FTP2166000R0
FTP-POS for Android
Function specification

Version 2

October 30, 2020

FUJITSU COMPONENT LIMITED

Table of Contents

- The contents of this document may be changed without notice.
- It is prohibited to duplicate and reprint part or all of this document without permission.
- Our company shall not be liable for any damages resulting from the use of information contained in this document.

- Android™ is a trademark of Google LLC.
- Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.
- The Bluetooth® word mark and logo is a registered trademark of Bluetooth SIG, Inc. FUJITSU COMPONENT LIMITED uses these trademarks under the terms of license. Other trademarks and trade names are the property of their respective owners.
- Bluetooth® is registered trademark or trademarks of Bluetooth SIG, Inc in the United States and other countries.
- QR Code is a registered trademark of Denso Wave Incorporated.
- Apache® is a trademark of the Apache Software Foundation.
- All other product names and service names are trademarks or registered trademarks of their respective companies.

1. Outline

FTP-POS is a Software Development Kit (SDK) for controlling the printer manufactured by FUJITSU COMPONENT LIMITED.

Library set name	FTP2166000R0_FTP-POS_Android_Vxxx
(Release file name)	(FTP2166000R0_ FTP-POS_Android_Vxxx.zip)
	"xxx" is file version
File list	libftp2166000r0_ftppos.jar
	SampleCode.zip
	A1NC40216-6000R0_xx.pdf
	A1NC40216-6000R0~6_xx.pdf
	("xx" is file version)
Supported operating system	Android 6.0 or lator
	Operation on all devices is not guaranteed.
Supported interface	USB

1.1. Restrictions

1.1.1. Common restrictions

- FTP-POS is not thread safe. It must be called from only one thread in the process.
- Do not allow multiple processes to communicate to the same printer at the same time.
- It does not automatically reconnect when the power is turned off or when communication is restored or when a hardware reset is performed. Please reconnect by the close method and open method. In the case of USB, the device name will change, so create an instance of the PrinterConfiguration class with a new device name.

1.2. Supported printer

	USB	Bluetooth SPP	Wi-Fi
FTP-62GDSL000 FTP-62GDSL100 FTP-62GDSL110 FTP-62GDSL120 FTP-62GUSL000 FTP-62GUSL070 FTP-62GUSL100 FTP-63GUSL000 FTP-63GUSL070 FTP-64GDSL130	✓		
FTP-629DSL350 FTP-639USL100 FTP-639USL200	✓		
FTP-62HDSL100	✓		

2. How to use

2.1. Include to Android Studio project

(1) Copy libftp2166000r0_ftppos.jar to app/libs directory in application project.

(2) Open [File] – [Project structure…].
Select [Dependencies].
Select module in [Modules].
Click [+] icon in [Declared Dependencies], then select [Jar Dependency].

(3) Select “libs” directory. And click [OK] button.

(4) Click [OK] to close [Project structure] window.

2.2. Connect USB device

If connect by USB, need to get permission to assscess the USB device.

Add code to AndroidManifest.xml as follows.

```
<manifest …>
<application>
<activity …>
<intent-filter>
<action android:name=”android.hardware.usb.action.USB_DEVICE_ATTACHED” />
</intent-filter>
<meta-data android:name=”android.hardware.usb.action.USB_DEVICE_ATTACHED”
android:resource=″@xml/device_filter″ />
```

Create device_filter.xml in res/xml folder. Add Vendor ID and Product ID of USB device as follows.

Please refer to the printer’s product specification for the vendor ID and product ID.

```
<?xml version=″1.0″ encoding=″utf-8″?>
<resources>

    <!--FTP-629DSL350-->
    <usb-device vendor-id=″1072″ product-id=″1565″ />

    <!--FTP-62GDSL000-->
    <usb-device vendor-id=″1072″ product-id=″1571″ />
```

2.3. Develop app

Contorolling printer flow is as follows. Refer the sample code for details.

1. **PosPrinter#open** method：Open device.
2. **PosPrinter#claim** method：Request exclusive access to the device.
3. **PosPrinter#deviceEnabled** property：Set to true for the device to enable.
4. Control the device. （Each property, method, event）
5. **PosPrinter#deviceEnabled** property：Set to false for the device to disable.
6. **PosPrinter#release** method：Release exclusive access to the device.
7. **PosPrinter#close** method：Close device

3. Package specification

Do not use any packages, classes, properties, methods or enums not listed in this document.

This SDK operates device attributes and setting values using properties.

The Properties exist two kinds of attributes: read-only and read-write. R/W means the writable properties and is described at next to the property name.

To get the value of a property, use a method of the form:

Type getSampleProperty();

If the property is writable, use the following form of method to set the value of the property:

void setSampleProperty(Type value);

“Type” is data type of property, “SampleProperty” is the name of property.

List

com.fujitsu.fcl.ftp2166000r0.ftppos package			
	Classes		
		FtpConst	Constants defined.
		PosConst	Constants defined.
		PosPrinter	Control the printer.
		PosPrinterConst	Constants defined.
		PrinterConfiguration	Configure the connected printer.
		PrintingQuality	Configure the printing quality.
	Enums		
		ConnectionType	Enum of connection type.
		PrinterModel	Enum of printer model.
		PrintingResolution	Enum of printing resolution.
		PrintingWidth	Enum of printing width.
	Exceptions		
		PosException	It is thrown when an exception occurs in this SDK.
com.fujitsu.fcl.ftp2166000r0.ftppos.event package			
	Interfaces		
		DirectIOListener	Listener interface for receive DirectIOEvent event.
		StatusUpdateListener	Listener interface for receive StatusUpdateEvent event.
	Classes		
		DirectIOEvent	This is an event that occurs at the time of notification of printer-specific status.
		PosEvent	Abstract class of the event class of this SDK.
		StatusUpdateEvent	This event occurs when the device status changes.

3.1. FtpConst class

public final class FtpConst extends Object

Constants for FTP−POS are defined.

3.2. PosConst class

public final class PosConst extends Object

Constants for FTP−POS are defined.

3.3. PosPrinterConst class

public final class PosPrinterConst extends Object

Constants for FTP−POS are defined.

3.4. PosPrinter class

public final class PosPrinter extends Object

Control the printer.

3.4.1. Data Characters and Escape Sequences

FTP-POS supports the following Ecape Sequences.

Perform indicated action.

Name	Data	Remarks
Paper cut	ESC |#P	Cuts receipt paper. The character '#' is replaced by an ASCII decimal string telling the percentage cut desired. If '#' is omitted, then a full cut is performed. If '#' is '1' to '99', a partial cut is performed. If '#' is '100' or is omitted, a full cut is performed. If there are data buffered to POS Printer (in the case that POS Printer does not print even if printing request is done), a cut is performed after buffered data is printed. When a rotated 90° left or right mode by **rotatePrint** method, this is not supported. The correspondence depends on the printer model. Please refer to Printer Specification.
Feed and Paper cut	ESC |#fP	Cuts receipt paper, after feeding the paper by the **RecLinesToPaperCut** lines. The character '#' is defined by the "Paper cut" escape sequence. If there are data buffered to POS Pprinter (in the case that POS Printer does not print even if printing request is done), a cut is performed after buffered data is printed. When a rotated 90° left or right mode by **rotatePrint** method, this is not supported. The correspondence depends on the printer model. Please refer to Printer Specification.
Print bitmap	ESC |#B	Prints the bitmap saved by **setBitmap** method. The character '#' is replaced by the bitmap number and supports the values from '1' to '255'. When a rotated 90° left or right mode by **rotatePrint** method, this is not supported. The correspondence depends on the printer model. Please refer to Printer Specification.
Print top logo	ESC |tL	Prints the top logo saved by **setLogo** method. When a rotated 90° left or right mode by **rotatePrint** method, this is not supported.
Print bottom logo	ESC |bL	Prints the bottom logo saved by **setLogo** method. When a rotated 90° left or right mode by **rotatePrint** method, this is not supported.
Fire stamp	ESC |sL	Not supported.
Feed lines	ESC |#lF	Feed 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. The character '#' supports the values from '1' to '255'. When print data is not buffered, specified lines are fed. When print data is buffered, buffered data is printed first, then specified lines are fed.

Feed units	ESC |#uF	Feed the paper forward by mapping mode units. 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. The character ‘#’ is supports the values from ‘1’ to ‘255’ which is converted with PosPrinterConst.PTR_MM_DOTS unit. The length of feed units is not affected by the length of feed line that was set in the printer.
Feed reverse	ESC |#rF	Feed the paper backward. 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. The character ‘#’ supports the values from ‘1’ to ‘255’. When print data is not buffered, specified lines are fed backward. When print data is buffered, buffered data is printed first, then specified lines are fed backward.

Print Line

Characteristics that are reset at the end of each print method or by a “Normal” sequence.

Name	Data	Remarks
Bold	ESC |bC	Prints in bold. The correspondence depends on the printer model. Please refer to Printer Specification.
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. Only one and two dots underline are supported. If ‘#’ is omitted, one dot underline is used. The correspondence depends on the printer model. Please refer to Printer Specification.
Reverse video	ESC |rvC	Prints in a reverse video format. The correspondence depends on the printer model. Please refer to Printer Specification.
Single high and wide	ESC |1C	Prints normal size.
Double wide	ESC |2C	Prints double-wide characters.
Double high	ESC |3C	Prints double-high characters.
Double high and wide	ESC |4C	Prints double-high/double-wide characters.
Scale horizontally	ESC |#hC	Prints with the width scaled ‘#’ times the normal size, where ‘#’ is replaced by an ASCII decimal string. It supports ‘1’ to ‘4’. If ‘#’ is omitted, ‘1’ is used. The correspondence depends on the printer model. Please refer to Printer Specification.
Scale vertically	ESC |#vC	Prints with the height scaled ‘#’ times the normal size, where ‘#’ is replaced by an ASCII decimal string. It supports the values from ‘1’ to ‘4’. If ‘#’ is omitted, ‘1’ is used. The correspondence depends on the printer model. Please refer to Printer Specification.
Center	ESC |cA	Aligns following text in the center. When a rotated 90° left or right mode by **rotatePrint** method, this is not supported. The correspondence depends on the printer model. Please refer to Printer Specification.
Right justify	ESC |rA	Aligns following text at the right. When a rotated 90° left or right mode by **rotatePrint** method, this is not supported. The correspondence depends on the printer model. Please refer to Printer Specification.
Normal	ESC |N	Restores printer characteristics to normal condition.

3.4.2. Constructors

Syntax

public PosPrinter()

Remarks

Create PosPrinter class instance.

3.4.3. Properties

BinaryConversion Property R/W

Remarks

This function is same as BinaryConversion property of OPOS. It is enabled only **printBarCode** method.

When specifying binary data with **printBarCode** method, please use this to change to nibble mode or decimal mode.

This setting is initialized to PosConst.PTR_DIO_BC_NONE by **claim** method

This property is read and written by the directIO method.

Please refer to PTR_DIO_GET_BINARY_CONVERSION and PTR_DIO_SET_BINARY_CONVERSION.

CharacterSet Property R/W

Syntax

int CharacterSet;

Remarks

The character set for printing characters.

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

This property will be set to one of the following values.

Value	Meaning
437	CP437(USA: Standard Europe) character set.
932	Japanese Shift-JIS (CP932).
950	Traditional Chinese Big5.
PosPrinterConst.PTR_CS_ASCII	The ASCII character set, supporting the ASCII characters between 0x20 and 0x7F.
PosPrinterConst.PTR_CS_ANSI	This supports the ASCII characters between 0x20 and 0x7F. Depending on the printer model, '€'(0xFF) can be printed. Please refer to Printer Specification.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	An invalid property value was used.

Claimed Property

Syntax

boolean Claimed;

Remarks

true: The device is claimed for exclusive access.

false: The device is released for sharing with other applications.

The value of **Claimed** is initialized to **false** by the **open** method.

Exception

PosException may be thrown. Please refer PosException class for details.

CoverOpen Property

Syntax

boolean CoverOpen;

Remarks

true: The printer's cover is open.

false: The printer's cover is close.

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

Exception

PosException may be thrown. Please refer PosException class for details.

DeviceEnabled Property　R/W

Syntax

boolean DeviceEnabled;

Remarks

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

false: The device has been disabled. If changed to **false**, then the device is physically disabled.

The application must set this propertyto **true** before using output devices.

The Device's power state may be reported while **DeviceEnabled** is **true.**

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

Exception

PosException may be thrown. Please refer PosException class for details.

Description Property

Syntax

String Description;

Remarks

Holds an identifier for the the company.

This property is always readable.

FreezeEvents Property　R/W

Syntax

boolean FreezeEvents;

Remarks

When **true**, the event will not be delivered. Events will be held until events are unfrozen.

When **false**, the application allows events to be fired. If some events have been held while events were frozen, then changing **FreezeEvents** to **false** will cause these events to be fired.

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.

Exception

PosException may be thrown. Please refer PosException class for details.

MapMode Property R/W

Syntax

int MapMode;

Remarks

Holds the mapping mode of the printer. The mapping mode defines the unit of measure used for other properties, such as line heights and line spacings.

The following map modes are supported. The value in () is dot-line unit whitch is converted from each unit.

Value	Meaning
PosPrinterConst.PTR_MM_DOTS	Dot-line unit of POS Printer. (1 dot)

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

The value of **MapMode** is initialized to PosPrinterConst.PTR_MM_DOTS by the **open** method.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	An invalid mapping mode was specified.

PowerNotify Property　R/W

Syntax

int PowerNotify;

Remarks

Contains the type power notification selection made by the Application.

The power notification values are:

Value	Meaning
PosConst.PN_DISABLED	The Control will not provide any power notifications to the application. No power notification **StatusUpdateEvent**s will be fired, and **PowerState** may not be set.
PosConst.PN_ENABLED	The Control will fire power notification **StatusUpdateEvent**s and update **PowerState**, beginning when **DeviceEnabled** is set to **true**.

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

This property is initialized to PosConst.PN_DISABLED by the **open** method.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	One of the following occurred. • The device is already enabled. • An invalid parameter value was specified.

PowerState Property

Syntax

int PowerState;

Remarks

Contains the current power condition, if **PowerNotify** is set to **PosConst.PN_ENABLED**.

The power reporting values are:

Value	Meaning
PosConst.PS_UNKNOWN	Cannot determine the device's power state, for one of the following reasons.(default) **PowerNotify** = **PosConst.PN_DISABLED**. Power notifications are disabled. **DeviceEnabled** = **false**. Power state monitoring does not occur until the device is enabled.
PosConst.PS_ONLINE	The device is powered on and ready for use.
PosConst.PS_OFF_OFFLINE	The device is off, offline or not connected. The condition reported depends on the printer model and the interface. Please refer to Printer Specification.

This property is initialized to PosConst.PS_UNKNOWN by the **open** method.

Exception

PosException may be thrown. Please refer PosException class for details.

RecEmpty Property

Syntax

boolean RecEmpty;

Remarks

true: The receipt is out of paper

false: The receipt paper is present.

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

Exception

PosException may be thrown. Please refer PosException class for details.

RecLetterQuality Property R/W

Syntax

boolean RecLetterQuality;

Remarks

true: Prints in high quality mode. (Midium speed mode printing)

false: Prints in high speed mode. (High speed mode printing)

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

Exception

PosException may be thrown. Please refer PosException class for details.

RecLineChars Property R/W

Syntax

int RecLineChars;

Remarks

Holds the number of characters that may be printed on a receipt line.

Please refer to "Font" in Printer Specification.

If changed to a line character width that can be supported, then the width is set to the specified value. If the exact width cannot be supported, then subsequent lines will be printed with a most close and larger character size supported. (For example, if set to 40 when Printable width 576 dots, then the Service should select the 48 characters per line size). If the character width cannot be supported, then an error is returned. Setting this property may also update **RecLineHeight**, **RecLineSpacing**, **RecSidewaysMaxChars**, and **RecSidewaysMaxLines.**

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

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	**Meaning**
PosConst.E_ILLEGAL	An invalid line character width was specified.

RecLineHeight Property R/W

Syntax

int RecLineHeight;

Remarks

Holds the receipt print line height, expressed in the unit of measure given by **MapMode**.

When **RecLineChars** is changed, this property is updated to the default line height for the selected width.

The value of **RecLineHeight** is initialized by the **open** method.

Please refer to Printer Specification.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	**Meaning**
PosConst.E_ILLEGAL	This property cannot be set. This property is always readable.

RecLineSpacing Property R/W

Syntax

int RecLineSpacing;

Remarks

Holds the spacing of each single-height print line, including both the printed line height plus the whitespace between each pair of lines. Line spacing is expressed in the unit of measure given by **MapMode**.

When **RecLineChars** is changed, if new **RecLineHeight** is bigger than **RecLineSpacing**, this property is updated to the value of **RecLineHeight.**

The value of **RecLineSpacing** is initializedby the **open** method.

The available range is from 16 (dot) to 255 (dot).

When the value of **RecLineHeight** and **RecLineSpacing** is equal, the printer cannot print smoothly. It's recommended that the value of **RecLineSpacing** is bigger than the value of **RecLineHeight.**

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	An invalid value was specified.

RecLinesToPaperCut Property

Syntax

int RecLinesToPaperCut;

Remarks

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**, and **RecLineSpacing** may cause this property to change.

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

Exception

PosException may be thrown. Please refer PosException class for details.

RecLineWidth Property

Syntax

int RecLineWidth;

Remarks

Holds the width of a line of **RecLineChars** characters, expressed in the unit of measure given by **MapMode**.

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

The value depends on the printer model. Please refer to Printer Specification.

Exception

PosException may be thrown. Please refer PosException class for details.

RecNearEnd Property

Syntax

boolean RecNearEnd;

Remarks

true: The receipt paper is low.

false: The receipt paper is not low.

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

Exception

PosException may be thrown. Please refer PosException class for details.

RecSidewaysMaxChars Property

Syntax

int RecSidewaysMaxChars;

Remarks

Holds the maximum number of characters that may be printed on each line in sideways mode (Rotated 90° left or right print mode).

The value depends on the printer model. Please refer to Printer Specification.

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

Exception

PosException may be thrown. Please refer PosException class for details.

RecSidewaysMaxLines Property

Syntax

int RecSidewaysMaxLines;

Remarks

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

This property is value which is **RecLineWidth** divided by **RecLineSpacing**. If the reminder equals to or bigger than **RecLineHeight** property (height of fonts), the reminder is plus 1.

Changing the properties **RecLineSpacing** may cause this property to change.

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

Exception

PosException may be thrown. Please refer PosException class for details.

RotateSpecial Property R/W

Syntax

int RotateSpecial;

Remarks

Holds the rotation orientation for barcodes.

This property is initialized to PosPrinterConst.PTR_RP_NORMAL by the **open** method.

Some printer models do not support the direction of rotation that can be set in this property. Please refer to Printer Specification.

It has one of the following values.

Value	Meaning
PosPrinterConst.PTR_RP_NORMAL	Print subsequent barcodes in normal orientation.
PosPrinterConst.PTR_RP_RIGHT90	Rotate printing 90º to the right (clockwise).
PosPrinterConst.PTR_RP_LEFT90	Rotate printing 90º to the left (counter-clockwise).
PosPrinterConst.PTR_RP_ROTATE180	Rotate printing 180º, that is, print upside-down.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	An invalid property value was used.

State Property

Syntax

int State;

Remarks

Contains the current state of the Control.

Value	Meaning
PosConst.S_CLOSED	The device is closed. (Default)
PosConst.S_IDLE	The devicel is in a good state and is not busy.

This property is always readable.

Version Property

Syntax

int Version;

Remarks

Holds version number.

This property is always readable.

3.4.4. Methods

addDirectIOListener Method

Syntax

public void addDirectIOListener (DirectIOListener *listener***);**

Parameter	Description
listener	DirectIOListener listener interface

Remarks

Add listener to be invoked when occur DirectIOEvent event.

If *listener* is null, do nothing.

Only one listener can be added. If this method is executed while the listener has already been set, it will be overwritten.

addStatusUpdateListener Method

Syntax

public void addStatusUpdateListener (StatusUpdateListener *listener***);**

Parameter	Description
listener	StatusUpdateListener listener interface

Remarks

Add listener to be invoked when occur StatusUpdateEvent event.

No exception is thrown if *listener* is null.

Only one listener can be added. If this method is executed while the listener has already been set, it will be overwritten.

claim Method

Syntax

public void claim (int *timeout*) throws PosException;

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

If zero, the method attempts to claim the device, then returns the appropriate status immediately.

If PosConst.FOREVER, the method waits as long as needed until exclusive access is satisfied.

Remarks

Call this method to request exclusive access to the device. Many devices require an application to claim them before they can be used.

When successful, the **Claimed** property is changed to **true**.

When the **claim** method is performed, a connection with a POS Printer device is established, and confirmed that processing is the possible situation. When it's possible to process it, the **claim** method is normally completed.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	An invalid *timeout* parameter was specified.
PosConst.E_NOHARDWARE	POS Printer is off-line. Perform again after recovering from error conditions. Even this error occurs when the printer status error like cover-open or paper-out.
PosConst.E_FAILURE	Could not establish a connection to POS Printer. The communication setting is wrong, or another instance may be already using.
PosConst.E_TIMEOUT	Another application has exclusive access to the device and did not relinquish control before *timeout* milliseconds expired. Or POS priner was not available condition before *timeout* milliseconds expired.

clearOutput Method

Syntax

public void clearOutput ();

Remarks

The rotation mode by **rotatePrint** method and the transaction mode by **transactionPrint** method are canceled.

Exception

PosException may be thrown. Please refer PosException class for details.

close Method

Syntax

public void close () throws PosException;

Remarks

Called to release the device and its resources.

If the **DeviceEnabled** property is true, then the device is first disabled.

If the **Claimed** property is **true**, then exclusive access to the device is first released.

Do not perform in event process (Event handler etc.).

Exception

PosException may be thrown. Please refer PosException class for details.

cutPaper Method

Syntax

public void cutPaper (int *percentage*) throws PosException;

The *percentage* parameter indicates the percentage of paper cut. The constant identifier PosPrinterConst.PTR_CP_FULLCUT or the value 100 causes a full paper cut.

Other values request a partial cut.

Remarks

Call to cut the receipt paper.

The correspondence depends on the printer. Please refer to Printer Specification.

An escape sequence embedded in a **printNormal** method call may also be used to cause a paper cut. If print data is buffered in the POS Printer (Requested to print but not printed yet), the cut is performed after printing the buffered data.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	An invalid percentage was specified, the printer does not have paper cuttter.
PosConst.E_NOHARDWARE	The POS Printer is either off or offline.
PosConst.E_BUSY	Cannot be executed while output is in progress.

directIO Method

Syntax

public void directIO (int *command*, int□ *data*, Object *object*) throws PosException;

Parameter	Description
command	Command number.
data	int array of size 1. Specific values vary by *command*.
object	Additional data. Specific values vary by *command*.

In this release, *command* parameters are as follows.

Value	Function
FtpConst.PTR_DIO_SET_QR_ENCODING	Set QR Code Encoding
FtpConst.PTR_DIO_GET_QR_ENCODING	Get QR Code Encoding.
FtpConst.PTR_DIO_SET_BINARY_CONVERSION	Set BinaryConversion property
FtpConst.PTR_DIO_GET_BINARY_CONVERSION	Get BinaryConversion property
FtpConst.PTR_DIO_SEND_BINARY_DATA	Send binary data

Remarks

Perform echa functionality following *command* parameter.

Exception

PosException may be thrown. Please refer PosException class for details.

PTR_DIO_GET_QR_ENCODING

Syntax

Parameter	Description
command	FtpConst.PTR_DIO_GET_QR_ENCODING
data	The current settings are retained.
object	Not used. Please set null

Remarks

Get QR Code encoding.

This setting is initialized to FtpConst.PTR_DIO_SHIFT_JIS by **claim** method.

Please refer to PTR_DIO_SET_QR_ENCODING for the meaning of *data*.

PTR_DIO_SET_QR_ENCODING

Syntax

Parameter	Description
command	FtpConst.PTR_DIO_SET_QR_ENCODING
data	FtpConst.PTR_DIO_SHIFT_JIS, FtpConst.PTR_DIO_UTF8 or FtpConst.PTR_DIO_GBK
object	Not used. Please set null

Remarks

Set QR Code encoding.

This setting is initialized to FtpConst.PTR_DIO_SHIFT_JIS by **claim** method.

If **BynaryConversion** is not PosConst.BC_NONE, the encoding is Shift JIS fixed.

If *data* is FtpConst.PTR_DIO_SHIFT_JIS, **printBarCode** method prints QR Code encoded in Shift JIS.

If *data* is FtpConst.PTR_DIO_UTF8, **printBarCode** method prints QR Code encoded in UTF-8.

If *data* is FtpConst.PTR_DIO_GBK, **printBarCode** method prints QR Code encoded in GBK.

PTR_DIO_GET_BINARY_CONVERSION

Syntax

Parameter	Description
command	FtpConst.PTR_DIO_GET_BINARY_CONVERSION
data	The current settings are retained.
object	Not used. Please set null

Remarks

Get **BinaryConversion**.

Please refer to PTR_DIO_SET_BINARY_CONVERSION for the meaning of *data*.

PTR_DIO_SET_BINARY_CONVERSION

Syntax

Parameter	Description
command	FtpConst.PTR_DIO_SET_BINARY_CONVERSION
data	PosConst.BC_NONE, PosConst.BC_NIBBLE or PosConst.BC_DECIMAL
object	Not used. Please set null

Remarks

Set **BinaryConversion**.

The binary conversion values are:

Value	Meaning
PosConst.BC_NONE	With no conversion.(This is the default.)
PosConst.BC_NIBBLE	Each byte is converted into two characters. Each data byte is converted as follows: First character = 0x30 + bits 7-4 of the data byte. Second character = 0x30 + bits 3-0 of the data byte. Example: Byte value 154 = 0x9A is converted into the characters U+0039 U+003A (= the string “9:”).
PosConst.BC_DECIMAL	Each byte is converted into three characters. Example 1: Byte value 154 = 0x9A becomes the characters U+0031 U+0035 U+0034 (= the string “154”). Example 2: Byte value 8 becomes the characters U+0030 U+0030 U+0038 (= the string “008”).

PTR_DIO_SEND_BINARY_DATA

Syntax

Parameter	Description
command	FtpConst. PTR_DIO_SEND_BINARY_DATA
data	Not used. Please set null
object	Specify the binary data as byte□ type Max array size is 192 KB (192 x 1024 bytes).

Remarks

Send the binary data.

The operation of other methods may be affected depending on the data to be sent.

Transacting by **transactionPrint** and buffering by **rotatePrint** are not performed.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	One of the following errors occurred. - *object* is not byte□ type - Data length is out of range. - *object* is null.
PosConst.E_NOHARDWARE	The POS Printer is either off or offline.
PosConst.E_FAILURE	FTP-POS is in error state. Perform again after recovering from error condition.
PosConst.E_BUSY	Cannot be executed while output is in progress.

markFeed Method

Syntax

public void markFeed (int *type*) throws PosException;

The *type* parameter indicates the type of mark sensored paper handling.

The parameter value of *type* is as follows.

Value	Meaning
PosPrinterConst.PTR_MF_TO_NEXT_TOF	Feed the Mark Sensed paper to the next paper's top of form.

Remarks

This method is used to utilize the printer's mark sensor for receipt paper..

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	An invalid value was specified.
PosConst.E_NOHARDWARE	The POS Printer is either off or offline.
PosConst.E_BUSY	Cannot be executed while output is in progress.

open Method

Syntax

public void open (PrinterConfiguration *printerConfiguration*, **Context** *context*) **throws PosException;**

Parameter	Description
printerConfiguration	Printer configuration.
context	Application context

Remarks

Call to open a device for subsequent I/O.

If it is not USB connection, *context* can be null.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	The target device is already open.
PosConst.E_EXTENDED	**ErrorCodeExtended** = FtpConst.EPTR_NEEDCONTEXT: Application context required.

printBarCode Method

Syntax

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

Parameter	**Description**
station	The printer station to be used. Set to PosPrinterConst.PTR_S_RECEIPT.
data	Character string of barcode.
symbology	Barcode symbol type to use. (Please refer to the following values.)
height	Barcode height. Expressed in the unit of measure given by **MapMode**. Set 1 to 255dot.
width	Barcode width. Expressed in the unit of measure given by **MapMode**. When normal / upside-down mode, *width* can be set in the range of **RecLineWidth**. When rotate printing 90° to the left / right, the range of *width* depends on the printer model.
alignment	Placement of the barcode. See values below.
textPosition	Placement of the readable character string.

Supported barcode depends on the printer model. Please refer to Printer Specification.

The *symbology* parameter has one of following values.

Value	**Symbol type**
PosPrinterConst.PTR_BCS_UPCA	UPC-A
PosPrinterConst.PTR_BCS_UPCE	UPC-E
PosPrinterConst.PTR_BCS_EAN8	EAN 8 (= JAN 8)
PosPrinterConst.PTR_BCS_JAN8	JAN 8 (= EAN 8)
PosPrinterConst.PTR_BCS_EAN13	EAN 13 (= JAN 13)
PosPrinterConst.PTR_BCS_JAN13	JAN 13 (= EAN 13)
PosPrinterConst.PTR_BCS_ITF	Interleaved 2 of 5
PosPrinterConst.PTR_BCS_Codabar	Codabar(NW-7)
PosPrinterConst.PTR_BCS_Code39	Code 39
PosPrinterConst.PTR_BCS_Code128	Code 128
PosPrinterConst.PTR_BCS_PDF417	PDF417
PosPrinterConst.PTR_BCS_MAXICODE	MaxiCode(Mode 2)
PosPrinterConst.PTR_BCS_GS1DATABAR	GS1DataBar Omnidirectional
PosPrinterConst.PTR_BCS_GS1DATABAR_S	GS1DataBar Stacked Omnidirectional
PosPrinterConst.PTR_BCS_GS1DATABAR_E	GS1DataBar Expanded
PosPrinterConst.PTR_BCS_GS1DATABAR_E_S	GS1DataBar Expanded Stacked
PosPrinterConst.PTR_BCS_QRCODE	QR Code
PosPrinterConst.PTR_BCS_OTHER(501)	MaxiCode(Mode 3)
PosPrinterConst.PTR_BCS_OTHER+1(502)	MaxiCode(Mode 4, Mode 5)
PosPrinterConst.PTR_BCS_OTHER+2(503)	MaxiCode(Mode 6)
PosPrinterConst.PTR_BCS_OTHER+5(506)	GS1DataBar Truncated
PosPrinterConst.PTR_BCS_OTHER+6(507)	GS1DataBar Stacked
PosPrinterConst.PTR_BCS_OTHER+8(509)	GS1DataBar Limited

The *alignment* parameter values are as follows. The correspondence depends on the printer. Please refer to Printer Specification. When a rotated 90° left or right mode by **rotatePrint**, please specify PosPrinterConst.PTR_BC_LEFT.

Value **Meaning**

PosPrinterConst.PTR_BC_LEFT Align with the left-most print column. (It is executed in the direction of the print data. Therefore, when turning 180 degrees, it will be right-most aligned in the POS Printer printing direction.)

PosPrinterConst.PTR_BC_CENTER

Align in the center of the station.

PosPrinterConst.PTR_BC_RIGHT

Align with the right-most print column. (It is executed in the direction of the print data. Therefore, when turning 180 degrees, it will be left-most aligned in the POS Printer printing direction.)

Other Values Distance from the left-most print column to the start of the barcode. Expressed in the unit of measure given by **MapMode**.

If the actual barcode width (calculating using *width*) + distance from left-most print column exceeds **RecLineWidth**, it returns the value PosConst.E_ILLEGAL.

The t*extPosition* parameter values are as follows. The correspondence depends on the printer. Please refer to Printer Specification.

Value **Meaning**

PosPrinterConst.PTR_BC_TEXT_NONE

No text is printed. Only print the barcode.

PosPrinterConst.PTR_BC_TEXT_ABOVE

Print the text above the barcode.

PosPrinterConst.PTR_BC_TEXT_BELOW

Print the text below the barcode.

Remarks

Call to print a barcode on the specified printer station.

The barcode printable condition for each *symbology* is indicated below.

symbology	Printable characters	String length	*width* (dots)
UPC-A	’0’ – ’9’	11 to 12	95 to 864
UPC-E		11 to 12	51 to 864
EAN8 JAN8		7 to 8	67 to 864
EAN13 JAN13		12 to 13	95 to 864
Code 39	‘0’ – ’9’, ‘A’ – ’Z’, space, ‘$’, ‘%’, ‘+’, ‘–‘, ‘.’, ‘/’ (Start and stop character ’*’ is automatically added.)	1 to 34	47 to 864
ITF	’0’ – ’9’	2 to 62	27 to 864
Codabar	’0’ – ’9’, ‘A’ – ’D, ‘$’, ‘+’, ‘–‘, ‘.’, ‘/’, ‘:’	3 to 71	41 to 864

Code 128	Code Set A: 0x00 – 0x5F Code Set B 0x20 – 0x7F Code Set C 0x00 – 0x63 The character including ″{″ is excluded. Details are described later.	3 to 51	46 to 864
PDF417	0x00 – 0xFF	1 to 410byte	158 to 864
MAXICODE (Mode 2)	Primary Message Number ‘0’ – ‘9’ Secondary Message 0x01 – 0xFF	18 to 85	228
MAXICODE (Mode 3)	Primary Message Postal code ‘0’ – ‘9’ , ‘A’ – ‘Z’ , space, “ “#$%&’()*+,–./ ” Country code Service code Number ‘0’ – ‘9’ Secondary Message 0x01 ～ 0xFF	15 to 85	228
MAXICODE (Mode 4, Mode 5)	0x01 – 0xFF	1 to 80	228
MAXICODE (Mode 6)	0x01 – 0xFF	1 to 80	228
QR Code	Numeric mode ‘0’ – ’9’ Alphanumeric mode ‘0’ – ’9’, ‘A’ – ’Z’, space, ‘$’, ‘%’, ‘*’, ‘+’, ‘–‘, ‘.’, ‘/’, ‘:’ 8-bit Byte mode 0x00 – 0xFF Shift JIS Kanji mode 1st byte : 0x81 – 0x9F , 0xE0 – 0xEA 2nd byte : 0x40 – 0x7E , 0x80 – 0xFC	1 to 2048byte	21 or more

GS1DataBar (Omnidirectional) GS1DataBar (Truncated) GS1DataBar (Stacked) GS1DataBar (Stacked Omnidirectional) GS1DataBar (Limited)	’0’ – ’9’	13 to 14	864 or less
GS1DataBar (Expanded) GS1DataBar (Expanded Stacked)	‘0’–‘9’, ‘A’–‘Z’ , ‘a’–‘z’, space, ‘!’, ‘‘‘’, ‘%’ , ‘&’, ‘‘’, ‘(‘, ‘)’, ‘*’, ‘+’, ‘,’ , ‘–‘, ‘.’, ‘/’, ‘:’, ‘;’, ‘<’, ‘=’, ‘>’, ‘?’, ‘_’, FNC1(0x1D)	1 to 77 (Only Numeric) *) Maximum data length changes according to the kind of character.	864 or less

The encoding of each barcode is as follows.

Type	Encoding
UPC-A	Ascii
UPC-E	
EAN8 JAN8	
EAN13 JAN13	
Code 39	
ITF	
Codabar	
Code 128	
PDF417	Windows 1252
MAXICODE	UTF-8
QR Code	If **BinaryConversion** is PosConst.BC_NONE, **printBarCode** method encode QR Code with the encoding specified in **directIO** method. Please refer to PTR_DIO_SET_QR_ENCODING. If **BinaryConversion** is not PosConst.BC_NONE, convert *data* parameter according to the value of **BinaryConversion** and print QR Code.

GS1DataBar	UTF-8

Print width (dot) of the final barcode is printed with the parameter value closest to *width* within the range not exceeding barcode printable condition.

Notes for the barcode printing

1. Print width (dot) of the final barcode is printed with the parameter value closest to *width* within the range not exceeding barcode printable condition.

2. When CODE39 is printed, "*" of the start character and the stop character is automatically added. It is not necessary to set it as a printing character.

3. When ITF is specified, the number of print characters must be specified as an even number. If you specify an odd number, PosExceptoin is thrown.

4. When specifying CODABAR, the beginning and the end of the character are always either "A" to "D". Therefore, it is necessary to specify the character of the print by three characters or more in total. If it is 2 characters or less, PosExceptoin is thrown.

5. When UPC-E is specified, the code is expanded based on the following table. Left code of UPC-A indicates the first 2 to 6 characters. Right code of UPC-A indicates the first 7 to 11 characters. Compressed code is printed as UPC-E. PosExceptoin is thrown if the first character of UPC-A is not 0 or if a character not based on the table below is specified.
   Example 05810000226 → It is converted into 58226.
   09859363583 → PosExceptoin is thrown.

Maker code Left cord of UPC-A					Item code Right cord of UPC-A					Compressed 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

6. When printing CODE 128, set the character by the followings.

   1. Specify either of {C "{A"" {B """ for the first data of the barcode. As a result, it is set to CODE A, CODE B, and CODE C respectively.
   2. Specify the function code as follows.
      FNC 1 = "{1", FNC 2 = "{2", FNC 3 = "{3", FNC 4 = "{4"
      In CODE C, only FUNC 1 is valid. In CODE C, PosExceptoin is thrown if you specify something other than FUNC 1.
   3. In CODE B, specify "{{" to set "{".
   4. Specify "{S" to set SHIFT. At this setting, the code set of one character is shifted to CODE A or CODE B. PosExceptoin is thrown when specifying it in CODE C.

The characters that can be printed at CODE A, CODE B, and CODE C are as follows.

【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)
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)	Followings are used by assigning ”{“		
P	P	48(30H)	FNC 3 “{3”	FNC 3 “{3”	
Q	Q	49(31H)	FNC 2 “{2”	FNC 2 “{2”	
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”

7. PDF417 increases the number of barcode rows as data length increases. Calculates the vertical and horizontal sizes and prints them according to the specified *width* parameter. *height* parameter is not used. Max height is 255 dots.
   If it is judged that printing is impossible, PosExceptoin will be thrown.
   If the **BinaryConversion** is PosConst.BC_NONE, PDF417 will be printed using *data* encoded with UTF-8.
   If the **BinaryConversion** is not PosConst.BC_NONE, PDF417 will be printed using *data* converted according to the value of **BinaryConversion**.

8. MAXICODE symbol size is fixed, so *width* and *height* parameters are not used. PosException is thrown only if *width* parameter is less than or equal to 0, or *height* parameter is less than or equal to 0 or greater than 255 dots. Since the data length after encoding is unknown for MAXICODE, the data length cannot be checked accurately. Judgment is made with a little margin, and data is printed when it is judged that printing is possible. For mode 2 and mode 3, PosException is thrown when the Secondary Message exceeds 70 bytes. For modes 4 to 6, PosException is thrown when data exceeds 80 bytes. The difference between mode 4 and mode 5 is the difference in error correction level. If the amount of data is small (50 bytes or less), use mode 5 with a high error correction level.

9. QR Code is the same size in both the vertical and horizontal, it prints QR Code by the approximate value of the value specified by the *width* parameter. If the *height* parameter is less than or equal to 0, PosExceptoin is thrown. The print size is decided by data length, the character mode, the module size, and the error correction level of the QR Code.
   The print size of the QR Code is determined by calculating an approximate value of the *width* parameter value from 3 x 3 or 4 x 4 dots of the module size. Error correction Level is M.

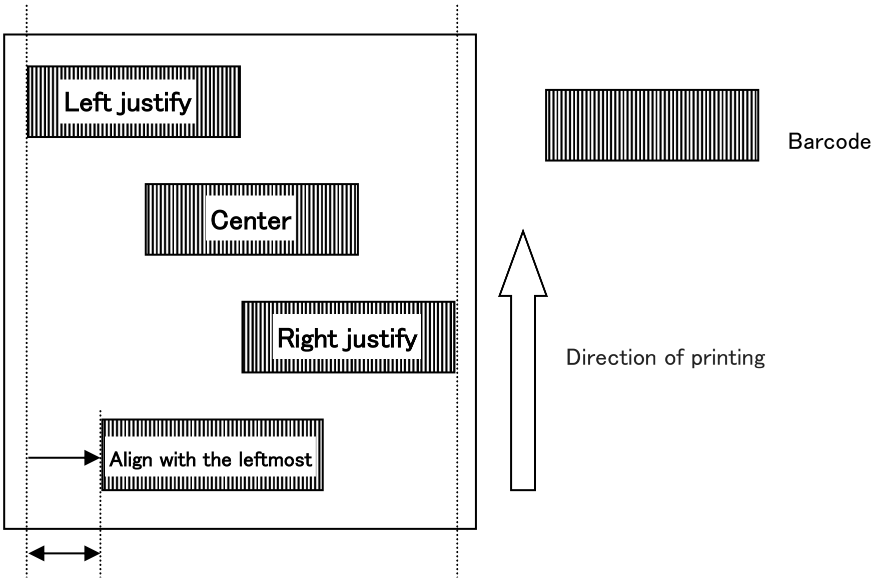
10. The size and the module size of GS1 DataBar are decided from the approximate value of the *width* parameter value. *height* parameter is not used.
    The data length printable by GS1 DataBar Expanded and Expanded Stacked is a maximum of 77 characters (numerals only). However, data length that can be printed is different according to the kind of the specified character.

11. Set the presence and the position of HRI character (Human Readable Interpretation) by specifying TextPosition.
    Specify the position of the HRI character in PosPrinterConst.PTR_BC_TEXT_ABOVE (upper side of the barcode) or PosPrinterConst.PTR_BC_TEXT_BELLOW (lower side of the barcode). The font size is changed according to the width of the barcode and the HRI character is printed. The HRI character is not printed in QR Code, MAXICODE, PDF417 and GS1DataBar of the two dimension barcode.

12. The escape sequence is canceled at ESC | N (normal) or at the beginning of the character string. Therefore, if barcode printing is done with bold designation, HRI characters will also be bold.

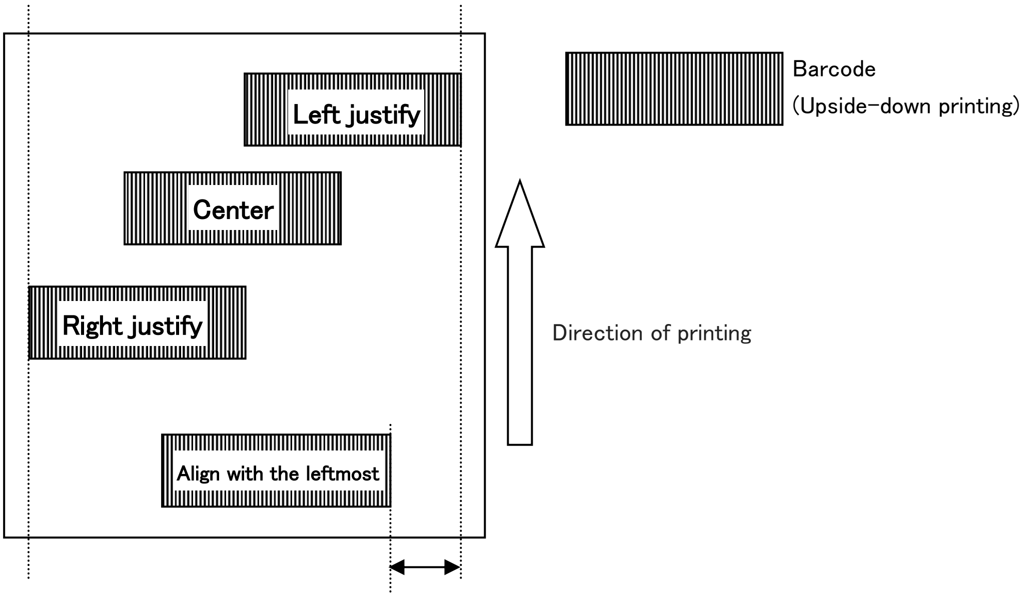
About the rotation print of the barcode using RotateSpecial and rotatePrint

At normal printing, the difference in print position by *alignment* specification is as follows.



Position set with *alignment* from left end.

At Upside-down printing, the difference in print position by *alignment* specification is as follows.



Position set with *alignment* from left end

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst..E_ILLEGAL	One of the following errors occurred. – *station* does not exist. – *station* does not support barcode printing. – *height* or *width* is zero or is too big. – *symbology* is not supported. – Not all characters in *data* are supported by *symbology*. – *alignment* is invalid or too big. (If you specify Alignment in absolute position, the total of the specified value of *alignment* and the actual printing width of the barcode (Value calculated with the value closest to *width*) exceeds the printable width.) – *textPosition* is invalid.
PosConst.E_NOHARDWARE	The POS Printer is either off or offline.
PosConst.E_BUSY	Cannot be executed while output is in progress.

printBitmap Method

Syntax

public void printBitmap (int *station*, Bitmap *bitmap*, int *width*, int *alignment*) throws PosException;

Parameter	Description
station	The printer station to be used. Set to PosPrinterConst.PTR_S_RECEIPT.
bitmap	The instance of android.graphics.Bitmap class
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
PosPrinterConst.PTR_BM_ASIS	Print the bitmap with one bitmap pixel per printer dot.
Other Values	Bitmap width expressed in the unit of measure given by **MapMode**. Specify it in units defined by **MapMode**. A valid value is from 1 to **RecLineWidth**.

The *alignment* parameter has one of the following values. When a rotated 90° left or right mode by **rotatePrint**, please specify PosPrinterConst.PTR_BM_LEFT.

Value	Meaning
PosPrinterConst.PTR_BM_LEFT	Align with the left-most print column.
PosPrinterConst.PTR_BM_CENTER	Align in the center of the station.
PosPrinterConst.PTR_BM_RIGHT	Align with the right-most print column.
Other Values	Distance from the left-most print column to the start of the bitmap. Expressed in the unit of measure given by **MapMode**. The sum of *width* and this does not exceed the parameter limit of *width*.

Remarks

Prints a bitmap on the specified printer station. The bitmap is converted to monochrome and then printed.
printBitmap sends the bitmap data to the printer at the time this method is called, cannot archieve good performance. It is recommended to use **setBitmap** and escape sequences.
The *width* parameter controls transformation of the bitmap. If *width* is PosPrinterConst.PTR_BM_ASIS, then no transformation is performed. The bitmap is printed with one bitmap pixel per printer dot.
If *width* is non-zero, 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.
The printable bitmap height is 1023 dot or less.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	One of the following errors occurred. - *station* does not exist. - *station* does not support bitmap printing. - *width* is too big. - *alignment* is invalid or too big.
PosConst.E_NOHARDWARE	The POS Printer is either off or offline.

printNormal Method

Syntax

public void printNormal (int *station*, String *data*) throws PosException;

Parameter	Description
station	The printer station to be used. Set to PosPrinterConst.PTR_S_RECEIPT.
data	The characters to be printed. May consist of printable characters, escape sequences, carriage returns(U+000D), and line feeds(U+000A).

Remarks

Prints *data* on the printer *station*.

Call this method when you print Data on the POS Printer. Print data exceeding the maximum charecters in one line is printed on the next line.

Special character values within *data* are:

Value	Meaning
Newline/Line Feed(U+000A)	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(U+000D)	If a Carriage Return immediately precedes a Line Feed, then it is ignored. The carriage return behaves like a line feed.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	The specified s*tation* does not exist.
PosConst.E_NOHARDWARE	The POS Printer is either off or offline.
PosConst.E_BUSY	Cannot be executed while output is in progress.

release Method

Syntax

public void release () throws PosException;

Remarks

Call this method to release exclusive access to the device.

If the **DeviceEnabled** property is true, and the device is an exclusive-use device, then the device is first disabled.

Do not perform in event process (Event handler etc.).

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	The application does not have exclusive access to the device.

removeDirectIOListener Method

Syntax

public void removeDirectIOListener (DirectIOListener ***listener*****);**

Parameter	Description
listener	DirectIOListener listener interface

Remarks

Remove listener.

If *listener* is null, do nothing.

removeStatusUpdateListener Method

Syntax

public void removeStatusUpdateListener (StatusUpdateListener *listener***);**

Parameter	Description
listener	StatusUpdateListener listener interface

Remarks

Remove listener.

If *listener* is null, do nothing.

rotatePrint Method

Syntax

public void rotatePrint (int *station*, int *rotation*) throws PosException;

Parameter	**Description**
station	Set to PosPrinterConst.PTR_S_RECEIPT.
rotation	Direction of rotation. See values below.

The *rotation* parameter has one of the following values.

Value	**Meaning**
PosPrinterConst.PTR_RP_RIGHT90	Start rotated printing 90° to the right (clockwise)
PosPrinterConst.PTR_RP_LEFT90	Start rotated printing 90° to the left (counter-clockwise)
PosPrinterConst.PTR_RP_ROTATE180	Start rotated printing 180°, that is, print upside-down
PosPrinterConst.PTR_RP_BARCODE	Start rotated the barcode printing. This value is ORed with one of the above start rotated print values.
PosPrinterConst.PTR_RP_BITMAP	Start rotated bitmap printing. This value is ORed with one of the above start rotated print values. The bitmap printed by **printBitmap** is rotated. The escape sequence (bitmap and stamp) specified in the **printNormal** is not rotated.
PosPrinterConst.PTR_RP_NORMAL	End rotated printing.

Remarks

If *rotation* includes PosPrinterConst.PTR_RP_ROTATE180, then upside-down print mode is entered. Subsequent calls to **printNormal** will print the data upside-down until **rotatePrint** is called with *rotation* set to PosPrinterConst.PTR_RP_NORMAL.

Each print line is rotated by 180°. Lines are printed in the order that they are sent, with the start of each line justified at the right margin of the printer station. Only print methods **printNormal** may be used while in upside-down print mode.

If *rotation* includes PosPrinterConst.PTR_RP_RIGHT90 or PosPrinterConst.PTR_RP_LEFT90, then sideways print mode is entered. Subsequent calls to **printNormal** will buffer the print data until **rotatePrint** is called with *rotation* set to PosPrinterConst.PTR_RP_NORMAL.

(In this case, the above method only buffers the data – it does not initiate printing. In this case, each method succeeds regardless of error conditions of the POS Printer. For example, while buffering print data by **rotatePrint**, if the POS Printer is turned off, it will not return an error when calling each method.)

When rotated printing 90° mode, the service object analyzes the buffered characters by **printNormal**, and determines the horizontal width according to the maximum width of all the lines.

The printable maximum width depends on the printer model. Please refer to Printer Specification. The overflowed characters will be printed on the next line in the page. Also, if the width of the characters is more than twice by the escape sequence, it calculates the value multiplied with the scaling factor.

If **printBitmap** is performed when the upside-down print mode, the upside-down bitmap is printed.

The saved bitmap by **setBitmap** when the upside-down print mode is also printed upside-down. (Even if in nomal print mode, it is printed upside-down.)

If *rotation* includes PosPrinterConst.PTR_RP_NORMAL, then rotated print mode is exited. 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 are treated as one message.

If *rotation* includes PosPrinterConst.PTR_RP_BARCODE and/or PosPrinterConst.PTR_RP_BITMAP, then any barcodes (printed with **printBarCode**) and/or bitmaps (printed with **printBitmap**) submitted for printing during the r**otatePrint** processing cycle will also be rotated. The rotated print of barcode and bitmap is canceled by specifying PosPrinterConst.PTR_RP_NORMAL.

When t**ransactionPrint** mode, **rotatePrint** cannot print correctly if the direction of rotation of character string, barcode or bitmap are different. When combination with **transactionPrint** mode, set both PosPrinterConst.PTR_RP_BARCODE and PosPrinterConst.PTR_RP_BITMAP.

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

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	The specified *station* does not exist, or the *station* does not support the specified rotation. It is in different rotation mode. Set to PosPrinterConst.PTR_RP_NORMAL or cancel the rotated print by **clearOutput** and then perform again.
PosConst.E_NOHARDWARE	The POS Printer is either off or offline.
PosConst.E_BUSY	Cannot be executed while output is in progress.

setBitmap Method

Syntax

public void SetBitmap (int *bitmapNumber*, **int** *station*, **Bitmap** *bitmap*, **int** *width*, **int** *alignment*) **throws PosException;**

Parameter	Description
bitmapNumber	The number to be assigned to this bitmap. Valid bitmap numbers are 1 through 255.
station	Set to PosPrinterConst.PTR_S_RECEIPT.
bitmap	Instance of android.graphics.Bitmap. If set to null, then the bitmap is unset from the POS Printer.
width	Printed width of the bitmap to be performed. (See **printBitmap** for values.)
alignment	Placement of the bitmap. (See **printBitmap** for values.)

Remarks

Saves information about a bitmap for later printing.

The bitmap may then be printed by calling the **printNormal** method with the print bitmap escape sequence in the print data.

The savable bitmap size is the w*idth* of less than **RecLineWidth** dot (if *alignment* is distance from the left-most print column, *width* + *alignment* <= **RecLineWidth**). The height depends on the printer model. Please refer to Printer Specification. If this condition is not satisfied, an exception indicating PosPrinterConst.EPTR_TOOBIG is thrown.

When setBitmap is performed, the bitmap is saved to the nonvolatile memory in POS Printer. Therefore after perfoming **release** method, the saved bitmap is valid. In other words, bitmap printing by escape sequence is possible after once saving the bitmap by **setBitmap**.

When this method is executed, the status may be temporarily released, and notified by **StatusUpdateEvent** or **DirectIOEvent**.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	One of the following errors occurred. – *station* does not exist. – *station* does not support bitmap printing. – *width* is too big. – *alignment* is invalid or too big.
PosConst.E_FAILURE	FTP-POS is in error state. Perform again after recovering from error condition.
PosConst.E_BUSY	Cannot be executed while output is in progress.
PosConst.E_EXTENDED	**ErrorCodeExtended** = PosPrinterConst.EPTR_TOOBIG: The bitmap is either too wide to print without transformation, or it is too big to transform.

setLogo Method

Syntax

public void setLogo (int *location*, String *data*) throws PosException;

Parameter	Description
location	The logo to be set. May be PosPrinterConst.PTR_L_TOP or PosPrinterConst.PTR_L_BOTTOM.
data	The characters that produce the logo. May consist of printable characters, escape sequences, carriage returns (U+000D), and line feeds (U+000A). It cannot contain logo escape sequences.

Remarks

Saves a data string as the top or bottom logo.

A logo may then be printed by calling the **printNormal** method with the print top logo or print bottom logo escape sequence in the print data.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_BUSY	Cannot be executed while output is in progress.
PosConst.E_ILLEGAL	An invalid *location* was specified.

transactionPrint Method

Syntax

public void transactionPrint(int *station*, int *control*) throws PosException;

Parameter	Description
station	Set to PosPrinterConst.PTR_S_RECEIPT.
control	Transaction control. See values below.

The *control* parameter has one of the following values.

Value	Meaning
PosPrinterConst.PTR_TP_TRANSACTION	Begin a transaction
PosPrinterConst.PTR_TP_NORMAL	End a transaction by printing the buffered data

Remarks

Enters or exits transaction mode.

If *control* is PosPrinterConst.PTR_TP_TRANSACTION, then transaction mode is entered. Subsequent calls to **printNormal**, **cutPaper**, **markFeed**, **rotatePrint**, **printBarCode**, and **printBitmap** will buffer the print data until **transactionPrint** is called with the *control* parameter set to PosPrinterConst.PTR_TP_NORMAL. (In this case, the print methods only validate the method parameters and buffer the data – they do not initiate printing. In this case, each method succeeds regardless of error conditions of the POS Printer. For example, while buffering print data by **transactionPrint**, if the POS Printer is turned off, it will not return an error when calling each method.)

If *control* is PosPrinterConst.PTR_TP_NORMAL, then transaction mode is exited. If some data was buffered by calls to the methods **printNormal**, **cutPaper**, **rotatePrint**, **printBarCode**, **printBitmap**, and **markFeed**, then the buffered data is printed. The entire transaction is treated as one message.

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

There is notes in execution of the **rotatePrint** method.

Printing by **rotatePrint** (Rotate printing mode), **printNormal**, and **rotatePrint** (PosPrinterConst.PTR_RP_NORMAL) is not performed until you execute the **transactionPrint** method to exit the transaction mode. Also, when calling **rotatePrint** (Rotate printing mode) and **transactionPrint** (PosPrinterConst.PTR_TP_TRANSACTION), the buffering by the **transactionPrint** method has a higher priority, so the data buffered during this time will not be rotated and printed correctly. Therefore, if you execute the **rotatePrint** method, execute it after the **tansactionPrint** method.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	The specified *station* does not exist.
PosConst.E_NOHARDWARE	The POS Printer is either off or offline.

3.4.5. Events

DirectIOEvent Event

Syntax

Interface

public interface DirectIOListener extends java.util.EventListener

Method

void directIOOccurred (DirectIOEvent *e*);

This event uses the following properties of parameter *e*.

Property	Description
EventNumber	Event number.
Data	Additional numeric data.
Object	Additional data.

EventNumber is set to one of the following values.

Value	Meaning
FtpConst.PTR_DIE_PRESENTER_ERROR	Presenter is error.
FtpConst.PTR_DIE_PRESENTER_OK	Presenter is OK.
FtpConst.PTR_DIE_LOW_BATTERY	The battery is low.
FtpConst.PTR_DIE_BATTERY_OK	The battery has recovered from low.

Remarks

Notified when an event of unique features of FTP-POS has occurred.

FtpConst.PTR_DIE_PRESENTER_ERROR

Property	Description
EventNumber	FtpConst.PTR_DIE_PRESENTER_ERROR
Data	Not used.
Object	Not used.

Remarks

Notify when a presenter error occurs.

FtpConst.PTR_DIE_PRESENTER_OK

Property	Description
EventNumber	FtpConst.PTR_DIE_PRESENTER_OK
Data	Not used.
Object	Not used.

Remarks

Notify when recovering from the presenter error.

StatusUpdateEvent Event

Syntax

Interface

public interface StatusUpdateListener extends java.util.EventListener

Method

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

This event uses the following properties of parameter *e*.

Property	Description
Status	Describing the type of status change.

Value	Meaning
PosPrinterConst.PTR_SUE_COVER_OPEN	Printer cover is open.
PosPrinterConst.PTR_SUE_COVER_OK	Printer cover is closed.
PosPrinterConst.PTR_SUE_REC_EMPTY	No receipt paper.
PosPrinterConst.PTR_SUE_REC_NEAREMPTY	Receipt paper is low.
PosPrinterConst.PTR_SUE_REC_PAPEROK	Receipt paper is ready.
PosConst.SUE_POWER_ONLINE	The device is powered on and ready for use. (Can be fired when **PowerNotify** = PosConst.PN_ENABLED.)
PosConst.SUE_POWER_OFF_OFFLINE	The device is either off or offline. (Can be fired when **PowerNotify** = PosConst.PN_ENABLED.)

Remarks

Notified when a significant device status change has occurred.

When a device is enabled, then the Control may fire the first **StatusUpdateEvent**s to inform the application of the device state.

3.5. PrinterConfiguration class

public final class PrinterConfiguration extends Object

Configure the connected printer.

3.5.1. Constructor

Syntacx

public PrinterConfiguration(PrinterModel *model*, **ConnectionType** *connection*,
String *deviceName*, **PrintingWidth** *width*, **PrintingResolution** *resolution*,
PrintingQuality *quality*)
throws PosException;

Parameter	Description
model	Select model.
connection	Select Connection type.
deviceName	Selec device name. See below.
width	Select printing width.
resolution	Select printing resolution.
quality	Configure printing quality.

Remarks

Create PrinterConfiguration class instance.

Specify *deviceName* as follows.

connection	*deviceName*
ConnectionType.Usb	The value of android.hardware.usb.UsbDevice#getDeviceName

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	Meaning
PosConst.E_ILLEGAL	Invalid parameter.

3.6. PrintingQuality class

public final class PrintingQuality extends Object

Set the printing quality

3.6.1. Static methods

createWithEnergy method

Syntax

public static PrintingQuality createWithEnergy(int *paperType*, **int** *energy*) **throw PosException**

Parameter	**Description**
paperType	Set the printing quality. The type that can be specified depends on the printer. Please refer to Printer Specification.
energy	Adjust the energy. To use it by default, set PrintingQuality.DEFAULT_ENERGY. When making adjustments, refer to the GS E command in the command specifications or the product specifications of the printer.

Remarks

Create an instance of PrintingQuality class.

Return

An instance of PrintingQuality class.

Exception

PosException may be thrown. Please refer PosException class for details.

ErrorCode	**Meaning**
PosConst.E_ILLEGAL	The parameter has an invalid value.

3.7. ConnectionType enum

public enum ConnectionType

Enum of connection type.

Value	Description
Usb	USB

3.8. PrintingRosolution enum

public enum PrintingResolution

Enum of resolution.

Value	Description
Resolution203	203dpi

3.9. PrintingWidth enum

public enum PrintingWidth

Enum of printing width.

Value	Description
Widith384Dots	384 dots
Width432Dots	432 dots
Width576Dots	576 dots
Width832Dots	832 dots

3.10. PrinterModel enum

pubic enum PrinterModel

Enum of supported printer model

Value	Description
Ftp_62gDsl000	FTP-62GDSL000 series
Ftp_62hDsl100	FTP-62HDSL100 series
Ftp_629Dsl350	FTP-629DSL350 series

3.11. PosException class

public final class PosException extends java.lang.Exception

Thrown if an exception occurs within this SDK.

3.11.1. Properties

ErrorCode Property

Syntax

int ErrorCode;

Remarks

Get error code

Value	**Description**
PosConst.E_CLOSED	Attempt was made to access a closed device.
PosConst.E_NOTCLAIMED	Attempt was made to access an exclusive-use device that must be claimed before the method or property set action can be used.
PosConst.E_DISABLED	Cannot perform operation while device is disabled.
PosConst.E_ILLEGAL	Attempt was made to perform an illegal or unsupported operation with the device, or an invalid parameter value was used.
PosConst.E_NOHARDWARE	The POS Printer is not powered on, or off-line.
PosConst.E_FAILURE	The device cannot perform the requested procedure, even though the device is connected to the system, powered on, and on-line.
PosConst.E_TIMEOUT	Timed out waiting for a response from the device,
PosConst.E_EXTENDED	A class-specific error condition occurred. The error condition code is available in the **ErrorCodeExtended** property.

ErrorCodeExtended Property

Syntax

int ErrorCodeExtended;

Remarks

Get extended error code.

When **ErrorCode** is PosConst.E_EXTENDED, this property is set to error value.

Value	Description
PosPrinterConst.EPTR_COVER_OPEN	The printer cover is opened. To recover from error, close the prnter cover. In this case, it is not necessary to disable, **release**, and **close**.
PosPrinterConst.EPTR_REC_EMPTY	The receipt is out of paper. To recover from error, replenish the receipt paper. In this case, it is not necessary to disable, **release** and **close**.
PosPrinterConst.EPTR_TOOBIG	The bitmap is either too wide to print without transformation, or it is too big to transform.
FtpConst.EPTR_POWERSUPPLY	Printer power-supply voltage error is occurred. There is no recovery method, so please **close**.
FtpConst.EPTR_CUTTER	The cutter error is occurred. The printer cannot normally cut because something stiff may be inserted to cutter. To recover from error, remove it. In this case, it is not necessary to disable, **release** and **close**.
FtpConst.EPTR_HARDWARE	The hardware error is occurred. There is no recovery method, so please **close**.
FtpConst..EPTR_HEADHOT	The abnormal head temperature is occurred. Possible to resume when the head temperature goes down. In this case, it is not necessary to disable, **release** and **close**.
FtpConst.EPTR_MARK	The mark can not be scaned. This error is occurred when **markFeed** method is performed. There is a possibility of paper jam. To recover from error, remove it. In this case, it is not necessary to disable, **release** and **close**.
FtpConst.EPTR_PRESENTER	The presenter error is occurred. It is possible that the presenter can not feed such as paper jam. To recover from error, remove it. In this case, it is not necessary to disable, **release** and **close**.
FtpConst.EPTR_NEEDCONTEXT	Application context is required.

3.12. DirectIOListener interface

public interface DirectIOListener extends java.util.EventListener

Listener interface to receive **DirectIOEvent** event.

3.12.1. Methods

directIOOccurred Method

Syntax

void directIOOccurred (DirectIOEvent *e*);

Remarks

For more information about this method, please refer to DirectIOEvent.

3.13. StatusUpdateListener interface

public interface StatusUpdateListener extends java.util.EventListener

Listener interface to receive **StatusUpdateEvent** event.

3.13.1. Methods

statusUpdateOccurred Method

Syntax

void statusUpdateOccurred (StatusUpdateEvent *e***);**

Remarks

For more information about this method, please refer to StatusUpdateEvent.

3.14. DirectIOEvent class

public class DirectIOEvent extends PosEvent

Class for events that notify you of printer-specific status, etc.

3.14.1. Properties

For more information about this property, please refer to DirectIOEvent.

3.15. PosEvent class

public abstract class PosEvent extends java.util.EventObject

Abstract class of event classes handled by this SDK.

3.15.1. Properties

SequenceNumber Property

Syntax

long SequenceNumber;

Remarks

The global sequence number of events in this SDK.

Increments each time an event occurs.

When Property

Syntax

long When;

Remarks

The timestamp of the event.

System.currentTimeMillis () is set to the value.

3.16. StatusUpdateEvent class

public class StatusUpdateEvent extends PosEvent

Event fired when the status of a device changes.

3.16.1. Properties

For more information about this property, please refer to StatusUpdateEvent.

4. Printer Specification

The features that are available depend on the printer model. Please refer to the following, product specification, command specification.

4.1. FTP-62GDSL000

FTP-62GDSL100, FTP-62GDSL110, FTP-62GDSL120, FTP-62GUSL000, FTP-62GUSL070, FTP-62GUSL100, FTP-62GUSL100, FTP-63GUSL000, FTP-63GUSL070 and FTP-64GDSL130 also has the same specification.

4.1.1. Printable width

Printable width
432 dots
576 dots
832 dots

4.1.2. Resolution

Resolution
203 dpi

4.1.3. Paper type

Type	Paper
1	Unused
2	Unused
3	TF50KS-E45
4	PD150R PD160R PD170R
5	TP60KS-F1 TP50KJ-R
6	TF60KS-E P220VBB-1
7	HA220AA
8	Unused
9	AFP-235
10	Unused
11	Unused
12	Unused

4.1.4. Supported barcode and 2D code

Barcode and 2D code	Supported	Remarks
UPC-A	✓	
UPC-E	✓	
JAN(EAN)13	✓	
JAN(EAN)8	✓	
CODE39	✓	
ITF	✓	
CODABAR	✓	
CODE128	✓	
QR Code	✓	Only the model equipped with the extended volatile memory.
MaxiCode		
PDF417	✓	Supported only with firmware of V1.01 or higher. Only the model equipped with the extended volatile memory.
GS1 Databar	✓	Supported only with firmware of V1.01 or higher.

4.1.5. Feature

(*The value if MapMode property is set to PosPrinterConst.PTR_MM_DOTS.)

Feature	Corresponding range	Remarks
Bold	Not supported	
Reverse video	Supported	
Scale horizontally	1 – 4	
Scale vertically	1 – 4	
Underline	1 – 2	
Paper cut	Full cut Partial cut	
Feed and Paper cut	Full cut Partial cut	
Print bitmap by escape sequence	Supported	Only the model equipped with the extended nonvolatile memory.
Center	Supported	If executed or cancelled in the middle of a line, a line feed will occur, and the next line will be reflected. Supporting is different depending on the FW version. Prease refer to *1.
Right justify	Supported	If executed or cancelled in the middle of a line, a line feed will occur, and the next line will be reflected. Supporting is different depending on the FW version. Prease refer to *1.
HRI of barcode	Supported	Supporting is different depending on the FW version. Prease refer to *1.
Printable characters		Depends on the printer model
Print '€'	Supported	

Max of printing width when a rotated 90° left or right mode by rotatePrint method	Printable width 832 dots: 1320 dots Printable width 576 dots: 1702 dots Printable width 432 dots: 1702 dots	Supporting is different depending on the FW version. Prease refer to *1.
Rotate right 90° and left 90°	Supported	Supporting is different depending on the FW version. Prease refer to *1. Only the model equipped with the extended volatile memory.
Rotate 180°	Supported	
Near end sensor	Supported	
RotateSpecial property	PosPrinterConst.PTR_RP_NORMAL PosPrinterConst.PTR_RP_RIGHT90 PosPrinterConst.PTR_RP_LEFT90 PosPrinterConst.PTR_RP_ROTATE180	Supporting of PosPrinterConst.PTR_RP_RIGHT90 and PosPrinterConst.PTR_RP_LEFT90 is different depending on the FW version. Prease refer to *1. If the extended volatile memory is not equipped, PosPrinterConst.PTR_RP_RIGHT90 and PosPrinterConst.PTR_RP_LEFT90 are not supported.
printBitmap method	Supported	Only the model equipped with the extended volatile memory.
Max printable height of **printBitmap** method	1023 dots	Max height when rotating 90° right or left is the value of **RecLineWidth** property.
Max savable height of **setBitmap** method	512 dots	
RecLineWidth property	Printable width 832 dots: 832 Printable width 576 dots: 576 Printable width 432 dots: 432	
Max of printing width of **printBarCode** method when a rotated 90° left or right mode	864 dots	
The condition when PosConst_PS_OFF_OFFLINE is reported by **PowerState** property	Off, disconnected or when printer error occurs.	

*1） Supported FW version is as follows.

Printer model	Supported FW version
FTP-62GDSL000 FTP-62GDSL100 FTP-62GDSL120 FTP-62GUSL000 FTP-62GUSL070 FTP-62GUSL100 FTP-63GUSL000 FTP-63GUSL070	V1.01 or higher.
FTP-62GDSL110 FTP-64GDSL130	V1.00 or higher.

4.1.6. Font

（*The value if the **MapMode** property is set to PosPrinterConst.PTR_MM_DOTS.）

Printable width	**RecLineChars** property	**RecLineHeight** property	**RecSidewaysMaxChars** property	Font
832 dots	69	24	110（Fullwidth 55）	12x24 dots
	104	16	165（Fullwidth 820）	8x16 dots
576 dots	48	24	141（Fullwidth 70）	12x24 dots
	72	16	212（Fullwidth 106）	8x16 dots
432 dots	36	24	141（Fullwidth 70）	12x24 dots
	54	16	212（Fullwidth 106）	8x16 dots

4.1.7. Remarks

・Near-end sensor is disabled depending on the printer model. To enable it, send the FS 9 command with PTR_DIO_SEND_BINARY_DATA of the directIO method.
For details on the FS 9 command, refer to the product specifications or command specifications of the printer.

4.2. FTP-629DSL350

FTP-639USL100 and FTP-639USL200 also has the same specification.

4.2.1. Printable width

Printable width
576 dots

4.2.2. Resolution

Resolution
203 dpi

4.2.3. Paper type

Type	Paper
1	Unused
2	Unused
3	TF50KS-E4
4	PD150R PD160R-N PD170R
5	TP60KS-F1
6	TF60KS-E P220VBB-1
7	HA220AA
8	Unused
9	AFP-235
10	Unused
11	Unused
12	Unused

4.2.4. Supported barcode and 2D code

Barcode and 2D code	Supported	Remarks
UPC-A	✓	
UPC-E	✓	
JAN(EAN)13	✓	
JAN(EAN)8	✓	
CODE39	✓	
ITF	✓	
CODABAR	✓	
CODE128	✓	
QR Code	✓	Only the model equipped with the extended volatile memory and the extended nonvolatile memory.
MaxiCode		
PDF417	✓	Only the model equipped with the extended volatile memory and the extended nonvolatile memory.
GS1 Databar		

4.2.5. Feature

(*The value if MapMode property is set to PosPrinterConst.PTR_MM_DOTS.)

Feature	Corresponding range	Remarks
Bold	Not supported	
Reverse video	Supported	
Scale horizontally	1 – 4	
Scale vertically	1 – 4	
Underline	1 – 2	
Paper cut	Full cut Partial cut	Please do not perform partial cut at FTP-639USL200.
Feed and Paper cut	Full cut Partial cut	Please do not perform partial cut at FTP-639USL200.
Print bitmap by escape sequence	Supported	Only the model equipped with the extended nonvolatile memory.
Center	Supported	If executed or cancelled in the middle of a line, a line feed will occur, and the next line will be reflected. Supported only with firmware of V1.05 or higher.
Right justify	Supported	If executed or cancelled in the middle of a line, a line feed will occur, and the next line will be reflected. Supported only with firmware of V1.05 or higher.
HRI of barcode	Not Supported	
Printable characters		Depends on the printer model
Print '€'	Supported	
Max of printing width when a rotated 90° left or right mode by **rotatePrint** method	Not supported	
Rotate left 90° and right 90°	Not supported	
Rotate 180°	Supported	It is not supported for bitmap printing by **printBitmap** method.
Near end sensor	Supported	
RotateSpecial property	PosPrinterConst.PTR_RP_NORMAL PosPrinterConst.PTR_RP_ROTATE180	
printBitmap method	Supported	Only the model equipped with the extended volatile memory.
Max printable height of **printBitmap** method	1023 dots	
Max savable height of **setBitmap** method	512 dots	
RecLineWidth property	Printable width 576 dots: 576	
Max of printing width of **printBarCode** method when a rotated 90° left or right mode	Not supported	

The condition when PosConst.PS_OFF_OFFLINE is reported by PowerState property	Power off, disconnected or when a printer error occurs	

4.2.6. Font

（*The value if the MapMode property is set to PosPrinterConst.PTR_MM_DOTS.）

Printable width	RecLineChars property	RecLineHeight property	RecSidewaysMaxChars property	Font
576 dots	48	24	141（Fullwidth 70）	12x24 dots
	72	16	212（Fullwidth 106）	8x16 dots

4.2.7. When using FTP-639USL200

・The receipt length that can be held by the presenter is limited. Please adjust the receipt contents and cut the paper so that the receipt length of one page is 50 mm or more and 250 mm or less.

・Partial cut is unusable.

・After the paper is cut, the receipt is automatically ejected.

・After enabled or after paper cut, if the following method is executed without pulling out the paper, the receipt will be released.

cutPaper method
markFeed method
printBarCode method
printBitmap method
printNormal method
setBitmap method

・When using the following escape sequence with **printNormal** method or **setLogo** method, please use only at the end of data. parameter

Paper cut escape sequence
Feed and Paper cut escape sequence

・When using **transactionPrint** method, buffer the following functions at the end of the buffered data.

cutPaper method
Paper cut escape sequence
Feed and Paper cut escape sequence

4.2.8. Remarks

・Near-end sensor is disabled depending on the printer model. To enable it, send the FS 9 command with PTR_DIO_SEND_BINARY_DATA of the directIO method.

For details on the FS 9 command, refer to the product specifications or command specifications of the printer.

4.3. FTP-62HDSL100

4.3.1. Printable width

Printable width
384 dots
576 dots
832 dots

4.3.1. Resolution

Resolution
203 dpi

4.3.2. Paper type

Type	Paper
1	Unused
2	Unused
3	TF50KS-E45
4	PD150R
5	PD160R TP50KJ-R
6	Unused
7	HA220AA
8	Unused
9	Unused
10	Unused
11	Unused
12	Unused

4.3.3. Supported barcode and 2D code

Barcode and 2D code	Supported	Remarks
UPC-A	✓	
UPC-E	✓	
JAN(EAN)13	✓	
JAN(EAN)8	✓	
CODE39	✓	
ITF	✓	
CODABAR	✓	
CODE128	✓	
QR Code	✓	
MaxiCode	✓	
PDF417	✓	
GS1 Databar	✓	

4.3.4. Feature

(*The value if MapMode property is set to PosPrinterConst.PTR_MM_DOTS)

Feature	Corresponding range	Remarks
Bold	Not supported	
Reverse video	Supported	
Scale horizontally	1 to 4	
Scale vertically	1 to 4	
Underline	1 to 2	
Paper cut	Not supported	
Feed and Paper cut	Not supported	
Print bitmap by escape sequence	Supported	
Center	Supported	
Right justify	Supported	If executed or cancelled in the middle of a line, a line feed will occur, and the next line will be reflected.
HRI of barcode	Supported	If executed or cancelled in the middle of a line, a line feed will occur, and the next line will be reflected.
Printable characters		Depends on the printer model
Print '€'	Supported	
Max of printing width when a rotated 90° left or right mode by **rotatePrint** method	1144 dots	
Rotate right 90° and left 90°	Supported	
Near end sensor	Supported	
RotateSpecial property	PosPrinterConst.PTR_RP_NORMAL PosPrinterConst.PTR_RP_RIGHT90 PosPrinterConst.PTR_RP_LEFT90 PosPrinterConst.PTR_RP_ROTATE180	
printBitmap method	Supported	Only the model equipped with the extended volatile memory.
Max printable height of **printBitmap** method	1023 dots	Max height when rotating 90° right or left is the vlue of RecLineWidth property.
Max savable height of **setBitmap** method	512dots	
RecLineCharsList property	Printing width 832 dots:"69,104" Printing width 576 dots:"48,72" Printing width 384 dots:"32,48"	
RecLineWidth property	Printing width 832 dots:832 Printing width 576 dots:576 Printing width 384 dots:384	
Max of printing width of **printBarCode** method when a rotated 90° left or right mode	864 dots	
The condition when PosConst.OFF_OFFLINE is reported by **PowerState** property	Power off, disconnected or when printer error occurs.	

4.3.5. Font

(*The value if the **MapMode** property is set to FtpPosConst.PTR_MM_DOTS.)

Printable width	**RecLineChars** property	**RecLineHeight** property	**RecSidewaysMaxChars** property	Font
832 dots	69	24	95(Fullwidth 47)	12×24 dots
	104	16	143(Fullwidth 71)	8×16 dots
576 dots	48	24	95(Fullwidth 47)	12×24 dots
	72	16	143(Fullwidth 71)	8×16 dots
384 dots	32	24	160(Fullwidth 80)	12x24 dots
	48	16	240(Fullwidth 120)	8x16 dots

4.3.6. Remarks

・Near-end sensor is disabled depending on the printer model. To enable it, send the FS 9 command with PTR_DIO_SEND_BINARY_DATA of the directIO method.

For details on the FS 9 command, refer to the product specifications or command specifications of the printer.

5. Update History

DRW NO.	A1NC40416-6000R0/6	
Edit	Date	Place / Reason / Content
01	20200828	First version (Preliminary version)
02	20201030	Update for Product version.
		Removed pass through embedded data escape sequence.
		Added the binary data sending by directIO method.

6. Driver Update History

Version	Content of change	Date	File name
V0.01	Alpha version	20200422	FTP2166000R0_FTP-POS_Android_V001.zip
V0.02	Beta version	20200828	FTP2166000R0_FTP-POS_Android_V002.zip
	Change all specifications.		
	Rename DeviceConfiguration class to PrinterConfiguration class.		
	Rename SupportedModel enum to PrinterModel enum.		
	Remove unsupported properties, methods and events		
	Change the package		
V1.00	Product version	20201030	FTP2166000R0_FTP-POS_Android_V100.zip
	Removed pass through embedded data escape sequence.		
	Added the binary data sending by directIO method.		