</p>

5.2. The explanation of XML items (Drawer)

Items	Remarks
JposEntry logicalName	The name of the logical device by open. Drawer1 connected to Serial I/F No.1: “[Printer name]SERDR1” Drawer1 connected to Serial I/F No.2: “[Printer name]SER2DR1” Drawer1 connected to USB I/F No.1: “[Printer name]USBDR1” Drawer1 connected to USB I/F No.2: “[Printer name]USB2DR1” Drawer2 connected to Serial I/F No.1: “[Printer name]SERDR2” Drawer2 connected to Serial I/F No.2: “[Printer name]SER2DR2” Drawer2 connected to USB I/F No.1: “[Printer name]USBDR2” Drawer2 connected to USB I/F No.2: “[Printer name]USB2DR2”
creation factoryClass	The factory class of the Device Service.
creation serviceClass	The class of the Device Service.
vendor name	The name of vendor.
vendor url	The URL of vendor.
jpos category	The device category of JavaPOS.
jpos version	The version number of JavaPOS.
product description	The description of the product.
product name	The name of the product.
product url	The URL of the product.
prop name="portName"	The port name.
prop name="baudRate"	The baud rate.
prop name="parity"	The parity.
prop name="dataBits"	The baud rate.
prop name="stopBits"	The stop bit.
prop name="flowControl"	The flow control.
prop name="vendorID"	The vendor ID set in the Printer.
prop name="productID"	The product ID set in the Printer.
prop name="serialNumber"	The serial number when the serial number in the printer setting is set to "EXTENDED". If "" (empty string) is set, the printer set to "STANDARD" for the serial number in the printer setting.
prop name="CapStatus"	Supports the open/close status of the Drawer. ("ture" is supported, and "false" is not supported.) This value should not be changed
prop name="DrawerNo"	The number of the Drawer. (1 or 2)
prop name="OffTimer"	The drawer kick on time (ms)
prop name="OnTimer"	The drawer kick off time (ms)
prop name="ID"	The identifier to share the port. This setting should not be changed.

Items	Remarks
prop name="Device"	Specifies the communication method between the device and the host. "1" is for the serial interface connection, and "3" is or the USB interface connection. This setting should not be changed
prop name="SendTimeout"	The timeout to send (ms). The period of timeout until JPOS_E_TIMEOUT(112) is delivered after the data cannot be sent to the printer while the data is being sent.
prop name="ResetTimeout"	The timeout to reset (ms). The period of timeout from when the printer start to reboot until when the Power ON/OFF is detected when the updateFirmware method is executed. (Only for USB).
prop name="DrawerOpenTimeout"	External Check (CheckHealthMethod)、 Timeout period of Drawer open detection used by OpenDrawer Method(ms)
prop name=" FP1000"	If this device is "FP-1000 Printer", set a value in "true". If this device isn't "FP-1000 Printer", set a value in "false".
prop name=" FP510II"	If this device is "FP-510II Printer", set a value in "true". If this device isn't "FP-510II Printer", set a value in "false".
prop name=" FP2000"	If this device is "FP-2000 Printer", set a value in "true". If this device isn't "FP-2000 Printer", set a value in "false".

6. Log Files

The Device Control has the function to output the log files. Setting prop name = "LogLevel" to the value from "0" thorough "3" will determines output of a log file.

The output levels of log and the results are provided in the table below. When a log is output, the java.util.logging package, the standard package of Java, is used. The operations described in the table represent the log levels in java.util.logging package. The output format is the format of the java.util.logging package generally.

Level	Operation
0	Outputs no log. (OFF)
1	Outputs a log when a severe error occurs. This is for future extension, and if this level is assigned, a log will be output when the operation of the thread fails. (SEVERE)
2	Outputs a log when a warning (such as general exceptions) is issued. (WARNING)
3	Outputs a information log. The methods of the Device Service that are executed and the data that are sent and received are provided. (INFO)

Example of a log output

2007/10/21 9:26:50 com.fujitsu.fit.jpos.services.POSPrinterService printNormal

Information: Enter

2007/10/21 9:26:50 com.fujitsu.fit.jpos.services.POSPrinterService printNormal

Information: station = 2

2007/10/21 9:26:50 com.fujitsu.fit.jpos.services.POSPrinterService printNormal

Information: data = |2hC

When the method is executed, "Enter" is output. The parameters are output as well.

2007/10/21 9:26:50 com.fujitsu.fit.jpos.services.POSPrinterDataThread run

Information: Send Data:

2007/10/21 9:26:50 com.fujitsu.fit.jpos.services.POSPrinterDataThread run

Information:

Addr : +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F

0000 : 1B 74 01 1B 52 08 1B 7B 00 1B 4D 00 1C 28 41 02

002000 30 00 1B 33 3C 1B 45 00 1B 2D 00 1C 2D 00 1D

0020 : 42 00 1D 21 00 1D 28 4E 02 00 30 31 1D 62 01 1B

0030 : 61 00 1D 42 10 66 66 65 0A

The send data are described in a dump list.

2007/10/21 9:27:18 com.fujitsu.fit.jpos.services.POSPrinterDataThread readComm

Information: Data Received.

2007/10/21 9:27:18 com.fujitsu.fit.jpos.services.POSPrinterDataThread readComm

Information:

Addr : +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F

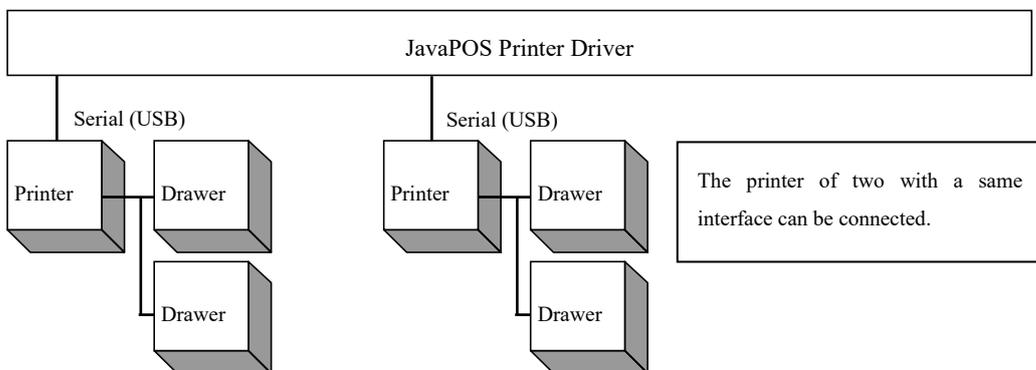
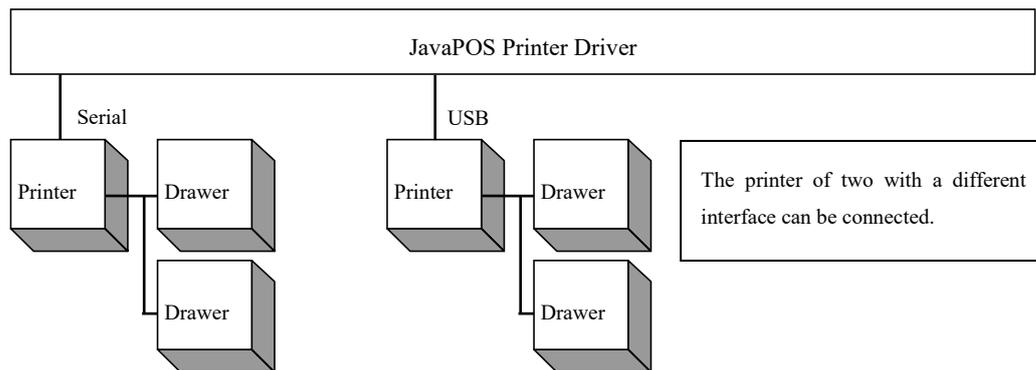
0000 : 14 00 00 0F

The received data are described in a dump list.

7. Using Multiple Printers

The FP-2000 POSPrinter Driver can use two of a serial and USB simultaneously.

In Windows, two of a serial can be used simultaneously.



* When you set two or more printers, connect printer one by one, and set it with the setup tool.

* The printer might not print correctly depending on PC or USB port when two printers are connected in USB interface. Please use the printer after confirming the correct print.

8. Replacement of printer

In case of Serial and USB interface, the replacement of the printer is completed only by connecting a new printer after the application is ended, and executing the application again. (After the Close method is issued, the Open method and the ClaimDevice method are issued, and the DeviceEnabled property is changed into TRUE.)

The setup tool need not be used.

Note: The printer connected with FPxxxUSBPRT cannot be replaced with FPxxxUSB2PRT. This opposite cannot be done. The purpose is to prevent the replacement of the printer from happening by mistake at two USB connection.

When you want to replace it, connect printer one by one, and set it with the setup tool.

Note: When replacing USB printer to another USB one, depending on the environment of PC, JPOS_E_TIMEOUT may occur to take time to recognize the USB device. And when you connect new printer, USB device driver install wizard may be displayed. Change the DeviceEnabled property into TRUE again, after finishing this wizard (*1).

*1: When the printer is connected to PC in USB interface and "Found New Hardware Wizard" was displayed, set it in the following procedures.

- (a) Check "No, not this time", and click [Next]
- (b) Check "install the software automatically", and click [Next].
- (c) Check "Don't prompt me again to install this software", and click [Finish].

Revision History			
Document	FP-2200/FP-2100/FP-2000 POSPrinter, CashDrawer Application Programmer's Guide of Java for Retail POS Driver for Serial/USB Interface		
Revision #	Date	Part/Reason/Contents	Revised by
1.0	October 03, 2017	First officially released to support JavaPOS 1.13.	FUJITSU ISOTEC LIMITED
1.1	January 05, 2022	Added the WPC1251	FUJITSU ISOTEC LIMITED
1.2	September 16, 2022	Added the WPC1250	FUJITSU ISOTEC LIMITED